DISCRETE APPLIED MATHEMATICS

Discrete Applied Mathematics 105 (2000) 273–289

Approximation algorithms for the shortest total path length spanning tree problem

Bang Ye Wu[a], Kun–Mao Chao[b,*], Chuan Yi Tang[c]

[a] *Chung-Shan Institute of Science and Technology, P.O. Box No. 90008-6-8, Lung-Tan, Taiwan, ROC*
[b] *Department of Life Science, National Yang-Ming University, Taipei, Taiwan 112, ROC*
[c] *Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan, ROC*

Received 30 May 1997; revised 17 December 1999; accepted 3 January 2000

Abstract

Given an undirected graph with a nonnegative weight on each edge, the shortest total path length spanning tree problem is to find a spanning tree of the graph such that the total path length summed over all pairs of the vertices is minimized. In this paper, we present several approximation algorithms for this problem. Our algorithms achieve approximation ratios of 2, 15/8, and 3/2 in time $O(n^2 + f(G))$, $O(n^3)$, and $O(n^4)$ respectively, in which $f(G)$ is the time complexity for computing all-pairs shortest paths of the input graph G and n is the number of vertices of G. Furthermore, we show that the approximation ratio of $(4/3 + \varepsilon)$ can be achieved in polynomial time for any constant $\varepsilon > 0$.

1. Introduction

Consider the following network design problem. We are given an undirected graph with a nonnegative weight on each edge, where the edge weight represents either the distance or the cost of the edge. The goal is to find a spanning tree of the graph such that the total path length summed over all pairs of the vertices is minimized.

The problem is called the *shortest total path length spanning tree* (SPST) problem and was proposed in [8]. The SPST problem is a classical network design problem and was proved to be NP-hard even when all the edge weights are equal [5,9]. For a minimization problem, a k-approximation algorithm is the algorithm which always finds a solution no more than k times of the optimum. In [11], Wong developed a 2-approximation algorithm for the SPST problem. However, to the best of our knowledge, no approximation algorithm achieving better approximation ratio for the SPST problem has been reported.

* Corresponding author.
E-mail address: kmchao@ym.pu.edu.tw (K.-M. Chao).

PII: S0166-218X(00)00185-2

Table 1
Results of this paper

	Approximation ratio	Time complexity
Section 3	2	$O(n^2 + f(G))$
Section 4	15/8	$O(n^3)$
Section 5	3/2	$O(n^4)$
Section 6	4/3+ε	$O(n^\delta)$, $\delta = \lceil (33\varepsilon + 8)/(9\varepsilon) \rceil$

The SPST problem may also have application for the *multiple sequence alignment* problem in computational biology. The multiple sequence alignment problem asks for an alignment optimizing a certain objective function. One of the most objectives, so-called sum-of-pair (SP) function [7], minimizes the sum of all pairwise distances between sequences. For the SP-alignment problem, the first approximation algorithm was due to Gusfield [6], and was improved by Pevzner [10]. The best-known approximation algorithm is due to Bafna et al. [1]. Gusfield's algorithm for the SP-alignment problem is based on the Wong's 2-approximation for the shortest total path length spanning trees. The approximation algorithms presented in this paper may be useful for the SP-alignment problem.

In this paper, we present several approximation algorithms for the SPST problem. Our method differs from Wong's 2-approximation algorithm: estimate the approximation ratio by comparing our solution with the best-possible tree while Wong's proof was based on the total shortest path length of the input graph. The concept in this paper can be summarized as follows: first, for any tree, we show that there exist special subtrees (called *separators*) which can break the tree into sufficiently small components. We then derive a lower bound on the optimal solution by using separators. Secondly, we show that there exists an approximation solution of a special type, called general stars. Our algorithms are based on finding such general stars. Approximation algorithms with different approximation ratios were developed based on different separators. The more precise the separator used, the better the approximation ratio achieved, while the order of runtime is increased.

We summarize the results in Table 1, in which $f(G)$ is the time complexity to compute all-pairs shortest paths of the input graph G and n is the number of vertices of G:

2. Preliminaries

In this paper, a graph $G=(V,E,w)$ is a simple, connected, undirected graph, in which w is a nonnegative edge weight function. For a graph G, $V(G)$ and $E(G)$ denote its vertex set and edge set, respectively. We use n to denote the number of vertices of the input graph. We first give some definitions and formulations:

Definition 1. Let $i,j \in V(G)$ be two vertices of $G=(V,E,w)$. We denote by $SP_G(i,j)$ a shortest path between i and j on G. Let $w(G)=\sum_{e\in E} w(e)$, and $d_G(i,j)=w(SP_G(i,j))$. The *total shortest path length* of G is defined to be $c(G)=\sum_{i,j\in V} d_G(i,j)$.

Definition 2 (*Shortest total path length spanning tree problem (SPST)*). Given a graph $G=(V,E,w)$, find a spanning tree T of G such that $c(T)$ is minimum among all possible spanning trees of G.

We use $SPST(G)$ to denote an optimal spanning tree of the SPST problem of G.

Definition 3. Let S be a subgraph of G and $i \in V(G)$. We use $SP_G(i,S)$ to denote a shortest path from i to S on G. We denote $d_G(i,S)=w(SP_G(i,S))$, that is, $d_G(i,S)=\min_{j\in V(S)}\{d_G(i,j)\}$.

Our approximation algorithm finds a certain spanning tree, called *general star*, of G, which is a generalization of the shortest-paths tree, and is defined as follows:

Definition 4. Let $G=(V,E,w)$ and R be a tree contained in G. T is a *general star of G with core R* if T is a spanning tree of G and $d_T(i,R)=d_G(i,R)\ \forall i \in V$. The set of all general stars of G with core R is denoted by $star(G,R)$.

A general star can be easily constructed by using the method similar to the Dijkstra's algorithm for the shortest-paths tree [2]. For the sake of completeness, we show the time complexity for constructing a general star in the following lemma.

