BUDGET CONSTRAINED MINIMUM COST CONNECTED MEDIANS

GORAN KONJEVOD [1], SVEN O. KRUMKE [2], AND MADHAV MARATHE [3]

ABSTRACT. Several practical instances of network design and location theory problems require the network to satisfy multiple constraints. In this paper, we address graph-theoretic problems that aim to simultaneously address a network design task and a location-theoretic constraint. The *Budget Constrained Connected Median Problem* is the following: We are given an undirected graph $G = (V, E)$ with two different edge-weight functions c (modeling the construction or communication cost) and d (modeling the service distance), and a bound B on the total service distance. The goal is to find a subtree T of G with minimum c-cost $c(T)$ subject to the constraint that the sum $\sum_{v \in V \setminus T} \text{dist}_d(v, T)$ of the service distances of all the remaining nodes $v \in V \setminus T$ does not exceed the specified budget B. Here, the *service distance* $\text{dist}_d(v, T)$ denotes the smallest shortest path distance of v to a vertex in T with respect to d. This problem has applications in optical network design and the efficient maintenance of distributed databases.

We formulate this problem as a bicriteria network design problem, and present bicriteria approximation algorithms. We also prove lower bounds on the approximability of the problem which demonstrate that our performance ratios are close to best possible.

Keywords: NP-hardness, Approximation Algorithms, Network Design, Median, k-Median Problem, Group Steiner Tree Problem

1. INTRODUCTION AND OVERVIEW

Given an undirected connected graph $G = (V, E)$ with two different edge-weight functions c (modeling the construction cost of the backbone/inter-database links) and d (modeling the service distance), the *Budget Constrained Connected Median Problem (*BCCMED*)* is to find a subtree T of G with minimum c-cost $c(T) = \sum_{e \in T} c(e)$ subject to the constraint that the sum of the service distances $\sum_{v \in V \setminus T} \text{dist}_d(v, T)$ of all the remaining nodes $v \in V \setminus T$ to a closest node in T does not exceed a specified budget B. The problem aims at combining an objective function typical for network design (cost of the tree) with a location-theoretic constraint (total cost of covering the nodes not in the tree).

A problem of this nature arises naturally in several practical settings. Power-aware network design is one such area. Due to recent advances in radio technology,

[1]Email: goran@asu.edu Department of Computer Science and Engineering, Arizona State University, Tempe, Az. Some of the work was done while the author was at Dept. of Mathematical Sciences, Carnegie Mellon University, Pittsburgh, PA 15213-3890 and visiting Los Alamos National Laboratory with support from the NSF CAREER grant CCR-9625297 and DOE Contract W-7405-ENG-36.

[2]Email: krumke@zib.de Konrad-Zuse-Zentrum für Informationstechnik Berlin Department Optimization Takustr. 7 14195 Berlin-Dahlem Germany. Research supported by the German Science Foundation (DFG, grant Gr 883/5-3)

[3]Email: marathe@lanl.gov. Basic and Applied Simulation Science, Los Alamos National Laboratory, P.O. Box 1663, MS M997, Los Alamos, NM 87545. Research supported by the Department of Energy under Contract W-7405-ENG-36.

researchers have begun to build ad-hoc networks with low-power radio devices (see [16] and the references therein for more details on this subject). We illustrate the kind of problems that arise in such systems in the context of the *information acquisition problem* in sensor networks. We are given a set of small radio devices that measure the relevant state of the system in a continuous fashion. This might correspond to the temperature and moisture content of the soil, or presence of a biological agent in the atmosphere. The data is periodically sent back to a base station for further analysis. The power consumed by the sensors is proportional to the amount of data transmitted as well as the range of the transmitter. We assume for uniformity that each sensor has the same amount of data to transmit. The goal is to select power levels (which is proportional to the d-edge costs in our formalism) and appropriate cluster head to relay information amongst the sensors so as to minimize the total power consumed.

Another area where such a problem might arise is in the design of modern opto-electronic networks. Interfacing optic and electronic networks has become an important problem in telecommunication network design [25, 27]. As an example, consider the following problem: Given a set of sites in a network, we wish to select a subset of the sites at which to place optoelectronic switches and routers. The backbone sites should be connected together using fiber-optic links in a minimum-cost tree, while the end users are connected to the backbone via direct links. The major requirement is that the total access cost for the users is within a specified bound, whereas the construction cost of the backbone network should be minimized.

Similar problems arise in the efficient maintenance of distributed databases [5, 6, 9, 20, 28]. Other applications of BCCMED include location theory and manufacturing logistics (see [25, 27] and the references cited therein).

In this paper, we study the complexity and approximability of BCCMED. The paper is organized as follows. In Section 2 we formally define the problem under study and the notion of bicriteria approximation. Section 3 contains a brief summary of the main results in the paper and a discussion of related work. In Section 4, we discuss how BCCMED and its "bicriteria dual" are related to other known problems in the literature. In Section 5 we prove hardness results. Section 6 contains a fully polynomial approximation scheme on trees. An approximation algorithm for the general case is presented in Section 7.

2. Problem Definition and Preliminaries

Throughout the paper $G = (V, E)$ denotes a finite connected undirected graph with $n := |V|$ vertices and $m := |E|$ edges. The *Budget Constrained Connected Median Problem (*BCCMED*)* problem considered in this paper is defined as follows:

Definition 2.1 (Budget Constrained Connected Median Problem)**.** An instance consists of a connected undirected graph $G = (V, E)$ with two different nonnegative edge-cost functions $c\colon E \to \mathbb{Q}_+$ (modeling the construction or communication cost) and $d\colon E \to \mathbb{Q}_+$ (modeling the service distance), and a bound B on the total service distance. The problem is to find a subtree T of G of minimum cost $c(T) := \sum_{e \in T} c(e)$ subject to the constraint that the total service distance of all vertices from V is at most B, that is,

$$\text{service}_d(T) := \sum_{v \in V} \text{dist}_d(v, T) \leq B,$$

where $\text{dist}_d(v, T) := \min_{u \in T} \text{dist}_d(v, T)$ and $\text{dist}_d(v, u)$ denotes the shortest path distance between vertices v and u with respect to the edge-cost function d.

