

---

Konrad-Zuse-Zentrum  
für Informationstechnik Berlin

Takustraße 7  
D-14195 Berlin-Dahlem  
Germany

GORAN KONJEVOD   SVEN O. KRUMKE   MADHAV MARATHE

## **Budget Constrained Minimum Cost Connected Medians**

# BUDGET CONSTRAINED MINIMUM COST CONNECTED MEDIANS

GORAN KONJEVOD<sup>1</sup>, SVEN O. KRUMKE<sup>2</sup>, AND MADHAV MARATHE<sup>3</sup>

ABSTRACT. Several practical instances of network design problems require the network to satisfy multiple constraints. In this paper, we address the *Budget Constrained Connected Median Problem*: 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 of the service distances of all the remaining nodes  $v \in V \setminus T$  to their closest neighbor in  $T$  does not exceed the specified budget  $B$ . This problem has applications in optical network design and the efficient maintenance of distributed databases.

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

## 1. INTRODUCTION AND OVERVIEW

The problem of interfacing optic and electronic networks has become an important problem in telecommunication network design [20, 21]. 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 be within a specified bound, whereas the construction cost of the backbone network should be minimized.

Problems of similar nature arise in the efficient maintenance of distributed databases [3, 4, 7, 15, 22]. Other applications of the *Budget Constrained Connected Median Problem* studied in this paper include location theory and manufacturing logistics (see [20, 21] and the references cited therein).

The above problems can be cast in a graph theoretic framework as follows: Given an undirected 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 goal is to find a subtree  $T$  of  $G$  with minimum  $c$ -cost  $c(T)$  subject to the

---

<sup>1</sup>Email: konjevod@andrew.cmu.edu Dept. of Mathematical Sciences, Carnegie Mellon University, Pittsburgh, PA 15213-3890 and Los Alamos National Laboratory. Supported by an NSF CAREER grant CCR-9625297 and DOE Contract W-7405-ENG-36.

<sup>2</sup>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)

<sup>3</sup>Email: marathe@lanl.gov. Los Alamos National Laboratory, P.O. Box 1663, MS B265, Los Alamos, NM 87545. Research supported by the Department of Energy under Contract W-7405-ENG-36.

constraint that the sum of the service distances of all the remaining nodes  $v \in V \setminus T$  to their closest neighbor in  $T$  does not exceed a specified budget  $B$ .

We study the approximability of the *Budget Constrained Connected Median Problem*. This 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 prove hardness results. Section 5 contains a fully polynomial approximation scheme on trees. An approximation algorithm for the general case is presented in Section 6.

## 2. PROBLEM DEFINITION AND PRELIMINARIES

Throughout the paper  $G = (V, E)$  denotes a finite 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 1** (Budget Constrained Connected Median Problem). An instance consists of an undirected graph  $G = (V, E)$  with two different edge-cost functions  $c$  (modeling the construction or communication cost) and  $d$  (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 each of the vertices from  $V$  is at most  $B$ , that is,

$$\text{median}_d(T) := \sum_{v \in V} \min_{u \in T} d(v, u) \leq B.$$

The problem BCCMED can be formulated within the framework developed in [13, 18]. A generic bicriteria network design problem,  $(A, B, S)$ , is defined by identifying two minimization objectives,  $-A$  and  $B$ , from a set of possible objectives, and specifying a membership requirement in a class of subgraphs,  $-S$ . The problem specifies a budget value on the first objective,  $A$ , and seeks to find a network having minimum possible value for the second objective,  $B$ , such that this network is within the budget on the first objective  $A$ . The solution network must belong to the subgraph-class  $S$ . In this framework BCCMED is stated as (total  $d$ -service distance, total  $c$ -edge cost, subtree). : the budgeted objective  $A$  is the total service distance  $\text{median}_d(T)$  with respect to the edge weights specified by  $d$ , the cost-minimization objective  $B$  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** (Bicriteria Approximation Algorithm). A polynomial time algorithm for a bicriteria problem  $(A, B, S)$  is said to have *performance*  $(\alpha, \beta)$ , if it has the following property: For any instance of  $(A, B, S)$ , the algorithm produces a solution from the subgraph class  $S$  for which the value of objective  $A$  is at most  $\alpha$  times the specified budget and the value of objective  $B$  is at most  $\beta$  times the minimum value of a solution from  $S$  that satisfies the budget constraint.

