Np-hardness proof and an approximation algorithm for the minimum vertex ranking spanning tree problem

Keizo Miyata\textsuperscript{a}, Shigeru Masuyama\textsuperscript{a, *}, Shin-ichi Nakayama\textsuperscript{b}, Liang Zhao\textsuperscript{c}

\textsuperscript{a}Knowledge-Based Information Engineering, Toyohashi University of Technology, Toyohashi 441-8580, Japan
\textsuperscript{b}Mathematical Sciences, Faculty of Integrated Arts and Sciences, The University of Tokushima, Tokushima 770-8502, Japan
\textsuperscript{c}Department of Information Science, Faculty of Engineering, Utsunomiya University, Utsunomiya 321-8585, Japan

Received 14 June 2004; received in revised form 8 November 2004; accepted 21 September 2005
Available online 5 July 2006

Abstract

The minimum vertex ranking spanning tree problem (MVRST) is to find a spanning tree of $G$ whose vertex ranking is minimum. In this paper, we show that MVRST is NP-hard. To prove this, we polynomially reduce the 3-dimensional matching problem to MVRST. Moreover, we present a $(\ceil{D_s/2} + 1)/(\lfloor\log_2(D_s + 1)\rfloor + 1)$-approximation algorithm for MVRST where $D_s$ is the minimum diameter of spanning trees of $G$.

Keywords: Vertex ranking; Spanning tree; Graph theory; NP-hard; Computational complexity; Approximation algorithm

1. Introduction

Consider a simple connected undirected graph $G = (V, E)$. A vertex ranking of $G$ is labeling $r$ from the vertices of $G$ to the positive integers such that for each path between any two vertices $u$ and $v$, $u \neq v$, with $r(u) = r(v)$, there exists at least one vertex $w$ on the path with $r(w) > r(u) = r(v)$. The value $r(v)$ of a vertex $v$ is called the rank of vertex $v$. A vertex ranking by integers $1, 2, \ldots, k$ is called a $k$-vertex ranking. A graph $G$ is said to be $k$-vertex-rankable if it has a $k$-vertex ranking. A vertex ranking $r$ of $G$ is minimum if the largest rank $k$ assigned by $r$ is the smallest among all rankings of $G$. Such rank $k$ is called the vertex ranking number of $G$, denoted by $\chi(G)$. The vertex ranking problem is to find a minimum ranking of given graph $G$. The vertex ranking problem has interesting applications, e.g., to, communication network design, planning efficient assembly of products in manufacturing systems \cite{5,6,12,17}, and VLSI layout design \cite{9,16}.

As for the complexity, this problem is NP-hard even when restricted to cobipartite graphs \cite{13} and bipartite graphs \cite{1}, and a number of polynomial time algorithms for this problem have been developed on several subclasses of graphs. Much work has been done in finding the minimum vertex ranking of a tree; a linear time algorithm for trees is proposed

\* Corresponding author. Tel.: +81 532 44 6894; fax: +81 532 44 6873.
E-mail addresses: miyata@smlab.tutkie.tut.ac.jp (K. Miyata), masuyama@tutkie.tut.ac.jp (S. Masuyama), shin@ias.tokushima-u.ac.jp (S. Nakayama), zhao@is.utsunomiya-u.ac.jp (L. Zhao).

0166-218X/S - see front matter © 2006 Elsevier B.V. All rights reserved.
in [14]. The problem is trivial on split graphs and is solvable in linear time on cographs [15]. Concerning interval graphs, although an $O(n^4)$ time algorithm was known, Deogun et al. has given an $O(n^3)$ time algorithm recently [3]. They also presented $O(n^6)$ algorithms on permutation graphs and on trapezoid graphs, respectively, and showed that a polynomial time algorithm on $d$-trapezoid graphs exists [3]. Moreover, a polynomial time algorithm on graphs with treewidth at most $k$ was developed [2].

The problem described above is the ranking with respect to vertices, while a ranking with respect to edges is similarly defined as follows. An edge ranking of $G$ is labeling $r$ from the edges of $G$ to the positive integers such that for each path between any two edges $e_1$ and $e_2$, $e_1 \neq e_2$, with $r(e_1) = r(e_2)$, there exists at least one edge $e_3$ on the path with $r(e_3) > r(e_1) = r(e_2)$. The value $r(e)$ of a vertex $e$ is called the rank of edge $e$. An edge ranking of $G$ is minimum if the largest rank $k$ assigned by $r$ is the smallest among all rankings of $G$. Such rank $k$ is called the edge ranking number of $G$, denoted by $\chi_e(G)$. The edge ranking problem is to find a minimum ranking of given graph $G$. Before the proof of this problem to be NP-hard was given, an $O(n^3)$ time algorithm for trees was known [17]. By now, a linear time algorithm for trees is shown in [8]. Recently, it has finally been shown that this problem on general graphs is NP-hard [7].

