Approximating the two-source minimum routing cost spanning trees of metric graphs

Bang Ye Wu
Dept. of Computer Science and Information Engineering,
Shu-Te University, YenChau, KaoShiung, Taiwan 824, R.O.C.
bangye@mail.stu.edu.tw

Abstract

Let G be an undirected graph with nonnegative edge lengths. Given two vertices as sources and all vertices as destinations, we investigated the problem how to construct a spanning tree of G such that the total distance from sources to destinations is minimum. In the paper, we show the NP-hardness of the problem and present a polynomial time approximation scheme. For any $\varepsilon > 0$, the approximation scheme finds a $(1+\varepsilon)$-approximation solution in $O(n^{\lceil 1/\varepsilon+1\rceil})$ time.

1 Introduction

Let $G = (V, E, w)$ be an undirected graph with nonnegative edge length function w. Given k vertices as sources and all vertices as destinations, the *k-source minimum routing cost spanning tree* (k-MRCT) problem is to construct a spanning tree T of G such that the total distance from sources to destinations is minimum. That is, we want to find a spanning tree T minimizing $\sum_{s\in S}\sum_{v\in V} d_T(s,v)$, where S is the set of sources and $d_T(s,v)$ is the distance between s and v on T.

If there is only one source, the problem is reduced to the *shortest-path tree* problem, and it is always possible to find a spanning tree such that the path between the source and each vertex is a shortest path on the given graph. The shortest-path tree problem has been well studied and efficient algorithms were developed. For example, Dijkstra's algorithm [2] finds a shortest path tree for a graph with nonnegative weights in $O(|V|^2)$ time. Other algorithms with different considerations can be found in [1]. For the other extreme case that all vertices are sources, the problem is reduced to the *minimum routing cost spanning tree* (MRCT) problem (also called the *shortest total path length spanning tree* problem), and is therefore NP-hard [5, 3]. The first constant ratio approximation algorithm for the MRCT appeared in [7]. It was shown that there is a shortest path tree which is a 2-approximation of the MRCT. In [8], the approximation ratio was improved to $(4/3+\varepsilon)$ for any fixed $\varepsilon > 0$, and then further improved to a *polynomial time approximation scheme* (PTAS) in [9].

The *optimum communication spanning tree* (OCT) problem[4] is a more general version of the MRCT problem. In the OCT problem, in addition to the edge length, we are also given the requirement for each pair of vertices, and the goal is to minimize the sum of the distances multiplied by the requirements, over all pairs of vertices. In [10], two vertex-weighted generalizations of the MRCT problem were studied; one is the *optimal product-requirement communication spanning tree* (PROCT) problem and the other is the *optimal sum-requirement communication spanning tree* (SROCT) problem. In the PROCT problem, the requirement between a pair of vertices is assumed to be the product of the given weights of the two vertices; while, in the SROCT problem, the requirement is the sum of the vertex weights. Both PROCT and SROCT problems are special cases of the OCT problem. A 1.577-approximation algorithm for the PROCT problem and a 2-approximation algorithm for the SROCT problem were shown in [10]. Furthermore the PROCT problem was shown to admit a PTAS in [11].

The k-source MRCT problem is a special case of the SROCT problem, in which the vertex weight of each source is one and the weights of all the other vertices are zeros. The relationship of the four problems is shown in Figure 1.

The k-MRCT problem for general k is obviously NP-hard since it contains the MRCT problem as

