1999年度日本オペレーションズ・リサーチ学会
秋季研究発表会

On Minimum Edge Ranking Spanning Trees

02601514 大阪大学 牧野 和久 (MAKINO Kazuhisa)
01012394 大阪府立大学 ○宇野 裕之 (UNO Yushi)
01001374 京都大学 茨木 俊秀 (IBARAKI Toshihide)

1 Introduction

Let $G = (V, E)$ be an undirected graph, which is simple and connected. An *edge ranking* of a graph $G = (V, E)$ is a labeling $r: E \to Z^+$, with the property that every path between two edges with the same label i contains an intermediate edge with label $j > i$. An edge ranking by integers $1, 2, \dots, k$ is called a *k-edge ranking*. A graph G is said to be *k-edge rankable* if it has a k-edge ranking. An edge ranking is *minimum* if the largest rank k in it is the smallest among all edge rankings of G; such k is called the *minimum edge rank* of G and is denoted by $\mathrm{rank}(G)$. The *minimum edge ranking problem* (MER) asks to compute a minimum edge ranking of a given graph G. It is known that MER is in general NP-hard [6], but it can be solved in polynomial time when the graph is a tree [1, 5].

In this paper, we newly consider the following problem, which resembles MER but is essentially different.

MERST (minimum edge ranking spanning tree problem)
Input: A simple undirected graph $G = (V, E)$ which is connected, and a nonnegative integer k.
Question: Does G have a k-edge rankable spanning tree (i.e., does there exist a spanning tree $T = (V, E_T)$ of G with $\mathrm{rank}(T) \le k$)?

Fig. 1 gives an example of a minimum edge ranking spanning tree of a graph G, together with its edge ranking. Problem MERST can be found in many practical applications.

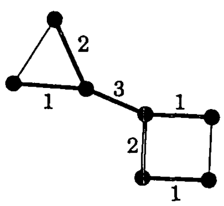

Figure 1: A minimum edge ranking spanning tree T of the graph G.

In this paper, we show that MERST is NP-hard, and present an approximation algorithm for MERST, with its worst case performance ratio $\min\{(\Delta^* - 1)\log n/\Delta^*, \Delta^* - 1\}/(\log(\Delta^* + 1) - 1)$, where n is the number of vertices in G and Δ^* is the maximum degree of a spanning tree whose maximum degree is minimum.

2 NP-hardness of MERST

In this section, we claim that MERST is intractable. The idea of our proof is based on the NP-hardness proof of the connected size-k-partition problem for planar bipartite graphs [2]. For a vertex set $W \subseteq V$, $G[W]$ denotes the subgraph of G induced by W.

Lemma 1 *Any connected graph G with* $\mathrm{rank}(G) = k$ *has at most* 2^k *vertices.*

For a graph $G = (V, E)$ and a positive integer k, a *size-k-partition* of V is a $(|V|/k)$-tuple $(V_1, V_2, \dots, V_{|V|/k})$ and $V = V_1 \cup V_2 \cup \cdots \cup V_{|V|/k}$, $V_i \cap V_j = \emptyset$ for all $i \ne j$ such that $|V_i| = k$ for $i = 1, 2, \dots, |V|/k$. Each V_i is called an *element* of the partition. A size-k-partition of V is *connected* if the graphs $G[V_i]$ are connected for all i. Let $G = (V, E)$ be a graph with $|V| = 2^k$, where $k \ge 0$. We say that G has a *nested partition* if it recursively satisfies one of the following conditions:
(i) $k = 0$, or
(ii) G has a connected size-2^{k-1}-partition (V_1, V_2) such that both $G[V_1]$ and $G[V_2]$ have nested partitions.

Lemma 2 *Let $G = (V, E)$ be a graph with $|V| = 2^k$ ($k \ge 0$). Then G has a k-edge rankable spanning tree if and only if it has a nested partition.*

This lemma provides the essential idea of NP-completeness proof of MERST, i.e., to find a k-edge rankable spanning tree of G is equivalent to find a nested partition of G.

Theorem 1 *MERST is NP-complete.*

3 An Approximation Algorithm for MERST

Since MERST is NP-hard, we propose an approximation algorithm, which is a combination of two existing algorithms for the minimum degree spanning tree problem (MDST) and for the minimum edge ranking problem of trees (which is MER whose input graphs are restricted to be trees). We state its approximation ratio here, and analyze the algorithm in the next section.