Makino et al. introduced a minimum edge ranking spanning tree problem (MERST) which is related to the minimum edge ranking problem but is essentially different [10]. MERST is to find a spanning tree of $G$ whose edge ranking is minimum. They proved that MERST is NP-hard and presented an approximation algorithm for MERST. MERST has interesting applications, e.g., to scheduling the parallel assembly of a multipart product from its components and relational databases [10].

In this paper, we consider the vertex version of MERST, i.e., the minimum vertex ranking spanning tree problem (MVRST). MVRST is to find a spanning tree of $G$ whose vertex ranking is minimum [11]. This problem is related to the minimum vertex ranking problem but is also essentially different as is the case of MERST. An example of a minimum vertex ranking spanning tree is illustrated in Fig. 1. Note that for each application of the vertex ranking problem introduced previously in this paper, MVRST corresponds to a variation of the problem in that each edge is optional and a solution requires connectivity among vertices. Nakayama et al. presented $O(n^3)$ time algorithms on interval graphs [11]. However, the complexity of MVRST on general graphs was left open. In this paper, we show that MVRST is NP-hard for general graphs. To prove this, we polynomially reduce the 3-dimensional matching problem (3DM) to MVRST. Moreover, we present a $((D_s/2) + 1)/(\lfloor \log_2(D_s + 1) \rfloor + 1)$-approximation algorithm for MVRST where $D_s$ is the minimum diameter of spanning trees of $G$. (We denote $\max_{u,v} d(u,v)$ by diameter where $d(u,v)$ is a distance from $u$ to $v$).
2. MVRST is NP-hard

2.1. Preliminary

The next three observations are obvious.

Observation 1. Given a rank, there exists exactly one vertex of the maximum label (see, e.g., [3]).

Observation 2. Let \( G' \) be a subgraph of \( G \). Then \( \chi(G') \leq \chi(G) \).

Observation 3. Let \( G \) be a path of \( n \) vertices. Then \( \chi(G) = \lfloor \log_2(n) \rfloor + 1 \) [11].

2.2. NP-hardness proof for the 4-rankable spanning tree problem

Problem (4-rankable spanning tree). Given a simple connected undirected graph \( G \), determine if it has a spanning tree that is 4-rankable.

Theorem 1. The 4-rankable spanning tree problem is NP-hard.

Proof. We show the theorem by reducing the NP-hard 3DM problem, which is defined as follows.

Problem (3DM). Given mutually disjoint sets \( X, Y \) and \( Z \), \( |X| = |Y| = |Z| = n \), and a set \( S = \{(x, y, z)| x \in X, y \in Y, z \in Z \}, |S| = m \), determine if there is a matching \( M \) with \( |M| = n \), where \( M \) is called a matching if \( M \subseteq S \) and no elements in \( M \) agree in any coordinate.

In the following, we will use the bipartite graph representation of 3DM, i.e., denote the elements in \( X \cup Y \cup Z \cup S \) by vertices and for each \( s \in S \), employ three edges \((s, x), (s, y), \) and \((s, z)\) (see Fig. 2 (a), (b) for an illustration).

Reducing 3DM to the 4-rankable spanning tree problem. We show how to transform an instance of 3DM to an instance of 4-rankable spanning tree problem. Using the instance in Fig. 2, the transformation is illustrated in Fig. 3 and is explained in the following.

Suppose we are given a 3DM instance \((X, Y, Z, S)\). For simplicity, let \( A = X \cup Y \cup Z \). Firstly, for each vertex \( a \in A \), we add two new vertices \( b_a \) and \( c_a \) and two edges \((a, b_a)\) and \((b_a, c_a)\). Let \( B = \{b_a| a \in A \} \) and \( C = \{c_a| a \in A \} \). Then we add \( n \) vertices \( p \in P, |P| = n \), and \( mn \) edges \((p, s)\) for all \( p \in P \) and \( s \in S \). Finally, we add nine vertices \( q \in Q = \{q_i| -4 \leq i \leq 4 \} \) and \( n + 8 \) edges \((q_0, p), p \in P \), and \((q_i, q_{i+1}), -4 \leq i \leq 3 \). See Fig. 3 for an illustration.

