Approximation algorithms for k-source bottleneck routing cost spanning tree problems (extended abstract)

Yen Hung Chen[1], Bang Ye Wu[2], and Chuan Yi Tang[1]

[1] Department of Computer Science, National Tsing Hua University, Hsinchu 300, Taiwan, R.O.C.
`{dr884336, cytang}@cs.nthu.edu.tw`

[2] Department of Computer Science and Information Engineering, Shu-Te University, YenChau Kaoshiung 824, Taiwan, R.O.C.
`bangye@mail.stu.edu.tw`

Abstract. In this paper, we investigate two spanning tree problems of graphs with k given sources. Let $G = (V, E, w)$ be an undirected graph with nonnegative edge lengths and $S \subset V$ a set of k specified sources. The first problem is the *k-source bottleneck vertex routing cost spanning tree* (k-BVRT) problem, in which we want to find a spanning tree T such that the maximum total distance from any vertex to all sources is minimized, i.e., we want to minimize $\max_{v \in V} \left\{\sum_{s \in S} d_T(s, v)\right\}$, in which $d_T(s, v)$ is the length of the path between s and v on T. The other problem is the *k-source bottleneck source routing cost spanning tree* (k-BSRT) problem, in which the objective function is the maximum total distance from any source to all vertices, i.e., $\max_{s \in S} \left\{\sum_{v \in V} d_T(s, v)\right\}$. In this paper, we present a polynomial time approximation scheme (PTAS) for the 2-BVRT problem. For the 2-BSRT problem, we first give $(2 + \varepsilon)$-approximation algorithm for any $\varepsilon > 0$, and then present a PTAS for the case that the input graphs are restricted to metric graphs. Finally we show that there is a simple 3-approximation algorithm for both the two problems with arbitrary k.

Keywords: combinatorial optimization problem, spanning tree, approximation algorithm, polynomial time approximation scheme

1 Introduction

Finding spanning trees of a given graph is an important problem in network design. Depending on the applications, problems are defined by different objectives. For example, a minimum spanning tree is the spanning tree of minimum total edge weight, and the objective function of the *minimum routing cost spanning tree* (MRCT) [9] is the total distance summed over all pairs of vertices.

Motivated by the applications of multicasting and broadcasting, several multi-source spanning tree problems have been studied [2, 4, 10]. In such problems, we are given an undirected graph $G = (V, E, w)$ with nonnegative edge length function w and $S \subset V$ a set of k specified sources, and asked for the spanning tree minimizing some distance related cost metric. The k-source MRCT (k-MRCT), also called as the k-source shortest path spanning tree, is the spanning tree T minimizing $\sum_{s\in S}\{\sum_{v\in V} d_T(s,v)\}$, where $d_T(s,v)$ is the length of the path between s and v in T. If there is only one source, the problem can be solved by finding the *shortest path tree*, a spanning tree in which 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 (see [3] or other text books for algorithms). The most efficient algorithms run in $O(|E|+|V|\log|V|)$ time on graphs with nonnegative edge lengths. And Thorup gave an $O(|E|)$ time algorithm for graphs with positive integer lengths [7].

In the case of more than one sources, the k-MRCT problem is a generalization of the MRCT problem (also called the *shortest total path length spanning tree* problem, [ND3] in [5]), in which all vertices are sources. The MRCT problem is NP-hard [5, 6] and admits a *polynomial time approximation scheme* (PTAS) [9]. The k-MRCT problem for general k is obviously NP-hard since it includes the MRCT problem as a special case. However, the NP-hardness of the MRCT does not imply the complexity of the k-MRCT problem for a fixed constant k. The NP-hardness of the 2-MRCT problem was shown independently in [4] and [10]. In the latter paper, a PTAS was also proposed. Recently, the k-MRCT problem for any constant k was also shown to be NP-hard [11] and a 2-approximation algorithm can be found in the previous work of the author [8].

While the k-MRCT problem is defined by a min-sum objective function, Connamacher and Proskurowski posted two variants of k-MRCT problem with bottleneck objective functions. They are the *k-source maximum vertex shortest paths spanning tree* (k-MVST) problem and the *k-source maximum source shortest paths spanning tree* (k-MSST) problem [2]. The objective function of the k-MVST and k-MSST problems are $\max_{v\in V}\{\sum_{s\in S} d_T(s,v)\}$ and $\max_{s\in S}\{\sum_{v\in V} d_T(s,v)\}$ respectively. Both the problems are shown to be NP-Complete in the strong sense even for $k=2$ [2]. To exhibit the objective and associate with the previous work, we shall use the name *k-source bottleneck vertex routing cost spanning tree* (k-BVRT) for k-MVST and *k-source bottleneck source routing cost spanning tree* (k-BSRT) for k-MSST.