We denote the maximum degree of vertices in a graph G by Δ_G, and the maximum degree of the minimum degree spanning tree T of G by Δ^* ($= \Delta_T$). Although MDST is known to be NP-hard [4], Fürer and Raghavachari [3] developed a polynomial time approximation algorithm which computes a spanning tree T satisfying

$$\Delta^* \le \Delta_T \le \Delta^* + 1 \ (\le \Delta_G). \qquad (1)$$

Our approximation algorithm for MERST first computes a spanning tree T_{Approx} of G satisfying (1) (by using the algorithm in [3]), and then computes its minimum edge ranking. Recall that MERT is polynomially solvable (e.g., [5]). Thus, our algorithm described below can be executed in polynomial time.

Algorithm APPROX_MERST
Input: A graph $G = (V, E)$.
Output: A spanning tree T of G and its edge ranking r.
Step 1: Compute a spanning tree T_{Approx} of G satisfying (1).
Step 2: Compute a minimum edge ranking r of T_{Approx}.
Step 3: Output $T = T_{\text{Approx}}$ and its edge ranking r.

Theorem 2 *For a graph $G = (V, E)$ with $|V| = n$, let T_{Min} denote a minimum edge ranking spanning tree of G, and let T_{Approx} denote a spanning tree of G computed by algorithm* APPROX_MERST *for the input G. Then, the approximation ratio of algorithm* APPROX_MERST *can be bounded from above by*

$$\frac{\text{rank}(T_{Approx})}{\text{rank}(T_{Min})} \le \frac{\min\{(\Delta^*-1)\log n/\Delta^*, \Delta^*-1\}}{\log(\Delta^*+1)-1},$$

where Δ^* is the maximum degree of the minimum degree spanning tree of G.

4 Analysis of Edge Ranking of Trees

In this section, we derive upper and lower bounds on $\text{rank}(T)$ of a tree $T = (V, E_T)$ in terms of the number of vertices $n = |V|$ and its maximum degree Δ_T, in order to prove the approximation ratio of algorithm APPROX_MERST.

Lemma 3 *For any tree $T = (V, E_T)$, $\text{rank}(T) \ge \max\{\Delta_T, \lceil \log n \rceil\}$ holds, where Δ_T is the maximum degree of vertices in T and $n = |V|$.*

Lemma 4 *Let $T = (V, E_T)$ be a tree with $|V| = n$. Then it holds that*

$$\text{rank}(T) = \lceil \log n \rceil \qquad \text{if } \Delta_T = 0,1,2 \qquad (2)$$

$$\text{rank}(T) \le \frac{(\Delta_T - 2)\log n}{\log \Delta_T - 1} \qquad \text{if } \Delta_T \ge 3. \qquad (3)$$

This lemma, together with Lemma 3, proves Theorem 2, since the algorithm of Fürer and Raghavachari [3] can find a spanning tree T of G such that $\Delta^* \le \Delta_T \le \Delta^* + 1$ in the first step of APPROX_MERST.

Let $T_{(d,h)}$ denote a tree in which all the inner vertices have the same degree d and there exists a vertex v_0 such that the distances between v_0 and all the leaves are exactly h. This $T_{(d,h)}$ attains the upper bound of Lemma 4.

Lemma 5 *Let d and h be integers such that $d \ge 3$ and $h \ge 2$. Then, $T_{(d,h)}$ satisfies* $\text{rank}(T_{(d,h)}) \ge \dfrac{(d-2)\log n}{\log(d-1)}$.

References

[1] de la Torre, P., Greenlaw, R. and Schäffer, A. A., Optimal edge ranking of trees in polynomial time, *Algorithmica*, 13, 592–618, 1995.

[2] Dyer, M. E. and Frieze, A. M., On the complexity of partitioning graphs into connected subgraphs, *Discrete Applied Mathematics*, 10, 139–153, 1985.

[3] Fürer, M. and Raghavachari, B., Approximating the minimum-degree Steiner tree to within one of optimal, *Journal of Algorithms*, 17, 409–423, 1994.

[4] Garey, M. R. and Johnson, D. S., *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman and Company, New York, 1979.

[5] Lam, T. W. and Yue, F. L., Optimal edge ranking of trees in linear time, *Proc. 9th ACM-SIAM SODA*, 436–445, 1998.

[6] Lam, T. W. and Yue, F. L., Edge ranking of graphs is hard, *Discrete Applied Mathematics*, 85, 71–86, 1998.