Lemma 1. *Let G be a graph, and let R be a tree contained in G. A spanning tree $T \in star(G,R)$ can be found in $O(n)$ time if shortest paths $SP_G(i,R)$ are given for all $i \in V(G)$.*

Proof. We give a constructive proof. Starting from $T=R$, we insert the other vertices into T one by one. At each iteration, we maintain the equality

$$d_T(i,R)=d_G(i,R) \quad \forall i \in V(T). \tag{1}$$

It is easy to see that the equality (1) is true initially. Let us consider the step of inserting a vertex. Let $SP_G(i,R)=(i=v_1,v_2,\ldots,v_k \in R)$ be a shortest path from i to R, and let v_j be the first vertex which is already in T. We set $T=T\cup(v_1,v_2,\ldots,v_j)$. Since $(v_1,v_2,\ldots,v_k)$ is a shortest path from i to R, $(v_a,v_{a+1},\ldots,v_j)$ is also a shortest path from v_a to v_j for any $a=1,\ldots,j$, and the equality (1) is true. It is easy to see that the time complexity is $O(n)$, if for every $i \in V$, a shortest path from i to R is given. □

Now we are going to define a separator of a tree. Intuitively, for $0<k\leqslant 1/2$, a k-separator is like a "center" of a tree. Starting from any node, there are sufficiently many nodes which cannot be reached without touching the separator.

Definition 5. Let T be a spanning tree of G, and let S be a connected subgraph of T. A *branch* of S is a connected component of the induced subgraph of T with vertices $V(T)-V(S)$. Let $0<k\leqslant 1/2$ be a real number. If $|V(B)|\leqslant kn$ for every branch B

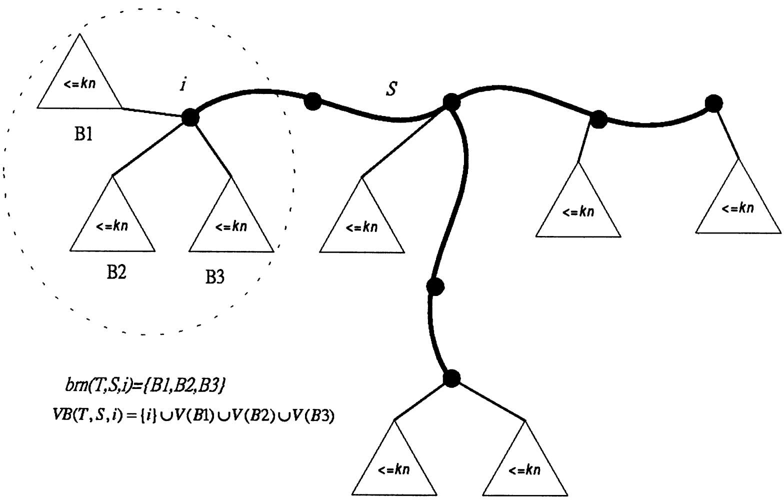

Fig. 1. The k-separator and branches of a tree. The bold line is the separator S and each triangle is a branch of S.

of S, then S is called a *k-separator* of T. A k-separator S is *minimal* if any proper subgraph of S is not a k-separator of T.

We define some notation below and illustrate it in Fig. 1.

Definition 6. Let T be a spanning tree of G and let S be a connected subgraph of T. Let i be a vertex in S. We use $brn(T,S,i)$ to denote the set of branches of S connected to i by an edge of T. We also use $brn(T,S)$ to denote the set of all branches of S. The set of vertices in the branches connected to i is denoted by $VB(T,S,i) = \{i\} \cup \{v \mid v \in B \in brn(T,S,i)\}$.

The separator can be thought as a generalization of the centroid. For any tree, there always exists a vertex such that if we delete the vertex, each resulting subtree contains at most one-half of the vertices. Such a vertex is usually called a *centroid* of a tree. By definition, a centroid is a 1/2-separator of the tree. The following lemma shows that there are sufficiently many vertices which are connected to the leaves of a minimal separator.

Lemma 2. *Let S be a minimal k-separator of T. If i is a leaf of S, then* $|VB(T,S,i)| > k \times |V(T)|$.

Proof. If S contains only one vertex, the result is trivial. If $\sum_{B\in brn(T,S,i)} |V(B)| < k \times |V(T)|$, then S is still a k-separator after deleting i. This is a contradiction to that S is minimal. Therefore we have $|VB(T,S,i)| = \sum_{B\in brn(T,S,i)} |V(B)| + 1 > k \times |V(T)|$. $\square$

Definition 7. Let T be a tree and $S \subset T$. We denote $w_S(T,i,j) = w(SP_T(i,j) \cap S)$ for any $i,j \in V(T)$.

The following lemma gives an upper bound on the total path length of a general star.

Lemma 3. *Let G be a graph and R be a tree contained in G. If $T \in star(G,R)$, then $c(T) \leqslant 2n \sum_{i \in V(G)} d_G(i,R) + (n^2/2)w(R)$.*

Proof. For any $i,j \in V(G)$,

$$\begin{aligned} d_T(i,j) &\leqslant d_T(i,R) + w_R(T,i,j) + d_T(j,R) \\ &= d_G(i,R) + d_G(j,R) + w_R(T,i,j). \end{aligned}$$

For any edge $e \in E(R)$, deleting e from T results in 2 subtrees T_1 and T_2. Let $h(e) = |V(T_1)| \times |V(T_2)|$. It should be noted that $2h(e)$ is the number of vertex pairs whose paths contain e, and $h(e) = |V(T_1)| \times (n - |V(T_1)|) \leqslant n^2/4$, $\forall e \in E(T)$. Therefore,

$$\begin{aligned} c(T) &\leqslant \sum_{i,j \in V} (d_G(i,R) + d_G(j,R) + w_R(T,i,j)) \\ &\leqslant 2 \sum_{i,j \in V} d_G(i,R) + \sum_{i,j \in V} w_R(T,i,j) \\ &\leqslant 2n \sum_{i \in V} d_G(i,R) + \sum_{i,j \in V} w_R(T,i,j). \end{aligned}$$

Notice that $w_R(T,i,j) = w(SP_T(i,j) \cap R)$. We now simplify the second term

$$\begin{aligned} \sum_{i,j \in V} w_R(T,i,j) &= \sum_{i,j \in V} \sum_{e \in E(R) \cap SP_T(i,j)} w(e) \\ &= \sum_{e \in E(R)} 2h(e)w(e) \\ &\leqslant (n^2/2) \sum_{e \in E(R)} w(e) \\ &= (n^2/2)w(R). \end{aligned}$$

This completes the proof. □

3. A simple 2-approximation algorithm

Based on the ideas of separators and general stars, we have developed several approximation algorithms for the SPST problem. In this section, we start with a simple 2-approximation algorithm. For a graph G, a *median* of G is a vertex $m \in V(G)$ such that $\sum_{i \in V(G)} d_G(i,m)$ is minimum. In fact, the 2-approximation algorithm in this section is the same as Wong's algorithm [11]. It returns the shortest-paths tree rooted at a median of the input graph. However, we use a different method to show the approximation ratio, and the method will be generalized in the following sections to obtain better approximation algorithms.

