THE COMPLEXITY OF MINIMIZING CERTAIN COST METRICS FOR K-SOURCE SPANNING TREES

by

HAROLD SCOTT CONNAMACHER

A THESIS

Presented to the Department of Computer
and Information Science
and the Graduate School of the University of Oregon
in partial fulfillment of the requirements
for the degree of
Master of Science

June 2000

"The Complexity of Minimizing Certain Cost Metrics for k-Source Spanning Trees," a thesis prepared by Harold Scott Connamacher in partial fulfillment of the requirements for the Master of Science degree in the Department of Computer and Information Science. This thesis has been approved and accepted by:

Andrzej Proskurowski

Dr. Andrzej Proskurowski

June 1, 2000

Date

Accepted by:

Marian Friestad

Dean of the Graduate School

An Abstract of the Thesis of

Harold Scott Connamacher for the degree of Master of Science

in the Department of Computer and Information Science

to be taken June 2000

Title: THE COMPLEXITY OF MINIMIZING CERTAIN COST METRICS FOR K-SOURCE SPANNING TREES

Approved: Andrzej Proskurowski

Dr. Andrzej Proskurowski

This thesis investigates multi-source spanning tree problems where, given a graph with edge weights and a subset of the nodes defined as sources, the object is to find a spanning tree of the graph that minimizes some distance related cost metric. This problem can be used to model multicasting in a network where messages are sent from a collection of senders, and the goal is to reach every receiver within minimum total cost. In this model, it is assumed that communication takes place along the edges of a single spanning tree. For a set of possible cost metrics for creating such a spanning tree, this thesis determines whether the problem is NP-hard; otherwise, it demonstrates the existence of an efficient algorithm to find an optimal tree.

CURRICULUM VITA

NAME OF AUTHOR: Harold Scott Connamacher

PLACE OF BIRTH: Pittsburgh, Pennsylvania

DATE OF BIRTH: July 26, 1969

GRADUATE AND UNDERGRADUATE SCHOOLS ATTENDED:

University of Oregon
State University of New York at Buffalo
Oberlin College

DEGREES AWARDED:

Master of Science in Computer and Information Science, 2000
University of Oregon
Bachelor of Arts in Computer Science, 1991, Oberlin College

AREAS OF SPECIAL INTEREST:

Theory of Computer Science
Applied Discrete Mathematics

PROFESSIONAL EXPERIENCE:

Teaching Assistant, Department of Computer and Information Science, University of Oregon, Eugene, 1999–2000

Senior Software Engineer, SunGard Data Systems, Eugene, 1997–2000

Software Engineer, ISTR Incorporated, Buffalo, 1992–1997

AWARDS AND HONORS:

Sixth Place, ACM International Programming Competition, Pacific Northwest Region, 1999

ACKNOWLEDGEMENTS

First, I wish to sincerely thank my advisor, Professor Andrzej Proskurowski, who not only presented me with these problems but also gave countless hours of advice, encouragement, and guidance throughout my entire studies at the University of Oregon.

I also wish to thank Professor Art Farley for his insight on a few of the problems, and I would like to thank John Fiskio-Lasseter and Shasta Willson who gave assistance to a new Latex user and Bart Massey and Takunari Miyazaki for their wonderful Latex templates.

Finally, I wish to thank my wife, Celeste, for proofreading this manuscript and for supporting me as I chase my dreams.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER I

THE K-SOURCE SPANNING TREE PROBLEM

A Brief Introduction

The motivation behind this thesis is the problem of multicasting in which a message is broadcast to multiple receivers across a network. One possible paradigm of multicasting is when several sources from a fixed set of vertices transmit the data, and every vertex in the network is a potential receiver. Multicast protocols often use a single routing tree which is shared by all transmissions. The goal is to minimize the time it takes to complete a message broadcast, and this thesis examines the feasibility of constructing optimal routing trees for such a protocol. The optimality of a tree is determined by minimizing some given cost function. Multiple cost metrics are considered because different situations may call for different requirements and because some of the metrics turn out to be intractable.

If there is only one source, an algorithm to find the single source shortest paths spanning tree, such as Dijkstra's or Bellman-Ford, will produce an optimal tree for each of the cost metrics considered. Therefore, the investigation considers only situations with more than one source. The problem is interesting because a shortest paths tree from one of the sources will probably not yield good results when used in conjunction with the other sources.

An obvious cost metric is to consider the total sum of the distances from each source to every vertex. With this cost function, this problem is an instance of the more general Optimum Communication Spanning Tree (cf. [ND7] in [2]) as defined in [3]. Also, if every vertex is a source, this problem becomes the Shortest Total Path Length Spanning Tree (cf. [ND3] in [2]). Unfortunately, both these problems were proven $\mathcal{NP}$-hard in [4], and even in the case of two sources and uniform edge weights, finding an optimal tree which minimizes this cost metric is still intractable. [1]

This thesis presents six distance related cost metrics and the problem of constructing an optimal spanning tree minimizing each of the metrics. One of the problems is known to be $\mathcal{NP}$-hard and another is known to have a polynomial time solution. [6] After the presentation of these results regarding the complexity status of the two problems and a strategy for proving that a spanning tree problem is polynomial, this thesis considers the four problems of, until now, unknown complexity status. Two of these are proven to be $\mathcal{NP}$-hard and two are proven to be solvable in polynomial time.

A Few Definitions

A graph is a pair $G = (V, E)$ where V is the set of vertices, also called nodes, and $E \subseteq V \times V$ is the set of edges. There is a length function defined on the edges, $l : E \rightarrow \Re$. This thesis will also make use of points where a point may be either a vertex of G or a location along an edge of G. The sources of a graph are a nonempty subset of the vertices.

A spanning tree T of G is a connected acyclic graph which connects all vertices of G using a subset of the edges of G.

The distance function, $d : V \times V \rightarrow \Re$, on nodes u and v is the sum of the lengths of each edge on the path from u to v. Depending on the set of edges considered, we distinguish between the tree distance $d_T(u, v)$ which is the distance of the unique path in T from u to v and the graph distance $d_G(u, v)$ which is defined to be minimum distance from u to v over all possible paths from u to v in G.

The Problems

The problems investigated in this thesis are generalized as a family of decision problems called k-Source Minimum Spanning Tree with Cost-i (k-SMSTi), parameterized by a positive integer k and given a cost metric $cost_i$. The basis of all the metrics is the tree distance between sources and vertices, and the operations combining the different distances are max and sum.

k-SMSTi:

Instance: A graph $G = (V, E)$ with a length function, $l : E \rightarrow \Re$, k sources $S = \{s_1, \ldots, s_k\} \subseteq V$, a positive integer K.