In this paper, we focus on the approximation algorithms for the two problems. For the 2-BVRT problem, we present a PTAS. For the 2-BSRT problem, we show a $(2+\varepsilon)$-approximation algorithm for the general graphs, and a PTAS for metric graphs. A metric graph is a complete graph in which the edge lengths obeying the triangle inequality. Finally, for any arbitrary $k>2$, we give simple 3-approximation algorithms for both the two problems.

2 Preliminaries

In this paper, a graph is simple, connected and undirected. 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. 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 $|V(G)|$. The following notations are used in this paper. 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 $d_G(u, v) = w(SP_G(u, v))$. Let H be a subgraph of G. For a vertex $v \in V(G)$, we let $d_G(v, H)$ 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. Let T be a spanning tree of G. For $v \in V$ and a vertex set $R \subset V$, we use $D_T(v, R)$ to denote the total distance from vertex v to R, i.e., $D_T(v, R) = \sum_{r \in R} d_T(v, r)$.

Definition 1. *Let T be a tree and $S \subset V(T)$ a set of k sources. The* bottleneck vertex routing cost *of T, denoted by $C_V(T)$, is the maximum total distances from any vertex to all sources, i.e.,* $\max_{v \in V} D_T(v, S)$. *Given a graph and a set of k sources, the* bottleneck vertex routing cost spanning tree *(k-BVRT) problem is to find a spanning tree T with minimum $C_V(T)$ among all possible spanning trees.*

Definition 2. *Let T be a tree and a set of k sources $S \subset V(T)$. The* bottleneck source routing cost *of T, denoted by $C_S(T)$, is the maximum total distances from any source to all vertices, i.e.,* $\max_{s \in S} D_T(s, V)$. *Given a graph and a set of k sources, the* bottleneck source routing cost spanning tree *(k-BSRT) problem is to find a spanning tree T with minimum $C_S(T)$ among all possible spanning trees.*

Let G be a graph and $r \in V(G)$. A *shortest path tree* rooted at r is a spanning tree T of G such that $d_T(r, v) = d_G(r, v)$ for each vertex $v \in V$. That is, on a shortest path tree, the path from the root to any vertex is a shortest path in G. Let $R \subset V$. A forest F is a *shortest path forest* with multiple roots R if $d_F(v, R) = d_G(v, R)$ for any vertex $v \in V$, i.e., each vertex is connected to the closest root by a shortest path. A shortest path forest can be constructed by an algorithm similar to the shortest path tree algorithm. The time complexity is the same as the shortest path tree algorithm.

3 A PTAS for the 2-BVRT Problem

In this section, we show a PTAS for the 2-BVRT problem. Throughout this section, we assume that s_1 and s_2 are the two given sources. The PTAS is a modification of the one designed in [10] for the 2-MRCT. First we show that the simple 2-approximation algorithm for the 2-MRCT also works for the 2-BVRT problem. The 2-approximation algorithm first finds a shortest path between the two sources, and then constructs a shortest path forest with all the vertices of the path as the multiple roots. It returns the union of the path and the forest as the approximation solution. The following property is useful for showing the performance of the algorithm.

Lemma 1. *Let T be any spanning tree of $G = (V, E, w)$ and P be the path between s_1 and s_2 on T. $C_V(T) = w(P) + 2\max_{v \in V}\{d_T(v, P)\}$.*

Proof. First, $D_T(v,S) = d_T(v,s_1)+d_T(v,s_2) = w(P)+2d_T(v,P)$ for any $v \in V$. Since $C_V(T)$ is the maximum $D_T(v,S)$ over all vertices in V, the result follows. $\square$

Now we show the performance of the simple algorithm.

Lemma 2. *A 2-approximation of the 2-BVRT can be found in $O(|V|\log|V| + |E|)$ time.*

Proof. First we show a lower bound of the optimum. Let Y and T be the optimal and approximation trees of the 2-BVRT problem. For any vertex $v \in V$, we have

$$C_V(Y) = \max_{u\in V}\{d_Y(u,s_1) + d_Y(u,s_2)\} \geq d_G(v,s_1) + d_G(v,s_2). \tag{1}$$

Since $d_G(s_1,s_2) \leq d_G(v,s_1) + d_G(v,s_2)$, we have $C_V(Y) \geq d_G(s_1,s_2)$. Together with Eq. (1), for any vertex v, we have $C_V(Y) \geq \frac{1}{2}(d_G(s_1,s_2) + d_G(v,s_1) + d_G(v,s_2))$. Since P is a shortest path between the two sources and each vertex is connected to P by a shortest path, for any vertex $v \in V$, we have $d_T(v,P) \leq \min\{d_G(v,s_1), d_G(v,s_2)\} \leq \frac{1}{2}(d_G(v,s_1) + d_G(v,s_2))$, and hence $2d_T(v,P) + d_T(s_1,s_2) \leq (d_G(v,s_1) + d_G(v,s_2)) + d_T(s_1,s_2) = (d_G(v,s_1) + d_G(v,s_2)) + d_G(s_1,s_2) \leq 2C_V(Y)$. By Lemma 1, we have $C_V(T) \leq 2C_V(Y)$ and T is a 2-approximation solution of the 2-BVRT. Note that the time-complexity is dominated by the construction of the shortest path forest. $\square$