The following lemma shows the existence of a centroid of a tree.

Lemma 4. *For any tree T, there exists a vertex $m \in V(T)$ such that m is a* 1/2-*separator of T.*

Proof. Let w be an edge weight function and $w(e)=1$ for any edge $e \in E(T)$. Let m be a median of T with respect to w. We prove that m is a 1/2-separator of T by contradiction. Assume that m is not a 1/2-separator. Then there must be a branch B with $|V(B)| > n/2$, where $n = |V(T)|$. Let (m, v) be the edge connecting m and B.

$$\begin{aligned}
&\sum_{i \in V} d_T(v,i) \\
&= \sum_{i \in V} d_T(m,i) - \sum_{i \in V(B)} w(m,v) + \sum_{i \in V - V(B)} w(m,v) \\
&= \sum_{i \in V} d_T(m,i) - (|V(B)| - (n - |V(B)|)) \\
&< \sum_{i \in V} d_T(m,i).
\end{aligned}$$

This contradicts to the fact that m is a median of T. □

The next lemma establishes a lower bound on $c(SPST(G))$.

Lemma 5. *Let G be a graph and $\hat{T} = SPST(G)$. There exists a vertex m in $V(G)$ such that $c(\hat{T}) \geqslant n \sum_{i \in V(G)} d_G(i,m)$.*

Proof. By Lemma 4, there exists a vertex m which is a 1/2-separator of $\hat{T}$. We have

$$\begin{aligned}
c(\hat{T}) &= \sum_{i,j \in V} d_{\hat{T}}(i,j) \\
&= \sum_{B \in brn(\hat{T},m)} \sum_{i \in V(B)} \left(\sum_{j \in V(B)} d_{\hat{T}}(i,j) + \sum_{j \notin V(B)} d_{\hat{T}}(i,j) \right) \\
&\geqslant \sum_{B \in brn(\hat{T},m)} \sum_{i \in V(B)} \sum_{j \notin V(B)} (d_{\hat{T}}(i,m) + d_{\hat{T}}(j,m)) \\
&\geqslant 2 \sum_{B \in brn(\hat{T},m)} \sum_{i \in V(B)} \sum_{j \notin V(B)} d_{\hat{T}}(i,m) \\
&\geqslant n \sum_{i \in V(G)} d_{\hat{T}}(i,m) \\
&\geqslant n \sum_{i \in V(G)} d_G(i,m). \qquad \square
\end{aligned}$$

Let M be a subgraph of G and contain only one vertex m. We use $star(G,m)$ to denote $star(G,M)$. The main result of this section is summarized in the following theorem.

Theorem 6. *There is a 2-approximation algorithm for the SPST problem with time complexity* $O(n^2 + f(G))$, *where* $f(G)$ *is the time complexity for finding all-pairs shortest paths in* G.

Proof. Let $\hat{T} = SPST(G)$. By Lemma 5, $c(\hat{T}) \geqslant n\sum_{i\in V(G)} d_G(i,u)$ for some vertex $u \in V(G)$. Let m be a median of G. For any $T \in star(G,m)$, by Lemma 3, $c(T) \leqslant 2n\sum_{i\in V(G)} d_G(i,m)$. Since m is a median of G, $\sum_{i\in V(G)} d_G(i,m) \leqslant \sum_{i\in V(G)} d_G(i,u)$, and $c(T) \leqslant 2c(\hat{T})$. If the shortest path lengths are available, a median m of G can be found in $O(n^2)$ time. Using an algorithm for computing single source shortest paths, we can find a shortest path from i to m for each vertex $i \in V$ in $O(n^2)$ time [2]. By Lemma 1, a $T \in star(G,m)$ can be constructed in $O(n)$ time. □

4. A 15/8-approximation algorithm

In the above section, we show that a 1/2-separator leads to a 2-approximation algorithm. We now generalize the idea and demonstrate that we can get better solution by using a 1/3-separator. A 1/3-separator of a tree is a path. The following lemma shows the existence of 1/3-separator. Note that a path may contain only one vertex.

Lemma 7. *For any tree* T, *there is a path* $P \subset T$, *such that* P *is a* 1/3-*separator of* T.

Proof. Let n be the number of vertices of T and m be a centroid of T. There are at most 2 branches of m, whose number of vertices exceed $n/3$. If there is no such branch, then m is already a 1/3-separator. Let A be a branch of m with $|V(A)| > n/3$. Since A itself is a tree with not more than $n/2$ vertices, a centroid m_a of A is a 1/2-separator of A, and each branch of m_a contains not more than $n/4$ vertices of A. Suppose there is another branch B of m, such that $|V(B)| > n/3$. We can also find a centroid m_b of B, such that each branch of m_b contains not more than $n/4$ vertices of B. Consider the path $P = SP_T(m_a,m) \cup SP_T(m,m_b)$. Since each branch of P contains no more than $n/3$ vertices, P is a 1/3-separator of T. Note that if B does not exist, then $SP_T(m_a,m)$ is already a 1/3-separator. □

In the following paragraphs, a path-separator of a tree T means a path of T, which is a minimal 1/3-separator of T. The following lemma shows that the path separator yields a better lower bound on an optimal solution.

Lemma 8. *Let* $G = (V,E,w)$, $\hat{T} = SPST(G)$. *If* P *is a path-separator of* $\hat{T}$, *then* $c(\hat{T}) \geqslant (4n/3)\sum_{i\in V} d_{\hat{T}}(i,P) + (4n^2/9)w(P)$.