Question: Is there a spanning tree T of G such that $cost_i(T) \leq K$?

The cost metrics are defined as:

$$
\begin{aligned}
cost_1(T) &= \sum_{s \in S} \sum_{v \in V} d_T(s, v) \\
cost_2(T) &= \max_{v \in V} \sum_{s \in S} d_T(s, v) \\
cost_3(T) &= \max_{s \in S} \sum_{v \in V} d_T(s, v) \\
cost_4(T) &= \sum_{v \in V} \max_{s \in S} d_T(s, v) \\
cost_5(T) &= \sum_{s \in S} \max_{v \in V} d_T(s, v) \\
cost_6(T) &= \max_{s \in S, v \in V} d_T(s, v).
\end{aligned}
$$

It should be noted that each of these problems is in $\mathcal{NP}$ as one can simply guess a spanning tree and, in polynomial time, calculate the appropriate cost metric.

The first of these metrics, k-SMST1, was proven to be $\mathcal{NP}$-complete in [1], and the last, k-SMST6, was shown to be polynomial in [5] and [6]. The remaining four metrics were presented as open problems in [6], and this thesis completely characterizes the complexity status of each of the four remaining problems, k-SMST2 through k-SMST5.

Known Results

The k-SMST1 Problem

The optimization version of the k-SMST1 problem seeks to minimize the total cost from every source to every vertex. This problem is also known as k-Source Shortest Paths Spanning Tree (k-SPST) and was proven $\mathcal{NP}$-complete in [1]. The proof is a reduction from the 3-SAT decision problem to the 2-SPST decision problem. The intractability of the more general k-SPST problem follows naturally.

An instance of 3-SAT (problem [LO2] in [2]) is a set of disjunctive clauses $(C_i, \ldots, C_m)$ involving literals of the variables $(x_1, \ldots, x_n)$, and we are asked whether there is a truth setting of the variables which satisfies every clause. The reduction involves constructing a variable gadget for each x_i. The gadget is a four-cycle with the vertices labeled, in order, $x_i^{'}$, x_i^T, $x_i^{''}$, and x_i^F. The variable gadgets are linked together in a lattice chain so that $x_i^{''} = x_{i+1}^{'}$. Each clause C_j is represented by a vertex c_j, and for each literal in C_j, there is a unique path of $n-1$ vertices from c_j to the appropriate literal vertex x_i^T or x_i^F. Finally, additional nodes are attached to one of the sources and to each clause vertex. These nodes serve to weight the graph and force an optimal tree to have a specific shape. See Figure 1 for an example of the reduction graph.

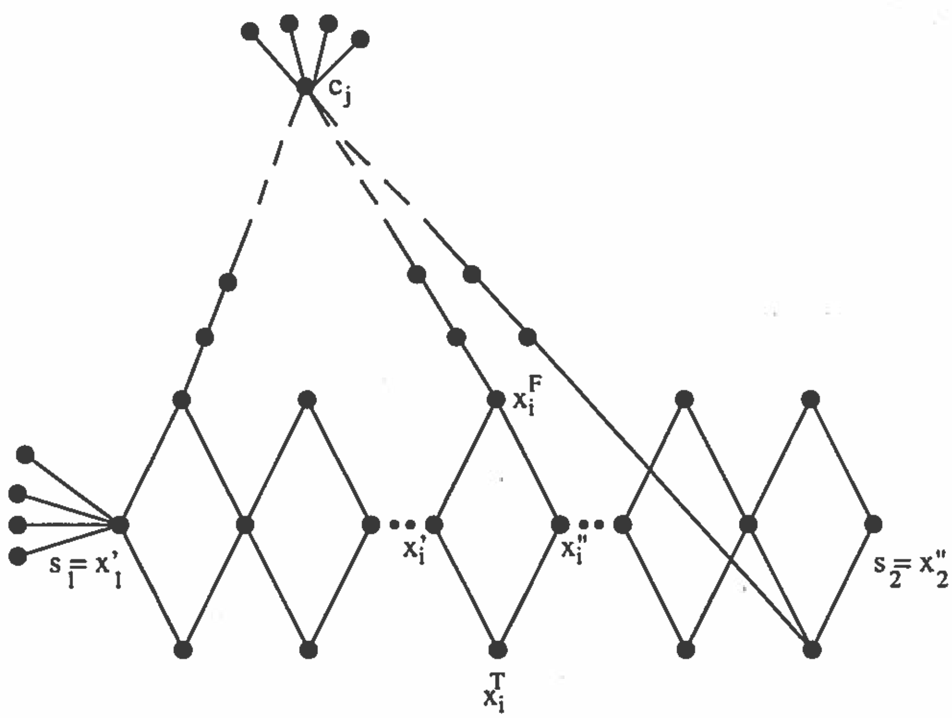


FIGURE 1. The construction for k-SMST1 (k-SPST); a clause $c_j = \neg x_1 \vee \neg x_i \vee x_n$.

The general idea is that by adjusting the number of vertices attached to each clause vertex and to the source vertex, we force the path linking the two sources in the tree to traverse the literal lattice and not include any clause vertex. As exactly one of each pair of nodes $\{x_i, \neg x_i\}$ will be on this path, the nodes included on the intrasource path correspond to a truth assignment to the variables. A clause vertex is linked to the intrasource path by its literals. If a clause has a true literal, the clause vertex is a shorter distance from the intrasource path than if all the clause literals are false. As the cost of the tree depends on the distance of the clause vertices from the path, the bound K on the cost of the tree is set so that an optimal tree exists only if every clause is satisfied by the truth assignment. A similar reduction is used to prove k-SMST2 and k-SMST3 are $\mathcal{NP}$-complete.

The k-SMST6 Problem

A solution to the k-SMST6 problem minimizes the source eccentricity of a spanning tree, defined to be the maximum distance between a source vertex and any other vertex. This problem is also known in the literature as the k-Source Maximum Eccentricity Spanning Tree (k-MEST) problem [1, 6] and the Minimum Eccentricity Multicast Tree (MEMT) problem. [5] An efficient solution exists for this problem, and [6] presents an $\mathcal{O}\left(|V|^3 + |E||V|\log|V|\right)$ algorithm while [5] presents an $\mathcal{O}\left(|V|^3\right)$ algorithm.