Now, we describe the PTAS and its performance analysis. By Lemma 1, it is easy to see that one can find the optimal if the path between the two sources is given. Similar to the previous PTAS for the 2-MRCT, our PTAS tries to guess some vertices of the path. For each guess, it first constructs a tree X spanning those guessed vertices, and then extends X to all other vertices by adding shortest paths to X. The performance is ensured by showing that at least one of the constructed trees is a good approximation solution and the approximation ratio approaches to 1 as the number of guessed vertices increasing. Although the algorithm is very similar to the previous one, the analysis of performance is different.

Let $\rho \geq 0$ be an integer parameter to be determined later. In the remaining paragraphs of this section, let Y be the optimal tree of the 2-BVRT. Also let $P = (p_1 = s_1, p_2, p_3, \ldots, p_h = s_2)$ be the path between the two sources on Y. The next lemma shows that we can choose only few vertices on P such that, for each vertex on P, there is a chosen vertex which is close enough to it. The lemma can be shown by induction on ρ but the proof is omitted in this abstract.

Lemma 3. *For any integer $\rho \geq 0$, there exists a subset $M \subset V(P)$ such that $|M| \leq \rho$ and $d_Y(v, M \cup \{s_1,s_2\}) \leq \frac{1}{\rho+2}w(P)$ for any $v \in V(P)$.*

The bound in the above lemma is tight. An extreme example is that the path have exactly $\rho+1$ internal vertices and all the edges are of the same length. Since we can only choose ρ vertices, there is one vertex left and the distance to its neighbor is $w(P)/(\rho+2)$.

Let M be a vertex set satisfying the property in Lemma 3 and $\bar{M} = M \cup \{s_1, s_2\}$. The next lemma can be shown by Lemma 3 but the proof is omitted here.

Lemma 4. *For any vertex v, $d_G(v, \bar{M}) \leq d_Y(v, P) + \frac{1}{\rho+2} w(P)$.*

By Lemma 4, we can design a good approximation algorithm for the 2-BVRT if the input graph is a metric graph. A metric graph is a complete graph with edge lengths satisfying the triangle inequality. In a metric graph, the edge between any pair of vertices is a shortest path. For a metric graph and an integer ρ, we try each possible path $(s_1, m_1, m_2, \ldots, m_i, s_2)$ for $i \leq \rho$. There exists at least one such paths whose internal vertices are also on the path P of the optimal tree in the same order and satisfy the property in Lemma 4. Since the graph is a metric graph, the length of the path is no more than $w(P)$. Connecting all other vertices to the path by a shortest path forest, we may have a spanning tree and it is a $(\rho + 4)/(\rho + 2)$-approximation of the 2-BVRT (shown later). However, a problem is encountered when the input is a general graph instead of a metric graph. For a guessed i-tuple $(m_1, m_2, \ldots, m_i)$ of vertices, there is no obvious way to construct such a desired path. We overcome the difficulty by a technique developed in [10] for 2-MRCT.

Lemma 5. *Suppose that P is a path on a general graph. Let s_1 and s_2 be the two endpoints and m_1, m_2,… ,m_i be i vertices such that P connects the consecutive m_j. Given the two endpoints of P and the i-tuple $(m_1, m_2, \ldots, m_i)$, there is an $O(in^2)$ time algorithm which constructs a tree X spanning the given vertices and having the property that $d_X(v, s_1) + d_X(v, s_2) \leq w(P)$ for any $v \in V(X)$.*

We list the PTAS below.

ALGORITHM PTAS-2-BVRT
Input: A graph $G = (V, E, w)$ and $s_1, s_2 \in V$, and an integer $\rho \geq 0$.
Output: A spanning tree T of G.
1: For each $i \leq \rho$ and i-tuple $(m_1, m_2, \ldots m_i)$ of vertices **do**
  Find a tree X as in Lemma 5.
  Find the shortest path forest spanning $V(G)$ with all vertices in $V(X)$ as roots.
  Let T be the union of the forest and X.
 end for
2: Output the tree T with minimum $C_V(T)$ among all constructed trees.

The performance of the PTAS is shown in the next lemma.

Lemma 6. *The algorithm returns a spanning tree T with $C_V(T) \leq (\frac{\rho+4}{\rho+2}) C_V(Y)$*

Proof. Since the algorithm tries all possible i-tuple of vertices for all $i \leq \rho$, it is sufficient to show the approximation ratio for the i-tuple satisfying the property in Lemma 4. Let M be the set of vertices in the i-tuple and $\bar{M} = M \cup \{s_1, s_2\}$.