Proof. Let $P=(p_1,p_2,\dots,p_k)$, $S_i=VB(T,P,p_i)$, and $n_i=|S_i|$. By Lemma 2, $n_1\geqslant n/3$, and $n_k\geqslant n/3$. Similar to the proof of Lemma 5,

$$\begin{aligned}c(\hat{T}) &\geqslant \sum_{B\in brn(\hat{T},m)}\sum_{i\in V(B)}\sum_{j\notin V(B)}(d_{\hat{T}}(i,P)+d_{\hat{T}}(j,P)+w_P(\hat{T},i,j))\\ &= 2\sum_{B\in brn(\hat{T},m)}\sum_{i\in V(B)}\sum_{j\notin V(B)}d_{\hat{T}}(i,P)+2\sum_{1\leqslant i<j\leqslant k}n_in_jd_{\hat{T}}(p_i,p_j).\end{aligned}$$

For the first term, since P is a 1/3-separator, we have

$$2\sum_{B\in brn(\hat{T},m)}\sum_{i\in V(B)}\sum_{j\notin V(B)}d_{\hat{T}}(i,P)\geqslant(4n/3)\sum_{i\in V}d_{\hat{T}}(i,P).$$

For the second term,

$$\begin{aligned}&2\sum_{1\leqslant i<j\leqslant k}n_in_jd_{\hat{T}}(p_i,p_j)\\ &\geqslant 2n_1n_kw(P)+2\sum_{1<i<k}n_i(n_1d_{\hat{T}}(p_1,p_i)+n_kd_{\hat{T}}(p_i,p_k))\\ &\geqslant 2w(P)(n_1n_k+(n-n_1-n_k)(n/3))\\ &=2w(P)((n_1-n/3)(n_k-n/3)+2n^2/9)\\ &\geqslant(4n^2/9)w(P). \qquad \Box\end{aligned}$$

Lemma 9. *For any graph $G=(V,E,w)$, there exist $m_1,m_2\in V$ such that $c(T)\leqslant(15/8)c(\hat{T})$, where $R=SP_G(m_1,m_2)$, $T\in star(G,R)$ and $\hat{T}=SPST(G)$.*

Proof. Let $P=(p_1,p_2,\dots,p_k)$ be a path separator of $\hat{T}$, $S_i=VB(\hat{T},P,p_i)$, and $n_i=|S_i|$. Let $m_1=p_1$, $m_2=p_k$, and $T\in star(G,R)$. Note that $w(R)\leqslant w(P)$ since R is a shortest path. First, for any $v\in S_1\cup S_k$,

$$\begin{aligned}d_G(v,R) &\leqslant \min\{d_G(v,p_1),d_G(v,p_k)\}\\ &\leqslant d_{\hat{T}}(v,P).\end{aligned}$$

For $v\in S_i, 1<i<k$

$$\begin{aligned}d_G(v,R) &\leqslant \min\{d_G(v,p_1),d_G(v,p_k)\}\\ &\leqslant (d_G(v,p_1)+d_G(v,p_k))/2\\ &\leqslant d_{\hat{T}}(v,P)+w(P)/2.\end{aligned}$$

Then, by Lemma 2, $n_1+n_k\geqslant 2n/3$. We have

$$\sum_{v\in V}d_G(v,R)\leqslant\sum_{v\in V}d_{\hat{T}}(v,P)+(n/6)w(P).$$

By Lemma 3,

$$\begin{aligned}c(T) &\leqslant 2n\sum_{v\in V}d_G(v,R)+(n^2/2)w(R)\\ &\leqslant 2n\sum_{v\in V}d_{\hat{T}}(v,P)+(5n^2/6)w(P).\end{aligned}$$

By Lemma 8, $c(\hat{T}) \geqslant (4n/3)\sum_{v\in V} d_{\hat{T}}(v,P) + (4n^2/9)w(P)$. Thus,

$$c(T) \leqslant \max\{3/2, 15/8\}c(\hat{T}) = (15/8)c(\hat{T}). \qquad \square$$

Lemma 9 implies that there exists a 15/8-approximation algorithm for the SPST problem with time complexity $O(n^4)$. All-pairs shortest paths can be found in $O(n^3)$. For every $m_1, m_2 \in V$, we construct a general star $T \in star(G, SP_G(m_1, m_2))$ including the degenerated cases for $m_1 = m_2$. To construct a star $T \in star(G, SP_G(m_1, m_2))$, we find a shortest path from i to $SP_G(m_1, m_2)$ for each $i \in V$. Since $SP_G(m_1, m_2)$ may have as many as $O(n)$ vertices, a direct method takes $O(n^2)$ time for each m_1, m_2. Thus, the total time complexity is $O(n^4)$. In the next lemma, we show that this can be done in $O(n^3)$.

Lemma 10. *Let $G = (V, E, w)$. There is an algorithm which constructs a general star $T \in star(G, SP_G(m_1, m_2))$ for every $m_1, m_2 \in V$ in $O(n^3)$ time.*

Proof. For any $m \in V$, if we can construct a general star $T \in star(G, SP_G(m, i))$ for each $i \in V$ with total time complexity $O(n^2)$, we can construct all the stars in $O(n^3)$ time by applying the algorithm n times for each $m \in V$. By Lemma 1, a star $T \in star(G, SP_G(m, i))$ can be constructed in $O(n)$ time if for every $j \in V$, a shortest path from j to $SP_G(m, i)$ is given. Define $A[i,j] = d_G(j, SP_G(m,i))$ and $B[i,j]$ to be the vertex $k \in SP_G(m,i)$ such that $SP_G(j,k) = SP_G(j, SP_G(m,i))$. Since the all-pairs shortest paths can be constructed in $O(n^3)$ time at the preprocessing stage, we need to compute $A[i,j]$, as well as $B[i,j]$, in $O(n^2)$ time for all $i,j \in V$. First, construct the single source shortest-paths tree, S, rooted at m. That is, S is a rooted tree and $d_S(i,m) = d_G(i,m)\ \forall i \in V$. S can be constructed in $O(n^2)$ [2]. Let $parent(i)$ denote the parent of i in S. It is not hard to see that $A[i,j] = \min\{A[parent(i), j], d_G(i,j)\}$ for $i,j \in V - \{m\}$, and $A[m,j] = d_G(m,j)$. Therefore by a top-down traversal of S, we can compute $A[i,j]$ as well as $B[i,j]$ for $\forall i,j \in V$ in $O(n^2)$ time. $\square$

We summarize the result of this section in the following theorem. Since it follows directly from Lemmas 9 and 10, we omit the proof.

