

Minimum vertex ranking spanning tree problem for chordal and proper interval graphs

Dariusz Dereniowski*

Department of Algorithms and System Modeling,
Gdańsk University of Technology, Poland

deren@eti.pg.gda.pl

Abstract: A vertex k -ranking of a simple graph is a coloring of its vertices with k colors in such a way that each path connecting two vertices of the same color contains a vertex with a bigger color. Consider the minimum vertex ranking spanning tree (MVRST) problem where the goal is to find a spanning tree of a given graph G which has a vertex ranking using the minimal number of colors over vertex rankings of all spanning trees of G . K. Miyata et. al. proved in [Np-hardness proof and an approximation algorithm for the minimum vertex ranking spanning tree problem, *Discrete Appl. Math.* 154 (2006) 2402-2410] that the decision problem: given a simple graph G , decide whether there exists a spanning tree T of G such that T has a vertex 4-ranking, is NP-complete. In this paper we improve this result by proving NP-hardness of finding for a given chordal graph its spanning tree having vertex 3-ranking. This bound is the best possible. On the other hand we prove that MVRST problem can be solved in linear time for proper interval graphs.

Keywords: computational complexity, vertex ranking, spanning tree

1 Introduction

A *vertex k -ranking* of a graph G is a function $c: V(G) \rightarrow \{1, \dots, k\}$ such that each path connecting two vertices of the same color contains a vertex with a bigger color. The symbol $\chi_r(G)$ is called the *vertex ranking number* of G and is equal to the smallest number k such that there exists a vertex k -ranking of G . A vertex ranking of G is *optimal* if it uses $\chi_r(G)$ colors. Finding an optimal vertex ranking is hard in general [15] and also for chordal graphs [7]. The problem can be solved in linear time for trees [16], in $O(n^{3d})$ time for d -trapezoid graphs [4], which implies an $O(n^3)$ time algorithm for interval and circular-arc graphs, and an $O(n^6)$ time algorithm for permutation

*Partially supported by the Foundation for Polish Science (FNP)

and trapezoid graphs. There exists an $O(n + m)$ time optimal algorithm for computing vertex ranking of a starlike graph [8].

An *edge k -ranking* of a graph G is a function $c: E(G) \rightarrow \{1, \dots, k\}$ such that each path connecting two edges of the same color contains an edge with a bigger color. The *edge ranking number* of G , denoted by $\chi'_r(G)$, equals the smallest integer k such that there exists an edge k -ranking of G .

Makino, Uno and Ibaraki introduced the Minimum Edge Ranking Spanning Tree (MERST) problem [11], where the goal is to find a spanning tree T of G , such that the edge ranking number of T is minimum over all spanning trees of G . This problem is hard for general graphs and there exists a polynomial time approximation algorithm with a sublinear approximation ratio [11]. The bound for performance guarantee of this algorithm has been improved asymptotically in [6]. There exist exact polynomial-time algorithms solving the MERST problem for threshold and split graphs [10]. The problem turns out to be NP-complete for series-parallel graphs [1]. In [9] an approximation algorithm for series-parallel graphs is given. The MERST problem has potential applications in parallel query processing in relational databases [11], and a special modification of this problem has been used to for creating search strategies in partial orders [5].

The vertex version of the MERST problem has been defined in [11]. Formally, given a simple graph G , the goal is to find a spanning tree T of G such that the vertex ranking number of T is minimum over the vertex ranking numbers of all spanning trees of G . The authors conjectured in [11] that the Minimum Vertex Ranking Spanning Tree (MVRST) problem is also hard in general and left their hypothesis as an open question. In the decision version of the problem a graph G and an integer k are given and we ask about the existence of a spanning tree T satisfying $\chi_r(T) \leq k$. We use the notation k -MVSRT to denote the problem with a fixed k and a simple graph G as an input. Authors in [12] proved that 4-MVRST problem is NP-complete. In this paper we improve this result: 3-MVRST problem is also NP-complete. Section 2 gives a polynomial time reduction from the Minimum Set Cover problem to the MVRST problem for chordal graphs with diameter at most 6. Few classes of graphs are known for which an optimal solution to this problem can be computed in polynomial-time, the examples are interval graphs [13] and outerplanar graphs [14]. In Section 3 we show a nontrivial class of graphs (proper interval graphs) for which the MVRST problem can be solved in linear time. Although the work of Nakayama et. al. [13] implies the existence of an optimal and polynomial-time algorithm, but the running time of their procedure is $O(n^3)$.