A Sufficient Set of Shortest Paths Spanning Trees

Given the general problem of finding spanning trees for a graph, if we can prove that a shortest paths spanning tree from some point on the graph will solve the problem, then a polynomial solution to the problem will exist.

Let Q be a set of spanning trees of G such that, for all points ϕ of G, Q contains a single source shortest paths spanning tree (SPST) from ϕ. An important result, for this thesis, from [6] is that we can construct Q in polynomial time. The key idea is that although there are an infinite number of points on a graph, we only need to construct a shortest paths tree from a polynomially bounded subset of these points. Therefore, to prove a problem is in $\mathcal{P}$, it suffices to show that a SPST from some point ϕ is optimal for the problem. Although this fact does not directly lead to efficient algorithms, it does at least present us with a naive polynomial time solution which is to generate a SPST from every necessary point and then choose the tree which has minimum cost. Because this result is crucial to the proofs given later, it is described in full here with two theorems of McMahan and Proskurowski. [6]

First, for each edge (p, q) in G, define a set of points $\Gamma_{(p,q)}$. For each vertex $v \in V$, let $\gamma_v \in \Gamma_{(p,q)}$ be a point on the edge (p, q) so that for any point $\alpha \in (p, \gamma_v)$ the shortest path from α to v is through the vertex p, and likewise for any point $\alpha \in (\gamma_v, q)$, the shortest path from α to v is through the vertex q. Each of these points can be located in polynomial time. Let $d_p(\gamma_v)$ be the distance along the edge (p, q) from p to γ_v. Then,

$$d_p(\gamma_v) = \frac{1}{2}(d_G(q, v) - d_G(p, v) + l(p, q)).$$

For the next two theorems, consider a set $\Gamma_{(p,q)}$ and index the vertices of G so that $d_p(\gamma_{v_1}), d_p(\gamma_{v_2}), \ldots, d_p(\gamma_{v_n})$ is a nondecreasing sequence and consider the $|V| + 1$ intervals $(\gamma_{v_i}, \gamma_{v_{i+1}})$.

<u>Theorem</u> 1.4.1. For any two points α_1 and α_2 in the interval $(\gamma_{v_i}, \gamma_{v_{i+1}})$ for $1 \leq i < n$, the set of SPSTs rooted at α_1 is the same as the set of SPSTs rooted at α_2.

Proof. For both α_1 and α_2, the shortest path to a vertex v_j goes through p if $j \leq i$ and through q if $j > i$. Thus, a SPST from either α_1 or α_2 contains a shortest paths tree from p to the vertices $v_1, \ldots, v_i$ and a shortest paths tree from q to the vertices $v_{i+i}, \ldots, v_n$. Therefore, the sets of all SPSTs from α_1 is identical to the set of all SPSTs from α_2. □

Theorem 1.4.2. Any SPST for a point $\alpha \in (\gamma_{v_{i-1}}, \gamma_{v_{i+1}})$ is also a SPST for the point γ_{v_i}.

Proof. The proof follows from Theorem 1.4.1 and the definition of γ_v. □

Therefore, to create Q, pick an arbitrary point $\alpha \in (\gamma_{v_i}, \gamma_{v_{i+1}})$ for each interval $(\gamma_{v_i}, \gamma_{v_{i+1}})$ along an edge, find a SPST from α, and repeat the process for each edge in G. As there are at most $|V| + 1$ intervals per edge, Q can be constructed in polynomial time. More efficient methods for forming Q are possible. Both [6] and [7] give such procedures, but these techniques are immaterial for the scope of this thesis.

CHAPTER II

THE NP-COMPLETE PROBLEMS

The k-SMST2 Problem

Instance: A graph $G = (V, E)$ with a length function, $l : E \to \Re$, k sources $S = \{s_1, \ldots, s_k\} \subseteq V$, a positive integer K.

Question: Is there a spanning tree T of G such that

$cost_2(T) = \max_{v \in V} \sum_{s \in S} d_T(s, v) \leq K$?

This metric minimizes the sum of the distance to each source from the farthest vertex in the tree. Unfortunately, this problem is $\mathcal{NP}$-hard. The proof is a reduction from 3-SAT and closely follows the proof of k-SMST1 in [1] (cf. p. 4).

Theorem 2.1.1. 2-SMST2 is $\mathcal{NP}$-complete even for graphs with unit edge lengths.

Proof. The proof is a reduction from 3-SAT. Given an instance of 3-SAT with m clauses $(C_1, \ldots, C_m)$ and n variables $(x_1, \ldots, x_n)$, construct a graph G. For each variable, x_i, create a 4-cycle gadget with vertices labeled, in order, $x_i^{'}$, x_i^T, $x_i^{''}$, x_i^F. We will connect these 4-cycles in a lattice such that $x_i^{''} = x_{i+1}^{'}$ for $i = 1 \ldots n-1$. For each clause, C_j, create a vertex labeled c_j, and for each of the three literals in the clause, connect the clause vertex to the associated variable gadget by a chain of n nodes so that if clause C_j contains the literal x_i, the chain will connect vertex c_j to vertex x_i^T and if clause C_j contains the literal $\neg x_i$, the chain will connect vertex c_j to vertex x_i^F. Finally,