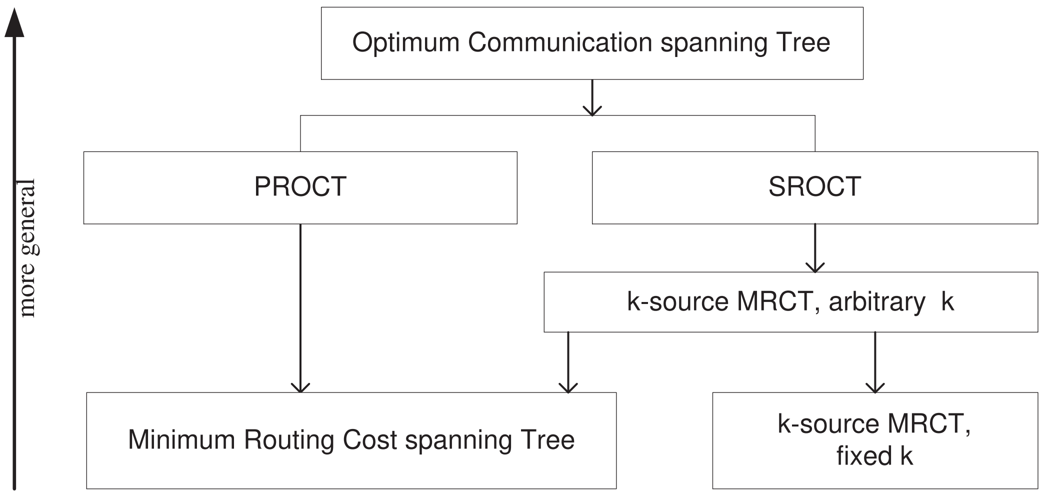

Figure 1: The relationship of the OCT, PROCT, SROCT, MRCT, and k-MRCT problems.

a special case. But the previous results do not tell us the complexity of the k-MRCT problem for a fixed constant k. Since the k-MRCT problem is a special case of the SROCT problem, the 2-approximation algorithm for the SROCT problem [10] ensures the same approximation ratio of the k-MRCT. In this paper, we show that the k-MRCT problem is NP-hard even for $k = 2$. We present a PTAS for the 2-MRCT problem. For any $\varepsilon > 0$, the PTAS finds a solution with approximation ratio $1 + \varepsilon$ in $O(n^{\lceil 1/\varepsilon+1 \rceil})$ time.

The remaining sections are organized as follows: In Section 2, some definitions and notations are given. The NP-hardness of the 2-MRCT problem is shown in Section 3, and the PTAS is presented in Section 4. Finally, concluding remarks are given in Section 5.

2 Preliminaries

In this paper, a graph is a simple, connected and undirected graph. By $G = (V, E, w)$, we denote a graph G with vertex set V, edge set E, and edge length function w. The edge length function is assumed to be nonnegative. For any graph G, $V(G)$ denotes its vertex set and $E(G)$ denotes its edge set. Let w be an edge length function on a graph G. For a subgraph H of G, we define $w(H) = w(E(H)) = \sum_{e \in E(H)} w(e)$. We shall also use n to denote $|V(G)|$.

Definition 1. Let $G = (V, E, w)$ be a graph. For $u, v \in V$, $SP_G(u, v)$ denotes a shortest path between u and v on G. The shortest path length is denoted by $d_G(u, v) = w(SP_G(u, v))$.

Definition 2. Let H be a subgraph of G. For a vertex $v \in V(G)$, we use $d_G(v, H)$ to denote the shortest distance from v to H, i.e., $d_G(v, H) = \min_{u \in V(H)} d_G(v, u)$. The definition also includes the case that H is a vertex set but no edge.

We now define the k-source routing cost of a tree.

Definition 3. For a tree T and $S \subset V(T)$, the *routing cost* of T with source set S is defined by $c(T, S) = \sum_{s \in S} \sum_{v \in V(T)} d_T(s, v)$. When there are only two sources s_1 and s_2, we also use $c(T, s_1, s_2)$ to denote the routing cost.

Lemma 1. Let T be a spanning tree of $G = (V, E, w)$ and P be the path between s_1 and s_2 on T. $c(T, s_1, s_2) = n \times w(P) + 2 \sum_{v \in V} d_T(v, P)$, in which $n = |V|$.

Proof: For any $v \in V$, $d_T(v, s_1) + d_T(v, s_2) = w(P) + 2d_T(v, P)$. Summing over all vertices in V, the result is obtained. $\square$

3 The computational complexity

In this section, we show the NP-hardness of the 2-MRCT problem by reducing the *exact cover by 3-sets* (X3C) problem to it.

Definition 4. Given a set X with $|X| = 3q$ and a collection C of 3-element subsets of X, the X3C problem asks if there is a subcollection $C' \subseteq C$ such that every element of X occurs in exactly one member of C'.