The problem BCCMED can be formulated within the framework developed in [18, 23]. A generic bicriteria network design problem, (f_1, f_2, Γ), is defined by identifying two minimization objectives, f_1 and f_2, from a set of possible objectives, and specifying a membership requirement in a class of subgraphs Γ. The problem specifies a budget value on the first objective, f_1, and seeks to find a network having minimum possible value for the second objective, f_2, such that this network is within the budget on the first objective f_1. The solution network must belong to the subgraph-class Γ. In this framework BCCMED is stated as (TOTAL d-SERVICE DISTANCE, TOTAL c-EDGE COST, SUBTREE): the budgeted objective f_1 is the total service distance $\text{service}_d(T)$ with respect to the edge weights specified by d, the cost-minimization objective f_2 is the total c-cost of the edges in the solution subgraph which is required to be a subtree of the original network.

Definition 2.2 (Bicriteria Approximation Algorithm)**.** A polynomial time algorithm for a bicriteria problem (f_1, f_2, Γ) is said to have *performance* (α, β), if it has the following property: For any instance of (f_1, f_2, Γ), the algorithm produces a solution from the subgraph class Γ for which the value of objective f_1 is at most α times the specified budget and the value of objective f_2 is at most β times the minimum value of a solution from Γ that satisfies the budget constraint.

We also use the term "(α, β)-approximation algorithm" to refer to an algorithm with performance (α, β).

Definition 2.3 (Fully Polynomial Approximation Scheme)**.** A family $\{A_\epsilon\}_\epsilon$ of approximation algorithms, is called a *fully polynomial approximation scheme* or *FPAS*, if algorithm A_ϵ is a $(1, 1+\epsilon)$-approximation algorithm and its running time is polynomial in the size of the input and $1/\epsilon$.

As discussed in [18, 23], for generic bicriteria problems, we can naturally formulate a variant of the BCCMED in which we interchange the two objectives. More precisely, in this case we put a bound on the cost of the tree and seek to minimize the cost of covering the nodes not in the tree. The resulting problem is (TOTAL c-EDGE COST, TOTAL d-SERVICE DISTANCE, SUBTREE). In [18, 23], the general relationship between such bicriteria problems is studied from the standpoint of approximability. It follows from these results that an (α, β)-approximation algorithm for BCCMED (stated as (TOTAL d-SERVICE DISTANCE, TOTAL c-EDGE COST, SUBTREE) in the form of the framework from [18, 23] as noted above) implies the existence of an $(\beta, (1+\epsilon)\alpha)$-approximation algorithm for its "bicriteria dual" (TOTAL c-EDGE COST, TOTAL d-SERVICE DISTANCE, SUBTREE) for any fixed $\epsilon > 0$.

3. SUMMARY OF RESULTS

In this paper, we study the complexity and approximability of the problem BCCMED. Our main results include the following:

(1) BCCMED is NP-hard even on trees. This result continues to hold even if the edge-weight functions c and d are identical. We strengthen this hardness result to obtain NP-hardness in the strong sense for bipartite graphs.

(2) We strengthen the above hardness results for general graphs further and show that unless $\mathsf{NP} \subseteq \mathsf{DTIME}(N^{\log\log N})$, there can be no polynomial-time approximation algorithm for BCCMED with a performance $(1, (1/20 - \epsilon)\ln n)$, where n denotes the number of vertices in the input graph.

Under the weaker assumption that $\mathsf{P} \neq \mathsf{NP}$, we show that there exists a constant $\gamma > 0$, such that there is no approximation algorithm for BCCMED with a performance of $(1, \gamma \ln\ln n)$.

Our hardness results are complemented by the following approximation results:

(1) There exists a FPAS for BCCMED on trees.
(2) For any fixed $\epsilon > 0$ there exists a $(1+\epsilon, (1+1/\epsilon)\mathcal{O}(\log^3 n \log\log n))$-approximation algorithm for BCCMED on general graphs.

As shown in the next section, the BCCMED problem is closely related to the *Traveling Purchaser Problem*. The recent approximation algorithms from [25] can be used to obtain results for BCCMED with metric c-cost. However, our results apply without this restriction and give a slightly better approximation guarantee.

4. Past Work and Relationship with Other Problems

In the past, several service-constrained minimum-cost network problems have been considered in [1, 8, 17, 21, 22]. These papers consider the variant that prescribes a budget on the service distance for each node. The goal is to determine a solution subgraph (which in some of the cited papers is a salesperson tour and in others is a subtree) subject to the budget constraints of the vertices not in the solution.

In case that the solution subgraph must be a salesperson tour, nodes represent customers and the service radius represents the distance a customer is willing to travel to meet the salesperson. The goal is to find a minimum-length salesperson tour so that all the (customer) nodes are strictly serviced. Restrictions of the problems to geometric instances were considered in [1, 17, 24].

These problems are similar to BCCMED studied here, the primary difference being in the way the location-theoretic constraint is formulated. In the current paper, we put a budget on the sum of the costs of the nodes not in the tree while the papers referred to above consider a budget on the maximum distance. This is similar to the difference between the *k-Center Problem* and *k-Median Problem* problems considered in the literature.

An important motivation for the problem considered in the current paper is to define and study problems that simultaneously address location-theoretic and connectivity constraints. We believe that such a generalization on the one hand allows more realistic models of several practical problems and on the other hand might prove useful in obtaining a more unified theory of approximation algorithms for these two closely related classes of problems.

4.1. **Relationship to other problems.** BCCMED problem is related to and generalizes several well known network design and location problems. We discuss a few of these problems in order to illustrate the connections.

4.1.1. *k-Median Problem.* Given a complete undirected graph $G = (V, E)$, an edge-weight function $d\colon E \to \mathbb{R}_{\geq 0}$ and a positive integer k, the *k-Median Problem* is to find a set of nodes $U \subseteq V$, $|U| = k$ such that $\sum_{v \in V-U} d(v, U)$ is minimized. Here, $d(v, U)$ is the smallest distance of the vertex v to a vertex in U.

The "bicriteria dual" problem to BCCMED, namely (TOTAL c-EDGE COST, TOTAL d-SERVICE DISTANCE, SUBTREE), can be seen as a generalization of the k-Median Problem. Namely, given an instance I of the *k-Median Problem*, we construct an instance $I' = (G', c', d', B')$ of (TOTAL c-EDGE COST, TOTAL d-SERVICE DISTANCE, SUBTREE) as follows: (1) the graph G' specified in I' coincides with G, (2) the d'-cost on the edges of G' is defined as in the instance of k-Median Problem, (3) we set $c'(e) = 1$ for all edges $e \in G'$ and (4) the budget B' on the tree cost is $B' := k - 1$. It is clear that the there is a solution of cost C for I if and only if there is solution of the same cost for I'. Using the known results on the non-approximability of k-Median Problem [19], we obtain the following result:

Theorem 4.1. *Unless* $\mathsf{P} = \mathsf{NP}$*, for any* $\beta > 0$ *there is no* $(1, \beta)$*-approximation algorithm for* (TOTAL c-EDGE COST, TOTAL d-SERVICE DISTANCE, SUBTREE). □

4.1.2. *Group Steiner Problem.* Given a complete undirected graph $G = (V, E)$ with edge weights $c(e)$ $(e \in E)$ and a collection $G_1, \ldots, G_k$ of (not necessarily disjoint) subsets of V, the *Group Steiner Tree Problem (GST)* consists of finding a subtree of G of minimum cost such that this tree contains at least one vertex from each of the groups $G_1, \ldots, G_k$. Garg, Konjevod and Ravi [14] devised a randomized poly-logarithmic approximation algorithm for the GST problem. Charikar et al. [7] showed how to derandomize the algorithm thus yielding a deterministic algorithm for GST with poly-logarithmic performance guarantee.