Clearly, this is a polynomial time reduction: the number of vertices and the number of edges in the new graph are \( 10n + m + 9 \) and \( mn + 7n + 4m + 8 \), respectively, whereas in the input 3DM instance, they are \( 3n + m \) and \( 3m \), respectively. Denote the new graph by \( G \). In the next, we show that the 3DM instance can be solved by solving the
4-rankable spanning tree problem in $G$. For this, we must show the next two facts, where spanning tree and matching are used with respect to $G$ and to the input 3DM instance, respectively.

(1) There must exist a 4-rankable spanning tree if there is a matching of cardinality $n$.
(2) There must exist a matching of cardinality $n$ if there is a 4-rankable spanning tree.

Let us show them in the following.

Consider (1). Suppose there is a matching $M$ of cardinality $n$. For simplicity, let $M = \{s_1, s_2, \ldots, s_n\}$ and $P = \{p_1, p_2, \ldots, p_n\}$ ($s_i \neq s_j, p_i \neq p_j$ if $i \neq j$). Let $T$ be the spanning tree obtained by the edges $(q_i, q_{i+1})$ ($-1 \leq i \leq 3$), $(q_0, p)$ (for all $p \in P$), $(p_i, s_i)$ ($1 \leq i \leq n$), $(p_i, s)$ ($s \in S - M$), $(s_i, a)$ ($1 \leq i \leq n, a \in A, (s_i, a)$ exists in $G$), $(a, b_a)$ and $(b_a, c_a)$ ($a \in A$). Notice that $T$ is a spanning tree since $M$ is a matching and is of cardinality $n$. Fig. 4 illustrates the constructed spanning tree for Fig. 3.
Let us label vertices in $C \cup A \cup (S - M) \cup \{q_{-4}, q_{-2}, q_2, q_4\}$ by 1, vertices in $B \cup P \cup \{q_{-3}, q_3\}$ by 2, vertices in $M \cup \{q_{-1}, q_1\}$ by 3, and vertices by 4. It is easy to verify that this is a 4 ranking for $T$. See also Fig. 4 for an illustration.

Thus we only need to show (2) in order to prove the theorem. Suppose there is a 4-rankable spanning tree $T'$ of $G$. We show how to construct a matching of cardinality $n$. In the following, let us mention edges and paths with respect to $T'$. Let $r$ be a rank of $T'$ with at most four labels.

**Observation 4.** The label of $q_0$ is 4.

**Proof.** Notice that the path $P_1$ from $q_{-4}$ to $q_4$ consists of nine vertices, and the path $P_2$ (resp., $P_3$) from an arbitrary vertex $c \in C$ to $q_{-4}$ (resp., to $q_4$) consists of at least 10 vertices. By Observation 3, there must exist a label-4 vertex on every one of paths $P_1$, $P_2$, and $P_3$. Here, we call a vertex with rank $k$ a label-$k$ vertex. Hence the only one common vertex $q_0$ of them must be labeled by 4 (Observation 1). □

**Observation 5.** There are at most $n$ label-3 vertices in $C \cup B \cup A \cup S$.

**Proof.** Consider the paths from label-3 vertices in $C \cup B \cup A \cup S$ to a label-3 vertex in $Q$ (the latter exist by the proof of Observation 3). If there are at least $n + 1$ label-3 vertices in $C \cup B \cup A \cup S$, then at least two paths must share a common vertex in $P \cup A$. (If there is no common vertex shared by at least two paths in $A$, at least two paths must share a common vertex in $P$ (since any path must have at least one vertex in $P$ and $|P| = n$).) This is a contradiction since it results in a path with two label-3 endpoints but no label-4 vertex on it. □

Now let $M$ be the set of vertices in $S$ that are labeled by 3. Let $k = |M|$. By the above observation, we have $k \leq n$. Let $A_M$ be the set of vertices in $A$ that are adjacent to $M$. Finally, we show $k = n$ and $A_M = A$, which will prove (2), $M$ is a matching of cardinality $n$ (notice that each vertex of $S$ can be adjacent to at most three vertices in $A$).