As in Lemma 5, let X be a tree spanning M and $d_X(v,s_1)+d_X(v,s_2) \leq w(P)$ for any $v \in V(X)$.

For each vertex v, since it is connected to X by a shortest path, there exists a vertex $x \in V(X)$ such that $d_T(v,s_1) = d_G(v,x) + d_T(x,s_1)$ and $d_T(v,s_2) = d_G(v,x)+d_T(x,s_2)$. Therefore $D_T(v,S) = 2d_G(v,X)+d_T(x,s_1)+d_T(x,s_2)$. By Lemma 5, we have $D_T(v,S) \leq 2d_G(v,X)+w(P)$. Since $\bar{M} \subset V(X)$, $d_G(v,X) \leq d_G(v,\bar{M})$, and then by Lemma 4 we have $d_G(v,X) \leq d_Y(v,P) + \frac{1}{\rho+2}w(P)$. Consequently we have, for any vertex v,

$$D_T(v,S) \leq 2d_Y(v,P) + \frac{2}{\rho+2}w(P) + w(P) \leq \left(1+\frac{2}{\rho+2}\right) D_Y(v,S),$$

since $w(P) \leq D_Y(v,S)$. Then the result is obtained by definition. □

The result of this section is summarized in the following theorem. The theorem can be easily shown by taking $\rho = \lceil \frac{2}{\varepsilon} \rceil - 2$ in the above lemma and the proof is omitted here.

Theorem 1. *The 2-BVRT problem admits a PTAS. For any constant $\varepsilon > 0$, a (1+ε)- approximation algorithm of the 2-BVRT of a graph G can be found in $O(n^{\lceil \frac{2}{\varepsilon} \rceil})$ time.*

4 The 2-BSRT problem

In the section, we discuss the 2-BSRT problem. In the 2-BSRT problem, we are given a graph $G = (V,E,w)$ with two source vertices and asked to find a spanning tree. The objective function of the problem is $C_S(T) = \max_{s \in S}\{\sum_{v \in V} d_T(s,v)\}$. First we shall consider the case that the input is a general graph, and give a $(2+\varepsilon)$-approximation algorithm for any fixed $\varepsilon > 0$. Then we show that the problem admits a PTAS if the input is restricted to a metric graph. Throughout this section, we assume that s_1 and s_2 are the two given sources.

4.1 On general graphs

First we present a $(2+\varepsilon)$-approximation algorithm on general graphs. The algorithm is basically the same as algorithm **PTAS-2-BVRT** except that, among all the constructed trees, it returns the tree T with minimum $C_S(T)$ instead of $C_V(T)$. The main idea is similar to the PTAS for the 2-BVRT, but we need a different analysis of the performance. First we establish a lower bound of the cost in the next lemma. But omit the proof in this abstract.

Lemma 7. *Let T be a spanning tree of $G = (V,E,w)$ and P be the path between s_1 and s_2 on T. $C_S(T) \geq \frac{n}{2}w(P) + \sum_{v \in V} d_T(v,P)$.*

In the remaining paragraphs of this section, we shall use the following notations. Let Y be the optimal tree of the 2-BSRT. Also 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. Now we introduce a partition of the

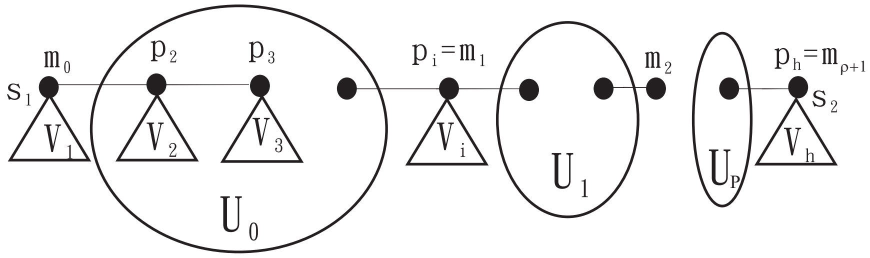

Fig. 1. The definitions of the partition of the vertices.

vertices, which appeared in the previous work for the 2-MRCT problem (Fig. 1).

