Math. Japonica 24, No. 6 (1980), 573–577.

AN APPROXIMATE SOLUTION FOR THE STEINER PROBLEM IN GRAPHS

Hiromitsu TAKAHASHI* and Akira MATSUYAMA*

(Received April 19, 1979)

Abstract. An $O(kn^2)$ time algorithm finding an approximate solution for the Steiner problem in graphs is considered, where n is the number of vertices in a given graph and k is the number of vertices that must be connected. The worst case cost-ratio of the obtained solution to the optimal solution is tightly $2\cdot(1-1/k)$.

1. Introduction. Let $G=(V,E)$ be a connected, undirected graph with a cost function c, where V is a finite set of *vertices*, E is a set of unordered pairs of distinct vertices in V called *edges*, and c maps each edge (v_i, v_j) of E to a positive number $c(v_i, v_j)$ called the *cost of edge* (v_i, v_j). A *subgraph* $G'=(V',E')$ of $G=(V,E)$ is a graph such that $V' \subseteq V$ and $E' \subseteq E$. The *cost of a subgraph* G' is the sum of the cost of edges in G'. The *Steiner problem in graphs* is: given graph $G=(V,E)$ and a subset S of V, find a subgraph with the minimum cost among all connected subgraphs that contain S. It is evident that the subgraph which is a solution of this problem must be a tree. We briefly call it an optimal tree.

Let $|V|=n$, and $|S|=k$ $(k \geqslant 2)$ ($|X|$ denotes the number of elements in set X). The Steiner problem in graphs is reduced to the "shortest path problem" when $k=2$, and to the "minimum-cost spanning tree problem" when $k=n$. These two problems are solved effectively by many authors [2], [3], [6], [7], etc. Dreyfus and Wagner [4] gave an algorithm solving the Steiner problem in graphs which requires time proportional to $n^3/2+n^2\cdot(2^{k-1}-k-1)+n\cdot(3^{k-1}-2^k+3)/2$. But this method is useful only for small values of k. No polynomial time algorithms of solving the Steiner problem in graphs are likely to exist, since Karp [5] showd that this problem is *NP*-complete. Hence it is of practical importance to obtain approximation methods which find trees whose costs are close to optimal.

Let $H=(S,E')$ be the complete graph on the vertices S, and let the cost of edge (u,v) in H be the length of a shortest path between u and v in G. It pointed out in [9] that a minimum-length spanning tree in H is an approximate solution of the Steiner tree problem for G whose worst case cost-ratio to an optimal trees is less than or equal to $1/2$.

AMS(MOS) subject classification (1970). 05-04.
* Department of Mathematical Sciences, University of Osaka Prefecture, Sakai 591, Japan.

In this paper we propose a more practical and reasonable algorithm to find an approximate solution for the Steiner problem in graphs and analyze it to bound the worst case cost-ratio of the obtained tree to an optimal tree.

2. An approximation algorithm. In this section we give an algorithm for finding an approximate solution for the Steiner problem in graphs.

At each step in this algorithm, a tree containing a subset of S has been built up, and a new vertex in S is inserted together with a shortest path connecting the tree and the vertex. Let $\mathrm{PATH}(W, v)$ denote a path whose cost is minimum among all shortest paths from vertices in W to vertex v where $W \subseteq V$ and $v \notin W$. Denote by $\hat{c}(W, v)$ the cost of $\mathrm{PATH}(W, v)$. Then the algorithm to find an approximate solution T_k may be described as follows:

Step 1. Start with subgraph $T_1 = (V_1, E_1)$ consisting a single vertex, say v_1, in S, that is, set $V_1 = \{v_1\}$ and $E_1 = \emptyset$.

Step 2. For each $i = 2, 3, \cdots, k$ do : Find a vertex in $S - V_{i-1}$, say v_i, such that $\hat{c}(V_{i-1}, v_i) = \min\{\hat{c}(V_{i-1}, v_j) \mid v_j \in S - V_{i-1}\}$. Construct tree $T_i = (V_i, E_i)$ by adding $\mathrm{PATH}(V_{i-1}, v_i)$ to T_{i-1}, i.e., set $V_i = V_{i-1} \cup \{$ vertices in $\mathrm{PATH}(V_{i-1}, v_i)\}$ and $E_i = E_{i-1} \cup \{$ edges in $\mathrm{PATH}(V_{i-1}, v_i)\}$.