Consider a vertex $a \in A - A_M$. It must be adjacent to some vertex $s$ in $S - M$, we need (at least) three labels to label them. Therefore, (at least) one of vertices $c_a, b_a$, and $a$ must be labeled by 3 (notice that $r(s) \leq 2$ as it is not in $M$).

Since $|A - A_M| \geq 3(n - k)$ (as $A_M \leq 3k$), the above discussion shows that there are at least $3(n - k)$ label-3 vertices in $C \cup B \cup A$. Therefore, the number of label-3 vertices in $C \cup B \cup A \cup S$ is at least $3(n - k) + k$. By Observation 4, we have

\[ 3(n - k) + k \leq n \iff n \leq k. \]

Combining the above inequality with $k \leq n$, we have $k = n$. Notice that the above discussion also shows $A_M = A$. Therefore, the proof for the theorem is complete. □

3. An approximation algorithm for MVRST

In this section, we consider MVRST as an optimization problem. Since MVRST is NP-hard, we propose an approximation algorithm APPROX_MVRST below.

**APPROX_MVRST**

**Step 1.** Given a graph $G = (V, E)$. For each $u \in V$, find a shortest path from $u$ to every other vertex in $G$. Let $L(u) = \max_{v \in V} d(u, v)$, where $d(u, v)$ is the distance between $u$ and $v$.

**Step 2.** Choose $s \in V$ such that $L(s) = \min_{u \in V} L(u)$. Obtain a spanning tree $T$ by breadth first search (BFS) starting from $s$, and obtain $\chi(T)$.

An example of a spanning tree of $G$ in Fig. 5 obtained by APPROX_MVRST is illustrated in Fig. 6, and a minimum vertex ranking spanning tree of $G$ is illustrated in Fig. 7.

**3.1. Analysis of complexity**

Let $|V| = n$ and $|E| = m$, respectively. As the undirected and unweighted version of the single source shortest paths problem can be solved in $O(m)$ by BFS. The time complexity of Step 1 is $O(mn)$. The time complexity to obtain a spanning tree by BFS is $O(m)$, and that to obtain the minimum vertex ranking of a tree is $O(n)$ [14]. Thus, the time complexity of APPROX_MVRST is $O(mn)$. 
3.2. Analysis of approximation ratio

We discuss some properties on the minimum vertex ranking of trees in order to obtain an approximation ratio of algorithm APPROX_MVRST. Let $T$ be the spanning tree of $G$ obtained by APPROX_MVRST, $D$ be the diameter of
Fig. 8. A path whose length is 6.

Fig. 9. A complete ternary tree whose tree depth is 3 ($D = 6$).

Let $T, T^*$ be an MVRST of $G$ and $D^*$ be the diameter of $T^*$, respectively. When the input graph $G$ consists of exactly one vertex, it is clear that the approximation ratio is 1. In the following, we consider the case where at least two vertices exist in $G$, and $\chi(T^*)$ is at least 2. In order to obtain the approximation ratio $\frac{D^*}{D^* + 1} \leq \frac{\chi(T)}{\chi(T^*)}$, we will examine a lower bound of $\chi(T^*)$ and an upper bound of $\chi(T)$.

First, we obtain a lower bound of $\chi(T^*)$. Let $P$ be a path in $T^*$ whose length is $D^*$. Clearly, $P$ is a subgraph of $T^*$, and $\chi(P) \leq \chi(T^*)$ by Observation 2. Thus, by Observation 3 and by noting that the number of vertices in $P$ is $D^* + 1$, we have $\left\lfloor \log_2(D^* + 1) \right\rfloor + 1 \leq \chi(T^*)$ (Figs. 8, 9).

Now, let $D_s$ be the minimum diameter of spanning trees of $G$. $\left\lfloor \log_2(D_s + 1) \right\rfloor + 1 \leq \chi(T^*)$.

Next, we consider an upper bound of $\chi(T)$.