Notice that a “standard”  $c$ -approximation algorithm is a  $(1, c)$ -bicriteria approximation algorithm. A family  $\{A_\epsilon\}_\epsilon$  of approximation algorithms, is called a *fully polynomial approximation scheme* or *FPAS*, if algorithm  $A_\epsilon$  is a  $(1, 1 + \epsilon)$ -approximation algorithm and its running time is polynomial in the size of the input and  $1/\epsilon$ .

### 3. SUMMARY OF RESULTS AND RELATED WORK

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

1. BCCMED is weakly 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 strict NP-hardness results for bipartite graphs.
2. We strengthen the above hardness results for general graphs further and show that unless  $\text{NP} \subseteq \text{DTIME}(N^{\log \log N})$ , there can be no polynomial time approximation algorithm for BCCMED with a performance  $(1, (1/10 - \epsilon) \ln n)$ .

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.

**3.1. Relationship to the Traveling Purchaser problem.** The BCCMED problem is closely related to a well studied variant of the classical traveling salesperson problem called the *Traveling Purchaser Problem* (see [20] 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 [20] aims at finding a minimum cost tour subject to a budget constraint on the purchasing costs.

It is easy to see that a  $(\alpha, \beta)$ -approximation algorithm for the budgeted traveling purchaser problem implies a  $(\alpha, 2\beta)$ -approximation for BCCMED: 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 [20] 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 6 uses the techniques from [20] directly and improves this result.

**3.2. Related Work.** Other service-constrained minimum cost network problems have been considered in [1, 6, 12, 16, 17]. These papers consider the variant that prescribes a budget on the service distance for each node not in the tree. The goal is to find a minimum length salesperson tour (or a tree as may be the case) so that all the (customer) nodes are strictly serviced. Restrictions of the problems to geometric instances were considered in [1, 12, 19]. Finally, the problem BCCMED can be seen as a generalization of the classical *k-Median Problem*, where we require the set of medians to be connected.

### 4. HARDNESS RESULTS

**Theorem 3.** *The problem BCCMED is weakly 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 [10, Problem SP12]. Given a multiset of (not necessarily distinct) positive integers  $\{a_1, \dots, a_n\}$ , the question is whether there exists a subset  $U \subseteq \{1, \dots, n\}$  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  $\{x, x_1, \dots, x_n\}$  and  $n$  edges  $(x, x_i)$ ,  $i = 1, \dots, n$ . 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.  $\square$

Next, we prove our inapproximability results for general graphs. Before stating the hardness result we recall the definition of the MIN SET COVER problem [10, Problem SP5] and cite the hardness results from [2, 9] 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 4** (Feige [9]). *Unless  $\text{NP} \subseteq \text{DTIME}(N^{\mathcal{O}(\log \log N)})$ , 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.*  $\square$

**Theorem 5** (Arora and Sudan [2]). *There exists a constant  $\eta > 0$  such that, unless  $P = \text{NP}$ , there is no approximation algorithm for MIN SET COVER with a performance of  $\eta \ln |U|$ , where  $U$  is the set of ground elements.*  $\square$

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

**Theorem 6.** *The problem BCCMED is strongly NP-hard even on bipartite graphs. If there exists an approximation algorithm for BCCMED on bipartite graphs with performance  $\alpha(|V|) \in \mathcal{O}(\ln |V|)$ , then there exists an approximation algorithm for MIN SET COVER with performance  $\alpha(2|U| + 2|\mathcal{S}|)$ . 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  $k \in \{2, \dots, n\}$  we construct an instance  $I_k$  of BCCMED as follows: 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  $X := k \lceil \alpha(|V_k|) \rceil + 1$ . The edges between element nodes and set nodes have weight  $X$ , all other edges have weight 1. The budget on the median cost for instance  $I_k$  is set to  $B_k := |L_k| + X|U| + |\mathcal{S}| - k$ . The construction is illustrated in Figure 1.

The bipartite graph constructed for instance  $I_k$  has  $|V_k| = 2|U| + 2|\mathcal{S}| + 2 - k \leq 2(|U| + |\mathcal{S}|)$  sets. Thus,  $\alpha(|V_k|) \leq \alpha(2|U| + 2|\mathcal{S}|)$ .

