DIMACS Series in Discrete Mathematics
and Theoretical Computer Science
Volume 00, 19xx

Improved Results on Service-Constrained Network Design Problems

Madhav V. Marathe, R. Ravi, and R. Sundaram

ABSTRACT. We focus on a class of problems that combine two classical objectives in network design: establishing connectivity at low cost and satisfying location theoretic constraints. Several practical instances of network design problems often require the network to satisfy such pairs of constraints. We build on our previous work in [3] and present an improved performance guarantee and an improved inapproximability result for a general service-constrained network design problem.

1. Introduction

In this paper, we continue [3] our study of "service-constrained network design problems." Informally, service-constrained network design problems involve both a location-theoretic objective and a cost-minimization objective subject to connectivity constraints. The location-theoretic objective requires that we choose a subset of nodes at which to "locate" services such that each node is within a bounded distance from at least one chosen location. The cost-minimization objective requires that the chosen locations be connected by a network minimizing parameters such as total cost, diameter or maximum edge cost. The two objectives are measured under two (possibly) different cost functions. As mentioned in [3], these problems find applications in modeling a variety of problems in managing replicated copies of data in distributed databases, optical network design and automatic inspection of PCBs.

1991 *Mathematics Subject Classification.* 68Q25, 68Q15, 68R10.

Key words and phrases. NP-hardness, Location Theory, Network Design, Approximation Algorithms, Bicriteria problems, Spanning trees, Steiner Trees, Combinatorial algorithms.

2. Problem Statement

The prototypical problem we consider in this paper is the following: We are given an undirected graph $G = (V, E)$ with two different cost functions c (modeling the service cost) and d (modeling the construction or communication cost) for each edge $e \in E$, and a bound $\mathcal{S}_v$ (on the service distance for each vertex v). The goal is to find a minimum d-cost tree such that every node v in the graph is *serviced* by some node in the tree, i.e. every node v is within distance $\mathcal{S}_v$ (under the c-costs) of some node in the tree.

We use the bicriteria framework developed in [**4**]. 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**, under one cost function, and seeks to find a network having minimum possible value for the second objective, **B**, under another cost function, such that this network is within the budget on the first objective **A**. The solution network must belong to the subgraph-class **S**.

There are two versions of the location-theoretic or service cost objectives: (i) Non-uniform maximum service cost (denoted by **Non-uniform service cost**) and (ii) Uniform service cost (denoted by **Uniform service cost**). In the *Non-uniform service cost* version a service constraint $\mathcal{S}_{v_k}$ is specified for each vertex. The *Uniform service cost* version is a special case where $\forall v_k,\ \mathcal{S}_{v_k} = \mathcal{S}$, i.e., all vertices have the same service constraint. Thus for the problems considered in this paper $\mathbf{A} \in \{$ Non-uniform service cost, Uniform service cost $\}$. For the cost-minimization objective we focus our attention on the total cost of the network. The *Total cost* objective is the sum of the costs of all the edges in the network.

In this paper, we continue [**3**] our study of service-constrained network design problems. Many such problems are **NP**-hard. Given the hardness of finding optimal solutions, we concentrate on devising approximation algorithms with worst case performance guarantees. Recall that an approximation algorithm for an optimization problem Π provides a *performance guarantee* of ρ if for every instance I of Π, the solution value returned by the approximation algorithm is within a factor ρ of the optimal value for I. Define an (α, β)-approximation algorithm for an (**A**, **B**, **S**)-bicriteria problem as a polynomial-time algorithm that produces a solution in which the first objective (**A**) value, is at most α times the budget, and the second objective (**B**) value, is at most β times the minimum for any solution that is within the budget on **A**. The solution produced must belong to the subgraph-class **S**.

