# Network Design Subject to Facility Location

J. Bhadury[*], R. Chandrasekaran[†], L. Gewali[‡]


## Abstract

We consider the problem of designing a transportation network to allow the residents of the network to avail service provided by a single facility whose location is predetermined. This problem is known to be NP-Complete for general graph. The problem remains NP-Complete even when the vertices of the network are required to lie on the boundary of a convex polygon in the Euclidean plane. We present a polynomial time algorithm for computing the median constrained minimum spanning tree when the edges of the required network are not allowed to cross with each other. Our technique is based on the integer programming formulation of the network design problem.


## 1 Introduction

Algorithms for computing median of networks and sets have many applications in facility location and robust estimation [1,3,5,6,9,10,11]. Median computing algorithms have been considered in operations research, computational geometry, and graph theory. In facility location applications, it is known that a service facility located at the median of a network of customers connected by road links tend to minimize travel time.

In this paper we consider the problem of computing the minimum spanning tree of a network constrained to have median at a given vertex. The general version of this problem is known to be NP-complete [3]. The problem remains NP-complete even for vertices on Euclidean plane. We consider the non-crossing version of the problem where the edges of the spanning tree are not allowed to cross. We develop a polynomial time algorithm to obtain a non-crossing median constrained minimum spanning tree by reducing the problem to the problem of computing minimum cover for a 0/1 matrix having circular-1 property.

[*]Industrial Engineering Program, University of California, Hayward email: jbhadury@csuhayward.edu

[†]Computer Science Program, University of Texas at Dallas, Texas e-mail: chandra@utdallas.edu

[‡] Department of Computer Science, University of Nevada, Las Vegas, Nevada. e-mail: laxmi@cs.unlv.edu.

Research supported in part by grants from ARI and Bonn Corporation

## 2 Preliminaries

Consider the problem of constructing a network connecting a service facility to customers. The $n$ clusters of customers are represented by a set of $n$ vertices $V = \{v_1, v_2, v_3, ...v_n\}$. The facility itself is required to be located at one of its vertices, say $v_n$. A weight $w_i$ is associated with vertex $v_i$. In the application of our interest, the weight $w_i$ represents the number of customers at the $i^{th}$ cluster. The desired network must have path between all pair of customer sites. i.e., the desired network must be connected. The network should be such that the sum of weighted distance from facility location $v_n$ to all customer sites is as minimum as possible. Let $N(V, E')$ be the desired network. The total weighted distance from $v_n$ to all other nodes, denoted by $F(v_n, N)$, is given by :

$$F(v_n, N) = \sum_{i=1}^{n} w_i d(n, i)$$

where $d(n, i)$ is the length of the shortest path between $v_n$ and $v_i$.

**Definition 1**: A vertex that minimizes the total weighted distance to other vertices is called the *1-median* of the network. Thus $v_n$ is 1-median if

$$F(v_n, N) \leq F(v_i, N) \forall v_i \in V$$

With these notation, the median constrained minimum spanning tree can be defined as follows.

**Definition 2**: Given a graph $G(V, E)$ with weighted nodes and weighted edges and a specified vertex $v_n \in V$, the minimum spanning tree of $G(V, E)$ with $v_n$ as its median is called the *median constrained minimum spanning tree (MCMST)* of the graph.

There is a nice characterization of 1-median of a weighted tree in term of the weights of subtrees of a node. Imagine removing a vertex $v_i$ and all edges incident on it from a tree $T$. The removal breaks the tree into several smaller trees which we call *subtrees spanned by* $v_i$. Goldman [8] has given a beautiful characterization for the median of a tree in term of the weights of subtrees spanned by a vertex. This characterization is stated in the following Lemma.

*Lemma 1:* [Goldman's Characterization] Let $T_1, T_2, T_3, ...T_k$ be the subtrees of vertex $v_i$ of a tree $T$. Let $w(T_j)$ denote the weight of tree $T_j$ (sum of weights of nodes in $T_j$) and let $T_r$ be the one with the largest weight, i.e., the heaviest subtree of $v_i$. Then $v_i$ is a 1-median of $T$ if and only if $w(T_j) \leq \frac{W}{2}$, where $W$ is the weight of the tree $T$.

Solutions to many optimization problem in Euclidean graphs satisfy the non-crossing property. (It may be noted that a solution consisting of edges $E$ in the plane is said to satisfy *non-crossing property* if the edges in $E$ do not cross in their interior.) For example the shortest watchman route inside a polygon can not intersect with itself [4]. However, it turns out that the solution to MCMST problem in the Euclidean plane could have crossing edges. This observation hints that the two dimensional version of the MCMST problem could possibly be solved in polynomial time if the edges of the tree are not allowed to cross. Such a solution is referred to as the *non-crossing solution*.