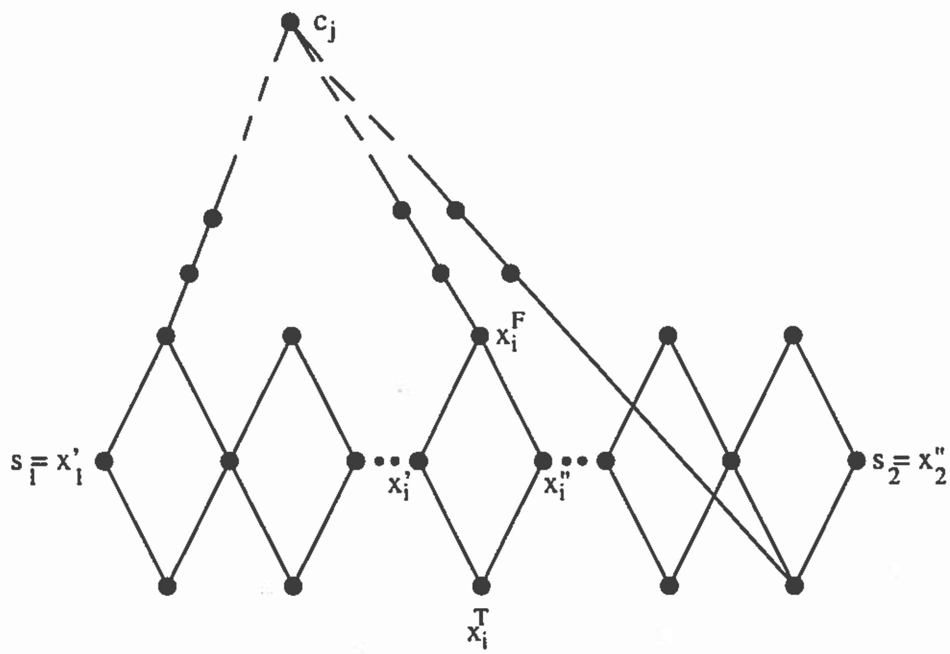


FIGURE 2. The construction for k-SMST2; a clause $C_j = \neg x_1 \vee \neg x_i \vee x_n$.

let $S = \{s_1, s_2\}$ with $s_1 = x_1^{'}$ and $s_2 = x_n^{''}$, and let $K = 4n + 2$. See Figure 2 for an example of the reduction graph.

This graph can be constructed in polynomial time because the lattice of variable gadgets has $3n+1$ vertices and $4n$ edges and each clause vertex is connected to the lattice by three paths of n vertices and $n + 1$ edges. So there is a total of $3n + 1 + m(3n + 1)$ vertices and $4n + 3m(n + 1)$ edges in G.

The instance of 3-SAT is satisfiable if and only if G has a spanning tree T with $cost_2(T) <= K$. If we have a satisfying assignment to the instance of 3-SAT, we can construct an optimal tree T. The path between s_1 and s_2 in T will traverse the variable gadget lattice according to the variable truth assignment. If x_i is assigned true, x_i^T will be on the path, and likewise if x_i is assigned false, x_i^F will be on the path. The tree is completed by noting which literal is critical to satisfying each clause C_j and removing the edges from vertex c_j to the noncritical literals of C_j. Finally, the extra edges in the lattice are removed. Define the cost of a vertex to be the sum of the distance to each source and note that the cost of a vertex is equal to twice the distance of the vertex from the s_1s_2-path plus the length of the s_1s_2-path. Each lattice vertex not on the path is at distance one from the path, each vertex in a chain between the clause vertex and the lattice is at distance at most $n+1$ from the intrasource path, and each clause vertex is at distance $n+1$ from the path. Therefore, $cost_2(T) = 4n+2 = K$.

Likewise, if we find an optimal tree for G, we can construct a satisfying 3-SAT assignment by noting which literals lie along the s_1s_2-path and setting each variable's truth value accordingly. This is possible because if G has a spanning tree T with $cost_2(T) <= K$, the path between s_1 and s_2 can not contain any of the clause vertices, and the nodes on the path must correspond to a satisfying assignment for the C_js. If we allow the path to contain two or more clause vertices, then the length of the s_1s_2-path will be at least $4n+6$, and if we allow the intrasource path to contain exactly one clause vertex, then the length of the path will be at least $2n+4$. In this case, the cost for some other clause vertex will be at least $4n+6$ because that vertex is at distance $n+1$ from the intrasource path, and so the cost of the tree will be at least $4n+6$. Thus, no clause vertex can be on the path from s_1 to s_2. Now, assume the tree does not correspond to a satisfying assignment for the C_js. Then, by the way G was constructed, some clause vertex must be at a distance $n+2$ from the intrasource path, and, thus, $cost_2(T) = 4n+6 > K$. Therefore, G has an optimal tree if and only if there is a satisfying assignment to the x_is. □

Corollary 2.1.2. k-SMST2 $\in \mathcal{NP}$-complete.

The k-SMST3 Problem

Instance: A graph $G = (V, E)$ with a length function, $l : E \rightarrow \Re$, k sources $S = \{s_1, \ldots, s_k\} \subseteq V$, a positive integer K.

Question: Is there a spanning tree T of G such that

$cost_3(T) = \max_{s \in S} \sum_{v \in V} d_T(s, v) \leq K$?

This problem is also $\mathcal{NP}$-hard. The proof uses a similar technique to the proof of Theorem 2.1.1, but in this case the reduction is from Exact Cover By 3-Sets (X3C) ([SP2] in [2]). In X3C we are given a set X with $|X| = 3m$ and a collection C of three element subsets of X, and we are asked whether there exists $C' \subseteq C$ such that every member of X occurs in exactly one member of C'.

Theorem 2.2.1. 2-SMST3 is $\mathcal{NP}$-complete even for graphs with unit edge lengths.

Proof. Given an instance of X3C with set X, $|X| = 3m$, and a collection C, $|C| = n$, of three element subsets of X, construct the graph G as follows. First, without loss of generality, assume n is odd since we can always supplement C by a duplicate member. The core of G will be m gadgets, and the gadgets are linked by $m+1$ separating vertices $(v_1, \ldots, v_{m+1})$. In gadget i there will be a vertex $c_{i,j}$ for each subset $c_j \in C$. The vertices are connected with an edge between v_i and each $c_{i,j}$ and an edge between each $c_{i,j}$ and v_{i+1}. Thus, for each subset $c_j \in C$, there are m vertices $c_{i,j}$ in G. The linking of the gadgets forms a diamond lattice in G. Also, for each of the $3m$ elements $x_k \in X$ there is a vertex x_k in G, and for every $C_j \in C$ and for each $x_k \in C_j$, we connect $c_{i,j}$ and x_k in G by a chain of $m - 1$ vertices, for $i = 1 \ldots m$. Finally, connected