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

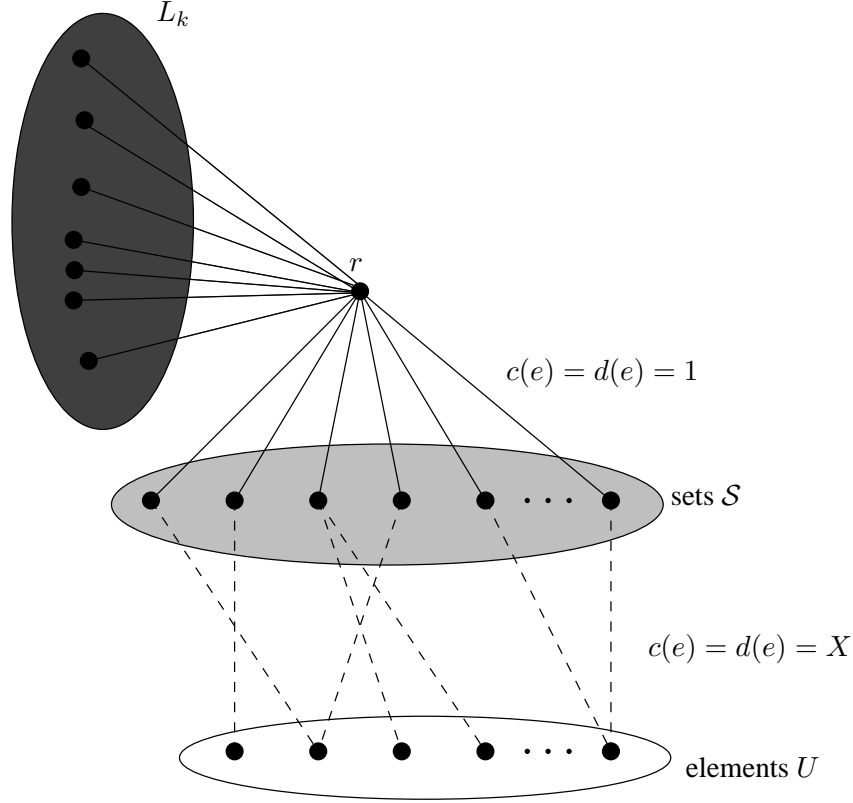


FIGURE 1. Construction used in the proof of Theorem 6. Solid edges are of weight 1, dashed edges have weight  $X = k[\alpha(|V_k|)] + 1$ .

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

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  $X$  from a vertex in the tree. Thus, the total median-cost of  $T$  is no more than  $X|U| + |\mathcal{S}| - k + |L_k| = B_k$ .

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

In fact, if this were not the case, then at least  $|L_k| - 1$  nodes from  $L_k$  can not be in  $T$  either. Moreover, these at least  $|L_k| - 1$  nodes are at distance at least two from any node in the tree. Moreover, since  $X \geq C + 1$ , the tree  $T$  can not contain any edge between set nodes and element nodes. Thus,  $T$  consists either of a single element node or  $T$  does not contain any element nodes. In the first case, the median cost of  $T$  is at least

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

which contradicts that  $T$  is feasible. In the second case, ( $T$  does not contain element nodes) either  $T$  consists of a single node from  $L_k$  or does not contain any node from  $L_k$ . Thus, we get that the median cost of  $T$  is at least

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

which is again a contradiction. (The additive terms in the above calculation stem from the fact that  $r$  is not in the tree and a set node or the remaining node from  $L_k$  is at distance at least two from the tree).

We now show that the collection  $\mathcal{C}$  of set nodes spanned by the new tree  $T'$  forms a valid set cover. Note that the size of  $\mathcal{C}$  can not exceed  $C$  since the cost  $c(T)$  of  $T$  is bounded by  $C$  and  $T$  contains the root node  $r$ . Suppose that  $u \in U$  is not covered by the sets on  $\mathcal{C}$ . As noted above, the tree  $T$  can not contain edges connecting set nodes and element nodes. Thus, the distance of  $u$  from any node in  $T$  is at least  $2X$ , whereas for all other element nodes the distance is at least  $X$ . The median cost of  $T$  thus satisfies:

$$(1) \quad \text{median}(T) \geq |L_k| + X|U| + X + |\mathcal{S}| - |T \cap (L_k \cup \mathcal{S})|.$$

