兩起點最佳通訊擴展樹之近似演算法

Approximation algorithms for the two-source optimum communication spanning trees

吳邦一(Bang Ye Wu)
樹德科技大學資訊工程系 E-mail: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, and aslo given arbitrary requirements between sources and destinations, we investigated the problem how to construct a spanning tree of G such that the total communication cost from sources to destinations is minimum, where the communication cost from a source to a destination is the path length multiplied by their requirement. In the paper, we present a 3-approximation algorithm for general graph inputs, and a 2-approximation algorithm for metric graphs.
Keywords: approximation algorithms, network design, spanning trees.

1 Introduction

Consider the following *optimum communication spanning tree* (OCT) problem formulated by Hu in [4]. Let $G = (V, E, w)$ be an undirected graph with nonnegative edge length function w. We are also given the requirements $\lambda(u, v)$ for each pair of vertices. For any spanning tree T of G, the communication cost between two vertices is defined to be the requirement multiplied by the path length of the two cities on T, and the communication cost of T is the total communication cost summed over all pairs of vertices. Our goal is to construct a spanning tree with minimum communication cost. That is, we want to find a spanning tree T such that $\sum_{u,v\in V} \lambda(u, v)d_T(u, v)$ is minimized, where $d_T(u, v)$ is the distance between u and v on T.

The requirements in the OCT problem are arbitrary nonnegative numbers. By restricting the requirements, several special cases of the problem have been studied.

- $\lambda(u, v) = 1$ for each $u, v \in V$. The problem is called the *minimum routing cost spanning tree* (MRCT) problem (also called the *shortest total path length spanning tree* problem), and is NP-hard [3, 5]. The first constant ratio approximation algorithm for the MRCT appeared in [6]. It was shown that there is a shortest path tree which is a 2-approximation of the MRCT. In [7], 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 [8].
- $\lambda(u, v) = r(u)r(v)$ for each $u, v \in V$, where $r(v)$ is the given nonnegative vertex weight for each vertex v. This version is the *optimal product-requirement communication spanning tree* (PROCT) problem. A 1.577-approximation algorithm for the PROCT problem was shown in [9], and then improved to a PTAS in [10].
- $\lambda(u, v) = r(u) + r(v)$ for each $u, v \in V$, where $r(v)$ is the given nonnegative vertex weight for each vertex v. It is called the *optimal sum-requirement communication spanning tree* (SROCT) problem. The problem was introduced in [9] and a 2-approximation algorithm for the SROCT problem were shown.

The k-source MRCT problem is a special case of the SROCT problem, in which k vertices are given as sources and all vertices (including the sources) are destinations. In other words, the vertex weight of each source is one and the weights of all the other vertices are zeros. The 2-source MRCT problem has been shown to be NP-hard even for metric inputs [11]. A PTAS for the problem was also proposed. In this paper, we investigate a generalization of the 2-source MRCT problem. We considered the case that there are also only two sources but the requirement for each pair of source and destination is arbitrary. We call it the *2-source optimum communication spanning tree* (2-OCT) problem. Given two vertices s_1, s_2 as sources and all vertices as destinations, the 2-OCT is a spanning tree T of G such that the total communication cost, defined by $\sum_v (r_1(v)d_T(v, s_1) + r_2(v)d_T(v, s_2))$, is minimum, in which, for each vertex v, $r_1(v)$ and $r_2(v)$ are the given requirements from v to s_1 and s_2 respectively.

In this paper, we present a 3-approximation algorithm for the 2-OCT when the input is a general graph, and a 2-approximation algorithm when the input is a metric graph, i.e., a complete undirected graph with edge lengths satisfying the triangle inequality. The relationship of the different versions of the OCT problems is illustrated in Figure 1, and the currently best approximation ratios are summarized in Table 1.

The remaining sections are organized as follows: In Section 2, some definitions and notations are given. The approximation algorithms for the 2-OCT problem on general and metric graphs are shown in Section 3 and 4 respectively.

2 Preliminaries

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. A metric graph is a complete undirected graph and the edge lengths satisfy the triangle inequality. Let s_1 and s_2 be the given two sources. For each vertex v, $r_1(v)$ and $r_2(v)$ are the given nonnegative requirements from v to s_1 and s_2 respectively. 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 $|V(G)|$ when there is no ambiguity.

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 communication cost of a spanning tree.

Definition 3: Let T be a tree. For any vertex $v \in V(T)$, $c_T(v) = r_1(v)d_T(v, s_1) + r_2(v)d_T(v, s_2)$. For any spanning tree T of G, the *communication cost* of T is defined by $c(T) = \sum_{v \in V(T)} c_T(v)$.

