在 metric 上尋找最短總和距離 2-star 之演算法
Algorithms for Finding the Shortest Total Path Length 2-star on a Metric Space

吳邦一
Bang Ye Wu
清華大學資訊系
Dept. of C.S., National Tsing Hua University, Hsinchu, Taiwan, R.O.C.
dr838305@cs.nthu.edu.tw

趙坤茂
Kun-Mao Chao
靜宜大學資管系
Dept. of C.S. and Inform. Mang., Providence University, Shalu, Taiwan, R.O.C.
kmchao@csim.pu.edu.tw

唐傳義
Chuan Yi Tang
清華大學資訊系
Dept. of C.S., National Tsing Hua University, Hsinchu, Taiwan, R.O.C.
cytang@cs.nthu.edu.tw

摘要

在 metric space 上的 2-star 即為不超過兩個 internal nodes 之 spanning tree. 在本論文中提出尋找最短距離總和 2-star 之演算法.

Abstract

A 2-star on a metric space is a spanning tree with at most 2 internal nodes. The total path length of a spanning tree T is defined to be $\sum_{\forall i,j} d(T,i,j)$, where $d(T,i,j)$ is the distance between i and j on T. Given an n by n metric, we show an algorithm to find the the shortest total path length 2-star in $O(n^3 \log n)$ time.

KEYWORDS: ALGORITHM, SPANNING TREE, NETWORK DESIGN, COMPUTATIONAL BIOLOGY.

1 Introduction

A k-star is a spanning tree with no more than k internal nodes. Given n nodes in a metric space, the shortest total path length k-star (minimum k-star in short) problem is to find a k-star such that the summation of the path lengths over all pairs of nodes is minimum. The problem is interseting because of its relation to the *shortest total path length spanning tree* (SPST) problem. The shortest total path length spanning tree problem is a special case of the *optimum communication spanning tree* problem ([4]) and was proved to be NP-hard in [5] (aslo listed in [3]). Recently, the SPST problem becomes more attractive not only in network design but also in multiple sequence alignments, which is an important problem in computational biology (e.g. see [7]). Exact and heuristic algorithms of the SPST problem were proposed in [1], and a 2-approximation algorithm was presented in [6]. Recently, a PTAS for the SPST problem was presented in [7]. The PTAS is based on (1) the minimum k-star is a good approximation solution and (2) the minimum k-star can be found in polynomial (with respect to n) time. So, any improvement on the time complexity of finding minimum k-star results in a more efficient approximation algorithm for the SPST problem.

In this paper, we restrict the problem to 2-star and give a more efficient algorithm. The minimum 2-star problem is different from the 2-median problem because of the cost definition. Given a set of vertices V, the 2-median problem is to find $a, b \in V$ such that $\sum_{v \in V} \min\{w(v,a), w(v,b)\}$ is minimum, where $w(a,v)$ is the edge length between a and v. In addition to the two vertices (a and b), the minimum 2-star problem also want to find a set partition V_a and V_b such that $(n-1)\left(\sum_{v \in V_a} w(v,a) + \sum_{v \in V_b} w(v,b)\right) + |V_a||V_b|w(a,b)$ is minimum. Note that the solution of the 2-median problem is not necessary a solution of the minimum 2-star problem. Furthermore, when the two vertices a and b are fixed, a vertex v may belong to V_b even for $w(v,a) < w(v,b)$. The above two points make the problem not trivial.

In this paper, we first present a naive algorithm with time complexity $O(n^{5.5})$, which is also presented in [7]. Then we show that the algorithm can be improved to $O(n^4)$ by dynamic programming. Although the time complexity is same as the one in [7], the dynamic programming algorithm is simpler. Finally, we give an $O(n^3 \log n)$ time algorithm.

The remaining sections are organized as follows: In Section 2, some definitions and notation are given. We present the three algorithms in Section 3 and give a concluding remark in Section 4.

2 Prelimiaris

In this paper, a graph $G = (V, E, w)$ is a simple, connected, undirected graph, in which w is the nonnegative edge weight. Any metric M can be represented by a complete graph in which the weight of edge (i,j) equals $M[i,j]$. We first give some definitions below:

Definition 1: A metric graph $G = (V, E, w)$ is a complete graph in which (1)$w(i,j) \geq 0$ $\forall i \neq j$, (2) $w(i,j) + w(j,k) \geq w(i,k)$ $\forall i,j,k$.

Definition 2: Let $G = (V, E, w)$ be a metric graph and $x, y \in V$. A 2-star $T = (V, E_t, w) = 2star(x, y, X, Y)$ is a spanning tree of G in which $E_t = \{(x,v)|\forall v \in X\} \cup \{(y,v)|\forall v \in Y\} \cup \{(x,y)\}$.