2 3-MVRST problem is hard for chordal graphs

In this section we propose a simple polynomial-time reduction from the Minimum Set Cover problem (MSC) to the 3-MVRST problem. The definition of the MSC problem is as follows:

Input: a set $S = \{a_1, \dots, a_n\}$, a collection C of subsets of S ($C = \{S_1, \dots, S_m\}$, where $S_i \subseteq S$ for each $i = 1, \dots, m$) and an integer $k > 0$;

Question: Does it exist $C' \subseteq C$ such that $|C'| \leq k$ and $\bigcup C' = S$?

Given an instance of the MSC problem, define a simple graph G .

$$V(G) = V[S] \cup V[C] \cup \{w_1, \dots, w_k\} \cup \{r_0, \dots, r_3\},$$

where $V[S] = \{v[a_1], \dots, v[a_n]\}$ contains the vertices corresponding to the elements in S while $V[C] = \{v[S_1], \dots, v[S_m]\}$ contains the vertices corresponding to the sets S_i . Then,

$$E(G) = \{\{v[a_i], v[S_j]\} : a_i \in S_j\} \cup \{\{x, y\} : x, y \in V[C] \cup \{w_1, \dots, w_k\}\} \cup \{\{r_0, w_i\} : i = 1, \dots, k\} \cup \{\{r_i, r_{i+1}\} : i = 0, 1, 2\}.$$

Let us give an example of a graph G for a given instance of the MSC problem. Let $n = 6$, $m = 5$, $k = 2$, $S_1 = \{a_1, a_2\}$, $S_2 = \{a_2, a_3, a_4\}$, $S_3 = \{a_3\}$, $S_4 = \{a_1, a_5, a_6\}$, $S_5 = \{a_4, a_5, a_6\}$. Fig. 1(a) depicts the corresponding graph G .

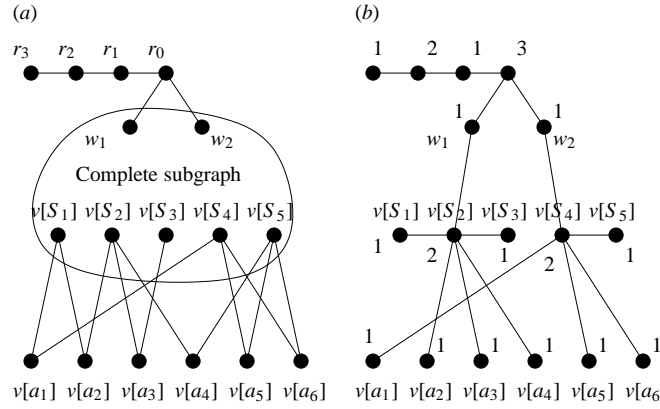


Figure 1: (a) a graph G and (b) a spanning tree T of G with $\chi_r(T) = 3$

Lemma 1 *If C' is a solution to the MSC problem satisfying $|C'| \leq k$ then there exists such a spanning tree T of G that $\chi_r(T) = 3$.*

Proof: We define a vertex 3-ranking c together with a spanning tree T . Clearly, $\{r_i, r_{i+1}\} \in E(T)$ for $i = 0, 1, 2$. Let $c(r_0) = 3, c(r_1) = c(r_3) = 1, c(r_2) = 2$. Let $c(v[S_j]) = 2$ if $S_j \in C'$. All the remaining vertices of G get color 1. Define $\{r_0, w_i\} \in E(T)$, $i = 1, \dots, k$. For each $v[S_j]$ such that $c(v[S_j]) = 2$ find a unique vertex w_l and add $\{w_l, v[S_j]\}$ to $E(T)$. Such a definition is correct, because $|C'| \leq k$. Finally, for each $i = 1, \dots, n$ find $j \in \{1, \dots, m\}$ such that $a_i \in S_j$, $S_j \in C'$, and add $\{v[a_i], v[S_j]\}$ to $E(T)$. If $c(v[S_j]) = 1$, $j \in \{1, \dots, m\}$ then find any vertex $v[S_l]$ colored with 2 and add $\{v[S_j], v[S_l]\}$ to $E(T)$. It is easy to see that T is a spanning tree. Note that $T - r_0$ is a union of stars with central vertices r_2 and $v[S_j]$, where $S_j \in C'$. So, the central vertex of each star has color 2 and the leaves are colored with 1. This proves that c is a ranking. \square

The construction of T and its vertex 3-ranking shown in the proof of Lemma 1 are given in Fig. 1(b). The only solution to the MSC problem in that case is $C' = \{S_2, S_4\}$.