We assume that when there are ties in step i, they can be broken arbitrarily.

We note that this algorithm requires at most $O(kn^2)$ time, since $\mathrm{PATH}(V_{i-1}, v_i)$ can be computed in time complexity $O(n^2)$ by Dijkstra's algorithm [3].

Let $d(u, v)$ be the cost of the path between vertices u and v in an optimal tree. We use OPTIMAL to represent the cost of an optimal tree.

Lemma 1. *There exists a permutation* $t_1, t_2, \cdots, t_k$ *of* $1, 2, \cdots, k$ *such that*

$$d(v_{t_1}, v_{t_2}) + \cdots + d(v_{t_{k-1}}, v_{t_k}) + d(v_{t_k}, v_{t_1}) = 2 \cdot \mathrm{OPTIMAL}$$

and

$$d(v_{t_k}, v_{t_1}) \geqslant (2/k) \cdot \mathrm{OPTIMAL}.$$

Proof. Suppose that v_{i_j} in S is visited after $v_{i_{j-1}}$ in S for each $2 \leqslant j \leqslant k$ by a

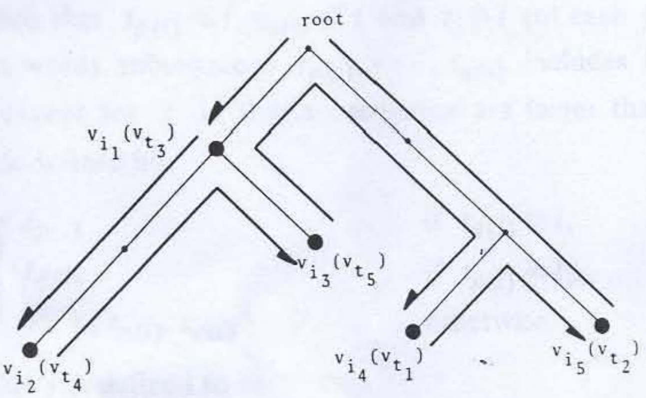


Fig. 1. An example of a preorder traversal of an optimal tree ($d(v_{i_3}, v_{i_4}) = \max\{d(v_{i_1}, v_{i_2}), \cdots, d(v_{i_4}, v_{i_5}), d(v_{i_5}, v_{i_1})\}$).

preorder traversal [1, p. 54] of an optimal tree from an arbitrary vertex (see Fig. 1). Then $d(v_{i_1}, v_{i_2}) + \cdots + d(v_{i_{k-1}}, v_{i_k}) + d(v_{i_k}, v_{i_1}) = 2\cdot\text{OPTIMAL}$. Assume $d(v_{i_{r-1}}, v_{i_r}) = \max\{d(v_{i_1}, v_{i_2}), \cdots, d(v_{i_{k-1}}, v_{i_k}), d(v_{i_k}, v_{i_1})\}$ for some r, $2 \leqslant r \leqslant k$. Then setting $t_1 = i_r, \cdots, t_{k-r+1} = i_k$, $t_{k-r+2} = i_1, \cdots, t_k = i_{r-1}$, we have $d(v_{t_{j-1}}, v_{t_j}) \leqslant d(v_{t_k}, v_{t_1})$ for all $2 \leqslant j \leqslant k$. Hence $d(v_{t_k}, v_{t_1}) \geqslant (2/k)\cdot\text{OPTIMAL}$. □

Let APPROXIMATE be the cost of the obtained tree T_k by the algorithm. Then APPROXIMATE is equal to $\Sigma_{i=2}^{k} \hat{c}(V_{i-1}, v_i)$.

Theorem 1. *For all n and $k\,(2 \leqslant k \leqslant n-1)$,*

$$\text{APPROXIMATE} / \text{OPTIMAL} \leqslant 2\cdot(1 - 1/k).$$

Moreover if $k = n$, APPROXIMATE is equal to OPTIMAL.

Proof. If $k = n$, the algorithm is Prim's algorithm [7] computing a minimum-cost spanning tree. Hence the latter half of the theorem is proved.