Theorem 11. *There is a 15/8-approximation algorithm for the SPST problem with time complexity $O(n^3)$.*

5. A 3/2-approximation algorithm

Let P be a path separator of the optimal tree. By Lemma 3, if $X \in star(G,P)$, then $c(X) \leqslant 2n\sum_{v\in V} d_G(v,P) + (n^2/2)w(P)$. Since $d_G(v,P) \leqslant d_{\hat{T}}(v,P)$ for any v, it can be shown that X is a 3/2-approximation solution by Lemma 8. However, we cannot try all possible paths in G since it leads to an exponential time algorithm. In fact, we need not know all the vertices in the path for a 3/2-approximation solution. The following lemma shows that a 3/2-approximation solution can be found if, in addition to the two end points of P, we know a centroid of the optimal tree.

Lemma 12. *For a graph $G=(V,E,w)$, there exist m_1, m_2, and $m_3 \in V$ such that $c(X) \leqslant (3/2)c(\hat{T})$, where $R = SP_G(m_1,m_2) \cup SP_G(m_2,m_3)$, $X \in star(G,R)$, and $\hat{T} = SPST(G)$.*

Proof. Let $P=(p_1,p_2,\dots,p_k)$ be a path separator of $\hat{T}$, $S_i = VB(T,P,p_i)$, and $n_i = |S_i|$ for $i=1,2,\dots,k$. Let p_q be the vertex such that $\sum_{1\leqslant i<q} n_i \leqslant n/2$ and $\sum_{q<i\leqslant k} n_i \leqslant n/2$. Let $R = SP_G(p_1,p_q) \cup SP_G(p_q,p_k)$ and $X \in star(G,R)$. Note that $w(R) \leqslant w(P)$. First, for any $v \in S_1 \cup S_q \cup S_k$,

$$\begin{aligned} d_G(v,R) &\leqslant \min\{d_G(v,p_1), d_G(v,p_q), d_G(v,p_k)\} \\ &\leqslant d_{\hat{T}}(v,P). \end{aligned}$$

For $v \in S_i, 1<i<q$,

$$\begin{aligned} d_G(v,R) &\leqslant \min\{d_G(v,p_1), d_G(v,p_q)\} \\ &\leqslant (d_G(v,p_1)+d_G(v,p_q))/2 \\ &\leqslant d_{\hat{T}}(v,P) + d_{\hat{T}}(p_1,p_q)/2. \end{aligned}$$

Similarly, for $v \in S_i,\ q<i<k$,

$$d_G(v,R) \leqslant d_{\hat{T}}(v,P) + d_{\hat{T}}(p_q,p_k)/2.$$

By Lemma 2 we can show that $n_1+n_k \geqslant 2n/3$, $\sum_{1<i<q} n_i \leqslant n/6$, and $\sum_{q<i<k} n_i \leqslant n/6$. Thus,

$$\sum_{i\in V} d_G(i,R) \leqslant \sum_{i\in V} d_{\hat{T}}(i,P) + (n/12)w(P).$$

By Lemmas 3 and 8,

$$\begin{aligned} c(X) &\leqslant 2n\sum_{i\in V} d_G(i,R) + (n^2/2)w(R) \\ &\leqslant 2n\sum_{i\in V} d_{\hat{T}}(i,P) + (2n^2/3)w(P) \\ &\leqslant (3/2)c(\hat{T}). \qquad \square \end{aligned}$$

Theorem 13. *There is a 3/2-approximation algorithm with time complexity $O(n^4)$ for the SPST problem.*

Proof. First, the all to all shortest paths can be found in $O(n^3)$ [2]. For every $m_1, m_2, m_3 \in V$, we construct a $X \in star(G,R)$, where $R = SP_G(m_1,m_2) \cup SP_G(m_2,m_3)$, including the degenerated cases for $m_i = m_j$. By Lemma 12, at least one of these stars is a 3/2-approximation solution of the SPST problem. We choose the minimum one among $O(n^3)$ stars to be constructed. Now let us show that each star can be constructed in $O(n)$ time. By Lemma 1, a $T \in star(G,R)$ can be constructed in $O(n)$ time if for every $i \in V$, a shortest path from i to R is given. Define $A[i,j,k]=d_G(i,SP_G(j,k))$ and $B[i,j,k]$ to be the vertex in $SP_G(j,k)$ which is closest to i. It is easy to see that $A[i,j,k]$ and $B[i,j,k]$ can be computed in $O(n^4)$ time. For any $R=SP_G(m_1,m_2)\cup SP_G(m_2,m_3)$, since $d_G(i,R)=\min\{A[i,m_1,m_2],A[i,m_2,m_3]\}$, we can compute $d_G(i,R)$, as well as the vertex in R which is closest to i, in $O(n^4)$ time for all $i \in V$ and for all such R at a preprocessing step. Finally, for any spanning tree T, it is not hard to compute $c(T)$ in $O(n)$ time. So the total time complexity is $O(n^4)$. □

6. A (4/3 + ε)-approximation algorithm for SPST

In this section, we generalize the idea in the above section and show that there is a polynomial time algorithm to approximate the SPST problem with ratio $(4/3+\varepsilon)$ for any constant $\varepsilon > 0$. We use a 1/4-separator to get a lower bound on the optimal solution.

Definition 8. Let $\{m_0,m_1,m_2,m_3\}\subset V(T)$, and $SP_T(m_0,m_i)\cap SP_T(m_0,m_j)=\{m_0\}$ $\forall 1\leqslant i<j\leqslant 3$. A *fork* is a subgraph of T, which is defined by $Fork(T,m_0,m_1,m_2,m_3)=\bigcup_{1\leqslant i\leqslant 3} SP_T(m_0,m_i)$.