Theorem 2. The two-source MRCT problem is NP-hard even for metric inputs.

Proof: We shall reduce the X3C problem to the 2-MRCT problem. Let $X = \{x_i | 1 \le i \le 3q\}$ and $C = \{Y_i | 1 \le i \le m\}$ be an instance of the X3C problem, in which each Y_i is a 3-element subset of X. We construct a complete graph $G = (V, E, w)$ as an instance of the 2-MRCT problem. The graph G is consisted of the following subgraphs as illustrated in Figure 2.

- For $1 \le i \le q$, G_i is a complete graph with vertex set $\{v_i^j | 1 \le j \le m\}$, in which v_i^j is corresponding to Y_j in C. Every edge in G_i has weight one.
- H is a complete graph with vertex set $\{u_i | 1 \le i \le 3q\}$, in which each u_i is corresponding to one element of X. Every edge in H has weight one.
- G_0 contains only one vertex s_1 and G_{q+1} contains only one vertex s_2.

The weights of edges between the subgraphs are defined as follows:

- For $u \in V(G_i)$, $v \in V(G_j)$ and $i \ne j$, $w(u, v) = |i - j|$.
- For $u_i \in V(H)$ and $v_k^j \in V(G_k)$, $1 \le k \le q$, $w(u_i, v_k^j) = L$ if $x_i \in Y_j$; and $w(u_i, v_k^j) = L+1$ otherwise, in which $L \ge q + 1$ is an integer.
- $w(u_i, s_1) = w(u_i, s_2) = L + 1$ for $u_i \in V(H)$.

Obviously G is a metric graph. Let $n = |V|$ be the number of vertices of G. We have $n = qm + 3q + 2 = q(m + 3) + 2$. We shall show that there is an exact cover of the X3C problem if and only if there is a spanning tree T of G such that $c(T, s_1, s_2) = n(q + 1) + 2(m - 1)q + 6qL$.

Suppose that there is an exact cover of the X3C problem. Without loss of generality, let $\{Y_i | 1 \le i \le q\} \subset C$ be an exact cover of X. Let $P = (s_1, v_1^1, v_2^2, \ldots, v_q^q, s_2)$. Construct a spanning tree T such that T contains P and all the other vertices are connected to P by the shortest edge to P. Each vertex in $V(G_i) \setminus \{v_i^i\}$ is connected to v_i^i by an edge of weight one, for $1 \le i \le q$. We can show that

$$c(T, s_1, s_2) = n(q + 1) + 2(m - 1)q + 6qL.$$

Conversely suppose that there is a spanning tree T of G such that $c(T, s_1, s_2) = n(q + 1) + 2(m - 1)q + 6qL$. Let P be the path between s_1 and s_2 on T. First we can show that P is a shortest path and $w(P) = q + 1$. By Lemma 1,

$$c(T, s_1, s_2) \ge n(q + 1) + 2(m - 1)q + 6qL$$

The equality holds if and only if $\sum_{1 \le i \le q} \sum_{v \in V(G_i)} d_T(v, P) = (m - 1)q$ and $\sum_{v \in V(H)} d_T(v, P) = 3qL$. It implies that each vertex in $V(H)$ is connected to some vertex of P by an edge of weight L. It also implies that the path contains at least q vertices in addition to the two sources. Consequently there are exact 3 vertices of $V(H)$ connected to each vertex of the path by edges of weight L. By the definition of G, there exists a corresponding exact cover of X.

We have shown that the X3C problem can be reduced to the 2-MRCT problem in polynomial time. Since the X3C problem is NP-complete [6, 3], the 2-MRCT problem is NP-hard even with metric inputs. □

4 A PTAS for 2-MRCT

Before showing our PTAS, we give a simple 2-approximation algorithm, and then generalize the idea to the PTAS.

Algorithm Simple_2-MRCT
Input: A graph $G = (V, E, w)$ and $s_1, s_2 \in V$.
Output: A spanning tree T of G.
1: Find a shortest path P between s_1 and s_2 on G.
2: Find the shortest path forest with multiple roots in $V(P)$.
3: Let T be the union of the forest and P.