## 3 Algorithm for Non-Crossing Solution

In this section we develop a polynomial time algorithm to obtain a non-crossing solution for the two dimensional Euclidean MCMST problem. We are given a set of points $v_1, v_2, v_3, ...v_n$ representing vertices in the plane. These vertices induce a complete graph. The weights of the edges of the graph is given by their Euclidean length. As before $w_i$ is the weight of vertex $v_i$. Without loss of generality, $v_n$ is taken as the vertex desired to be the median of the spanning tree. The remaining *n-1* vertices are the customer vertices. By fixing one of the customers vertex (say $v_1$) as the reference vertex, let $\angle v_i$ denote the angle $v_1 v_n v_i$. We relabel (see Figure 1) the vertices such that

$$\angle v_1 = 0 \leq \angle v_2 \leq \angle v_3 \leq ... < \angle v_{n-1}$$

**Vertex Cluster:** Based on the angular ordering of the customer vertices we define vertex cluster $V(i,j), i \leq j, i \neq n$ as the set of vertices encountered when the ray originating at $v_n$ and passing through $v_i$ is swept in the counterclockwise direction up to and including $v_j$ (see Figure 2). A vertex cluster $V(i,j)$ is called *feasible* if the total weight in the cluster is no more than half the total weight $W$ of all customer vertices, i.e. if

$$\sum_{v_k \in V(i,j)} w_k \leq \frac{W}{2}$$

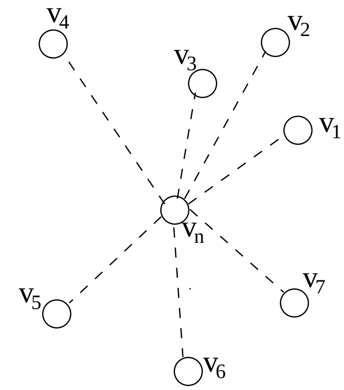


Figure 1: Labeling Vertices by Angular Order

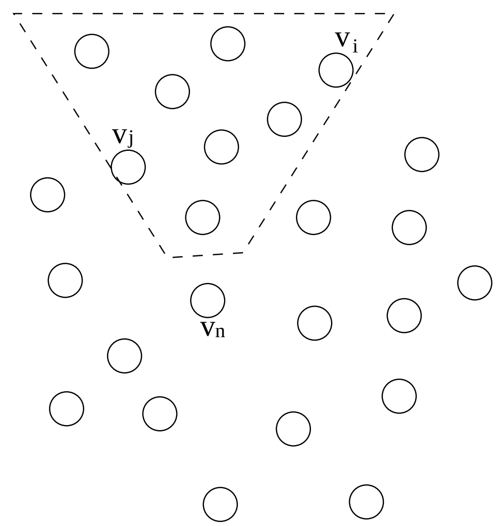


Figure 2: Illustrating Vertex Cluster V(i,j)

Let $MST(i,j)$ denote the minimum spanning tree of the feasible cluster $V(i,j)$. We need to identify feasible subtrees rooted at the planned median $v_n$. Let $v'$ be the vertex in $V(i,j)$ closest to $v_n$. Then $T(i,j)$, the feasible subtree of $v_n$ induced by $V(i,j)$ is given by adding segment $(v_n, v')$ to $MST(i,j)$, i.e., $T(i,j) = MST(i,j) \cup \{(v_n, v')\}$ (Figure 3).

Consider the angular range $\angle T(i,j)$ formed by the rays originating at $v_i$ and bounding the feasible subtree $T(i,j)$. The angular range can be written in vector form as:

$$\angle T(i,j) = \begin{bmatrix} \angle v_i \\ \angle v_j \end{bmatrix} \quad (1)$$

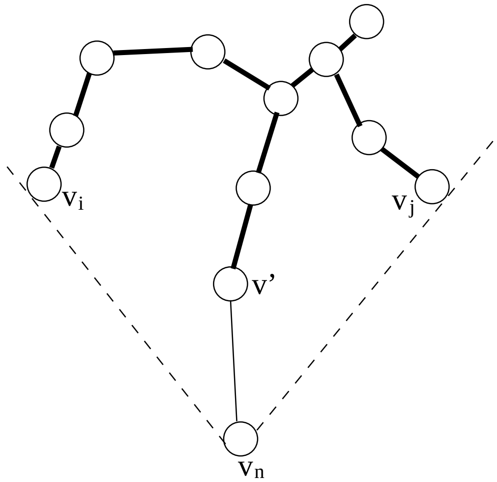


Figure 3: Constructing a Feasible Subtree

Consider all feasible subtrees rooted at $v_n$. We renumber these subtrees as $T_1^{'}$, $T_2^{'}$, $T_3^{'}$, ... $T_m^{'}$ such that

$\angle T_1^{'} \leq_{lex} \angle T_2^{'} \leq_{lex} \angle T_3^{'} \leq_{lex} \ldots \leq_{lex} Tm^{'}$

where $\leq_{lex}$ denotes *"lexicographically less than or equal to"* and $m$ is the total number of feasible subtrees rooted at $v_n$.