We assume that the four vertices (for specifying the fork) are always m_0, m_1, m_2, and m_3 in the remaining paragraphs. The statement, "Y is a fork of T", means $Y=Fork(T,m_0,m_1,m_2,m_3)$. We also use M to denote $\{m_0,m_1,m_2,m_3\}$. A fork may degenerate to a path or only a vertex. Y is a fork-separator of a tree T if Y is a fork of T and Y is also a minimal 1/4-separator of T. The following lemma shows the existence of a fork-separator.

Lemma 14. *For any tree T, there is a fork $Y\subset T$, such that Y is a minimal* 1/4-*separator of T.*

Proof. Let $n=V(T)$ and m_0 be a centroid of T. There are at most 3 branches of m_0, whose number of vertices exceed $n/4$. Let A be a branch of m_0 with $|V(A)|>n/4$. Since A is a tree with no more than $n/2$ vertices, each branch of its centroid m_1 contains no more than $n/4$ vertices of A. If there are other branches B and C of m_0, such that $|V(B)|>n/4$ and $|V(C)|>n/4$. We can also find a centroid m_2 of B and m_3 of C. Consider the fork $Y=Fork(T,m_0,m_1,m_2,m_3)$ of T. Since each branch of Y contains no more than $n/4$ vertices, Y is a 1/4-separator of T. Note that even if A (and B, C) does not exist, Y is still a 1/4-separator by setting $m_1=m_0$ (and $m_2=m_0$, $m_3=m_0$, respectively). □

In the case that the fork-separator is a path, there may be different choices of m_0 to denote the same fork. In the following, we assume that m_0 is always the centroid of the tree. We now define some notations and then derive a lower bound on $c(SPST(G))$ using the fork-separator.

Definition 9. Let R be a connected subgraph of a tree T and $A \subset V(R)$. We define $Nhang(T,R,A) = V(T) - \bigcup_{u \in A} VB(T,R,u)$. For a path $P = SP_T(i,j)$, $Nhang(T,P)$ denotes $Nhang(T,P,\{i,j\})$ in brief. For a fork Y of T, $Nhang(T,Y)$ denotes $Nhang(T,Y,M)$.

The set $Nhang(T,R,A)$ contains all the vertices not hung on any vertex in A.

Lemma 15. *Let $G = (V,E,w)$ and $\hat{T} = SPST(G)$. If Y is a fork-separator of $\hat{T}$, then $c(\hat{T}) \geqslant (3n/2)\sum_{v \in V} d_{\hat{T}}(v,Y) + (3n^2/8)w(Y) + (n/2)\sum_{v \in Q} w_Y(\hat{T},v,m_0)$, where $Q = Nhang(\hat{T},Y)$.*

Proof. Similar to Lemma 8, since the fork Y is a 1/4-separator,

$$c(\hat{T}) \geqslant (3n/2)\sum_{v \in V} d_{\hat{T}}(v,Y) + \sum_{u,v \in V} w_Y(\hat{T},u,v).$$

Note that the fork Y may be a path or a vertex. If Y is only a vertex, then the proof is completed. We now assume that Y is not a path, that is, $m_i \neq m_0$ for $i = 1, 2, 3$. For $1 \leqslant i \leqslant 3$, by Lemma 2, $|VB(\hat{T},Y,m_i)| \geqslant n/4$. Let $L_i \subset VB(\hat{T},Y,m_i)$ and $|L_i| = n/4$ for $1 \leqslant i \leqslant 3$. We set $S = V - L_1 - L_2 - L_3$. Note that L_1, L_2, L_3, and S form a partition of V and each of them contains $n/4$ vertices. Also let $L = L_1 \cup L_2 \cup L_3$. For any $v \in S$, without loss of generality, we assume that v is hung on a vertex of the path from m_0 to m_1. Since m_0 is a centroid of $\hat{T}$, we have

$$\begin{aligned}
&\sum_{u \in V} w_Y(\hat{T},u,v)\\
&\quad \geqslant (n/2)w_Y(\hat{T},m_0,v) + (n/4)(w_Y(\hat{T},m_1,v) + w_Y(\hat{T},m_0,m_2) + w_Y(\hat{T},m_0,m_3))\\
&\quad \geqslant (n/4)(w(Y) + w_Y(\hat{T},m_0,v)).
\end{aligned}$$

Similarly we can show that the above inequality holds for any $v \in S$. That is,

$$\sum_{u \in V} w_Y(\hat{T},u,v) \geqslant (n/4)(w(Y) + w_Y(\hat{T},m_0,v)) \quad \forall v \in S. \tag{2}$$

Using this bound, we obtain

$$\begin{aligned}
&\sum_{u,v \in V} w_Y(\hat{T},u,v)\\
&\quad \geqslant \sum_{u,v \in L} w_Y(\hat{T},u,v) + 2\sum_{u \in V, v \in S} w_Y(\hat{T},u,v)\\
&\quad \geqslant (n^2/4)w(Y) + (n/2)\sum_{v \in S}(w(Y) + w_Y(\hat{T},m_0,v))\\
&\quad \geqslant (3n^2/8)w(Y) + (n/2)\sum_{v \in Q} w_Y(\hat{T},v,m_0).
\end{aligned}$$

This completes the proof for the case when Y is not a path.

When Y is only a path, one or two of the vertices m_i may be identical to m_0. We modify the definitions of L_i such that $L_i = \emptyset$ if $m_i = m_0$ for $i = 1, 2, 3$. It can be easily

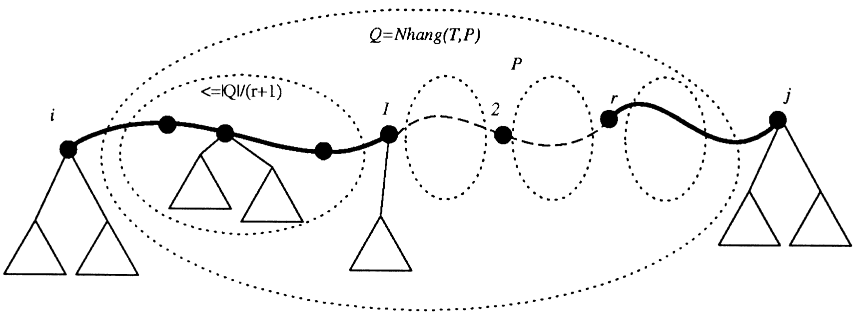

Fig. 2. Breaking $Nhang(T,P)$ into small components.