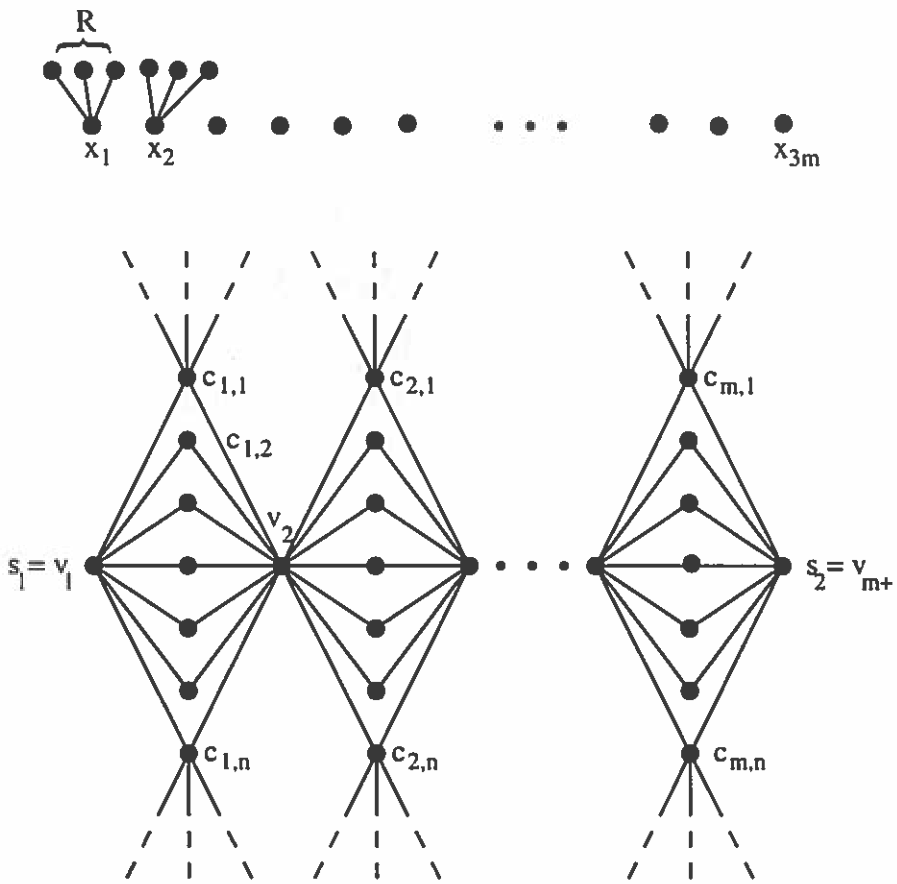


FIGURE 3. The construction for k-SMST3.

to each vertex x_k there will be $R = \frac{9}{2}m^2n + \frac{5}{2}m$ additional vertices of degree one. See Figure 3 for a diagram of this graph.

Now, let

$$K = 4m^2 + 3m + \frac{9}{2}m^3n - \frac{1}{2}m^2n - 2mn + 6m^2R + 3mR.$$

Note that

$$
\begin{aligned}
|V(G)| &= 3m(R+1) + mn(1 + 3(m-1)) + m + 1 \\
|E(G)| &= 2mn + 3mn(m-1) + 3mR.
\end{aligned}
$$

Thus, the reduction is in polynomial time. To complete the proof, it suffices to show that X has an exact cover if and only if G has a 2-SMST3 tree of cost $\leq K$.

In the remainder of the proof, we show that for G to have a 2-SMST3 tree of cost $\leq K$, the $s_1 s_2$-path in this optimal tree must pass along the diamond lattice and not include any element vertex x_k. Also, the path from an element vertex x_k to a source vertex will not include any vertex $c_{i,j}$ which is not on the $s_1 s_2$-path. Let T^* be such a tree. If the $s_1 s_2$-path does not include any x_k, there will be exactly m vertices $c_{i,j}$ on the path, and since each $c_{i,j}$ vertex has a direct path to only three of the x_k vertices, T^* exists if and only if X has an exact cover.

First, look at the cost of T^*. The tree is symmetric so the cost is the same for either source. The cost of the $s_1 s_2$-path is

$$\sum_{i=1}^{2n} i = 2m^2 + m.$$

The cost of the paths between each $c_{i,j}$ on the $s_1 s_2$-path and its element vertices is

$$
\begin{aligned}
3[\sum_{i=1}^{m}(i+1)+\sum_{i=1}^{m}(i+3)+\ldots+\sum_{i=1}^{m}(2m-1)] \\
&= 3[m\sum_{i=1}^{m} i + m\sum_{i=1}^{m} m^2] \\
&= \frac{9}{2}m^3 + \frac{3}{2}m^2.
\end{aligned}
$$

The cost of the degree one nodes attached to each x_k is

$$
\begin{aligned}
3R[\sum_{i=1}^{m}(m+1+1)+\sum_{i=1}^{m}(m+1+3)+\ldots+\sum_{i=1}^{m}(m+1+2m-1)] \\
&= 3R[m(m-1)+\sum_{i=1}^{m}(2i-1)] \\
&= 3R[m^2+m+m^2] \\
&= 6Rm^2+3Rm.
\end{aligned}
$$

The remaining $c_{i,j}$ nodes which are not on the s_1s_2-path will hang from either vertex v_j or v_{j+1}, and from each of these $c_{i,j}$ nodes will hang three chains of $m-1$ vertices. T^* will be balanced so if a node hangs from v_j, then another node will hang from v_{n-j+1}. Since n is odd, there will be an even number of these extra $c_{i,j}$s to distribute so it will be possible to balance the tree. The cost of "garbage collecting" these extra nodes is

$$
\begin{aligned}
m(n-1)[(m+1)+3\sum_{i=1}^{m-1}(i+m+1)] \\
&= (mn-m)[\frac{9m^2-m}{2}-2] \\
&= \frac{9}{2}m^3n-\frac{1}{2}m^2n-2mn-\frac{9}{2}m^3+\frac{1}{2}m^2+2m.
\end{aligned}
$$

So the total cost of T^* is

$$cost_3(T^*) = 4m^2 + 3m + \frac{9}{2}m^3n - \frac{1}{2}m^2n - 2mn + 6m^2R + 3mR = K,$$

and thus T^* is optimal for k-SMST3.

Now, we show that the s_1s_2-path in an optimal tree can not contain any element vertices. If the s_1s_2-path contains an element vertex, x_r, then the length of the intra-source path is at least $2m+2$. Also, for some source, at least half of the paths in the tree from that source to the element vertices must include x_r, and after taking into account the paths to the remaining element vertices, there are still $m(n-1)(3m-2)$ additional nodes in the graph to be counted. From this, we can estimate the cost of such a tree T' to be

$$\begin{aligned}
cost_3(T') &> \sum_{i=1}^{2m+2} i + (\frac{3m}{2} - 1)\left[\sum_{i=1}^{m}(i + 2m + 1) + (3m+2)R\right] \\
&\quad + (\frac{3m}{2} + 1)\left[\sum_{i=1}^{m}(i+1) + (m+2)R\right] + m(n-1)(3m-2) \\
&= \frac{9}{2}m^3 + \frac{3}{2}m^2 + 7m + 3m^2n - 2mn + 6m^2R + 4mR \\
&= \frac{69}{2}m^3 + \frac{23}{2}m^2 + 7m + 27m^4n + 18m^3n + 3m^2n - 2mn \\
&> 30m^3 + \frac{23}{2}m^2 + 3m + 27m^4n + 18m^3n - \frac{1}{2}m^2n - 2mn \\
&= 4m^2 + 3m + \frac{9}{2}m^3n - \frac{1}{2}m^2n - 2mn + 6m^2R + 3mR \\
&= K.
\end{aligned}$$

Therefore, the s_1s_2-path in an optimal tree can not include any element vertex x_r.