Given an instance I of GST, we construct an instance $I' = (G', c', d', B')$ of BCCMED as follows. To create G', remove from G all Steiner vertices (those not contained in any group). Define $c'(u, v)$ for $u, v \in V(G')$ to be their shortest-path c-distance in G. For each group G_i, set $d'(e) = 0$ for all edges such that both end points are in G_i. The d'-cost of each remaining edge is set to some large number and the budget B' is set to 0. Now given a solution of cost C for I, it is easy to construct a solution of cost $\leq 2C$ for I': simply use the vertices of the tree T generated for I as the vertices of the solution tree T' for I'. The c'-cost of T' is no more than twice the c-cost of T because the cost of terminal-to-terminal edges of T has not increased, and the cost of each Steiner-to-terminal or Steiner-to-Steiner edge of T may in the worst case be counted twice in T'. Conversely, assume that G' contains a tree T' of cost C' for I'. The budget constraint implies that we will pick at least one node from each group G_i. Thus, T' can be used as a solution for I no more expensive than C'. This approximation-preserving reduction from GST to BCCMED implies the following result:

Theorem 4.2. *A* $(1, \beta)$*-approximation algorithm for the* BCCMED *problem implies a* 2β*-approximation algorithm for the Group Steiner tree problem.* □

The theorem should be compared to the approximation algorithm for BCCMED given in Section 7; it implies a nearly best possible solution to BCCMED given the current status of the GST problem.

4.1.3. *k-Minimum Spanning Tree.* Given a complete undirected graph $G = (V, E)$ with nonnegative edge weights $c(e)$ $(e \in E)$, the *k-Minimum Spanning Tree Problem (k-MST)* asks to find a tree spanning at least k nodes in V of minimum cost. There has been a substantial amount of research on providing upper and lower bounds for the k-MST problem (see [2, 13, 26]; currently the best results are those in [2, 3] for both the geometric and non-geometric case). Given an instance I of

the k-MST problem we can transform it to an instance $I' = (G', c', d', B')$ of BCCMED as follows: Once more, the graph G' is identical to G specified in I. The c'-costs in I' are equal to the respective c-costs given in I. The d'-cost of each edge in I' is set to 1. We place a budget $B' := (n - k)$ on the total d'-service cost. This in particular implies that any feasible solution for I' can leave at most $n - k$ vertices out of the tree; in other words, it must contain at least k vertices. It is now easy to see how to obtain—given a $(1, \beta)$-approximation algorithm for BCCMED—a solution with performance guarantee β for the k-MST problem.

4.1.4. *Traveling Purchaser Problem.* The BCCMED problem is also closely related to a well studied variant of the classical traveling salesperson problem called the *Traveling Purchaser Problem* (see [25] and the references therein). In this problem we are given a bipartite graph $G = (M \cup P, E)$, where M denotes a set of markets and P denotes the set of products. There is a (metric) cost c_{ij} to travel from market i to market j. An edge between market i and product p with cost d_{ip} denotes the cost of purchasing product p at market i. A *tour* consists of starting at a specified market visiting a subset of market nodes, thereby purchasing *all* the products and returning to the starting location. The cost of the tour is the sum of the travel costs used between markets and the cost of buying each of the products. The budgeted version of this problem as formulated by Ravi and Salman [25] aims at finding a minimum-cost tour subject to a budget constraint on the purchasing costs.

It is easy to see that an (α, β)-approximation algorithm for the budgeted traveling purchaser problem implies an $(\alpha, 2\beta)$-approximation for BCCMED with metric c-cost: just delete one edge of the tour to obtain a tree. Using the $(1 + \epsilon, (1 + 1/\epsilon)\mathcal{O}(\log^3 m \log\log m))$-approximation algorithm from [25] we get a $(1 + \epsilon, 2(1 + 1/\epsilon)\mathcal{O}(\log^3 m \log\log m))$-approximation for BCCMED. Our algorithm given in Section 7 uses the techniques from [25] directly and improves this result by a factor of 2. In addition, it does not require that the cost c be metric.

5. HARDNESS RESULTS

Theorem 5.1. *The problem* BCCMED *is* NP*-hard even on trees. This result continues to hold even if we require the two cost functions c and d to coincide.*

Proof. We use a reduction from the PARTITION problem, which is well known to be NP-complete [12, Problem SP12]. Given a multiset of (not necessarily distinct) positive integers $\{a_1, \ldots, a_n\}$, the question is whether there exists a subset $U \subseteq \{1, \ldots, n\}$ such that $\sum_{i \in U} a_i = \sum_{i \notin U} a_i$.

Given any instance of PARTITION we construct a star-shaped graph G having $n + 1$ nodes $\{x, x_1, \ldots, x_n\}$ and n edges (x, x_i), $i = 1, \ldots, n$. We define $c(x, x_i) := d(x, x_i) := a_i$. Let $D := \sum_{i=1}^{n} a_i$. We set the budget for the median cost of the tree to be $B := D/2$. It is easy to see that there exists a feasible tree T of cost $c(T)$ at most $D/2$ if and only if the instance of PARTITION has a solution. □

We can easily make the tree binary by adding dummy nodes and adding edges of c-costs and d-costs equal to 0. These results immediately implies NP-hardness of the BCCMED problem for graphs that are simultaneously of bounded treewidth, planar and of bounded degree.

Next, we give inapproximability results for general graphs. Before stating the hardness result we recall the definition of the MIN SET COVER problem [12, Problem SP5] and cite the hardness results from [4, 11] about the hardness of approximating MIN SET COVER. An instance $(U, \mathcal{S})$ of MIN SET COVER consists of a finite set U of ground elements and a family $\mathcal{S}$ of subsets of U. The objective is to find a subcollection $\mathcal{C} \subseteq \mathcal{S}$ of minimum size $|\mathcal{C}|$ which contains all the ground elements.