3 On general graphs

In this section, we investigate the 2-OCT problem in the case that the input is a general graph. Our approximation algorithm is to construct a shortest path forest rooted at the two sources, and then connect the two source by a shortest path. The algorithm is presented below.

Algorithm Approx1
Input: A graph G, two vertices s_1, s_2, and requirements $r_1(v), r_2(v)$.
Output: A spanning tree T of G.
Find a shortest-path forest with roots s_1, s_2. Let T_1, T_2 be the two trees containing s_1, s_2 respectively.
Find a shortest path P between s_1 and s_2. Let $P = (p_1 = s_1, p_2, \ldots, s_2)$.
Find the first vertex p_i not in T_1.
Join T_1 and T_2 into a tree T by adding edge (p_{i-1}, p_i).
Output T.

To analyze the performance of the algorithm, we first give a trivial lower bound of the optimal in the next lemma.

Lemma 1: Let Y be the 2-OCT. For any vertex v, $c_Y(v) \geq r_1(v)d_G(v, s_1) + r_2(v)d_G(v, s_2)$.

Proof: Since $d_G(v, s_1)$ is the shortest distance from v to s_1, $d_G(v, s_1) \leq d_Y(v, s_1)$. Similarly $d_G(v, s_2) \leq d_Y(v, s_2)$. The result is obvious by the definition of $c_Y(v)$. □

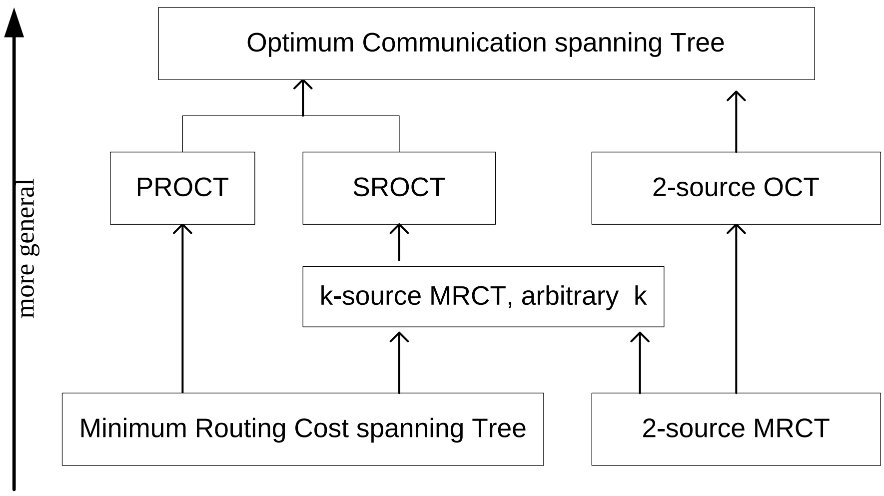

Figure 1: The relationship of the OCT problems.

Table 1: The restrictions and currently best ratios of the OCT problems

problem	restriction	ratio	reference
OCT	no	$O(\log n \log\log n)$	[8]
PROCT	$\lambda(u,v) = r(u)r(v)$	PTAS	[10]
SROCT	$\lambda(u,v) = r(u) + r(v)$	2	[9]
MRCT	$\lambda(u,v) = 1$	PTAS	[8]
2-source MRCT	$\lambda(u,v) = r(u) + r(v)$ $r(s_1) = r(s_2) = 1$ $r(v) = 0$ for $v \notin \{s_1, s_2\}$	PTAS	[11]
2-source OCT	$\lambda(u,v) = 0$ for $u, v \notin \{s_1, s_2\}$	2 (metrics) 3 (general graphs)	this paper

We now show the performance ratio in the next lemma, in which T is the spanning tree obtained by the approximation algorithm.

Lemma 2: Let Y be the 2-OCT. For any vertex v, $c_T(v) \le 3c_Y(v)$.

Proof: First we show that the path between s_1 and s_2 is a shortest path. By the algorithm, $d_T(s_1,s_2) = d_T(s_1,p_{i-1}) + w(p_{i-1},p_i) + d_T(p_i,s_2)$. Since T_1 and T_2 are shortest path trees, $d_T(s_1,p_{i-1}) = d_G(s_1,p_{i-1})$ and $d_T(p_i,s_2) = d_G(p_i,s_2)$. Therefore $d_T(s_1,s_2) = w(P) = d_G(s_1,s_2)$.