The remainder of the proof makes use of the following observation.

Observation 2.2.2. $cost_3(T) \geq \frac{1}{|S|} \cdot cost_1(T)$ where $cost_1(T)$ is defined as $cost_1(T) = \sum_{s \in S} \sum_{v \in V} d_T(s, v)$.

Define $\mathcal{T} = \{T | s_1 s_2\text{-path in } T \text{ does not contain an element vertex } x_k\}$. Therefore, T^* is optimal for k-SMST3 if $cost_3(T^*) = \frac{1}{|S|} \cdot \min_{T \in \mathcal{T}}\{cost_1(T)\}$.

To calculate $\min_{T \in \mathcal{T}}\{cost_1(T)\}$, we make use of an additional observation from [1, Observation 1].

Observation 2.2.3. The cost of a 2-SMST1 spanning tree T of a graph G with N vertices is equal to $N \cdot d(p) + 2\sum_{v \in V} d(v, p)$ where p is the $s_1 s_2$-path, $d(p)$ is the length of p, and $d(v, p)$ is the shortest distance from v to a vertex of p.

To calculate $\min_{T \in \mathcal{T}}\{cost_1(T)\}$, fix an arbitrary $s_1 s_2$-path along the diamond lattice and look at the minimum distance in G of each vertex v to the path. By Observation 2.2.3, we can use this distance to find the minimum cost. Note that there are $2m + 1$ vertices on the path, $3m + m(n-1)$ vertices at distance 1 from the path, $3mn$ vertices each at distance from 2 to m from the path, and $3mR$ vertices at distance $m + 1$ from the path. Summing these up,

$$\begin{aligned}
\min_{T \in \mathcal{T}}\{cost_1(T)\} &= (2m)[3mR + 4m + 3m^2 n - 2mn + 1] \\
&\quad + 2[(3m + m(n-1)) + \sum_{i=2}^{m} [3mn(i)] + 3mR(m+1)] \\
&= 8m^2 + 6m + 9m^3 n - m^2 n - 4mn + 12m^2 R + 6mR \\
&= 2cost_3(T^*) = 2K.
\end{aligned}$$

Thus, by Observation 2.2.2, T^* is minimal for $cost_3$, and we have proven that $cost_3(T^*) = K$. Note that in T^* the path from each element vertex x_k to a source only includes subset vertices $c_{i,j}$ which lie on the s_1s_2-path. If the path from x_k to a source included a subset vertex not on the s_1s_2-path, the distance from x_k to the s_1s_2-path would increase by one. By Observation 2.2.3, $cost_1(T^*)$ of the tree would increase and, by Observation 2.2.2, so would $cost_3(T^*)$.

Thus, a spanning tree T with $cost_3(T) \leq K$ can only exist if the s_1s_2-path does not include any element vertices and if the path from each element vertex to a source does not include any subset vertex not on the s_1s_2-path. As there are $3m$ element vertices, m subset vertices on the s_1s_2-path, and each subset vertex directly connects to exactly 3 element vertices, G will have a spanning tree T with $cost_3(T) \leq K$ if and only if X has an exact match. □

Corollary 2.2.4. k-SMST3 is $\mathcal{NP}$-complete.

CHAPTER III

THE POLYNOMIAL PROBLEMS

The key strategy used in this chapter for proving that a k-SMST problem has an efficient solution is to prove that a single source shortest paths spanning tree (SPST) from some point ϕ is optimal for the given cost metric. By the argument presented in Chapter I (p. 6), if some SPST is optimal, we can find the tree in polynomial time.

The k-SMST4 Problem

Instance: A graph $G = (V, E)$ with a length function, $l : E \to \Re$, k sources $S = \{s_1, \ldots, s_k\} \subseteq V$, a positive integer K.

Question: Is there a spanning tree T of G such that

$cost_4(T) = \sum_{v \in V} \max_{s \in S} d_T(s, v) \leq K$?

Define a vertex to be critical to a source if it is at the maximum distance from the source. The first lemma shows that the paths between two sources and their critical points must intersect along the $s_1 s_2$-path.

Lemma 3.1.1. Given a tree T and $s_1, s_2 \in V(T)$, then for all $c_1, c_2 \in V(T)$ such that $d_T(s_i, c_i) = \max_{v \in V} d_T(s_i, v)$, we have $d_T(s_1, p(c_1)) \geq d_T(s_1, p(c_2))$ where $p(v)$ is the projection of v on the $s_1 s_2$-path.

Proof. Let $d_1 = d_T(s_1, p(c_1))$, $D_1 = d_T(c_1, p(c_1))$, $d_2 = d_T(s_2, p(c_2))$, $D_2 = d_T(c_2, p(c_2))$, and let $d' = d_T(s_1, p(c_2)) - d_T(s_1, p(c_1))$. Figure 4 sketches this situation. Assuming the theorem does not hold, $d' > 0$. However,

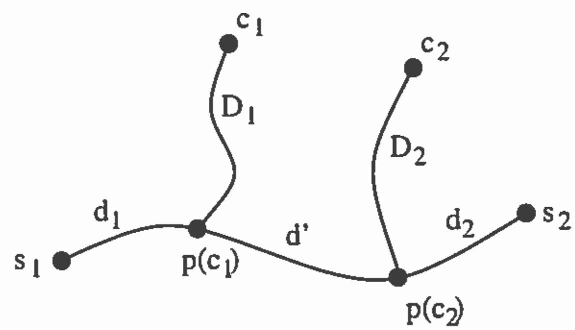


FIGURE 4. Diagram for the proof of Lemma 3.1.1

$$
\begin{array}{lrcl}
 & d_1 + D_1 & \geq & d_1 + d' + D_2 \\
+ & d_2 + D_2 & \geq & d_2 + d' + D_1 \\
\hline
 & d_1 + d_2 + D_1 + D_2 & \geq & d_1 + d_2 + D_1 + D_2 + 2d' \\
 & 0 & \geq & d'.
\end{array}
$$

A contradiction is reached, and the lemma is proved. □

Corollary 3.1.2. s_1c_1-path $\cap$ s_2c_2-path $\neq \emptyset$.