Theorem 5.2 (Feige [11])**.** *Unless* $\mathsf{NP} \subseteq \mathsf{DTIME}(N^{\mathcal{O}(\log\log N)})$*, for any* $\epsilon > 0$ *there is no approximation algorithm for* MIN SET COVER *with a performance of* $(1-\epsilon)\ln|U|$*, where* U *is the set of ground elements.* □

Theorem 5.3 (Arora and Sudan [4])**.** *There exists a constant* $\eta > 0$ *such that, unless* $\mathsf{P} = \mathsf{NP}$*, there is no approximation algorithm for* MIN SET COVER *with a performance of* $\eta \ln|U|$*, where* U *is the set of ground elements.* □

We are now ready to prove the result about the inapproximability of BCCMED on general graphs.

Theorem 5.4. *The problem* BCCMED *is strongly* NP*-hard even on bipartite graphs. If there exists an approximation algorithm for* BCCMED *on bipartite graphs with performance* $(1, f(|V|))$*, where* $f(|V|) \in \mathcal{O}(\ln|V|)$*, then there exists an approximation algorithm for* MIN SET COVER *with performance* $2f(2|U| + 2|\mathcal{S}|)$*. All results continue to hold even if we require the two cost functions* c *and* d *to coincide.*

Proof. Let $(U, \mathcal{S})$ be an instance of MIN SET COVER. We assume without loss of generality that the minimum-size set cover for this instance contains at least two sets (implying also that $|U| \geq 2$).

For each $k \in \{2, \ldots, n\}$ we construct an instance I_k of BCCMED as follows: The bipartite graph constructed for instance I_k has $|V_k| = 2|U| + 2|\mathcal{S}| + 2 - k \leq 2(|U| + |\mathcal{S}|)$ vertices. First construct the natural bipartite graph with node set $U \cup \mathcal{S}$. We add an edge between an element node $u \in U$ and a set node $S \in \mathcal{S}$ if and only if $u \in S$. We now add a root node r which is connected via edges to all the set nodes from $\mathcal{S}$. Finally, we add a set L_k of $|U| + |\mathcal{S}| - k + 1$ nodes which are connected to the root node via the edges (l, r), $l \in L_k$. Let $\Delta := k\lceil f(|V_k|)\rceil + 1$. The edges between element nodes and set nodes have weight Δ, all other edges have weight 1. The budget on the median cost for instance I_k is set to $B_k := |L_k| + \Delta|U| + |\mathcal{S}| - k$. The construction is illustrated in Figure 1.

As noted above, the bipartite graph constructed for instance I_k has $|V_k| = 2|U| + 2|\mathcal{S}| + 2 - k \leq 2(|U| + |\mathcal{S}|)$ vertices. Thus, $f(|V_k|) \leq f(2|U| + 2|\mathcal{S}|)$.

The main goal of the proof is to show that (i) if there exists a set cover of size k, then instance I_k of BCCMED has a solution with value at most k; (ii) any feasible solution for instance I_k of BCCMED with cost $C \leq f(|V_k|)k$ can be used to obtain a set cover of size at most $2C$. Using these two properties of the reduction, we can show that any $f(|V|)$-approximation to BCCMED transforms into a $2f(2|U| + 2|\mathcal{S}|)$-approximation for MIN SET COVER: Find the minimum value $k^* \in \{1, \ldots, n\}$ such that the hypothetical f-approximation algorithm A for BCCMED outputs a solution of cost at most $f(|V_{k^*}|)k^*$ for instance I_{k^*}. By property (i) and the performance of A it follows that k^* is no greater than the optimal size set cover. By property (ii) we get a set cover of size at most $2f(|V_{k^*}|)k^*$ which is at most $2f(2|U| + 2|\mathcal{S}|)$ times the optimal size cover.

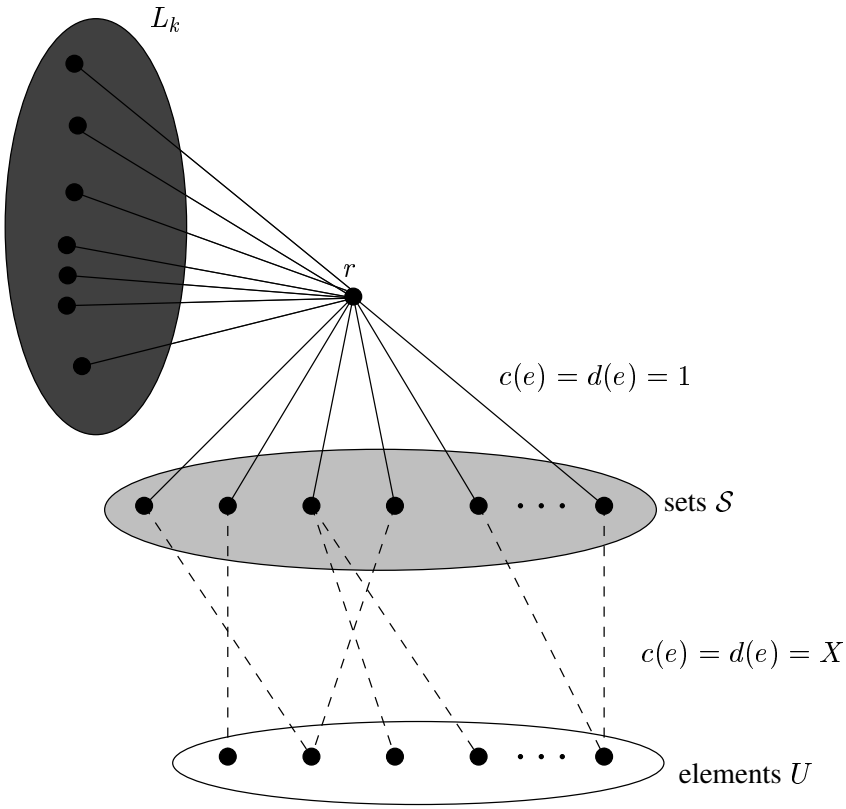

FIGURE 1. Construction used in the proof of Theorem 5.4. Solid edges are of weight 1, dashed edges have weight $\Delta = k\lceil f(|V_k|)\rceil + 1$.

We first prove (i). Any set cover $\mathcal{C}$ of size k can be used to obtain a tree by choosing the subgraph induced by the set nodes corresponding to the sets in $\mathcal{C}$ and the root node r. Clearly, the cost of the tree is k. Since the sets in $\mathcal{C}$ form a cover, each element node is within distance of Δ from a vertex in the tree. Thus, the total median-cost of T is no more than $\Delta|U| + |\mathcal{S}| - k + |L_k| = B_k$.