**Observation 6.**

\[ \chi(T) \leq \left\lfloor \frac{D}{2} \right\rfloor + 1. \]

**Proof.** We prove Observation 6 by reductio ad absurdum.

Assume that $\chi(T) \geq \left\lfloor \frac{D}{2} \right\rfloor + 2$. Let $u$ be the vertex satisfying $\max_{v \in V} d(u, v) = \left\lfloor \frac{D}{2} \right\rfloor$ (if $D$ is odd, exactly two vertices satisfy this condition and let $u$ be either of them). Now, let $r(u) = \left\lfloor \frac{D}{2} \right\rfloor + 2$, and $r(v) = \left\lfloor \frac{D}{2} \right\rfloor + 2 - d(u, v)$ ($\forall v \in V (u \neq v)$), and we can assign label $r$ to the vertices in $T$ without contradiction to the definition of vertex ranking. Note that $\min_{v \in V} r(v) \geq 2$ by $\max_{v \in V} d(u, v) = \left\lfloor \frac{D}{2} \right\rfloor$. Thus, as $T$ can be ranked without contradiction even if 1 is subtracted from the rank of each vertex, $T$ is $\left\lfloor \frac{D}{2} \right\rfloor + 1$-rankable, which contradicts our initial assumption. \qed

We now show that the upper bound of $\chi(T)$ in Observation 6 is tight.

**Observation 7.** A complete ternary tree $T'$ whose diameter is $D$ satisfies $\chi(T') = \left\lfloor \frac{D}{2} \right\rfloor + 1$. 

Proof. Let $d$ be the tree depth of a complete ternary tree $T'$. By configuration of the complete ternary tree, $d = \lceil D/2 \rceil$. We show $\chi(T') = d + 1$. Note that we also call an undirected tree obtained from a complete ternary tree by neglecting the direction of each edge, also a complete ternary tree, when no confusion may arise.

We prove Observation 7 by induction with respect to $d$.

If $d = 1$, clearly $\chi(T') = 2$. Assume that the formula holds for $d \leq k$. We show that the formula holds for $d = k + 1$. Let $u$ be the root of $T'$ with tree depth $k + 1$. Let $T_1, T_2, T_3$ be the subtrees obtained from $T'$ by removing $u$ and edges incident to $u$ where $T_i$ has the longest tree depth, and $d_i$ is the tree depth of $T_i$, $i = 1, 2, 3$. Then $k = d_1 \geq d_2, d_3 \geq k - 1$.

Now, consider the subtree $(T' - T_1)$ which is a subtree obtained by removing all vertices and edges in $T_1$ and edge $(u, v_1)$ where $v_1$ is the root of $T_1$. By the assumption, there exists a vertex whose rank is at least $k$ in each $T_2, T_3$, thus $\chi(T' - T_1) \geq k + 1$ by the definition of vertex ranking. Moreover, $\chi(T_1) = k + 1$ by the assumption. Thus, there exists a vertex whose rank is at least $k + 1$ in each $(T' - T_1), T_1$, and $\chi(T') \geq k + 2$.

By Observation 6 and $\chi(T') \geq k + 2$, $\lceil D/2 \rceil + 2 = d + 2 = k + 3 > \chi(T') \geq k + 2$, thus $\chi(T') = k + 2 = (k + 1) + 1$. Hence, we have shown that, if Observation 7 hold for $d \leq k$, then it holds for $d = k + 1$, and therefore, Observation 7 holds for all $d$. □

Applying APPROX_MVRST to $T'$, the upper bound of (3) in Observation 6 is obviously attained as $\chi(T') = \lceil D/2 \rceil + 1$. Now, consider $D_s$ and $D$. By APPROX_MVRST, if $D_s$ is even, $D_s = D$. Moreover, if $D_s$ is odd, $D_s \leq D \leq D_s + 1$.

Thus, $\lceil D/2 \rceil = \lceil D_s/2 \rceil$.

By $\lceil D/2 \rceil = \lceil D_s/2 \rceil$ and Observations 6 and 8,

$$\chi(T) \leq \left\lceil \frac{D_s}{2} \right\rceil + 1.$$  

(4)

By multiplying both sides of inequalities (2) and (4), we have the following theorem. Note that $\chi(T^*) \leq \chi(T)$ is obvious by definition.

Theorem 2. A spanning tree $T$ obtained by APPROX_MVRST satisfies

$$\chi(T^*) \leq \chi(T) \leq \delta \chi(T^*),$$  

(5)

where $T^*$ is an MVRST and

$$\delta = \frac{\lceil D_s/2 \rceil + 1}{\lceil \log_2(D_s + 1) \rceil + 1}.$$  

(6)

4. Conclusion

In this paper, we showed that MVRST is NP-hard for general graphs and presented a $(\lceil D_s/2 \rceil + 1)/\lceil \log_2(D_s + 1) \rceil + 1)$-approximation algorithm for MVRST where $D_s$ is the minimum diameter of spanning trees of $G$.

References


Further reading