Define V_i, $1 \leq i \leq h$, as the set of the vertices connected to P at p_i and also let $p_i \in V_i$. Let $\rho \geq 0$ be an integer parameter to be determined later. For $0 \leq i \leq \rho+1$, define $m_i = p_j$ in which j is the minimal index such that $\sum_{1\leq q\leq j} |V_q| \geq \lceil i\frac{n}{\rho+1}\rceil$. By definition, $s_1 = m_0$ and $s_2 = m_{\rho+1}$. For $0 \leq i \leq \rho$, let $U_i = \bigcup_{a<j<b} V_j$ and P_i be the path from p_a and p_b, in which $p_a = m_i$ and $p_b = m_{i+1}$. Also let $U = V \setminus \bigcup_{0\leq i\leq\rho} U_i$ and $M = \{m_i : 0 \leq i \leq \rho+1\}$. 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. By the above definitions, the vertex set V is partitioned into $U, U_0, U_1, \ldots, U_\rho$ satisfying the following properties. The lemma comes from [10]. We state the result but the proof is omitted.

Lemma 8. $\sum_{v\in U_i} d_G(v,M) \leq \sum_{v\in U_i} d_Y(v,P) + \frac{n}{2(\rho+1)} w(P_i)$, *for any* $0 \leq i \leq \rho$ *and* $\sum_{v\in U} d_G(v,M) \leq \sum_{v\in U} d_Y(v,P)$.

Our algorithm tries to correctly guess $m_1, m_2, \ldots m_\rho$ and construct a tree X spanning these vertices with the property that $d_X(v,s_1) + d_X(v,s_2) \leq w(P)$ for any $v \in V(X)$. Once such a tree X has been constructed, we extend it to a spanning tree T by adding the shortest path forest with vertices in $V(X)$ as the multiple roots. In the next lemma, we show that T is a $(2+\frac{1}{\rho+1})$-approximation of the 2-BSRT.

Lemma 9. *Let X be a tree spanning M and $d_X(v,s_1) + d_X(v,s_2) \leq w(P)$ for any $v \in V(X)$. The spanning tree T, which is the union of X and the shortest path forest with vertices in $V(X)$ as the multiple roots, is a $(2+\frac{1}{\rho+1})$-approximation solution of the 2-BSRT problem.*

Proof. (Outline) By definition, the vertices $m_1, m_2, \ldots, m_\rho$ partition the vertex set V into $(U, U_0, U_1, \ldots, U_\rho)$ and the subsets satisfy the properties in Lemma 8. Since $d_X(v,s_1) + d_X(v,s_2) \leq w(P)$ for any $v \in V(X)$, we have

$$C_S(T) \leq \sum_{v\in U} d_T(v,X) + \sum_{0\leq i\leq\rho} \sum_{v\in U_i} d_T(v,X) + n\, w(P). \tag{2}$$

For $v \in U$, since $d_T(v,X) = d_G(v,X) \leq d_G(v,M)$ and by Lemma 8, we have

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

Similarly $d_T(v,X) \leq d_G(v,M)$ for $v \in U_i$, and then by Lemma 8, we have

$$\sum_{0\leq i\leq \rho} \sum_{v\in U_i} d_T(v,X) \leq \sum_{0\leq i\leq \rho} \sum_{v\in U_i} d_Y(v,P) + \frac{n}{2(\rho+1)} w(P). \tag{4}$$

By Eqs. (2)–(4) and Lemma 7, we can obtain

$$C_S(T) \leq C_S(Y) + \left(\frac{\rho+2}{\rho+1}\right) C_S(Y) = \left(2 + \frac{1}{\rho+1}\right) C_S(Y).$$

□

Theorem 2. *For any constant $\varepsilon > 0$, there is an algorithm finding a $(2+\varepsilon)$-approximation of the 2-BSRT of a general graph in $O(n^{\lceil \frac{1}{\varepsilon}+1 \rceil})$ time.*

4.2 On metric graphs

Although the 2-BVRT problem and the 2-BSRT problem seem very similar, they are actually different. In the 2-BVRT problem, by Lemma 1, if the path between the two sources is given, the optimal tree can be easily obtained by connecting all other vertices to their closest vertices on the path. However, it does not hold for the 2-BSRT problem. There is no obvious way to find an optimal even the path is given. For this reason, the 2-BSRT seems more difficult to be approximated than the 2-BVRT. In the last section, we show how to approximate the 2-BSRT on general graphs within approximation ratio $2+\varepsilon$. We shall show that the 2-BSRT admits a PTAS if the input graph is restricted to a metric graph. The key point is that we employ an algorithm to find how to connect vertices to the vertices of the path so as to minimize the cost. The algorithm runs in polynomial time if the number of vertices of the path is constant.

The main idea of the PTAS is similar to the algorithm for the problem on general graphs. The definitions of Y, P, P_i, V_i, M, U, $U_0, U_1 \ldots U_\rho$ are also the same as in the previous section. To show the performance, we first construct a spanning tree $\bar{Y}$ from the optimal tree Y as follows. Note that the tree $\bar{Y}$ does not appear in the PTAS. It is used to show the performance of our PTAS.