Since the cost of $\text{PATH}(V_{i-1}, v_i)$ is minimum among all paths between vertices in V_{i-1} and vertices in $S - V_{i-1}$, we have

(1) $$\hat{c}(V_{i-1}, v_i) \leqslant d(v_p, v_q) \qquad \text{for all } 2 \leqslant i \leqslant k$$

if $1 \leqslant \min\{p, q\} \leqslant i-1$ and $i \leqslant \max\{p, q\} \leqslant k$. By Lemma 1 there is a permutation $t_1, t_2, \cdots, t_k$ of $1, 2, \cdots, k$ such that

(2) $$d(v_{t_1}, v_{t_2}) + \cdots + d(v_{t_{k-1}}, v_{t_k}) + d(v_{t_k}, v_{t_1}) = 2\cdot\text{OPTIMAL}$$

and

(3) $$d(v_{t_k}, v_{t_1}) \geqslant (2/k)\cdot\text{OPTIMAL}.$$

We can construct a one-to-one correspondence between numbers $i, i = 2, 3, \cdots, k$ and pairs (t_{j-1}, t_j), $j = 2, 3, \cdots, k$, such that

$$\hat{c}(V_{i-1}, v_i) \leqslant d(v_{t_{j-1}}, v_{t_j}).$$

Such a correspondence can be established by the method which Rosenkrantz, et al. used in more general case [8, Proof of Lemma 3]. For each i with $i \geqslant 2$, consider the longest subsequence $t_{p(i)}, t_{p(i)+1}, \cdots, i, \cdots, t_{q(i)-1}, t_{q(i)}$ including i of sequence $t_1, t_2, \cdots, t_k$ such that $t_{p(i)} \leqslant i$, $t_{q(i)} \leqslant i$ and $t_j \geqslant i$ for each $j, j = p(i)+1, \cdots, q(i)-1$. In other words, subsequence $t_{p(i)}, \cdots, t_{q(i)}$ includes i, and all the intermediate numbers except for i in that subsequence are larger than i. The *critical number* i^* for i is defined by

$$i^* = \begin{cases} t_{p(i)} & \text{if } t_{q(i)} = i, \\ t_{q(i)} & \text{if } t_{p(i)} = i, \\ \max\{t_{p(i)}, t_{q(i)}\} & \text{otherwise.} \end{cases}$$

The *critical pair* for i is defined to be

$$(t_{p(i)}, t_{p(i)+1}) \qquad \text{if } i^* = t_{p(i)},$$
$$(t_{q(i)-1}, t_{q(i)}) \qquad \text{if } i^* = t_{q(i)}.$$

Next we show that no two numbers can have the same critical pair. Assume to the contrary that i and j $(i<j)$ have the same critical pair (t_{m-1}, t_m). Assume that $t_m < t_{m-1}$. Then t_m is critical for i and j, and $m = q(i) = q(j)$. Since all the intermediate numbers in the subsequence from j to t_m of subsequence $t_1, t_2, \cdots, t_k$ are larger than j, number i can not be in that sequence. This implies number j is in the sequence from i to t_m. Since $t_m < i$, all the numbers in the sequence from i to j are larger than t_m. Thus $t_{p(j)} > t_m = t_{q(j)}$. This contradicts the assumption that t_m is critical for j. The same contradiction is concluded when $t_m > t_{m-1}$.

Let $[t_{m(i)-1}, t_{m(i)}]$ be the critical pair for i, then from (1) we have, since $\min\{t_{m(i)-1}, t_{m(i)}\} < i \leqslant \max\{t_{m(i)-1}, t_{m(i)}\}$ holds,

$$\hat{c}(V_{i-1}, v_i) \leqslant d(v_{t_{m(i)-1}}, v_{t_{m(i)}}). \tag{4}$$

From (2), (3) and (4), we have

$$\begin{aligned} \text{APPROXIMATE} &= \sum_{i=2}^{k} \hat{c}(V_{i-1}, v_i) \\ &\leqslant \sum_{i=2}^{k} d(v_{t_{m(i)-1}}, v_{t_{m(i)}}) = \sum_{p=2}^{k} d(v_{t_{p-1}}, v_{t_p}) \\ &= 2\cdot\text{OPTIMAL} - d(v_{t_k}, v_{t_1}) \\ &\leqslant 2\cdot(1-1/k)\cdot\text{OPTIMAL}. \quad \square \end{aligned}$$

If $k \leqslant n-1$, we can construct graphs for which the ratio is equal to $2\cdot(1-1/k)$.