As mentioned before, the two objectives are measured with respect to different edge-cost functions. The (budgeted) service cost objective is measured using the c-cost function while the cost-minimization objective is measured using the d-cost function. As stated before, a node u is said to *service* node v if u is within distance S_v of v, under the c-cost. The *service-degree* of a node is defined to be the number of nodes that can service it. All our results come in two flavors: (i) Different cost functions and (ii) Identical cost functions. The *Identical cost functions* version is a special case of the *Different cost functions* case where the two cost functions are the same, i.e. $c_e = d_e, \forall e$. We do not address the identical cost functions case in this paper.

In the next section, we summarize our previous work in [**3**] on this problem. For a list of related results on these problems, we refer the reader to [**3**]. We then present our improvements on our previous results in the next section and close with some open questions. As described in the next section, our improved results apply to the case of the **(Non-uniform service cost, Total edge cost, Spanning Tree)** problem.

3. Previous Work

In [**3**], we presented a $(1, O(\Delta \ln n))$-approximation algorithm for the (**Non-uniform service cost, Total cost, Spanning Tree**) problem (where Δ is the maximum service-degree of any node in the graph). We counterbalanced this by showing that even the uniform service cost version of the problem does not have an (α, β)-approximation algorithm for any $\alpha < 3$ and $\beta < \ln n$ unless $\mathbf{NP} \subseteq \mathbf{DTIME}(n^{\log \log n})$.

In this paper, we improve these results as follows: we improve the performance guarantee for the (**Non-uniform service cost, Total edge cost, Spanning Tree**) problem to $(1, 2\Delta)$. We improve the non-approximability bound for the uniform service cost version of the problem to show NP-hardness of (α, β)-approximations for any $\alpha \geq 1$ and $\beta = o(\ln n)$.

4. Hardness results

First we show the improved hardness result for spanning trees under different cost functions. We use the recent results on the non-approximability of **MIN SET COVER** problem.

As an instance of the **MIN SET COVER** problem we are given a universe $Q = \{q_1, q_2, \ldots, q_n\}$ and a collection $Q_1, Q_2, \ldots, Q_m$ of subsets of Q. The problem is to find a minimum size collection of the subsets whose union is Q. Recently [**1, 5**] have independently shown the following non-approximability result.

THEOREM 4.1. *It is NP-hard to find an approximate solution to the* **MIN SET COVER** *problem, with a universe of size k, with performance guarantee better than $\Omega(\ln k)$.*

THEOREM 4.2. *Unless* **NP** $\subseteq$ **P**, *the (***Uniform service cost, Total cost, Spanning Tree***) problem, with different cost functions, cannot be approximated to within (α, β), for any $\alpha \geq 1$ and any $\beta = o(\ln n)$.*

Proof: We show that for any $\alpha \geq 1$, if there is a polynomial-time (α, β)-approximation algorithm for the (**Uniform service cost, Total cost, Spanning Tree**) problem, then there is a polynomial-time β-approximation algorithm for the **MIN SET COVER** problem.

We construct a slightly expanded version of the natural bipartite graph for set cover: we include one node for each set, and one node for every (element,set) pair where the element occurs in the set. Note that we have as many copies of an element node as the number of sets it is contained in. We include edges from a set node for s to all the element nodes represented by (element,s). Moreover, we add a clique of edges between all the nodes representing the same element in the set cover problem. In addition, we add an enforcer node with edges to all the set nodes, and a new mate node with a single edge to the enforcer node.

Recall that the d-costs are the construction costs and the c-costs are the service distances on the edges. We set the d-cost of an edge from the enforcer to a set node to be 1. The edges from a set node to the element nodes it contains have zero d-cost. The edge from the enforcer to the mate node is also given very high d-cost, say $\beta m + 1$, where m is the total number of sets. Finally all the edges between the different node copies of an element are also assigned very high d-costs.

We now specify the c-costs (service distances) for the edges. We set the c-cost for the edge between a set node and the element nodes contained in this set to be some fixed positive value, say S. We set the c-cost of all the edges from the enforcer to the set nodes to be zero. The edge between the enforcer and the mate is also assigned zero c-cost. Finally, we set of the c-costs of the edges between the different nodes representing the same element to be zero. Let G denote the resulting instance of the (**Uniform service cost, Total cost, Spanning Tree**) problem with the c and d cost functions as specified above and a uniform service budget of zero for all nodes.