We now address (ii). Assume conversely, that T is a solution for I_k with value C, i.e., a tree with service$_d(T) \leq B_k$ and $c(T) = C \leq f(|V_k|)k$. We first show that the root node r must be contained in the tree.

Assume for the sake of a contradiction that $r \notin T$. Since $\Delta \geq C + 1$, the tree T can not contain any edge between set nodes and element nodes. Hence, it follows that T consists of a single node $v \in U \cup \mathcal{S} \cup L_k$. If $T = \{v\}$ with $v \in U$, then each element node from $U \setminus \{v\}$ is at distance at least 2Δ from T and each node from L_k is at distance at least $2 + \Delta \geq 3$ from T. Consequently, the service distance cost of T is at least

$$2\Delta(|U| - 1) + 3|L_k| = B_k + (|U| - 2)\Delta + 2|U| + |\mathcal{S}| - k + 2 > B_k,$$

which contradicts that T is feasible. If $T = \{v\}$ with $v \in \mathcal{S}$, then the service distance cost of T is at least

$$\Delta|U| + 2|L_k| + 2(|\mathcal{S}| - 1) + 1 = B_k + |L_k| + |\mathcal{S}| + k - 1 > B_k,$$

which is again a contradiction. In the above calculation, the terms in the first expression account for the element nodes, the nodes from L_k, the $|\mathcal{S}|-1$ set nodes not in T and the root in this order. It remains to cover the case that $T=\{v\}$ with $v \in L_k$. Then, the service distance cost of T is no smaller than

$$(\Delta+2)|U|+2(|L_k|-1)+1+2|\mathcal{S}| = B_k+2|U|+L_k+|\mathcal{S}|+k-1 > B_k.$$

This completes the proof of the fact that the root node r must be contained in the tree T.

We now show that the collection $\mathcal{C}$ of set nodes spanned by the tree T can be used to obtain a cover of size at most $2C$. Let $\bar{U}_{\mathcal{C}} \subseteq U$ be the subset of element nodes not covered by the collection $\mathcal{C}$ of sets. For each element $u \in \bar{U}_{\mathcal{C}}$ its distance to any node in T is at least $\Delta+1$. The median cost of T thus satisfies:

$$\text{service}_d(T) \geq |L_k|+\Delta|U|+|\bar{U}_{\mathcal{C}}|+|\mathcal{S}|-|T\cap(L_k\cup\mathcal{S})| = B_k+k+|\bar{U}_{\mathcal{C}}|-C. \quad (1)$$

On the other hand, since T is feasible, $\text{service}_d(T) \leq B_k$, and hence we get from (1) that $|\bar{U}_{\mathcal{C}}| \leq C-k < C$. In words, the number of elements left uncovered by the collection $\mathcal{C}$ of sets is at most C, the cost of the tree T. Hence, we can augment $\mathcal{C}$ to a valid cover by adding at most $|\bar{U}_{\mathcal{C}}| \leq C$ sets. This leads to a set cover of size at most $2C$. □

Corollary 5.5. (i) *Unless* $\mathsf{NP} \subseteq \mathsf{DTIME}(N^{\log\log N})$, *for any* $\epsilon > 0$ *there can be no polynomial time approximation algorithm for* BCCMED *with a performance* $(1,(1/20-\epsilon)\ln n)$, *where* n *denotes the number of vertices in the input graph.*

(ii) *There exists a constant* $\gamma > 0$ *such that, unless* $\mathsf{P}=\mathsf{NP}$, *there is no approximation algorithm for* BCCMED *with a performance of* $(1,\gamma\ln\ln n)$.

Proof. To prove part (i), we use the following fact: The instances of MIN SET COVER used in [11] have the property that the number of sets is at most $|U|^5$, where U is the ground set (see [10] for an explicit computation of the number of sets used). Hence for (i), it suffices to show that any approximation for BCCMED can be used to obtain an approximation for those instances $(U,\mathcal{S})$ of MIN SET COVER with $|\mathcal{S}| \leq |U|^5$. For the sake of reference we call those MIN SET COVER-instances *low set number* MIN SET COVER*-instances*.

By Theorem 5.4 the existence of an approximation algorithm for BCCMED with performance $(1,c\ln n)$ for some $c>0$, implies that there exists an approximation algorithm for all low set number MIN SET COVER instances with performance $2c\ln(2|U|+2|U|^5)$. Notice that for $|U| \geq 2$ we have $2c\ln(2|U|+2|U|^5) \leq 4c\ln|U|^5 = 20c\ln|U|$. By the hardness result of [11] we must have $20c \geq 1$, or $c \geq 1/20$ (under the assumption that $\mathsf{NP} \not\subseteq \mathsf{DTIME}(N^{\log\log N})$). This proves part (i) of the corollary.

Part (ii) is proved in a similar way. Notice that for any instance $(U,\mathcal{S})$ of MIN SET COVER the number of sets is bounded by $2^{|U|}$. By Theorem 5.4, an approximation for BCCMED with performance $(1,c\ln\ln n)$ implies an approximation algorithm for (the general) MIN SET COVER-problem with performance $2c\ln\ln(2|U|+2|\mathcal{S}|) \leq 2c\ln\ln(2|U|+2\cdot 2^{|U|}) \leq 8c\ln|U|$. The claim follows because unless $\mathsf{P}=\mathsf{NP}$ [4], we have $8c \geq \eta$ for some constant $\eta > 0$. □

6. Approximation Scheme on Trees

We first consider the problem BCCMED when restricted to trees. We present an FPAS for a slightly more general problem than BCCMED, called *generalized* BCCMED in the following: We are additionally given a subset $U \subseteq V$ of the vertex set and the budget constraint requires that the total service distance of all vertices in U (instead of V) does not exceed B.

Theorem 6.1. *There is a FPAS for the generalized* BCCMED *on trees with running time $\mathcal{O}(\log(nC)n^3/\epsilon^2)$, where C denotes the maximum c-weight of an edge in the given instance.*

Proof. Let I be a given instance of BCCMED and let $T = (V, E)$ be the tree given in I. We root the tree at an arbitrary vertex $r \in V$. In the sequel we denote by T_v the subtree of T rooted at vertex $v \in V$. So $T_r = T$. Without loss of generality we can assume that r is contained in some optimal solution I (we can run our algorithm for all vertices as the root vertex). We can also assume without loss of generality that the rooted tree T is binary (since we can add zero cost edges and dummy nodes to turn it into a binary tree). Finally, we can assume that all edge weights $c(e)$ ($e \in E$), are integral, since otherwise we can multiply the rational numbers by their common denominator and consider the resulting problem.