- Let $\bar{P}$ be the path $(s_1 = m_0, m_1, m_2, \ldots, m_{\rho+1} = s_2)$. Since the graph is a metric graph, the path exists.
- For each $v \in V_j \subset U$ ($p_j \in M$), v is connected to p_j, i.e., edge $(v, p_j) \in E(\bar{Y})$.
- For each $v \in U_i$, v is connected to either m_i or m_{i+1} depended on which is closer to v on Y, i.e., edge $(m_i, v) \in E(\bar{Y})$ if $d_Y(v, m_i) \leq d_Y(v, m_{i+1})$, and $(m_{i+1}, v) \in E(\bar{Y})$ otherwise. Note that decision is made by the distances on Y but not on the original graph.

We shall show that $\bar{Y}$ is a $(\frac{\rho+3}{\rho+1})$-approximation of the 2-BSRT Y.

Lemma 10. *For* $s \in \{s_1, s_2\}$ *and* $0 \leq i \leq \rho$, $D_{\bar{Y}}(s, U_i) \leq D_Y(s, U_i) + \frac{n}{(\rho+1)} w(P_i)$

Proof. We show the result for s_1. The case of s_2 can be shown similarly. First, since the graph is a metric graph, $d_{\bar{Y}}(s_1, m) \leq d_Y(s_1, m)$ for any $m \in M$. For each $v \in V_j \subset U_i$, v is connected to either m_i or m_{i+1}. When v is connected to m_i, it is clear that $d_{\bar{Y}}(v, s_1) \leq d_Y(v, s_1)$ since the path in Y is just replaced with consecutive short-cut edges. Now we consider the case that v is connected to m_{i+1}. It only happens when $d_Y(v, m_{i+1}) < d_Y(v, m_i)$. Since $d_Y(v, m_i) = d_Y(v, p_j) + d_Y(p_j, m_i)$ and $d_Y(v, m_{i+1}) = d_Y(v, p_j) + d_Y(p_j, m_{i+1})$, the condition implies $d_Y(p_j, m_{i+1}) < d_Y(p_j, m_i)$, or equivalently $d_Y(p_j, m_{i+1}) < w(P_i)/2$. We have

$$\begin{aligned} d_{\bar{Y}}(v, s_1) &= d_{\bar{Y}}(s_1, m_{i+1}) + w(m_i, v) \\ &\leq ((d_Y(s_1, p_j) + d_Y(p_j, m_{i+1})) + (d_Y(v, p_j) + d_Y(p_j, m_{i+1})) \\ &= d_Y(v, s_1) + 2d_Y(p_j, m_{i+1}) \leq d_Y(v, s_1) + w(P_i). \end{aligned}$$

Consequently $d_{\bar{Y}}(v, s_1) \leq d_Y(v, s_1) + w(P_i)$ in both the cases. Since $|U_i| \leq \frac{n}{\rho+1}$, we have

$$\begin{aligned} D_{\bar{Y}}(s_1, U_i) &\leq \sum_{v \in U_i} (d_Y(v, s_1) + w(P_i)) \leq \sum_{v \in U_i} d_Y(v, s_1) + \frac{n}{(\rho+1)} w(P_i) \\ &= D_Y(s_1, U_i) + \frac{n}{(\rho+1)} w(P_i). \end{aligned}$$

□

The next lemma gives the approximation ratio of $\bar{Y}$, which can be shown by Lemma 10 and Lemma 7.

Lemma 11. *The spanning tree* $\bar{Y}$ *is a* $(\frac{\rho+3}{\rho+1})$*-approximation solution of* Y.

Proof. (Outline) For any vertex $v \in V_j \subset U$, it is connected to $p_j \in M$ in $\bar{Y}$, and $d_{\bar{Y}}(v, s_1) \leq d_Y(v, s_1)$. By Lemma 10, $D_{\bar{Y}}(s_1, V) \leq D_Y(s_1, V) + \frac{n}{(\rho+1)} w(P)$. Similarly $D_{\bar{Y}}(s_2, V) \leq D_Y(s_2, V) + \frac{n}{(\rho+1)} w(P)$. By definition, we have $C_S(\bar{Y}) = \max\{D_{\bar{Y}}(s_1, V), D_{\bar{Y}}(s_2, V)\} \leq C_S(Y) + \frac{n}{(\rho+1)} w(P) \leq (1 + \frac{2}{\rho+1}) C_S(Y)$, since $C_S(Y) \geq \frac{n}{2} w(P)$ by Lemma 7. □

We have transformed a 2-BSRT Y into a tree $\bar{Y}$ and have shown that $\bar{Y}$ is a good approximation of Y. Since the input is a metric graph, we may assume that the vertices of P are the only possible internal vertices of Y. As a result, the tree $\bar{Y}$ belongs to a special kind of spanning trees in which there are only $(\rho + 2)$, or less, internal vertices and the internal vertices form a path with the two sources as the endpoints. Our PTAS for the 2-BSRT is designed to find the best spanning tree of the kind.