It is easy to see that any collection of k subsets which form a set cover correspond to a tree in G that strictly services all the nodes and has a d-cost of k. This is because the tree consisting of the enforcer, the nodes corresponding to the sets in the collection, and the elements in these sets strictly services all the nodes – nodes corresponding to sets not in the cover are at zero service distance from the enforcer, and the covered copy of each element is at zero distance from all the uncovered copies of nodes representing elements – this tree has a d-cost of k.

Let OPT denote the size of a minimum set cover to the original instance. Now we show that if there exists a tree T which is an (α, β)-approximation to the resulting instance G of the (**Uniform service cost, Total cost, Spanning Tree**) problem, then from it we can derive a β-approximation to the original set cover instance. Such a tree T must satisfy the following properties:

1. The d-cost of T is at most $\beta \cdot \text{OPT}$. This follows from the definition of β-approximation and the fact that there exists a tree in G corresponding to OPT with d-cost at most OPT.
2. The nodes of G must be serviced by T within budget zero. This is because the c-cost of any edge is either 0 or S, but T violates the budget constraint by at most a factor α.
3. The mate node cannot be in T. This is because the d-cost of the edge from the enforcer to the mate node is $\beta \cdot m + 1$ which is greater than the d-cost of T. Since only the enforcer node can service the mate node with a service cost of zero, the enforcer must be in T.
4. Using the same reasoning as that for the mate node, none of the edges between nodes representing the same ground element can be in T. To service all such nodes, at least one node corresponding to every ground element must be in T.

We thus conclude that T consists of the enforcer node, some of the set nodes, and the element nodes corresponding to elements in these sets. Since the d-cost of T is at most $\beta \cdot$ OPT, it follows that the number of set nodes in T is at most $\beta \cdot$ OPT. Since all the element nodes are serviced by the chosen set nodes with a service distance of at most $\alpha \cdot 0 = 0$, the corresponding sets must form a set cover. We thus have a β-approximation algorithm for set cover and this completes the proof.
□

5. Different Cost Functions

In this section, we present an $(1, 2\Delta)$-approximation algorithm for the (**Non-uniform service cost, Total edge cost, Tree**) problem with different cost functions. We first recall a few basic definitions and preliminaries.

DEFINITION 5.1. *A node u is said to* **service** *a node v if u is within distance S_v of v. Call the set of nodes that can service a node u the service neighborhood of u and denote it by $N(u)$. The service-degree of a node is the number of nodes that can service it, i.e., the size of its service neighborhood. The service-degree of the graph is the maximum over all nodes of the service-degree of the node and is denoted by Δ.*

Our approximation is a simple consequence of rounding a linear programming relaxation for the problem. Before we present the LP relaxation, we recall a rounding result [**2**] for the Steiner tree problem. Given an undirected graph with nonnegative weights d on the edges, and a subset of nodes $\{t_1, t_2, \ldots, l_k\}$ called the terminals, the Steiner tree problem is to find a minimum weight tree spanning all the terminals. Consider the following integer program for the Steiner tree problem. For any node subset X, let $\delta(X)$ denote the set of edges with exactly one endpoint in X. Let us consider the problem rooted at $r = t_1$ and formulate the problem as that of finding a set of edges (choice variable x_e) that can support a flow of one unit from every other terminal t_i to the root. By the max-flow min-cut theorem, we may write this as a set of cut conditions as follows.

$$
\begin{array}{llrcll}
(ST-IP) & \text{minimize} & \sum_{e \in E} d_e x_e & & & \\
& \text{subject to} & \sum_{e \in \delta(X)} x_e & \geq & 1 & (\text{whenever } r \notin X, |X \cap T| \geq 1) \\
& & x_e & \in & \{0,1\} & (e \in E)
\end{array}
$$

The LP relaxation of the above IP is obtained by replacing the last set of constraints with $0 \le x_e \le 1 \quad (e \in E)$. Let its optimum solution have value $z_{(ST-LP)}$.