Let $T^* = (V^*, E^*)$ be an optimal solution for I which contains r. Denote by $\mathsf{OPT} = c(T^*)$ its cost. Define $C := \max_{e \in E} c(e)$ and let $K \in [0, nC]$ be an integral value. The value K will act as "guess value" for the optimal cost in the final algorithm. Notice that due to our integrality assumption on the c-weights the optimal cost is an integer between 0 and nC.

For a vertex $v \in V$ and an integer $k \in [0, K]$ we denote by $D[v, k]$ the minimum service cost of a tree $T^*_{v,k}$ servicing all nodes in U contained in the subtree T_v rooted at v and which has following properties: (1) $T^*_{v,k}$ contains v, and (2) $c(T^*_{v,k}) \leq k$. If no such tree exists, then we set $D[v, k] := +\infty$. Notice that

$$c(T^*) = \min\{\, k : D[r, k] \leq B \,\}.$$

Let $v \in V$ be arbitrary and let v_1, v_2 be its children in the rooted tree T. We show how to compute all the values $D[v, k]$, $1 \leq k \leq B$ given the values $D[v_i, \cdot]$, $i = 1, 2$.

For $i = 1, 2$ let $S_i := \sum_{w \in T_{v_i} \cap U} \mathrm{dist}_d(w, v)$. If v_i is not in the tree $T^*_{v,k}$ then none of the vertices in T_{v_i} can be contained in $T^*_{v,k}$. Let

$$X_k := S_1 + S_2$$

and

$$Y_k := \min\{\, D[v_1, k'] + D[v_2, k''] : k' + k'' = k - c(v, v_1) - c(v, v_2) \,\}.$$

Then we have that

$$D[v, k] = \min\{S_2 + D[v_1, k - c(v, v_1)], S_1 + D[v_2, k - c(v, v_1)], X_k, Y_k\}.$$

The first term in the last equation corresponds to the case that v_1 is in $T^*_{v,k}$ but not v_2. The second term is the symmetric case when v_2 is in the tree but not v_1. The third term concerns the case that none of v_1 and v_2 is in the tree. Finally, the fourth term models the case that both children are contained in $T^*_{v,k}$.

It is straightforward to see that this way all the values $D[v, k]$, $0 \leq k \leq K$ can be computed in $\mathcal{O}(K^2)$ time given the values for the children v_1 and v_2. Since

the table values for each leaf of T can be computed in time $\mathcal{O}(K)$, the dynamic programming algorithm correctly finds an optimal solution within time $\mathcal{O}(nK^2)$.

Let $\epsilon > 0$ be a given accuracy requirement. Now consider the following test for a parameter $M \in [0, (n-1)C]$: First we scale all edge costs $c(e)$ in the graph by setting

$$c^M(e) := \left\lceil \frac{(n-1)c(e)}{M\epsilon} \right\rceil. \tag{2}$$

We then run the dynamic programming algorithm above with the scaled edge costs and $K := (1 + 1/\epsilon)(n-1)$. We call the test *successful* if the algorithm gives the information that $D[r, K] \leq B$. Observe that the running time for one test is $\mathcal{O}(\frac{n^3}{\epsilon^2})$.

We now prove that the test is successful if $M \geq \mathsf{OPT}$. To this end we have to show that there exists a tree of cost at most K such that its service cost is at most B. Recall that T^* denotes an optimal solution. Since we have only scaled the c-weights, it follows that T^* is also a feasible solution for the scaled instance with service cost at most B. If $M \geq \mathsf{OPT}$ we have

$$\sum_{e \in T^*} c^M(e) \leq \sum_{e \in T^*} \left(\frac{(n-1)c(e)}{M\epsilon} + 1 \right) \leq \frac{n-1}{\epsilon} + |T^*| \leq \left(1 + \frac{1}{\epsilon}\right)(n-1).$$

Hence for $M \geq \mathsf{OPT}$, the test will be successful. We now use a binary search to find the minimum integer $M' \in [0, (n-1)C]$ such that the test described above succeeds. Our arguments from above show that the value M' found this way satisfies $M' \leq \mathsf{OPT}$. Let T' be the corresponding tree found which has service cost no more than B. Then

$$\sum_{e \in T'} c(e) \leq \frac{M'\epsilon}{n-1} \sum_{e \in T'} c^{M'}(e) \leq \frac{M'\epsilon}{n-1} \left(1 + \frac{1}{\epsilon}\right)(n-1) \leq (1+\epsilon)\mathsf{OPT}. \tag{3}$$

Thus, the tree T' found by our algorithm has cost at most $1 + \epsilon$ times the optimal cost. The running time of the algorithm can be bounded as follows: We run $\mathcal{O}(\log(nC))$ tests on scaled instances, each of which needs time $\mathcal{O}(n^3/\epsilon^2)$ time. Thus, the total running time is $\mathcal{O}(\log(nC)n^3/\epsilon^2)$, which is bounded by a polynomial in the input size and $1/\epsilon$. □

7. Approximation Algorithm on General Graphs

In this section we use a linear programming relaxation in conjunction with filtering techniques (cf. [19]) to design an approximation algorithm. The algorithm and its analysis are very similar to those given in [25] for the *Traveling Purchaser Problem*. The basic outline of our algorithm is as follows:

1. Formulate BCCMEDas an *integer linear program (IP)*.
2. Solve the *linear programming relaxation (LP)* of this (IP).
3. With the help of the optimal fractional solution define a *service-cluster* for each vertex. The goal is to service each vertex by one node from its cluster.
4. Solve a Group Steiner Tree problem on the clusters to obtain a tree.

