# Constructing Median Constrained Minimum Spanning Tree

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

## Abstract

The Median Constrained Minimum Spanning Tree (MCMST) problem is the determination of a minimum cost spanning tree of a given graph constrained to have a median at a specified node. We show that MCMST is NP-complete by reducing the partitioning problem to MCMST. We also present heuristics to solve MCMST when nodes are randomly distributed in the Euclidean plane. Implementation (in C++/LEDA) of a proposed heuristic yields solutions which are no more than 1.3 times the optimum solution for randomly distributed points in the Euclidean plane.

## 1 Introduction

Consider a complete graph $G(V, E)$ with node set $V = \{v_1, v_2, \ldots, v_n\}$. In the problem which we consider, both edges and nodes are assigned numbers to reflect their lengths and weights. Associated with each edge $(v_i, v_j)$ is a non-negative number $d(i, j)$ that represents its length. Similarly, associated with each node $v_i$ is a non-negative number $w_i$ that represents its weight. Let $W$ be the sum of the weights of all nodes.

Suppose we want to locate a facility on a network represented by this graph $G(V, E)$. The facility will be used by consumers residing on the nodes (the weight of each node may be viewed as the number of consumers residing there.) The objective is to locate the facility so that the total "travel disutility" is minimized. The "total disutility" of locating a facility at a point $x$, denoted by $F(x)$, is given by the total weighted distance of all nodes from $x$, i.e., $F(x) = \sum_{i=1}^{n} w_i d(x, v_i)$.

**Definition 1**: The *1-median* (or simply, the median) of a graph $G(V, E)$ is a point $x$ in the graph that minimizes $F(x)$ over $G(V, E)$. In other words, it is a point $x^*$ in $G(V, E)$ such that $F(x^*) \leq F(x)\ \forall x \in G(V, E)$.

Hakimi [1] gave the *node optimality theorem* which states that the 1-median can always be found on a node.

The main problem we address in this paper is the determination of a spanning tree of $G(V, E)$ so that (i) a given node is a 1-median of the tree and (ii) the length of the tree (sum of the length of its edges) is minimized.

**Median Constrained Minimum Spanning Tree Problem (MCMST)**
*Instance*: A graph $G(V, E)$ and a specified node $v*$.
*Question*: Determine a minimum length spanning tree of $G(V, E)$ constrained to have a median at node $v^*$.

When the nodes are points in the Euclidean plane and the weight of an edge $(v_i, v_j)$ is taken as the Euclidean distance between $v_i$ and $v_j$ we get the Euclidean Median Constrained Minimum Spanning Tree problem (EMCMST).

Goldman [2] has given a rather pleasing condition to check whether a node of a tree is a 1-median or not. His condition can be described in term of the subtree spanned by a vertex. 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* $v_i$.

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

## 2 Complexity of MCMST

**2-Partitioning Problem (2-P)**
*Instance:* Set $A$ of $n$ numbers $a_1, a_2, \ldots, a_n$
*Question:* Does there exist a subset $B$ of $A$ such that $\sum_{a_i \in B} a_i = \sum_{a_i \in A \setminus B} a_i$.

[*]University of New Brunswick, Fredericton, NB. email: bhadury@unb.ca. Research supported in part by a grant from NSERC

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

[‡]University of Nevada, Las Vegas, Nevada, e-mail: laxmi@cs.unlv.edu. Research supported in part by a grant from DOE

From an instance of 2-P we construct in polynomial time an Euclidean graph $G'(V', E')$ that satisfies the following lemma.

**Lemma 2** *There exists a 2-partition of $A$ if and only if there exists a minimum spanning tree of $G'(V', E')$ with vertex $v'_{n+1}$ as median and total length $\leq U_2$, where $U_2$ is a quantity determined by the construction of $G'(V', E')$.*

**Proof**: (omitted)

**Theorem 1** *EMCST is NP-Complete.*

# 3 Heuristics

The above NP-complete result motivates us to look for heuristics that can be used to generate approximate solutions for EMCMST. We propose two heuristics for nodes distributed in Euclidean plane: (a) The Sweep-Connect Heuristic and (b) The Connect-Split Heuristic.