Any vertex may be in either T_1 or T_2. There are two cases.

- **Case** $d_G(v,s_1) \le d_G(v,s_2)$**:** Since $d_G(s_1,s_2) \le d_G(v,s_1) + d_G(v,s_2)$ by the triangle inequality of shortest path lengths,

$$\begin{aligned} & c_T(v) \\ \le\ & (r_1(v)+r_2(v))d_G(v,s_1) \\ & +r_2(v)d_G(s_1,s_2) \\ \le\ & (r_1(v)+r_2(v))d_G(v,s_1) \\ & +r_2(v)(d_G(v,s_1)+d_G(v,s_2)) \\ =\ & r_1(v)d_G(v,s_1)+r_2(v)d_G(v,s_2) \\ & +2r_2(v)d_G(v,s_1) \end{aligned}$$

  Since $d_G(v,s_1) \le d_G(v,s_2)$, we have

$$\begin{aligned} & c_T(v) \\ \le\ & r_1(v)d_G(v,s_1)+r_2(v)d_G(v,s_2) \\ & +2r_2(v)d_G(v,s_2) \\ \le\ & 3(r_1(v)d_G(v,s_1)+r_2(v)d_G(v,s_2)) \end{aligned}$$

  By Lemma 1, $c_Y(v) \ge r_1(v)d_G(v,s_1) + r_2(v)d_G(v,s_2)$, and we have $c_T(v) \le 3c_Y(v)$ for any vertex v in T_1.

- **Case** $d_G(v,s_1) > d_G(v,s_2)$**:** Similarly,

$$\begin{aligned} & c_T(v) \\ \le\ & (r_1(v)+r_2(v))d_G(v,s_2) \end{aligned}$$

$$
\begin{aligned}
& +r_1(v)d_G(s_1,s_2) \\
\leq\ & (r_1(v)+r_2(v))d_G(v,s_2) \\
& +r_1(v)(d_G(v,s_1)+d_G(v,s_2)) \\
=\ & r_1(v)d_G(v,s_1)+r_2(v)d_G(v,s_2) \\
& +2r_1(v)d_G(v,s_2) \\
\leq\ & r_1(v)d_G(v,s_1)+r_2(v)d_G(v,s_2) \\
& +2r_1(v)d_G(v,s_1) \\
\leq\ & 3c_Y(v)
\end{aligned}
$$

We have that $c_T(v) \leq 3c_Y(v)$ for any vertex v in T_2.

□

The following theorem summarizes our result for the 2-OCT problem on general graphs.

Theorem 3: In $O(n^2)$ time, the algorithm **Approx1** computes a 3-approximation of the 2-OCT of a general graph.

Proof: By Lemma 2, $c_T(v) \leq 3c_Y(v)$ for any vertex v. Since $c(T) = \sum_v c_T(v)$, $c(T) \leq 3c(Y)$. For the time complexity, the shortest path forest can be found by an algorithm similar to the one for the shortest path trees. The time complexity is $O(n^2)$ [2, 1]. Since all other steps can be done in $O(n^2)$ obviously, the total time complexity is $O(n^2)$. □

4 On metric inputs

In this section, we give a 2-approximation algorithm for the 2-OCT problem with metric inputs. A metric graph is a complete graph and the edge between any pair of vertices is a shortest path. Our algorithm greedily connects the vertices to one of the sources. In each iteration, we connect a vertex v to either s_1 or s_2 depending on which has the smaller communication cost. By the properties of metric graphs and the 2-OCT problem, we shall show the performance ratio is two in the worst case. The algorithm is given in the following.

Algorithm Approx2
Input: A metric graph G, two vertices s_1, s_2, and requirements $r_1(v), r_2(v)$.
Output: A spanning tree T of G.
Initially T contains only one edge (s_1, s_2).
For each vertex $v \in V$ do
/* Connect each v to either s_1 or s_2 */
  If $(r_1(v)+r_2(v))w(v,s_1)+r_2(v)w(s_1,s_2)$
  $\leq (r_1(v)+r_2(v))w(v,s_2)+r_1(v)w(s_1,s_2)$
    Insert edge (v, s_1) into T
  else
    Insert edge (v, s_2) into T
  endif
Output T.

For convenience, we first define some notations.

Definition 4: Let Y be the 2-OCT and P be the path between s_1 and s_2 on Y. We define $f_1(v) = d_Y(v,s_1) - d_Y(v,P)$ and $f_2(v) = d_Y(v,s_2) - d_Y(v,P)$ for each each vertex v.