Lemma 3. The algorithm Simple_2-MRCT finds a 2-approximation of the 2-MRCT of a graph in $O(n^2)$ time.

Proof: First we show a lower bound of the optimum. Let Y be the optimal tree of the 2-MRCT problem with input G and s_1, s_2.

$$c(Y, s_1, s_2) \ge \frac{1}{2} \sum_v (d_G(v, s_1) + d_G(v, s_2)) + \frac{n}{2} d_G(s_1, s_2) \quad (1)$$

Let T be the tree constructed by Algorithm Simple_2-MRCT and P be the shortest between s_1

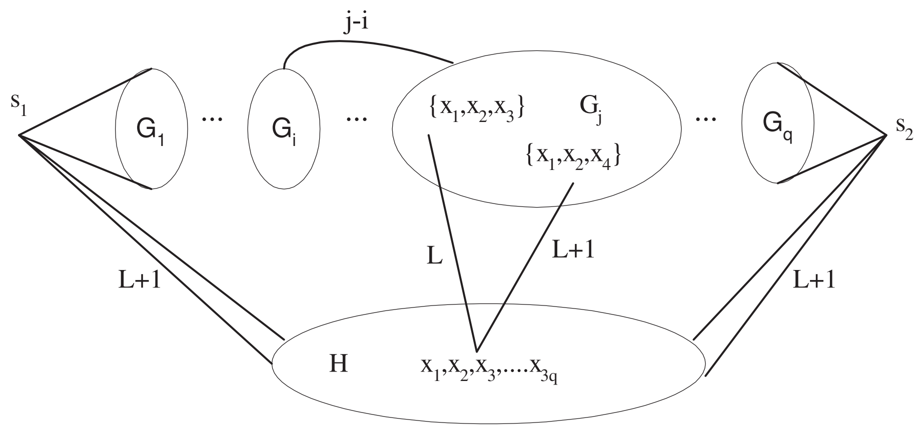

Figure 2: The transformation of an instance of the X3C problem to an instance of the 2-MRCT problem.

and s_2. Since each vertex is connected to P by a shortest path to any of the vertices of P at Step 2, for any vertex v,

$$d_T(v, P) \le (d_G(v, s_1) + d_G(v, s_2))/2$$

By Lemma 1,

$$\begin{aligned} & c(T, s_1, s_2) \\ \le \quad & n d_G(s_1, s_2) + \sum_v (d_G(v, s_1) + d_G(v, s_2)) \end{aligned}$$

Comparing with Eq. (1), we have $c(T, s_1, s_2) \le 2c(Y, s_1, s_2)$ and T is a 2-approximation of the 2-MRCT. □

As mentioned in Section 1, the 2-MRCT is a special case of the SROCT problem. By the result in [10], the 2-MRCT can be approximated with error ratio 2 and time complexity $O(n^3)$. The simple algorithm ensures the same error ratio and is more efficient in time. The ratio shown in Lemma 3 is tight in the sense that there is an instance such that the spanning tree constructed by the algorithm has routing cost twice as the optimum. Consider the complete graph in which $w(v, s_1) = w(v, s_2) = 1$ and $w(s_1, s_2) = 2$ for each vertex v. The distance between any other pair of vertices is zero. At Step 1, the algorithm may find edge (s_1, s_2) as the path P, and then each other vertex is connected to one of the two sources. The routing cost of the constructed tree is $4n - 4$. On the optimal tree, the path between the two sources is a two-edge path, and all other vertices are connected to the middle vertex of the path. The optimal routing cost is therefore $2n$. The increased cost is resulted by missing the vertex on the path. The existence of the vertex reduces the cost at an amount of $w(P)$ for each vertex.

The worst case instance of the simple algorithm give us the intuition to improve the approximation ratio. To improve the result of Simple_2-MRCT, we may try to guess some vertices of the path. Let m be an vertex of the path between the two sources on the optimal tree, and U be the set of vertices connected to the path at m. If the path P found in Step 1 of the simple algorithm includes m, the distance from any vertex in U to each of the sources will be no more than the corresponding distance on the optimal tree. In addition, the vertex m partitions the path into two subpaths. The maximal increased cost by one of the vertices is the length of the subpath instead of the whole path. Our PTAS is to guess some of the vertices of the path, which partition the path such that the number of vertices connected to each subpath is small enough. We now describe the PTAS and the analysis precisely.