Since the tree  $T$  has cost at most  $C$  and contains the root  $r$  it follows that  $T$  can contain at most  $C$  vertices from  $L_k \cup \mathcal{S}$ . Thus from (1) we get that

$$\text{median}(T) \geq |L_k| + X|U| + |\mathcal{S}| - C + X \geq |L_k| + X|U| + |\mathcal{S}| + 1 > B_k.$$

which contradicts once more the fact that  $T$  was a feasible solution of median cost at most  $B_k$ .  $\square$

The instances of MIN SET COVER used in [9] have the property that the number of sets is at most  $|U|^5$ , where  $U$  is the ground set (see [8] for an explicit computation of the number of sets used). Thus from Theorem 6 we get a lower bound for BCCMED  $(1/10 - \epsilon) \ln |V|$  (assuming that  $\text{NP} \not\subseteq \text{DTIME}(N^{\log \log N})$ ). Since the number of sets in any instance of MIN SET COVER is bounded by  $2^U$ , we can use the result from [2] to obtain a result under the weaker assumption that  $\text{P} \neq \text{NP}$ :

**Theorem 7.** (i) *Unless  $\text{NP} \subseteq \text{DTIME}(N^{\log \log N})$ , for any  $\epsilon > 0$  there can be no polynomial time approximation algorithm for BCCMED with a performance  $(1/10 - \epsilon) \ln n$ .*  
(ii) *Unless  $\text{P} = \text{NP}$ , for any  $\epsilon > 0$  there is no approximation algorithm for BCCMED with a performance of  $(1/4 - \epsilon) \ln \ln n$ .*  $\square$

## 5. 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 8.** *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 a given instance.*

*Proof.* Let  $T = (V, E)$  be the tree given in the instance  $I$  of BCCMED. 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).

Let  $T^* = (V^*, E^*)$  be an optimal solution for  $I$  which contains  $r$ . Denote by  $\text{OPT} = c(T^*)$  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 optimum cost in the final algorithm. Notice that the optimum 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  $S_i := \sum_{w \in T_{v_i} \cap U} c(w, v)$ . 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

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

We then run the dynamic programming algorithm from 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  $M \geq \text{OPT}$ . 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 optimum 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  $M \geq \text{OPT}$  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  $M \geq \text{OPT}$ , 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  $M' \leq \text{OPT}$ . Let  $T'$  be the corresponding tree found which has service cost no more than  $B$ . Then

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

Thus, the tree  $T'$  found by our algorithm has cost at most  $1 + \epsilon$  times the optimum 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  $1/\epsilon$ .  $\square$

## 6. APPROXIMATION ALGORITHM ON GENERAL GRAPHS

In this section we use a Linear Programming relaxation in conjunction with filtering techniques (cf. [14]) to design an approximation algorithm. The techniques used in this section are similar to those given in [20] for the *Traveling Purchaser Problem*.

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  $v$ . 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 an edge of  $T$  crossing the cut induced by  $S$  (this is expressed by the second term).

$$\begin{aligned}
(\text{IP}) \quad & \min \sum_{e \in E} c(e)z_e \\
(4) \quad & \sum_{v \in V} x_{vw} = 1 \quad (w \in V) \\
(5) \quad & \sum_{v \in V} \sum_{w \in V} d(v, w)x_{vw} \leq D \\
(6) \quad & \sum_{v \notin S} x_{vw} + \sum_{v \in S, u \notin S} z_{vu} \geq 1 \quad (w \in V, S \subset V, r \notin S) \\
(7) \quad & z_e \in \{0, 1\} \quad (e \in E) \\
(8) \quad & x_{vw} \in \{0, 1\} \quad (v \in V, w \in V)
\end{aligned}$$

**Lemma 9.** *The relaxation (LP) of (ILP) 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 [11] 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  $x_{vw}$  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  $w$ .  $\square$

Let  $\epsilon > 0$ . Denote by  $(\hat{x}, \hat{z})$  the optimal fractional solution of (LP). For each vertex  $w \in V$  define the value

$$D_w := \sum_{v \in V} d(v, w)\hat{x}_{vw}$$

and the subset

$$G_w(\epsilon) := \{v \in V : d(v, w) \leq (1 + \epsilon)D_w\}.$$

The value  $D_w$  is the contribution of vertex  $w$  to the total service cost in the optimum fractional solution of the Linear Program. The set  $G_w(\epsilon)$  consists of all those vertices that are “sufficiently close” to  $w$ .