7.1. Integer Linear Programming Formulation and Relaxation.

In the following we assume again that there is one node r (the root) that must be included in the tree. This assumption is without loss of generality. Consider the following integer

linear program (IP) which we will show to be a relaxation of BCCMED. The meaning of the binary decision variables is as follows: $z_e = 1$ if and only if edge e is included in the tree; furthermore $x_{vw} = 1$ if and only if vertex w is serviced by v.

$$\text{(IP)} \qquad \min \sum_{e \in E} c(e) z_e$$

$$\sum_{v \in V} x_{vw} = 1 \qquad (w \in V) \tag{4}$$

$$\sum_{v \in V} \sum_{w \in V} \text{dist}_d(v, w) x_{vw} \leq B \tag{5}$$

$$\sum_{v \notin S} x_{vw} + \sum_{v \in S, u \notin S} z_{vu} \geq 1 \qquad (w \in V, S \subset V, r \notin S) \tag{6}$$

$$z_e \in \{0, 1\} \qquad (e \in E) \tag{7}$$

$$x_{vw} \in \{0, 1\} \qquad (v \in V, w \in V) \tag{8}$$

The constraints (4) ensure that every vertex is serviced, constraint (5) enforces the budget-constraint on the service distance. Inequalities (6) are a relaxation of the connectivity and service requirements: For each vertex w and each subset S which does not contain the root r either w is serviced by a node in $V \setminus S$ (this is expressed by the first term) or there must be a an edge of T crossing the cut induced by S (this is expressed by the second term).

The linear programming relaxation (LP) of (IP) is obtained by replacing the integrality constraints (7) and (8) by the constraints $z_e \in [0, 1]$ $(e \in E)$ and $x_{vw} \in [0, 1]$ $(v \in V, w \in V)$.

Lemma 7.1. *The relaxation (LP) of (IP) can be solved in polynomial time.*

Proof. We show that there is a polynomial time separation oracle for the constraints (6). Using the result from [15] implies the claim.

Suppose that (x, z) is a solution to be tested for satisfying the constraints (6) for a fixed w. We set up a complete graph with edge capacities z_{vu} $(u, v \in U)$. We then add a new node $\tilde{w}$ and edges $(\tilde{w}, v)$ of capacity x_{wv} for all $v \in V$. It is now easy to see that there exists a cut separating r and $\tilde{w}$ of capacity less than one if and only if constraints (6) are violated for w. □

7.2. **Service Clusters and Group Steiner Tree Construction.** Let $\epsilon > 0$. Denote by $(\hat{x}, \hat{z})$ the optimal fractional solution of (LP) and by $Z_{\text{LP}} := \sum_{e \in E} c(e) \hat{z}_e$ the optimal objective function value. For each vertex $w \in V$ define the value

$$B_w := \sum_{v \in V} \text{dist}_d(v, w) \hat{x}_{vw}$$

and the subset (service cluster)

$$G_w(\epsilon) := \{\, v \in V : \text{dist}_d(v, w) \leq (1 + \epsilon) B_w \,\}.$$

The value B_w is the contribution of vertex w to the total service cost in the optimal fractional solution of the linear program. The set $G_w(\epsilon)$ consists of all those vertices that are "sufficiently close" to w.

Lemma 7.2. *For each $w \in V$ we have $\sum_{v \in G_w(\epsilon)} \hat{x}_{vw} \geq \epsilon/(1 + \epsilon)$.*

Proof. If the claim is false for $w \in V$ then we have $\sum_{v \notin G_w(\epsilon)} \hat{x}_{vw} > 1 - \epsilon/(1+\epsilon) = 1/(1+\epsilon)$. Thus

$$B_w = \sum_{v \in V} \text{dist}_d(v,w)\hat{x}_{vw} \geq \sum_{v \notin G_w(\epsilon)} \text{dist}_d(v,w)\hat{x}_{vw} \geq (1+\epsilon)B_w \sum_{v \notin G_w(\epsilon)} \hat{x}_{vw} > B_w.$$

This is a contradiction. Hence the claim must hold. □

Next, we will use the Group Steiner Tree Problem (GST) as a subroutine to complete the solution (see Section 4.1.2 for a definition of GST). Consider the instance of GST on the graph G given in the instance of BCCMED where the groups are the sets $G_w(\epsilon)$ $(w \in V)$, and the edge weights are the c-weights. This problem is formulated as an integer linear program as follows:

$$\begin{aligned}
\text{(GST)} \qquad & \min \sum_{e \in E} c(e) z_e & \\
& \sum_{v \in S, w \notin S} z_{vw} \geq 1 & (S \subset V, r \notin S, G_w(\epsilon) \subseteq S \text{ for some } w) \\
& z_e \in \{0,1\} & (e \in E)
\end{aligned}$$

The algorithm from [7] finds a group Steiner tree of cost $\mathcal{O}(\log^3 m \log\log m)$ times $\sum_{e \in E} c(e) z^*_e = Z_{\text{LP-GST}}$, where z^*_e denotes the optimal fractional solution of the LP-relaxation (LP-GST) and $Z_{\text{LP-GST}}$ denotes its objective function value.

Lemma 7.3. *Denote by $Z_{LP\text{-}GST}$ and Z_{LP} the optimal values of the LP-relaxations of the integer linear programs (GST) and (IP), respectively. Then $Z_{LP\text{-}GST} \leq (1 + 1/\epsilon) Z_{LP}$.*

Proof. We show that the vector $\bar{z}$ defined by $\bar{z}_{vw} := (1+1/\epsilon)\hat{z}_{vw}$ is feasible for the LP-relaxation of (GST). This implies the claim of the lemma. To this end let S be an arbitrary subset such that $r \notin S$, $G_w(\epsilon) \subseteq S$ for some w and $\sum_{v \in S, w \notin S} z_{vw} < 1$. Since $(\hat{x}, \hat{z})$ is feasible for (LP), it satisfies constraint (6), i.e., $\sum_{v \notin S} \hat{x}_{vw} + \sum_{v \in S, w \notin S} \hat{z}_{vw} \geq 1$. Hence we get that

$$\begin{aligned}
\sum_{v \in S, w \notin S} \hat{z}_{vw} &\geq 1 - \sum_{v \notin S} \hat{x}_{vw} \\
&\geq 1 - \sum_{v \notin G_w(\epsilon)} \hat{x}_{vw} && \text{(since } G_w(\epsilon) \subset S\text{)} \\
&\geq 1 - \left(1 - \frac{\epsilon}{1+\epsilon}\right) && \text{(by Lemma 7.2)} \\
&= \frac{\epsilon}{1+\epsilon}.
\end{aligned}$$

Multiplying the above chain of inequalities by $1 + 1/\epsilon$ shows that $\bar{z}$ is feasible for the LP-relaxation of (GST). □

Hence we know that $Z_{\text{LP-GST}} \leq \left(1 + \frac{1}{\epsilon}\right) Z_{\text{LP}} \leq \left(1 + \frac{1}{\epsilon}\right) \mathsf{OPT}$. Here OPT denotes the optimal solution for the instance of the BCCMEDproblem. We can now use the algorithm from [7] to obtain a group Steiner tree. By the last chain of inequalities this tree is within a factor $(1 + 1/\epsilon)\mathcal{O}(\log^3 m \log\log m)$ of the optimal solution value for the instance of BCCMED while the budget constraint on the service distance is violated by a factor of at most $1 + \epsilon$:

Theorem 7.4. *For any fixed $\epsilon > 0$ there is a $(1+\epsilon, (1+1/\epsilon)\mathcal{O}(\log^3 m \log\log m))$-approximation algorithm for* BCCMED. □

Acknowledgments We thank the anonymous referees for a carefuly reading of the manuscript and pointing out a number of inconsistencies. Their comments have substantially improved the presentation. We thank Professor R. Ravi (Carnegie Mellon University) for his collaboration in early stages of this work and providing us with a copy of the paper on the traveling purchaser problem. We also acknowledge useful conversations with Professor S.S. Ravi and Professor H.B. Hunt III (SUNY Albany) and Ravi Sundaram (Akamai Technologies) during the course of writing this paper.