Let Y be the 2-MRCT of $G = (V, E, w)$ and $n = |V|$. Let $P = (p_1 = s_1, p_2, p_3, \ldots, p_h = s_2)$ be the path between s_1 and s_2 on Y. Define V_i, $1 \le i \le h$, as the set of the vertices connected to P at p_i. Let $k \ge 0$ be an integer. For $0 \le i \le k+1$, define $m_i = p_j$ in which j is the minimal index such that $\sum_{1 \le q \le j} |V_q| \ge \lceil in/(k+1) \rceil$. By definition, $s_1 = m_0$ and $s_2 = m_{k+1}$. For $0 \le i \le k$, let $U_i = \bigcup_{a<j<b} V_j$ and P_i be the path from p_a to p_b, in which $p_a = m_i$ and $p_b = m_{i+1}$. Also let $U = V \setminus \bigcup_{0 \le i \le k} U_i$ and $M = \{m_i | \, 0 \le i \le k+1\}$. Note that the above definitions include the case $m_i = m_{i+1}$. In such a case, P_i contains only one vertex and U_i is empty. The definitions are shown in Figure 3. The above definitions partition the vertex set V into $U, U_0, U_1, \ldots, U_k$ to satisfying the properties in the following lemmas.

The proof of the following lemma is omitted.

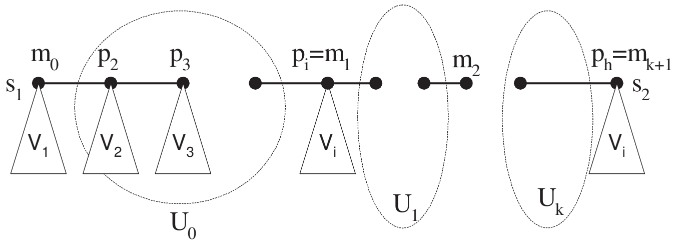

Figure 3: The definitions of the partition of the vertices.

Lemma 4. For any $v \in U$, $d_G(v, M) \leq d_Y(v, P)$.

Lemma 5. For any $0 \leq i \leq k$,

$$\sum_{v \in U_i} d_G(v, M) \leq \sum_{v \in U_i} d_Y(v, P) + \frac{n}{2(k+1)} w(P_i).$$

The PTAS is listed below.

Algorithm PTAS_2-MRCT
Input: A graph $G = (V, E, w)$, $s_1, s_2 \in V$, and an integer $k \geq 0$.
Output: A spanning tree T of G.
For each k-tuple $(m_1, m_2, \ldots, m_k)$ of not necessarily distinct vertices, use the following steps to find a spanning tree T, and output the tree of minimal routing cost.
1: Let $m_0 = s_1$ and $m_{k+1} = s_2$.
2: Find a tree $X = \bigcup_{0 \leq i \leq k} SP_G(m_i, m_{i+1})$.
3: Find the shortest path forest spanning $V(G)$ with all vertices in $V(X)$ as the multiple roots.
4: Let T be the union of the forest and X.

Lemma 6. For any constant k, the algorithm PTAS_2-MRCT finds a $(\frac{k+2}{k+1})$-approximation for the 2-MRCT problem in $O(n^{k+2})$ time.

Proof: The algorithm tries each possible k-tuple in each iteration. There exists a k-tuple $(m_1, m_2, \ldots, m_k)$ which partitions the vertex set V into $(U, U_0, U_1, \ldots, U_k)$ and the subsets satisfy the properties in Lemma 4 and 5.

$$\begin{aligned} c(T, s_1, s_2) &= n \times w(P) + 2 \sum_{v \in U} d_T(v, X) \\ &\quad + 2 \sum_{0 \leq i \leq k} \sum_{v \in U_i} d_T(v, X) \qquad (2) \end{aligned}$$