Given a weighted graph G and a subset of its nodes U, define the metric completion of U as a complete graph on the node set U with edge weights equal to the shortest distance of a path between the endpoints in G.

THEOREM 5.2 ([**2**]). *The weight of a minimum spanning tree on the metric completion of the terminal nodes is at most twice the optimum value of the LP relaxation of (ST-IP).*

We now formulate an integer program for the (**Non-uniform service cost, Total edge cost, Tree**) problem. We use a rooted formulation where a predetermined node r is always required to be in the tree solution. Note that this is without loss of generality since we may try every node in the graph as the root and use the best solution obtained over all such trials. We introduce a integer binary variable y_v for every node v in the graph that represents whether the node is in the tree or not. As before we have edge-choice variables x_e for the edges of the graph. As in the Steiner tree formumation, we continue to require that for any cut in the graph, we must choose at least one edge in a cut if it separates a node chosen to be in the tree (via y_v variables) and the root r. To model the service constraints we introduce new assignment variables z_{uv}. This is set to one whenever u is assigned to be serviced by the node v in the tree. Note that in such a case, the node v must be in the service neighborhood of u, i.e., within c-distance of S_u of u, and also v must be chosen to be in the tree, i.e., $y_v = 1$. Thus we have the following IP formulation.

$$
\begin{array}{llrcll}
(CT-IP) & \text{minimize} & \sum_{e\in E} d_e x_e & & & \\
 & \text{subject to} & \sum_{e\in\delta(X)} x_e & \ge & y_u & (\text{for all } X : r \notin X, u \in X) \\
 & & z_{uv} & \le & y_v & (\forall u, v \in V) \\
 & & \sum_{v\in N(u)} z_{uv} & = & 1, & (\forall u \in V) \\
 & & x_e & \in & \{0,1\} & (e \in E) \\
 & & y_v & \in & \{0,1\} & (v \in V) \\
 & & z_{uv} & \in & \{0,1\} & (u, v \in V).
\end{array}
$$

The LP relaxation is obtained by relaxing the last three sets of integrality constraints to linear inequalities specifying lower and upper bounds on these varibles. Let its value be denoted $z_{(CT-LP)}$.

Notice that by the assignment constraints, for any node u, we have $\sum_{v\in N(u)} y_u \ge \sum_{v\in N(u)} z_{uv} = 1$. Since the service-degree of any node is at most Δ, for any node u, we get $|N(u)| \leq \Delta$. Thus, for every node u, we must have that $\max_{v \in N(u)} y_v \geq \frac{1}{\Delta}$. By reinterpreting the fractional values y_v on the nodes as "flow" values from v to the root r, we can perform the following rounding. Define $U = \{r\} \cup \{v / y_v \geq \frac{1}{\Delta}\}$. By the previous observation, if we build a Steiner tree on these nodes, then this tree will cover every node within its service distance, since at least one node in every node's neighborhood has fractional flow at least $\frac{1}{\Delta}$ and this node is in U. We simply use a minimum spanning tree on the metric completion of U using the d-costs as the tree connecting them up, which is the final heuristic solution. To summarize, we solve the above LP, and use the solution to define U, and output the minimum d-cost spanning tree on the metric completion of U as the approximate solution.