**Theorem 1:** A non-crossing solution for Euclidean MCMST problem can be found in polynomial time.

**Proof:** Define a 0/1 matrix $A = (a_{ij})$ of dimension $(n-1)$ *by* $m$ as follows. Row $i, 1 \leq i \leq n$, represents vertex $v_i$ and column $j, 1 \leq j \leq m$, represents feasible subtree $T_j^{'}$ rooted at $v_n$. The elements $a_{ij}$'s of matrix $A$ are defined as:

$$a_{ij} = \left\{ \begin{array}{ll} 1 & \text{if vertex i is in subtree } T_j^{'} \\ 0 & \text{otherwise} \end{array} \right\} \qquad (2)$$

Observe that the 0/1 matrix constructed in this way has *"circular 1s"* property. In other words, in every rows all ones occur in consecutive column, with the first and the last columns being considered consecutive. We introduce variables $x_1,\ x_2,\ x_3,\ \ldots\ x_j,\ \ldots\ x_m$ such that

$$x_j = \left\{ \begin{array}{ll} 1 & \text{if feasible subtree } T_j^{'} \text{is selected} \\ 0 & \text{otherwise} \end{array} \right\} \qquad (3)$$

Consider two vectors

$X = \{x_1,\ x_2,\ \ldots\ x_j,\ \ldots\ x_m\}$
$C = \{w(T_1^{'}),\ w(T_2^{'}),\ w(T_3^{'}),\ \ldots\ w(T_m^{'})\}$

Then the problem of finding non-crossing solution can be written as the following covering problem.

$$\text{Min} \sum_{j=1}^{m} CX \qquad (4)$$
$$\text{subject to } AX = 1$$

where matrix $A$ has circular 1s property.

When matrix $A$ has circular 1s property, the above covering problem can be solved in $O(n \log m \log n(n + m \log m))$ by using the algorithms developed by Orlin et. al. [2,12]. This implies that a non-crossing solution for the Euclidean MCMST problem can be solved in polynomial time. $\square$

## 4 Discussion

We presented a polynomial time algorithm for solving the median constrained minimum spanning tree problem when the edges of the tree are not allowed to cross. The algorithm is obtained by reducing the problem to the cyclic integer programming problem. These algorithms are difficult for practical implementation and it would be interesting to develop an easily implementable algorithm. When the edges are allowed to intersect the median constrained minimum spanning tree problem is NP-Hard. It would therefore be interesting to develop an approximation algorithm for the general version of the problem.

## References

[1 ] Aloupis, G., Soss, M., and G. Toussaint, "On the Computation of Bivariate Median and Fermat-Torricelli Problem for Lines,"*Proceedings of the 13th Canadian Conference on Computational Geometry,"*2001, pp. 21-24.

[2 ] Bartholdi III, J. J., J. B. Orlin, and H. D. Ratlife, "Cyclic Scheduling via Integer Programming with Circular Ones," *Operations Research*, 28, 1980, pp. 1074-1085.

[3 ] Bhadury, J., R. Chandrasekaran, and L. Gewali, "Constructing Median Constrained Minimum Spanning Tree," *Proceedings of the Tenth Canadian Conference on Computational Geometry,* 1998, pp. 22-23.

[4 ] Chin, W. P. and S. Ntafos, "Shortest Watchman Routes in Simple Polygons," *Discrete Computational Geometry*, 1991, pp. 19-31.

[5 ] Daskin M.S. "Network and Discrete Location: Models, Algorithms and Applications", John Wiley and Sons, New York, 1995

[6 ] Drezner, Z. “Facility Location: A Survey of Applications and Methods”, Springer-Verlag New York, Inc, 1995

[7 ] Garey M.R. and Johnson D.S., “Computers and Intractability: A Guide to the Theory of NP-Completeness”, W.H. Freeman and Company, New York, 1979

[8 ] Goldman A.J. (1971). ”Optimum Center Location In Simple Networks,” *Transportation Science*, 5, 1971, pp. 12-221.

[9 ] Hakimi, S.L. (1964). ”Optimum Locations of Switching Centers and the Absolute Centers and Medians of a Graph,” *Operations Research*, 12, 1964, pp.450-459.

[10 ] Hurtado, F., V. Sacristan, and G. Toussaint, “Some Constrained Minimax and Maximin Location Problems,” *Studies in Locational Analysis: Special Issue on Computational Geometry in Locational Analysis,*” Guest Editor: Juan Mesa, December 2000, pp. 17-35.

[11 ] Niinimaa, A., Oja, H., and Nyblom, J., “Algorithm AS 277: The Oja Bivariate Mean”, *Applied Statistics*, 41:611-617, 1992.

[12 ] Orlin, J. B., “A Faster Strongly Polynomial Minimum Cost Flow Algorithm,” *Proceedings of the 20th ACM Symposium on Theory of Computing,* Chicago, 377-387.