Since $d_T(v, X) = d_G(v, X) \leq d_G(v, M)$ for any $v \in U$, by Lemma 4, we have

$$\sum_{v \in U} d_T(v, X) \leq \sum_{v \in U} d_G(v, M) \leq \sum_{v \in U} d_Y(v, P) \quad (3)$$

Similarly by Lemma 5,

$$\begin{aligned} &\sum_{0 \leq i \leq k} \sum_{v \in U_i} d_T(v, X) \\ \leq &\sum_{0 \leq i \leq k} \sum_{v \in U_i} d_Y(v, P) + \frac{n}{2(k+1)} w(P) \quad (4) \end{aligned}$$

By Eq. (2), (3), and (4),

$$c(T, s_1, s_2) \leq \frac{k+2}{k+1} c(Y, s_1, s_2),$$

since $c(Y, s_1, s_2) = n \times w(P) + 2 \sum_{v \in V} d_Y(v, P)$ by Lemma 1.

The number of all possible k-tuples is $O(n^k)$ for constant k. Similar as the proof of Lemma 3, each iteration takes $O(n^2)$ time. The time complexity of the algorithm is therefore $O(n^{k+2})$. □

Theorem 7. The 2-MRCT problem admits a PTAS. For any $\varepsilon > 0$, a $(1+\varepsilon)$-approximation of the 2-MRCT can be found in $O(n^{\lceil 1/\varepsilon + 1 \rceil})$ time.

Proof: For any $\varepsilon > 0$, we choose an integer $k = \lceil 1/\varepsilon - 1 \rceil$ and run the algorithm PTAS_2-MRCT. By Lemma 6, the algorithm takes $O(n^{k+2})$ time to finds a $(k+2)/(k+1)$-approximation for the 2-MRCT problem. The approximation ratio is $(k+2)/(k+1) = 1 + \varepsilon$ and the time complexity is $O(n^{\lceil 1/\varepsilon + 1 \rceil})$, which is polynomial for constant ε. □

5 Concluding remarks

The fastest implementation of Dijkstra's algorithm is $O(|E| + |V| \log |V|)$. This may improve the time complexities of our approximation algorithms for sparse graphs. An interesting problem is the computational complexity of the k-source MRCT for any fixed k. By the NP-hardness of 2-MRCT, we may easily show that the k-MRCT problem is also NP-hard for any fixed even number

$k = 2q$. For an instance of the 2-MRCT problem, we duplicate q copies of the two sources. If there is a polynomial time algorithm for the k-MRCT, it can also solve the 2-MRCT problem in polynomial time. However, we did not find any obvious reduction for fixed odd k. By the previous result for the SROCT problem, the k-MRCT admits a 2-approximation algorithm, either for any fixed $k > 2$ or for arbitrary k. But the approximability for either case is also still open.

References

[1] T.H. Cormen, C.E. Leiserson, and R.L. Rivest, *Introduction to Algorithms*, the MIT Press, 1994.

[2] E.W. Dijkstra, A note on two problems in connexion with graphs, *Numerische Mathematik* 1 (1959), 269–271.

[3] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman and Company, San Francisco, 1979.

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

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

[6] R.M. Karp, Reducibility among combinatorial problems, in R.E. Miller and J.W. Thatcher (eds.) *Complexity of Computer Computations*, pp. 85–103, Plenum Press, New York, 1972.

[7] R. Wong, Worst-case analysis of network design problem heuristics. *SIAM J. Algebraic Discrete Mathematics* 1 (1980), 51–63.

[8] B. Y. Wu, K. M. Chao, and C. Y. Tang, Approximation algorithms for the shortest total path length spanning tree problem, *Discrete Applied Mathematics* 105 (2000), 273–289.

[9] B. Y. Wu, G. Lancia, V. Bafna, K. M. Chao, R. Ravi, and C. Y. Tang, A polynomial time approximation scheme for minimum routing cost spanning trees, *SIAM J. Computing* 29 (2000), 761–778.

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

[11] B. Y. Wu, K. M. Chao, and C. Y. Tang, A polynomial time approximation scheme for optimal product-requirement communication spanning trees, *Journal of Algorithms* 36 (2000), 182–204, doi:10.1006/jagm.2000.1088.