We have already seen that the tree output is feasible for all the service constraints. To show that it has near-optimal d-cost, we make the following simple observation. Let $\tilde{z}_{(CT-LP)}$ denote the optimum value of the LP relaxation of (CT-IP). Let $\tilde{x}$ denote the induced solution on the edge variables. Notice that by the cut constraints, for any node $v \in U$, the edge values $\tilde{x}$ support a flow of at least $\frac{1}{\Delta}$ from v to the root r. Note that the value of this solution is $\tilde{z}_{(CT-LP)} = \sum_{e \in E} d_e \tilde{x}_e$. Thus, if we scale up the solution $\tilde{x}$ by a factor of Δ to get $\hat{x}_e = \Delta \cdot \tilde{x}_e$ for every $e \in E$, the scaled solution will support a flow of value at least one from every node $v \in U$ to the root r. By the max-flow min-cut theorem and the earlier formulation of the Steiner tree problem, we see that $\hat{x}$ is a feasible solution to the Steiner tree problem with the set of nodes in U as terminals. Let the value of this scaled solution be $\hat{z}_{(ST-LP)} = \sum_{e \in E} d_e \hat{x}_e = \Delta \tilde{z}_{(CT-LP)}$. By Theorem 5.2, the value of the MST on the metric completion of U is at most twice that of the optimum value of the LP formulation of this Steiner tree problem, and hence at most twice the value of any feasible solution, in particular $\hat{z}_{(CT-LP)}$. Thus the d-cost of the heuristic solution is at most $2\hat{z}_{(ST-LP)} = 2\Delta\tilde{z}_{(CT-LP)}$, which is at most 2Δ times the optimal, since the LP relaxation is a lower bound on the optimal value for the given minimization problem. Thus we have the following theorem.

THEOREM 5.3. *There is a $(1, 2\Delta)$-approximation algorithm for the (***Non-uniform service cost, Total edge cost, Tree***)-bicriteria problem with different cost functions, where $\tilde{\Delta}$ is the maximum service-degree of any node in the graph.*

Remark. Note that the bounds of Theorem 5.3 also extend to the Steiner version where only a set of terminal sites need to be serviced. The Steiner version reduces to the regular version by setting the service budgets of the nonterminal nodes to some large value, such as the diameter of the graph.

Acknowledgement: We thank Samir Khuller for pointing out the error in an earlier version of [**3**] which claimed the hardness result finally proved here. Ravi acknowledges gratefully a conversation with Petr Slavik which led to the rounding result.

Research of the first author was supported by the Department of Energy under Contract W-7405-ENG-36. The second author acknowledges support from an NSF CAREER grant CCR-9625297.

References

[1] S. Arora and M. Sudan, "Improved low-degree testing and its applications," *Proc. 29th Annual ACM Symp. on Theory of Comput.*, 485-496 (1997).

[2] M. X. Goemans and D. J. Bertsimas, "Survivable networks, linear programming relaxations and the parsimonious property," *Math. Programming*, 60, 145-166 (1993).

[3] M. V. Marathe, R. Ravi and R. Sundaram, "Service-constrained network design problems," *Nordic Journal of Computing* **3**, 367-387 (1996). A preliminary version appeared in *Proceedings of the 5th Scandinavian Workshop on Algorithms Theory (SWAT '96)*, LNCS 1097, 28-40 (1996).

[4] M. V. Marathe, R. Ravi, R. Sundaram, S. S. Ravi, D. J. Rosenkrantz and H. B. Hunt, "Bicriteria network design problems,"*Proceedings of the International Colloquium on Automata, Languages and Processing (ICALP '95)*, LNCS 944, 487-498 (1995).

[5] R. Raz and S. Safra, "A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP," *Proc. 29th Annual ACM Symp. on Theory of Comput.*, 475-484 (1997).

LOS ALAMOS NATIONAL LABORATORY P.O. BOX 1663, MS B265, LOS ALAMOS NM 87545

E-mail address: `madhav@c3.lanl.gov`

GRADUATE SCHOOL OF INDUSTRIAL ADMINISTRATION, CARNEGIE MELLON UNIVERSITY, 5000 FORBES AVENUE PITTSBURGH, PA 15213, USA

E-mail address: `ravi@cmu.edu`

PART OF THE WORK WAS DONE WHILE THE AUTHOR WAS AT MIT. CURRENT ADDRESS: DELTA GLOBAL TRADING L. P. FOUR CAMBRIDGE CENTER, CAMBRIDGE MA 02142, USA

E-mail address: `koods@delta-global.com`