The next lemma gives a formula of the optimal cost. The formula directly follows the above notations and the definition of the communication cost. We omit the proof.

Lemma 4: Let Y be the 2-OCT and P be the path between s_1 and s_2 on Y. $c(Y) = \sum_{v \in V}((r_1(v)+r_2(v))d_Y(v,P)+r_1(v)f_1(v)+r_2(v)f_2(v)$.

The next theorem shows the result of the approxiamtion algorithm.

Theorem 5: Except for the time of input, the algorithm **Approx2** computes a 2-approximation of the 2-OCT of a metric graph in $O(n)$ time.

Proof: The time complexity is obvious and we shall show the performance ratio. Let Y be the 2-OCT and P be the path between s_1 and s_2 on Y. By the triangle inequality, $w(v,s_1) \leq d_Y(v,s_1) = d_Y(v,P) + f_1(v)$. We have

$$
\begin{aligned}
& (r_1(v)+r_2(v))w(v,s_1)+r_2(v)w(s_1,s_2) \\
\leq\ & (r_1(v)+r_2(v))(d_Y(v,P)+f_1(v)) \\
& +r_2(v)(f_1(v)+f_2(v)) \\
=\ & (r_1(v)+r_2(v))d_Y(v,P) \\
& +(f_1(v)r_1(v)+f_2(v)r_2(v))+2f_1(v)r_2(v) \\
=\ & c_Y(v)+2f_1(v)r_2(v)
\end{aligned}
$$

Similarly $w(v,s_2) \leq d_Y(v,s_2) = d_Y(v,P) + f_2(v)$ and we have

$$
\begin{aligned}
& (r_1(v)+r_2(v))w(v,s_2)+r_1(v)w(s_1,s_2) \\
\leq\ & (r_1(v)+r_2(v))(d_Y(v,P)+f_2(v))
\end{aligned}
$$

$$
\begin{aligned}
& +r_1(v)(f_1(v)+f_2(v)) \\
= \; & (r_1(v)+r_2(v))d_Y(v,P) \\
& +(f_1(v)r_1(v)+f_2(v)r_2(v))+2f_2(v)r_1(v) \\
= \; & c_Y(v)+2f_2(v)r_1(v)
\end{aligned}
$$

Since the vertex v is connected to either s_1 or s_2 by choosing the minimum of the two costs,

$$
\begin{aligned}
& c_T(v) \\
= \; & \min\{(r_1(v)+r_2(v))w(v,s_1)+r_2(v)w(s_1,s_2), \\
& (r_1(v)+r_2(v))w(v,s_2)+r_1(v)w(s_1,s_2)\} \\
= \; & c_Y(v)+\min\{2f_1(v)r_2(v), 2f_2(v)r_1(v)\}
\end{aligned}
$$

Since the minimum of two number is no more than their weighted mean, we have

$$
\begin{aligned}
& c_Y(v)+\min\{2f_1(v)r_2(v), 2f_2(v)r_1(v)\} \\
\leq \; & c_Y(v)+\frac{r_1(v)^2}{r_1(v)^2+r_2(v)^2}2f_1(v)r_2(v) \\
& +\frac{r_2(v)^2}{r_1(v)^2+r_2(v)^2}2f_2(v)r_1(v) \\
= \; & c_Y(v)+\frac{2r_1(v)r_2(v)}{r_1(v)^2+r_2(v)^2} \\
& \times(f_1(v)r_1(v)+f_2(v)r_2(v))
\end{aligned}
$$

Since $r_1(v)^2+r_2(v)^2-2r_1(v)r_2(v)=(r_1(v)-r_2(v))^2 \geq 0$, we have

$$
\begin{aligned}
c_T(v) & \leq c_Y(v)+f_1(v)r_1(v)+f_2(v)r_2(v) \\
& \leq 2c_Y(v)
\end{aligned}
$$

We have shown that $c_T(v) \leq 2c_Y(v)$ for any vertex v. Therefore $c(T)=\sum_{v\in V} c_T(v) \leq 2\sum_{v\in V} c_Y(v) = 2c(Y)$, and T is a 2-approximation of the optimal.

□

5 Concluding remarks

In this paper, we present approximation algorithms for the two-source optimum communication spanning trees. An interesting open question is the approximability of the problem. Another interesting problem is if the technique developed in this paper can be extended to k-source OCT for any fixed integer k.

Acknowledgements

The work was partially supported by grant NSC 90-2213-E-366-005 from the National Science Council.

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

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] B.Y. Wu, A polynomial time approximation scheme for the two-source minimum routing cost spanning trees, manuscript.