The next lemma shows that in a tree the path from each vertex to its critical source must include the midpoint of the s_1s_2-path.

Lemma 3.1.3. Given a tree T and $s_1, s_2 \in V(T)$, let χ be the midpoint of the s_1s_2-path in T. Then for all $v \in V(T)$, if $d_T(s_1, v) > d_T(s_2, v)$, then $d_T(s_1, \chi) < d_T(s_1, p(v))$ where p is the projection of v onto the s_1s_2-path.

Proof. Let $d_T(s_1, v) > d_T(s_2, v)$ and $d_T(s_1, \chi) \geq d_T(s_1, p(v))$. Then

$$\begin{aligned} d_T(s_1, v) &= d_T(s_1, p(v)) + d_T(p(v), v) \\ &\leq d_T(s_1, \chi) + d_T(\chi, v) \\ &= d_T(s_2, \chi) + d_T(\chi, v) \\ &= d_T(s_2, v). \end{aligned}$$

Thus, a contradiction is reached, and the lemma is proved. □

Corollary 3.1.4. If $d_T(s_1, v) = d_T(s_2, v)$, then $d_T(s_1, \chi) = d_T(s_1, p(v))$.

Corollary 3.1.5. The path from v to its critical source must pass through χ.

We are now ready to prove that there exists a SPST from some point in G which is optimal for k-SMST4.

Theorem 3.1.6. Given a graph G with sources $s_1, \ldots, s_k \in V(G)$, there exists a point χ such that any SPST rooted at χ is an optimal tree for k-SMST4.

Proof. Let T^* be an optimal tree for k-SMST4 and let s_1 and s_2 be the sources with maximum intrasource distance in T. Pick χ to be the midpoint on the s_1s_2-path in T.

First, we prove that for any vertex v and any source s_i, $i = 1 \ldots k, i \neq 1, i \neq 2$, either $d_T(v, s_1) \geq d_T(v, s_i)$ or $d_T(v, s_2) \geq d_T(v, s_i)$. Assume, without loss of generality, that $d_T(v, s_1) \geq d_T(v, s_2)$. Then by Corollary 3.1.5, $d_T(v, s_1) = d_T(v, \chi) + d_T(\chi, s_1)$.

$$\begin{aligned} d_T(v, s_i) &\leq d_T(v, \chi) + d_T(\chi, s_i) \\ &\leq d_T(v, \chi) + d_T(\chi, s_1) \\ &= d_T(v, s_1) \end{aligned}$$

Given T_χ, a SPST from χ, assume $cost_4(T^*) < cost_4(T_\chi)$. Thus, there exists some vertex v and a source s_1 of greatest distance from v in $T\chi$ for which $d_{T^*}(v, s_1) < d_{T_\chi}(v, s_1)$. This implies, from Corollary 3.1.5, $d_{T^*}(v, \chi) + d_{T^*}(\chi, s_1) < d_{T_\chi}(v, \chi) + d_{T_\chi}(\chi, s_1)$ but contradicts the fact that $T\chi$ is a shortest paths tree. Therefore, $cost_4(T^*) = cost_4(T_\chi)$ so a SPST from χ is optimal for k-SMST4. □

Theorem 3.1.7. k-SMST4 $\in \mathcal{P}$.

Proof. The proof follows directly from Theorems 3.1.6, 1.4.1, and 1.4.2. □

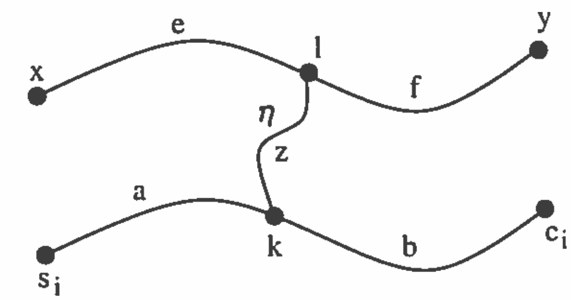
The k-SMST5 Problem

Instance: A graph $G = (V, E)$ with a length function, $l : E \to \Re$, k sources $S = \{s_1, \ldots, s_k\} \subseteq V$, a positive integer K.

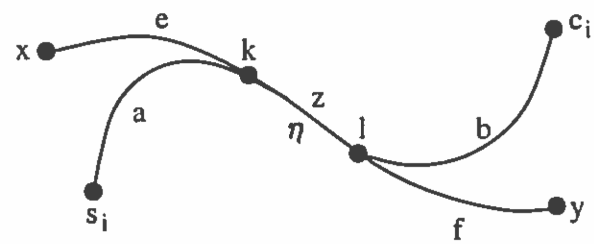
Question: Is there a spanning tree T of G such that

$cost_5(T) = \sum_{s \in S} \max_{v \in V} d_T(s, v) \leq K$?

To prove this problem is polynomially solvable we will again prove that there exists some single source shortest paths spanning tree T for which $cost_5(T)$ is optimal. The diameter of a graph is the maximum distance between any two vertices, and the next



(a) Assume paths do not intersect.



(b) Assume paths do intersect.

FIGURE 5. Diagrams for the proof of Lemma 3.2.1

lemma proves that a path from a source to a critical vertex must intersect the diametric path in T. Also, one of the endpoints of this diametric path will be critical to the source. From this lemma, we can prove that all paths between source nodes and their critical vertices will intersect at the midpoint of the longest path in T.

Lemma 3.2.1. For a tree T with $s_i \in V(T), i = 1 \ldots k$, let $c_i = \max_{v \in V}\{d_T(s_i, v)\}$, and let $x \rightarrow y$ be the path of maximum distance in T. Let χ be the midpoint on this path. Then $\chi \in s_i c_i$-path. Moreover, if $C_i = \{u | d_T(s_i, u) = \max_{v \in V} d_T(s_i, v)\}$, then $\{x, y\} \cup C_i \neq \emptyset$.