Definition 3: Let $G = (V, E, w)$ be a graph. $w(G) = \sum_{e\in E} w(e)$. Let T be a tree and i, j be two nodes of T. $P(T,i,j)$ denotes the unique path between i and j on T and $d(T,i,j)$=$w(P(T,i,j))$. The total path length of T is define as $c(T) = \sum_{\forall i,j \in V} d(T,i,j)$.

Definition 4: Minimum 2-Star Problem (M2S) Given a metric graph $G = (V, E, w)$, find a 2-star T of G such that $c(T)$ is minimum.

The following lemma is trivial, and we omit the proof.

Lemma 1: If $T = 2star(x, y, X, Y)$,

$$\begin{aligned} c(T) &= 2(|X|+1)(|Y|+1)w(x,y) \\ &\quad +2(n-1)\left(\sum_{v\in X} w(x,v) + \sum_{v\in Y} w(y,v)\right) \end{aligned}$$

.

3 Algorithms

Clearly, the M2S is to find x, y, X, and Y. For any x, y, if we can find the optimal partition in $O(f(n))$ time, we can solve the problem in $O(f(n)n^2)$ time by trying all possible node pairs. We fist present a naive algorithm in the following subsection.

3.1 A naive algorithm

For specified x and y, if $|X| = k$, $0 \leq k \leq n-2$,

$$\begin{aligned} c(T) &= 2(k+1)(n-k-1)w(x,y) \\ &\quad +2(n-1)\left(\sum_{v\in X} w(x,v) + \sum_{v\in Y} w(y,v)\right) \end{aligned}$$

. So our goal is to find a partition X, Y such that $|X| = k$, $|Y| = n-k-2$, and $\sum_{v\in X} w(x,v) + \sum_{v\in Y} w(y,v)$ is minimum. We now show that such a partition can be solved by matching.

Definition 5: Let $G = (V \cup \{x,y\}, E, w)$ be a complete bipartite graph with edge weight w, $|V| = n$. $0 \leq k \leq n$. Given G and k, the U-partition(G,k) problem is to partition V into X and Y such that $|X| = k$ and $\sum_{v\in X} w(x,v) + \sum_{v\in Y} w(y,v)$ is minimum.

Lemma 2: The U-partition(G,k) problem can be solved in $O(n^{2.5})$ by solving a minimum perfect matching problem on a complete bipartite graph.

Proof: Assume $G = (V \cup \{x,y\}, E, w_1)$. Construct a complete bipartite graph $H = (V \cup U^*, E^*, w_2)$ in which $U^* = \{u_i | 1 \leq i \leq n\}$, and $w_2(v, u_i) = w_1(v, x)$ $\forall i \leq k$, and $w_2(v, u_i) = w_1(v, y)$ $\forall i > k$ That is, U^* contains k copies of x and $(n-k)$ copies of y. It is easy to see that every feasible solution of the original problem corresponds to a perfect matching on H. For solving the minimum perfect matching on H, since the perfect matching contains exactly n edges, we can solve it by an algorithm for maximum matching. Let $b = \max\{w_2(e)|\forall e \in E^*\}$. Consider the complete bipartite graph $H^* = (V \cup U^*, E^*, w_3)$, in which $w_3(e) = b - w_2(e)$, $\forall e \in E^*$. If A is the maximum matching in H^*, since H^* is complete bipartite and the edge weights are nonnegative, A must contains n edges. So, A is also a perfect matching and $w_3(A) = nb - w_2(A)$, which implies $w_2(A)$ is minimum if and only if $w_3(A)$ is maximum. The time complexity then depends on the algorithm for maximum matching, which is $O(n^{2.5})$ for a graph with n vertices [2]. □

Here comes our first algorithm:

Algorithm I
Input: An n by n metric M
Output: The minimum 2-star

```
For each vertex pair (x, y) do
  For k = 0 to n - 2 do
    By solving U-partition problem, find the minimum 2-star under the constraints that (x, y)
      are centers and k leaves hanged at x
    Keep the best solution found so far
```

We have the following lemma:

Lemma 3: Algorithm I solves the M2S problem in $O(n^{5.5})$ time.

3.2 A dynamic programming algorithm

In this section, we show that the U-partition(G,k) can be solved by dynamic programming for all $0 \leq k \leq n$. That is, we solve the inner loop of Algorithm I with dynamic programming. Let $G = (V \cup \{x,y\}, E, w)$ be a complete bipartite graph and the solutions of U-partition(G,k) are (X_k, Y_k) for all $0 \leq k \leq n$. Assume H be the super graph of G with vertex set $V_h \cup \{x,y\}$ and $V_h = V \cup \{u\}$. It is not hard to prove the solution of U-partition(H,k) is either $(X_{k-1} \cup \{u\}, Y_{k-1})$ or $(X_k, Y_k \cup \{u\})$. Using this property, we can insert the vertices one by one (in any order) and find the solutions of the U-partition(G,k) problem for all $0 \leq k \leq n$. The algorithm is as follows:

Algorithm II
Input: An n by n metric M
Output: The minimum 2-star
For each vertex pair (x, y) do
/* Assume the vertex set is $\{x, y, 1, 2, ..., n-2\}$. $A[i, j]$ denotes the cost of the best 2-star with leaf set $\{1..i\}$ and there are j leaves adjacent to x. $B[k]$ denotes the best solution with exactly k leaves adjacent to x */
$A[i, -1] = \infty$ for all i; $A[i, j] = \infty$ for $j > i$.
$A[1, 0] = M[y, 1]; A[1, 1] = M[x, 1]$;
For $i = 2$ to $n-2$ do
For $j = 0$ to i do

$$A[i, j] = \min \left\{ \begin{array}{l} A[i-1, j-1] + M[x, i] \\ A[i-1, j] + M[y, i] \end{array} \right\}$$

$B[k] = 2(n-1)A[n-2, k] + 2(k+1)(n-1-k)M[x, y], \forall 0 \le k \le n-2$
Keep the best solution found so far

We get the following lemma.

Lemma 4: Algorithm II solves the M2S problem in $O(n^4)$ time.

3.3 An efficient algorithm

To derived a more efficient algorithm, we found the following property.

Lemma 5: Let Y be the minimum 2-Star of a metric graph $G = (V, E, w)$ with internal nodes a and b. Assume $S_i = \{v | (v, i) \in Y, v \notin \{a, b\}\}$, for $i \in \{a, b\}$. For any $v_1 \in S_a$ and $v_2 \in S_b$, $w(a, v_1) - w(b, v_1) \le w(a, v_2) - w(b, v_2)$.

Proof: If the inequality does not hold, we can chang the two vertices and get a solution with less cost. □

Based on the above property, we can obtain an efficient algorithm. Let $f_{a,b}(v) = w(a, v) - w(b, v)$, $\forall a, b, v \in V$. For any specified a and b, relabel the vertices such that $V = \{a, b, 1, 2, \ldots n-2\}$ and $f_{a,b}(i-1) \le f_{a,b}(i)$. This can be done by sorting the value $f_{a,b}(v)$ and takes $O(n \log n)$ time. Let $Y_{a,b}$ be the minimum 2-Star with internal nodes a and b. Assume S_a and S_b be the set of vertices hanged at a and b on $Y_{a,b}$ respectively. Then, from Lemma 5, there exists an integer $k \in \{1..n-1\}$ such that $S_a = \{i | 1 \le i < k\}$ and $S_b = \{i | k \le i \le n-2\}$. Let $A[k]$ denote the cost of the 2-star in which $S_a = \{i | 1 \le i < k\}$ and $S_b = \{i | k \le i \le n-2\}$. Clearly,

$$\begin{aligned} A[i] &= A[i-1] + 2(n-1)f_{a,b}(i) \\ &\quad + 2(n-2i-1)M[a, b]. \forall i > 1 \end{aligned}$$

. So, the array A can be computed in $O(n)$ time, and $Y_{a,b}$ can be found by searching the minimum among $A[i]$. The algorithm is listed below and the main result of this paper is in the following theorem.

Algorithm III
Input: An n by n metric M
Output: The minimum 2-star
For each vertex pair (a, b) do
/* Assume the vertex set is $\{a, b, 1, 2, ..., n-2\}$ */
$f_{a,b}(v) = w(a, v) - w(b, v)$ for all $v \in \{1..n-2\}$
Sorting and relabel the vertices such that $f_{a,b}(i-1) \le f_{a,b}(i)$
$A[0] = 2(n-1)\sum M[b, v] + 2(n-1)M[a, b]$
For $i = 1$ to $n-2$ do
$A[i] = A[i-1] + 2(n-1)f_{a,b}(i) + 2(n-2i-1)M[a, b]$
Keep the best solution found so far

Theorem 6: The minimum 2-star problem can be solved by Algorithm III with time complexity $O(n^3 \log n)$.

4 Concluding remark

In [7], there is an $O(n^{2k})$ algorithm for the minimum k-star. Algorithm II can be also generalized to k-star with the same time complexity. The most interesting question is how to generalize Algorithm III to k-star and results in a more efficient PTAS for the SPST problem. However, we have not found such a generalization with a more efficient time complexity.

References

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

[2] S. Even and O. Kariv, An $O(n^{2.5})$ algorithm for maximum matching in general graph, *Proceedings of the 16th Annual Symposium on Foundations of Computer Science*, 100-112, 1979.

[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 Fransisco, 1979.

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

[5] D.S. Johnson, J.K. Lenstra, and A.H.G. Rinnooy kan, The Complexity of the network Design problem, *Networks*, 8:279-285, 1978.

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

[7] Bang Ye Wu, Giuseppe Lancia, Vineet Bafna, Kun-Mao Chao, R. Ravi, and Chuan Yi Tang, A polynomial time approximation scheme for minimum routing cost spanning trees, to appear in *SODA98*, 1997.