checked that equation (2) holds in this case. When there is exactly one of m_i identical to m_0, since $|S|=n/2$ in this case, we have

$$\begin{aligned}
&\sum_{u,v\in V} w_Y(\hat{T},u,v)\\
&\quad\geqslant \sum_{u,v\in L} w_Y(\hat{T},u,v)+2\sum_{u\in V,v\in S} w_Y(\hat{T},u,v)\\
&\quad\geqslant (n^2/8)w(Y)+(n/2)\sum_{v\in S}(w(Y)+w_Y(\hat{T},m_0,v))\\
&\quad\geqslant (3n^2/8)w(Y)+(n/2)\sum_{v\in Q} w_Y(\hat{T},v,m_0).
\end{aligned}$$

Finally, when there are exactly two of m_i identical to m_0, since $|S|=3n/4$, it is easy to see that the above inequality also holds and the proof is completed. □

Let T be a tree and $P=SP_T(i,j)$. We can find a vertex set $A=\{i,j,1,2,\ldots,r\}\subset V(P)$ such that each connected component of the induced subgraph of T on $Nhang(T,P,A)$ contains no more than $|Nhang(T,P)|/(r+1)$ vertices. Fig. 2 illustrates the concept. The following lemma is based on this property and will be used to achieve a better approximation algorithm. In the lemma, k is a parameter and will be determined later.

Lemma 16. *Let $G=(V,E,w)$ and $\hat{T}=SPST(G)$. Assume Y be a fork-separator of $\hat{T}$ and $Q=Nhang(\hat{T},Y)$. For any integer constant $r\geqslant 0$ and any real number $0\leqslant k\leqslant 1$, there exists a set $A\subset V(Y)$ with $|A|\leqslant r+4$ and $M\subset A$, such that $\sum_{u\in V} d_G(u,A)\leqslant \sum_{u\in V} d_{\hat{T}}(u,Y)+(k/2(r+1))w(Y)|Q|+(1-k)\sum_{u\in Q} w_Y(\hat{T},u,m_0)$.*

Proof. Let $L=\bigcup_{v\in M} VB(\hat{T},Y,v)$. For any $u\in L$, $d_G(u,A)\leqslant d_{\hat{T}}(u,Y)$. Let $Q_i=Nhang(\hat{T},SP_Y(m_0,m_i))$ for $1\leqslant i\leqslant 3$. Q_1, Q_2 and Q_3 form a partition of Q. Consider $P=SP_Y(m_0,m_i)$. As in Fig. 2, we can find $A_i=\{m_0,y_1,\ldots,y_{x_i},m_i\}\subset V(P)$ such that $|Nhang(\hat{T},SP_Y(y_j,y_{j+1}))|\leqslant|Q|/(r+1)$ for $0\leqslant j\leqslant x_i$, where $y_0=m_0$ and $y_{x_i+1}=m_i$ and $x_i=\lceil(r+1)|Q_i|/|Q|\rceil-1$. Let $A=A_1\cup A_2\cup A_3$. We have $|A|\leqslant\sum_{1\leqslant i\leqslant 3}\lceil(r+1)|Q_i|/|Q|\rceil-3+4\leqslant r+4$. Note that if $u\in VB(\hat{T},Y,v)$ and $v\in SP_Y(y_j,y_{j+1})$, we have

$$\begin{aligned} & d_G(u,A) \\ & \quad \leqslant d_{\hat{T}}(u,Y) + \min\{d_Y(y_j,y_{j+1})/2, w_Y(\hat{T},u,m_0)\} \\ & \quad \leqslant d_{\hat{T}}(u,Y) + kd_Y(y_j,y_{j+1})/2 + (1-k)w_Y(\hat{T},u,m_0) \end{aligned}$$

for any $0\leqslant k\leqslant 1$. Thus,

$$\begin{aligned} & \sum_{u\in V} d_G(u,A) \\ & \quad = \sum_{u\in L} d_G(u,A) + \sum_{u\in Q} d_G(u,A) \\ & \quad \leqslant \sum_{u\in V} d_{\hat{T}}(u,Y) + (k/2)w(Y)|Q|/(r+1) + (1-k)\sum_{u\in Q} w_Y(\hat{T},u,m_0) \end{aligned}$$

for any $0\leqslant k\leqslant 1$. □

Assume that Y is a fork-separator of the optimal tree and A is the vertex set indicated in Lemma 16. We will show how to construct a tree R which spans A and $w(R)\leqslant w(Y)$. Our approximation algorithm is to construct a general star with core R. A possible method for constructing R is to solve the *Steiner Minimum Tree* problem. However, in general, the Steiner minimum tree problem is NP-hard [5]. Fortunately, we need not solve such a general problem. If the *depth-first-search* (DFS) sequence of A on Y is given, R can be constructed by a simple algorithm. Since $|A|$ is constant in our application, trying all possible sequences (permutations) only takes polynomial time. The following algorithm takes a sequence of vertices as input and returns a tree spanning those vertices. It will be shown later that it returns the desired tree if the input is a DFS sequence of A on Y.

Algorithm CORE

Input: a graph G and a sequence S of $A\subset V(G)$. Assume $S=(v_1,v_2,\ldots,v_r)$

Output: a tree R spanning A.

Step 1: Initially set R be the tree containing only one vertex v_1.

Step 2: for $i=2$ to r do

$R=R\cup SP_G(v_i,V(R))$

enddo

We use $CORE(G,S)$ to denote the output tree of the algorithm **CORE** with input G and S.