Definition 3. *A* λ*-path is a path between* s_1 *and* s_2 *and containing exactly* $(\lambda + 2)$ *vertices.*

Definition 4. *A spanning tree T is a λ-path tree if $SP_T(s_1, s_2)$ is a λ-path and all the internal vertices are on the path.*

By the definition and Lemma 11, we have the next corollary.

Corollary 1. *For a metric graph and any integer $\rho \geq 0$, there exists a λ-path tree, $\lambda \leq \rho$, which is a $(\frac{\rho+3}{\rho+1})$-approximation of the 2-BSRT.*

Our PTAS finds the λ-path tree with minimum bottleneck source routing cost for each $\lambda \leq \rho$, and returns the best of them. By Corollary 1, it is a $(\frac{\rho+3}{\rho+1})$-approximation. Now we show how to find the best λ-path tree of a metric graph for a fixed λ. The idea is from [9], which is used to find the minimum routing cost tree with constant number of internal nodes.

A λ-path tree can be described by a λ-path $Q = (s_1 = q_0, q_1, \ldots, q_{\lambda+1} = s_2)$ and a partition $\mathcal{L} = (L_0, L_1, \ldots, L_{\lambda+1})$ of $\bar{V}$, in which $\bar{V} = V \setminus V(Q)$ and L_i is the set of vertices in $\bar{V}$ and adjacent to q_i. We denote a λ-tree by $(Q, \mathcal{L})$. Let $l = (l_0, l_1, \ldots . l_{\lambda+1})$ be a nonnegative $(\lambda+2)$-vector such that $\sum_{i=0}^{\lambda+1} l_i = n - \lambda - 2$. We say that a λ-path tree $(Q, \mathcal{L})$ has the configuration (Q, l) if $l_i = |L_i|$ for all $0 \leq i \leq \lambda + 1$. We define $\alpha(Q, l)$ to be the λ-path tree of minimum bottleneck source routing cost with a configuration (Q, l). We shall show that $\alpha(Q, l)$ can be found in polynomial time.

Lemma 12. *For a given configuration (Q, l), the optimal λ-path tree $\alpha(Q, l)$ can be found in $O(n^3)$ time.*

Proof. (outline) Let $T = (Q, \mathcal{L})$ be a λ-path spanning tree with configuration (Q, l). We show that

$$D_T(s_1, V) = D_T(s_1, V(Q)) + \sum_{i=0}^{\lambda+1} l_i \; d_T(s_1, q_i) + \sum_{i=0}^{\lambda+1} \sum_{v \in L_i} w(v, q_i).$$

Similarly,

$$D_T(s_2, V) = D_T(s_2, V(Q)) + \sum_{i=0}^{\lambda+1} l_i \; d_T(s_2, q_i) + \sum_{i=0}^{\lambda+1} \sum_{v \in L_i} w(v, q_i).$$

Define $f_1(Q, l) = D_T(s_1, V(Q)) + \sum_{i=0}^{\lambda+1} l_i \; d_T(s_1, q_i)$ and $f_2(Q, l) = D_T(s_2, V(Q)) + \sum_{i=0}^{\lambda+1} l_i \; d_T(s_2, q_i)$. Note that the two function depend only on Q and l, and therefore two trees with the same configuration have the same values of the two functions. We can have $C_S(T) = \max\{f_1(Q, l), f_2(Q, l)\} + \sum_{i=0}^{\lambda+1} \sum_{v \in L_i} w(v, q_i)$. For a specified configuration (Q, l), the cost depends only on the second term. As a result, $\alpha(Q, l)$ can be constructed by finding a minimum-cost way of matching up the vertices of $\bar{V}$ to those in $V(Q)$ which obeys the degree constraints imposed by l. It is equivalent to find the minimum-cost perfect matching on the auxiliary complete bipartite graph H with vertex set $\bar{Q} \cup \bar{V}$ in which $\bar{Q}$ contains l_i copies of q_i for each i. Since the bipartite perfect matching problem, also called as the assignment problem, can be solved in $O(n^3)$ time [1], $\alpha(Q, l)$ can be solved with the same time complexity for a given configuration. $\square$

For a λ-path Q, we may find the best λ-path tree by trying every possible nonnegative $(\lambda+2)$-vector l. Since there are $O(n^{\lambda+1})$ such vectors and each vector corresponds to one instance of the assignment problem, it takes $O(n^{\lambda+4})$ time if we solve the assignment problems individually. As in [9], by carefully ordering the assignment problems for the vectors and exploiting the common structure of two consecutive problems, we can obtain the optimal solution of every vector in this order by performing a single augmentation on the optimal solution of the previous vector. By this method, the optimal solution for each vector can be found in $O(\lambda n)$ time from the solution for its predecessor. We state the result in the next lemma but the proof is omitted in this abstract.