REFERENCES

[1] E. M. Arkin and R. Hassin, *Approximation algorithms for the geometric covering salesman problem*, Discrete Applied Mathematics **55** (1994), 197–218.

[2] S. Arora, *Polynomial-time approximation schemes for euclidean TSP and other geometric problems*, Journal of the ACM **45** (1998), no. 5, 753–782.

[3] S. Arora and G. Karakostas, *A (2+ ϵ) approximation algorithm for the k-MST problem*, Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms, 2000, pp. 754–759.

[4] S. Arora and M. Sudan, *Improved low-degree testing and its applications*, Proceedings of the 29th Annual ACM Symposium on the Theory of Computing, 1997, pp. 485–496.

[5] B. Awerbuch, Y. Bartal, and A. Fiat, *Competitive distributed file allocation*, Proceedings of the 25th Annual ACM Symposium on the Theory of Computing, 1993, pp. 164–173.

[6] Y. Bartal, A. Fiat, and Y. Rabani, *Competitive algorithms for distributed data management*, Journal of Computer and System Sciences **51** (1995), no. 3, 341–358.

[7] M. Charikar, C. Chekuri, A. Goel, and S. Guha, *Rounding via tree: Deterministic approximation algorithms for group Steiner trees and k-median*, Proceedings of the 30th Annual ACM Symposium on the Theory of Computing, 1998, pp. 114–123.

[8] J. T. Current and D. A. Schilling, *The covering salesman problem*, Transportation Science **23** (1989), 208–213.

[9] L. W. Dowdy and D. V. Foster, *Comparative models of the file assignment problem*, ACM Computing Surveys **14** (1982), no. 2, 287–313.

[10] S. Eidenbenz, C. Stamm, and P. Widmayer, *Positioning guards at fixed height above a terrain – an optimum inapproximability result*, Proceedings of the 6th Annual European Symposium on Algorithms, Lecture Notes in Computer Science, vol. 1461, Springer, 1998, pp. 187–198.

[11] U. Feige, *A threshold of* $\ln n$ *for approximating set cover*, Proceedings of the 28th Annual ACM Symposium on the Theory of Computing, 1996, pp. 314–318.

[12] M. R. Garey and D. S. Johnson, *Computers and intractability (a guide to the theory of* NP*-completeness)*, W.H. Freeman and Company, New York, 1979.

[13] N. Garg, *A 3-approximation algorithm for the minimum tree spanning k vertices*, Proceedings of the 37th Annual IEEE Symposium on the Foundations of Computer Science, 1996, pp. 302–309.

[14] N. Garg, G. Konjevod, and R. Ravi, *A polylogarithmic approximation algorithm for the group steiner tree problem*, Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms, 1998, pp. 253–259.

[15] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric algorithms and combinatorial optimization*, Springer-Verlag, Berlin Heidelberg, 1988.

[16] D. Hall, *Mathematical techniques in multisensor data fusion*, Artech House Boston, MA, 1992.

[17] K. Iwano, P. Raghavan, and H. Tamaki, *The traveling cameraman problem, with applications to automatic optical inspection*, Proceedings of the 5th International Symposium on Algorithms and Computation, Lecture Notes in Computer Science, vol. 834, Springer, 1994, pp. 29–37.

[18] S. O. Krumke, *On the approximability of location and network design problems*, Ph.D. thesis, Lehrstuhl für Informatik I, Universität Würzburg, December 1996.

[19] J. H. Lin and J. S. Vitter, *ε-approximations with minimum packing constraint violation*, Proceedings of the 24th Annual ACM Symposium on the Theory of Computing, 1992, pp. 771–781.

[20] C. Lund, N. Reingold, J. Westbrook, and D. C. K. Yan, *On-line distributed data management*, Proceedings of the 2nd Annual European Symposium on Algorithms, Lecture Notes in Computer Science, vol. 855, Springer, 1994, pp. 202–214.

[21] M. V. Marathe, R. Ravi, and R. Sundaram, *Service constrained network design problems*, Nordic Journal on Computing **3** (1996), no. 4, 367–387.

[22] ———, *Improved results for service constrained network design problems*, Network Design: Connectivity and Facilities Location (P. M. Pardalos and D. Du, eds.), AMS-DIMACS Volume Series in Discrete Mathematics and Theoretical Computer Science, vol. 40, American Mathematical Society, 1998, pp. 269–276.

[23] M. V. Marathe, R. Ravi, R. Sundaram, S. S. Ravi, D. J. Rosenkrantz, and H. B. Hunt III, *Bicriteria network design problems*, Journal of Algorithms **28** (1998), no. 1, 142–171.

[24] C. Mata and J. B. Mitchell, *Approximation algorithms for geometric tour and network design problems*, Proceedings of the 11th Annual Symposium on Computational Geometry, ACM Press, June 1995, pp. 360–369.

[25] R. Ravi and F. S. Salman, *Approximation algorithms for the traveling purchaser problem and itsvariants in network design*, Proceedings of the 7th Annual European Symposium on Algorithms, Lecture Notes in Computer Science, vol. 1643, Springer, 1999, pp. 29–40.

[26] R. Ravi, R. Sundaram, M. V. Marathe, D. J. Rosenkrantz, and S. S. Ravi, *Spanning trees short or small*, SIAM Journal on Discrete Mathematics **9** (1996), no. 2, 178–200.

[27] S. Voss, *Designing special communication networks with the traveling purchaser problem*, Proceedings of the FIRST ORSA Telecommuncation Conference, 1990, pp. 106–110.

[28] O. Wolfson and A. Milo, *The multicast policy and its relationship to replicated data placement*, ACM Transactions on Database Systems **16** (1991), no. 1, 181–205.