Lemma 17. *Let G be a graph and T a spanning tree of G. Assume Y be a fork of T and $M\subset A\subset V(Y)$. If $S=(v_1=m_0,v_2,\ldots,v_r)$ is a* DFS *sequence of A on Y, then $w(CORE(G,S))\leqslant w(Y)$.*

Proof. Let $X=(m_0=u_1,u_2,\ldots,u_h=m_3)$ be a DFS sequence of all vertices of Y. Let S be the subsequence of X on A. That is, S is obtained by deleting the vertices of $V(Y)\backslash A$ from X. In other words, $v_i=u_{f(i)}$ where $f(i)$ is a monotonically increasing function mapping from $\{1..r\}$ to $\{1..h\}$. Also let Y_i be the induced subgraph of Y on $\{u_j \mid 1 \leqslant j \leqslant f(i)\}$. Since X is a DFS sequence, Y_i is a tree. Let R_i be the tree constructed at ith iteration of Algorithm **CORE**. We shall prove that $w(R_i) \leqslant w(Y_i)$. Initially, $w(R_1) = 0 = w(Y_1)$. Suppose that $w(R_i) \leqslant w(Y_i)$ for some i. Consider Y_{i+1} and R_{i+1}. Since Y is a fork and X is a depth first search sequence starting at m_0,

$$\begin{aligned} w(Y_{i+1}) &\geqslant w(Y_i) + \min\{d_Y(v_{i+1}, v_i), d_Y(v_{i+1}, m_0)\} \\ &\geqslant w(Y_i) + d_Y(v_{i+1}, V(R_i)) \\ &= w(R_{i+1}). \end{aligned}$$

By induction, the lemma follows. □

Lemma 18. *For a graph $G = (V, E, w)$ and an integer r, there exists a sequence S of no more than $(r+4)$ vertices, such that if $R = CORE(G, S)$ and $X \in star(G, R)$, then $c(X) \leqslant (4/3 + 8/(9r+12))c(\hat{T})$, where $\hat{T} = SPST(G)$.*

Proof. Let Y be a fork-separator of $\hat{T}$ and $Q = Nhang(\hat{T}, Y)$. By Lemma 16, there exists a set $A \subset V(Y)$ such that $M \subset A$, $|A| \leqslant r+4$, and for any $0 \leqslant k \leqslant 1$,

$$\sum_{u \in V} d_G(u, A) \leqslant \sum_{u \in V} d_{\hat{T}}(u, Y) + (k/2)w(Y)|Q|/(r+1) + (1-k)\sum_{u \in Q} w_Y(\hat{T}, u, m_0)$$

Let S be a DFS sequence of A on Y. By Lemma 17, $w(R) \leqslant w(Y)$. Then by Lemma 3, we have

$$\begin{aligned} c(X) &\leqslant 2n \sum_{i \in V} d_G(i, Y) + (n^2/2)w(Y) \\ &\leqslant 2n \sum_{i \in V} d_{\hat{T}}(i, Y) + (kn|Q|/(r+1) + n^2/2)w(Y) + 2n(1-k)\sum_{i \in Q} d_{\hat{T}}(i, m_0). \end{aligned}$$

Since $|Q| \leqslant n/2$, we set $k = (2r+2)/(3r+4)$ and obtain

$$c(X) \leqslant 2n \sum_{i \in V} d_{\hat{T}}(i, Y) + \left((3n^2/2)w(Y) + 2n \sum_{i \in Q} d_{\hat{T}}(i, m_0)\right) \frac{r+2}{3r+4}.$$

By Lemma 15, we can conclude that

$$c(X) \leqslant \max\{4/3, 4(r+2)/(3r+4)\}c(\hat{T}) = (4/3 + 8/(9r+12))c(\hat{T}). \quad \square$$

Theorem 19. *For any $\varepsilon > 0$, there is an approximation algorithm for the SPST problem with approximation ratio $(4/3 + \varepsilon)$ and time complexity $O(n^{\delta})$, where $\delta = \lceil (33\varepsilon + 8)/(9\varepsilon) \rceil$.*

Proof. The approximation ratio directly comes from Lemma 18 for $\varepsilon = 8/(9r+12)$. It remains to show that the total running time is $O(n^{r+5})$. Basically, we examine all possible sets with no more than $r+4$ vertices. For each possible set and each possible sequence S, we construct a core $R = CORE(G, S)$ and find a spanning tree $T = star(G, R)$ for each R. It takes $O(n^{r+5})$ time. We then choose the one with minimum total path length among these spanning trees. The time complexity is therefore $O(n^{r+5})$, and the result follows. □

7. Concluding remarks

In this paper, we present several approximation algorithms for the shortest total path length spanning tree problem. The best achieved approximation ratio is $(4/3+\varepsilon)$ for any $\varepsilon > 0$. The algorithms developed in this paper also work for the *shortest total path length Steiner tree* problem which asks for a tree T with minimum $c(T)$ spanning a subset of the vertices of the input graph. An interesting open problem is whether there are approximation algorithms for the SPST problem that provide better approximation ratios than those presented in this paper. Very recently, Bafna et al. and the authors of this paper gave a *polynomial time approximation scheme* for the problem [12]. Another problem is whether the idea of separators can be applied to other tree construction problems such as the *minimum increment to additive problem under* L_1*-norm* [4], which is an important problem in computational biology.

8. For Further reading

The following reference is also of interest to the reader: [3].

Acknowledgements

We thank the anonymous referees for their careful reading and many useful comments.

References

[1] V. Bafna, E.L. Lawler, P. Pevzner, Approximation Algorithms for Multiple Sequence Alignment, Proceedings of the 5th Combinatorial Pattern Matching Conference, Lecture Notes in Computer Science, Vol. 807, Springer, Berlin, 1994, pp. 43–53.

[2] T.H. Cormen, C.E. Leiserson, R.L. Rivest, Introduction to Algorithms, MIT Press, Cambridge, MA, 1994.

[3] R. Dionne, M. Florian, Exact and approximate algorithms for optimal network design, Networks 9 (1) (1979) 37–60.

[4] M. Farach, S. Kannan, T. Warnow, A robust model for finding optimal evolutionary trees, Algorithmica 13 (1995) 155–179.

[5] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman, San Francisco, CA, 1979.

[6] D. Gusfield, Efficient methods for multiple sequence alignment with guaranteed error bounds, Bull. Math. Biol. 55 (1993) 141–154.

[7] D. Gusfield, Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology, Cambridge University Press, Cambridge, 1997.

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

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

[10] P. Pevzner, Multiple alignment, communication cost, and graph matching, SIAM J. Appl. Math. 52 (1992) 1763–1779.

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

[12] B.Y. Wu, G. Lancia, V. Bafna, K.M. Chao, R. Ravi, C.Y. Tang, A polynomial time approximation scheme for minimum routing cost spanning trees, Proceedings of Ninth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'98), 1998, pp. 21–32.