Lemma 13. *For any constant $\lambda \geq 0$, the λ-path tree of minimum bottleneck source routing cost can be found in $O(n^{2\lambda+2})$.*

In summary, we show the 2-BSRT problem on metric graphs admits a PTAS.

Theorem 3. *There exists a PTAS for the 2-BSRT problem on metric graphs, which finds a $(1+\varepsilon)$-approximation solution in $O(n^{\lceil 4/\varepsilon \rceil})$ time.*

5 The k-BVRT and the k-BSRT for $k > 2$

In this section we show that both the k-BVRT and the k-BSRT problems can be approximated with ratio 3 by a shortest path tree. First, we show the result for the k-BVRT problem.

Theorem 4. *Any shortest path tree rooted at any vertex is a 3-approximation of the k-BVRT.*

Proof. Let Y be the optimal tree of the k-BVRT, we have

$$C_V(Y) = \max_{v \in V} \{D_Y(v,S)\} \geq \max_{v \in V} \{D_G(v,S)\}.$$

That is, $C_V(Y) \geq D_G(v,S)$ for any vertex v. Let T be any shortest path tree rooted at an arbitrary vertex r.

$$\begin{aligned} C_V(T) &= \max_{v \in V} \left\{ \sum_{s \in S} d_T(v,s) \right\} \leq \max_{v \in V} \left\{ \sum_{s \in S} (d_T(v,r) + d_T(r,s)) \right\} \\ &\leq \max_{v \in V} \left\{ \sum_{s \in S} ((d_G(v,s) + d_G(r,s)) + d_G(r,s)) \right\} \\ &= \max_{v \in V} \{D_G(v,S)\} + 2D_G(r,S) \leq 3C_V(Y). \end{aligned}$$

□

Corollary 2. *Given a general graph and a set of sources, a 3-approximation of the k-BVRT can be found in $O(|V| \log |V| + |E|)$ time.*

Similarly we have the next result for the k-BSRT problem. Note that the shortest path tree must be rooted at a source to ensure the performance, which is different from the k-BVRT problem. The proof is omitted in this abstract.

Theorem 5. *Any shortest path tree rooted at any source is a 3-approximation of the k-BSRT.*

Corollary 3. *Given a general graph and a set of sources, a 3-approximation of the k-BSRT can be found in* $O(|V|\log|V| + |E|)$ *time.*

6 Conclusion

In this paper, we investigate the k-BVRT and the k-BSRT problems and propose approximation algorithms. While the 2-BVRT problem having been shown to admit a PTAS, the 2-BSRT seems more difficult to approximate. We show a PTAS for the 2-BSRT only for metric graphs. For general graph, we only have a $(2+\varepsilon)$-approximation algorithm. Improved approximation algorithms will be interesting. Another open problem left in the paper is the approximabilities of the two problems with more than two sources. The approximation algorithms in this paper are very simple, and better results are expected.

References

1. Ahuja, R.K., Magnanti, T.L., and Orlin, J.B.: Network Flows –Theory, Algorithms, and Applications. Prentice–Hall (1993).
2. Connamacher, H.S. and Proskurowski, A.: The complexity of minimizing certain cost metrics for k-source spanning trees. Discrete Applied Mathematics **131** (2003) 113–127.
3. Cormen, T.H., Leiserson, C.E., Rivest, R.L. and Stein, C.: Introduction to Algorithm. 2nd edition, MIT Press, Cambridge (2001).
4. Farley, A.M., Fragopoulou, P., Krumme, D.W., Proskurowski, A., and Richards, D.: Multi-source spanning tree problems. Journal of Interconnection Networks **1** (2000) 61–71.
5. Garey,M.R., and Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman and Company, San Francisco (1979).
6. Johnson, D.S., Lenstra, J.K., and Rinnooy Kan, A.H.G.: The complexity of the network design problem. Networks **8** (1978) 279–285.
7. Thorup,M.: Undirected single source shortest paths with positive integer weights in linear time. Journal of ACM **46** (1999) 362–394.
8. Wu, B.Y., Chao, K.M., and Tang, C.Y.: Approximation algorithms for some optimum communication spanning tree problems. Discrete Applied Mathematics **102** (2000) 245–266.
9. Wu, B.Y., Lancia, G., Bafna, V., Chao, K.M., Ravi, R., and Tang, C.Y.: A polynomial time approximation scheme for minimum routing cost spanning trees. SIAM Journal on Computing **29** (2000) 761–778.
10. Wu, B.Y.: A polynomial time approximation scheme for the two-source minimum routing cost spanning trees. Journal of Algorithm **44** (2002) 359–378.
11. Wu, B.Y.: Approximation algorithms for the optimal p-source communication spanning tree, to appear in Discrete Applied Mathematics.