Consider $n$ point nodes distributed in the Euclidean plane. We refer to such points as "sites" and denote them by $s_1, s_2, s_3, \ldots, s_n$. As before, let $w(i)$ be the weight associated with each site $s_i$. These sites induce the complete graph, where each pair of sites $s_i$ and $s_j$ are connected by line segment to form edge $e(i, j)$. The length of edge $e(i, j)$ is taken as the Euclidean distance $d(i, j)$ between them. One of the advantages of working with points in two dimensions is that we need not consider the complete graph to find a minimum spanning tree connecting the sites. (It is known that the minimum spanning tree of points in two dimensional Euclidean plane is a subset of the delaunay triangulation.)

## 3.1 The Sweep-Connect Heuristic

We partition the plane into three wedges anchored at the central-site $s^*$, where the facility is to be located. The partitioning is done in such a way that the total weight of the sites in each wedge is no more than $\frac{W}{3}$. We refer to such a partitioning as *weight balanced partitioning*. Weight balanced partitioning can be easily obtained by sweeping the plane by a ray $r$ originating at $s^*$ and rotating (say) in the counterclockwise direction. During the sweep wedges are constructed by keeping track of the total weight of the sites encountered by the ray. Minimum spanning tree of sites in each wedge (excepting $s^*$) is determined next. (We refer to the minimum spanning tree in each wedge as *wedge-tree*). The final tree $T^*$ is formed by connecting $s^*$ to respective nearest site in each wedge-tree.

**Lemma 3** *The sweep-connect heuristic executes in $O(n \log n)$ time and generates a feasible spanning tree whose length is no more than three times the length of the optimum tree.*

## 3.2 The Connect-Split Heuristic

In this approach we first determine the minimum spanning tree $T^m$ of the input sites. If $s^*$ is found to be the median of $T^m$ then we have the optimal solution. Otherwise, the heaviest subtree of $T^m$ must have weight greater than $\frac{W}{2}$. Let $T^h$ denote the heaviest subtree. In order to make $s^*$ a median we need to split $T^h$ into two subtrees of comparable size. This can be done by processing the sites of $T^h$ separately. The sites of $T^h$ are partitioned into two balanced wedges and then separately connected to form spanning trees $T_1^h$ and $T_2^h$ as done in the sweep-connect heurictic. The final tree is obtained by connecting $s^*$ to nearest sites in $T_1^h$ and $T_2^h$.

## 3.3 Experimental Result

We implemented the sweep-connect heuristic in C++/LEDA running in Silicon Graphics O2 workstation. The location of input sites were randomly generated and the weight of each site was a randomly picked number between 1 and 100. The delaunay triangulation of the input sites is determined by using the delaunay function provided in LEDA. In order to measure the quality of the generated solution we compared the length $l^*$ of the generated tree with the length $l^m$ of the corresponding minimum spanning tree. This is valid since $l^*$ is a lower bound of the length of the optimum tree. The approximation factor $f$ is calculated as $\frac{l^* - l^m}{l^*} 100$. The experiment was conducted for several input size ($n = 50, 100, 150, \ldots, 800$) and the generated solution is observed to be within thirty percent of the optimum solution.

# References


[1] S. L. Hakimi, " Locations with Spatial Interaction", in Discrete Location Theory, P. B. Mirchandani and R. L. Francis (eds.), John Wiley and Sons, 1990.

[2] A. J. Goldman, "Optimal Simple Location in Simple Networks", Transportation Science, 5, pp. 212-221, (1971).

[3] F. P. Preparata and M. I. Shamos, "Computational Geometry: An Introduction", Springer Verlag, 1985.