Proof. First, show the xy-path in tree T intersects with the $s_i c_i$-path for some i. If the two paths do not intersect, they must be joined by some path η. Let k be the terminus of η on the s_ic_i-path, and let l be the terminus of η on the xy-path. Now, let $a = d_T(s_i, k), b = d_T(k, c_i), e = d_T(x, l), f = d_T(l, y)$, and $z = d_T(k, l)$. See Figure 5a for a diagram of these assumptions. Note that $f \geq z + b$, otherwise the xc_i-path would be longer than the xy-path. Also note that $b \geq z + f$, otherwise the s_iy-path would be longer than the s_ic_i-path. Yet, these two facts imply $z \leq 0$, and thus the xy-path must intersect the s_ic_i-path.

The next step is to show $y \in C_i$. Let η be the intersection of the xy-path and the s_ic_i-path, and let k and l be the endpoints of this intersection although the intersection could be a single vertex. Assume the situation is as in Figure 5b with $e = d_T(x, k)$, $b = d_T(l, c_i)$, and a, f, and z defined as before. Let q be the diameter of T and thus the length of the xy-path. As $c_i \in C_i$, $a+z+b \geq a+z+f$ which implies $b \geq f$, and as the xy-path is the path of maximum length in T, $e+z+f \geq e+z+b$ which implies $f \geq b$. Thus, $b = f$ so $a + z + b = a + z + f$ and $d_T(s_i, y) = d_T(s_i, c_i) = \max_{v \in V} d_T(s_i, v)$. Therefore, $y \in C_i$.

Finally, let χ be the midpoint of the xy-path so $d_T(x, \chi) = d_T(y, \chi) = \frac{q}{2}$, and show that χ lies in η and thus on the s_ic_i-path. If $\chi \notin \eta$, then either $e > \frac{q}{2}$ or $f > \frac{q}{2}$. If we let $e > \frac{q}{2}$, then

$$\begin{aligned} z + f &< \tfrac{q}{2} && \Rightarrow \\ z + b &< \tfrac{q}{2} && \Rightarrow \\ e &> z + b && \Rightarrow \\ a + e &> a + z + b, \end{aligned}$$

and a contradiction is reached. Also, if we let $f > \frac{q}{2}$, then

$$\begin{aligned} b &> \tfrac{q}{2} \quad \Rightarrow \\ b + f &> q, \end{aligned}$$

and a contradiction is again reached. Therefore, $e, f \leq \frac{q}{2}$, so $\chi \in \eta$, and thus χ is on the $s_i c_i$-path. □

With this lemma, we can prove k-SMST5 $\in \mathcal{P}$.

Theorem 3.2.2. A SPST rooted at χ is optimal for k-SMST5.

Proof. Let T be an optimal tree, let q be the diameter of T, and let x and y be the endpoints of a diametric path. From Lemma 3.2.1, all $s_i c_i$-paths include χ, and x or y is critical for each source. Let T_χ be a SPST from χ. By definition, $c_i \in V$ is defined such that $d_T(s_i, c_i) = \max_{v \in V} d_T(s_i, v)$. Define $c_i^\chi \in V$ such that $d_{T_\chi}(s_i, c_i^\chi) = \max_{v \in V} d_{T_\chi}(s_i, v)$.

$$\begin{aligned}
cost_5(T) &= \sum_i (d_T(s_i, \chi) + d_T(\chi, c_i)) \\
cost_5(T_\chi) &= \sum_i (d_{T_\chi}(s_i, c_i^\chi)) \\
&\leq \sum_i (d_{T_\chi}(s_i, \chi) + d_{T_\chi}(\chi, c_i^\chi)) \\
&\leq \sum_i (d_T(s_i, \chi) + \frac{q}{2}) \\
&= \sum_i (d_T(s_i, \chi) + d_T(\chi, c_i)) \\
&= cost_5(T)
\end{aligned}$$

By the assumption that T is optimal, $cost_5(T) \leq cost_5(T_\chi)$. Therefore, $cost_5(T) = cost_5(T_\chi)$ so T_χ is optimal. □

Theorem 3.2.3. k-SMST5 $\in \mathcal{P}$.

Proof. The proof follows directly from Theorems 3.2.2, 1.4.1, and 1.4.2. □

CHAPTER IV

CONCLUSION

This thesis considered the problems of constructing optimal multi-source spanning trees with six different distance related cost metrics. One of these problems is known to be $\mathcal{NP}$-hard and another is known to have an efficient solution. After summarizing these two results, this thesis presented proofs of the complexity of the other four problems which until now had unknown complexity status. Two of these metrics were shown to yield $\mathcal{NP}$-hard problems, but the other two metrics could be minimized in polynomial time.

Further possible research on this topic includes finding efficient algorithms for the polynomially solvable problems and approximation algorithms for the $\mathcal{NP}$-hard problems. There has been some work on approximation algorithms for the more general Optimum Communication Spanning Tree in [8] which can be applied to constructing a tree for k-SMST1 (k-SPST). Also, this thesis made no judgement as to the relative goodness of each of the metrics nor did this thesis attempt to apply any of the metrics to specific problems.

Beyond the practical results of demonstrating which cost metrics are feasible for possible multicast routing trees, this thesis may have some theoretical significance by further defining the boundary between the polynomially solvable and the $\mathcal{NP}$-complete problems for multi-source spanning trees.

BIBLIOGRAPHY

[1] A. M. Farley, P. Fragopoulou, D. Krumme, A. Proskurowski, and D. Richards. Multi-source spanning tree problems. In *Proceedings of the 6th International Colloquium on Structural Information and Communication Complexity*, pages 126–136, 1999.

[2] 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, NY, 1979.

[3] T. C. Hu. Optimum communication spanning trees. *SIAM Journal of Computing*, 3(3):188–195, 1974.

[4] D. S. Johnson, J. K. Lenstra, and A. H. G. Rinnooy Kan. The complexity of the network design problem. *Networks*, 8(4):279–285, 1978.

[5] D. W. Krumme and P. Fragopoulou. Minimum eccentricity multicast trees. Submitted, 1999.

[6] B. McMahan and A. Proskurowski. Multi-source spanning trees: Algorithms for minimizing source eccentricities. Submitted, 1999.

[7] R. Ravi, R. Sundaram, M. V. Marathe, D. J. Rosenkrantz, and S. S. Ravi. Spanning trees – short or small. *SIAM Journal of Discrete Mathematics*, 9(2):178–200, 1996.

[8] B. Y. Wu, K. Chao, and C. Y. Tang. Approximation algorithms for some optimum communication spanning tree problems. *Discrete Applied Mathematics*, 102:245–266, 2000.