**Lemma 10.** *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

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

This is a contradiction. Hence the claim must hold.  $\square$

The *Group Steiner Tree Problem* (GST) is defined as follows: Given a complete undirected graph  $G = (V, E)$  with edge weights  $c(e)$  ( $e \in E$ ) and a collection  $G_1, \dots, G_k$  of (not necessarily disjoint) subsets of  $V$ , find a subtree of  $G$  of minimum cost such that this tree contains at least one vertex from each of the groups  $G_1, \dots, G_k$ . Charikar et al. [5] gave an approximation algorithm for GST with polylogarithmic performance guarantee. We will use this algorithm as a subroutine.

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:

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

The algorithm from [5] finds a group Steiner tree of cost at most  $\mathcal{O}(\log^3 m \log \log m)$  times  $\sum_{e \in E} c(e) z_e^*$ , where  $z_e^*$  denotes the optimal fractional solution of the LP-relaxation LP-GST.

**Lemma 11.** *Denote by  $Z_{\text{LP-GST}}$  the optimal value of the LP-relaxation of the Integer Linear Program (GST). Then  $Z_{\text{LP-GST}} \leq (1 + 1/\epsilon) Z_{\text{LP}}$ .*

*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  $r \notin V$ ,  $G_w(\epsilon) \subseteq S$  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

$$\begin{aligned}
\sum_{v \in S, w \notin S} \hat{z}_{vw} & \geq 1 - \sum_{v \notin S} \hat{x}_{vw} \\
& \geq 1 - \sum_{v \notin G_w(\epsilon)} \hat{x}_{vw} && (\text{since } G_w(\epsilon) \subset S) \\
& \geq 1 - \left(1 - \frac{\epsilon}{1 + \epsilon}\right) && (\text{by Lemma 10}) \\
& = \frac{\epsilon}{1 + \epsilon}.
\end{aligned}$$

Multiplying the above chain of inequalities by  $1 + 1/\epsilon$  yields the claim.  $\square$

Hence we know that  $Z_{\text{LP-GST}} \leq (1 + \frac{1}{\epsilon}) Z_{\text{LP}} \leq (1 + \frac{1}{\epsilon}) \text{OPT}$ . We can now use the algorithm from [5] 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 12.** *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.  $\square$*

**Acknowledgements** We thank Professor R. Ravi (Carnegie Mellon University) for his collaboration in early stages of this work.

## 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 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.
- [3] 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.
- [4] 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.
- [5] 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.
- [6] J. T. Current and D. A. Schilling, *The covering salesman problem*, Transportation Science **23** (1989), 208–213.
- [7] L. W. Dowdy and D. V. Foster, *Comparative models of the file assignment problem*, ACM Computing Surveys **14** (1982), no. 2, 287–313.
- [8] S. Eidenbenz, Ch. 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.
- [9] 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.
- [10] 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.
- [11] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric algorithms and combinatorial optimization*, Springer-Verlag, Berlin Heidelberg, 1988.
- [12] 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.
- [13] S. O. Krumke, *On the approximability of location and network design problems*, Ph.D. thesis, Lehrstuhl für Informatik I, Universität Würzburg, December 1996.
- [14] J. H. Lin and J. S. Vitter,  *$\epsilon$ -approximations with minimum packing constraint violation*, Proceedings of the 24th Annual ACM Symposium on the Theory of Computing, 1992, pp. 771–781.
- [15] 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.
- [16] M. V. Marathe, R. Ravi, and R. Sundaram, *Service constrained network design problems*, Nordic Journal on Computing **3** (1996), no. 4, 367–387.
- [17] ———, *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.
- [18] 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.
- [19] 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.
- [20] R. Ravi and F. S. Salman, *Approximation algorithms for the traveling purchaser problem and its variants in network design*, Proceedings of the 7th Annual European Symposium on Algorithms, Lecture Notes in Computer Science, vol. 1643, Springer, 1999, pp. 29–40.
- [21] S. Voss, *Designing special communication networks with the traveling purchaser problem*, Proceedings of the FIRST ORSA Telecommunication Conference, 1990, pp. 106–110.
- [22] 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.