Theorem 2. *For all n and k $(2 \leqslant k \leqslant n)$, there exists a graph for which*

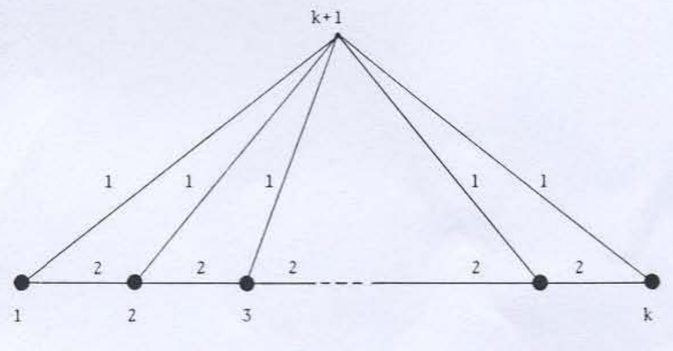
$$\text{APPROXIMATE} / \text{OPTIMAL} = 2\cdot(1-1/k).$$

Proof. Let V be the set of integers, $\{1, 2, \cdots, n\}$, E be the set $\{(i, j) \mid i = 1, 2, \cdots, n, j = 1, 2, \cdots, n\}$, and S be the set $\{1, 2, \cdots, k\}$. Suppose that

$$c(i,j) = \begin{cases} 1 & i = 1, 2, \cdots, k,\ j = k+1, \\ 2 & i = 1, \cdots, k-1,\ j = i+1, \\ 10 & \text{otherwise} \end{cases}$$

(see Fig. 2). It is evident that the tree $(S \cup \{k+1\}, \{(i, k+1) \mid i = 1, 2, \cdots, k\})$

Fig. 2.



is obtainable by the algorithm and the cost of this tree is $2 \cdot (k-1)$. The ratio is then established by dividing $2 \cdot (k-1)$ by OPTIMAL. □

By Theorems 1 and 2, the worst case ratio of APPROXIMATE to OPTIMAL is $2 \cdot (1 - 1/k)$.

The authors have studied two other types of approximate solutions which can be computed in time complexity $O(n^2)$; (1) a tree obtained from a minimum-cost spanning tree for $G = (V, E)$ by deleting edges not essential in order to connect vertices in S, and (2) a union of $k-1$ shortest paths from a single vertex in S. We have been able to show that cost ratios for these solutions are tightly bounded by $n - k + 1$ and $k - 1$, respectively. It follows that there is little reason to consider these types of approximations further.

Acknowledgment. Sincere thanks are due to Professor E. L. Lawler for his suggestion of a simple proof technique in Theorem 1.

References

[1] A. V. Aho, J. E. Hopcroft and J. D. Ullman: The Design and Analysis of Computer Algorithms, Addison-Wesley, Reading, Mass. (1974).

[2] D. Cheriton and R. E. Tarjan: Finding minimum spanning trees, SIAM J. Comput., **5** (1976), 724–742.

[3] E. W. Dijkstra: A note on two problems in connexion with graphs, Numer. Math., **1** (1959). 269–271.

[4] S. E. Dreyfus and R. A. Wagner: The Steiner problem in graphs, Networks, **1** (1972), 195–207.

[5] R. M. Karp: Reducibility among combinatorial problems, Complexity of Computer Computations, R. E. Miller and J. W. Thatcher, eds., Plenum Press, New York (1972).

[6] J. B. Kruskal, Jr.: On the shortest spanning subtree of a graph and the traveling salesman problem, Proc. Amer. Math. Soc., 7 (1956), 48–50.

[7] R. C. Prim: Shortest connection networks and some generalizations, Bell System Tech. J., **36** (1957), 1389–1401.

[8] D. J. Rosenkrantz, R. E. Stearns and P. M. Lewis II: An analysis of several heuristics for the traveling salesman problem, to appear in SIAM J. Comput..

[9] E. N. Gilbert and H. O. Pollak: Steiner Minimal Trees, SIAM J. Appl. Math., **16** (1968), 1–20.