Lemma 2 *If T is a spanning tree of G such that $\chi_r(T) = 3$ then there exists a solution to the MSC problem.*

Proof: Let c be a vertex 3-ranking of T . Note that $\{r_i, r_{i+1}\} \in E(T)$. So, one of the vertices r_0, \dots, r_3 gets color 3. If $c(r_0) \neq 3$ then we may modify c in such a way that $c(r_0) = 3, c(r_1) = c(r_3) = 1$ and $c(r_2) = 2$. Clearly, c is a valid vertex 3-ranking of T . Define $C' = \{S_j : c(v[S_j]) = 2, j = 1, \dots, m\}$.

Note that $\chi_r(T) = 3$ implies that the color 3 is unique under c , so only r_0 gets this color. It is not possible that T contains a subpath P with three consecutive vertices $v[S_j], v[a_i], v[S_l]$, because then $v[a_i]$ would require color 2 while $c(v[S_j]) = c(v[S_l]) = 1$ and none of these vertices is connected to r_0 – a contradiction. So, the vertices $v[a_i], i = 1, \dots, n$ are leaves in T and consequently $c(v[a_i]) = 1, i = 1, \dots, n$. Since for each $i = 1, \dots, n$ there exists $j \in \{1, \dots, m\}$ such that $\{v[a_i], v[S_j]\} \in E(T)$ it must be the case $c(v[S_j]) = 2$. So, $\bigcup C' = S$.

If $c(v[S_j]) = 2$ then there exists $l \in \{1, \dots, k\}$ such that $\{v[S_j], w_l\}, \{w_l, r_0\} \in E(T)$, because $v[S_j]$ is not adjacent to r_0 , and a path connecting $v[S_j]$ to r_0 in T may contain at most one vertex, because this path cannot use colors other than 1. So, for such a vertex w_l we have $c(w_l) = 1$, which implies that for each w_l there exists at most one $v[S_j]$ such that $\{w_l, v[S_j]\} \in E(T)$. This means that $|C'| \leq k$. \square

Let G be a simple graph. Given a cycle $C \subseteq G$, an edge between two nonadjacent vertices in C is called a *chord*. We say that G is *chordal* if each cycle of length at least 4 has a chord.

Lemma 3 *The graph G is chordal.*

Proof: Let $C_p \subseteq G$ be a cycle in G . If $v[a_i] \in V(C_p)$ then $\{v[a_i], v[S_j]\}, \{v[a_i], v[S_l]\} \in E(C_p), j \neq l, j, l \in \{1, \dots, m\}$, and by the definition of G , $\{v[S_j], v[S_l]\}$ is a chord of C_p . So, if C_p is chordless then $v[a_i] \notin V(C_p)$ for $i = 1, \dots, n$. Similarly one can show that $r_0 \notin V(C_p)$. So, $V(C_p) \subseteq V[C] \cup \{w_1, \dots, w_k\}$ and C_p cannot be chordless, because the vertices of C_p are pairwise adjacent. This completes the proof. \square

Clearly, the size of G is polynomial in $n + m$. Moreover, by Lemmas 1 and 2 there exists a solution to the MSC problem if and only if there exists a spanning tree T of G with the property $\chi_r(T) \leq 3$. By Lemma 3 G is chordal. It is also easy to verify that the diameter of the graph G in our reduction is bounded by 6. So, we have proved the following.

Theorem 1 *The 3-MVRST problem is NP-complete for chordal graphs with diameter at most 6.* \square

3 Proper interval graphs

An *interval graph* is such a graph I that for each vertex $v \in V(I)$ there exists an interval $I_v = (l_v, r_v)$, $l_v < r_v$ such that for any two vertices u, v of I , $\{u, v\} \in E(I)$ if and only if $I_u \cap I_v \neq \emptyset$. An interval graph is *proper* if the intervals I_v for $v \in V(I)$ can be defined in such a way that there are no two vertices u, v such that $I_u \subseteq I_v$. In the following we assume that an *interval diagram*, i.e. a mapping of the intervals to the vertices, is given. This is not a strong assumption since an interval diagram can be computed for a given interval graph in linear time [2]. We also assume that I is connected, because the problem for disconnected graphs reduces to solving MVRST for each connected component separately.

We start by proving a property which is true for any graph.

Lemma 4 *Let u, v be some vertices of a graph G . If P is a shortest path between u and v in G then $\chi_r(P) \leq \chi_r(T)$, where T is any spanning tree of G .*

Proof: Denote by P' the path connecting u and v in T . Clearly, P' connects u and v in G which means that $|V(P)| \leq |V(P')|$. The vertex ranking number of a graph is not bigger than the vertex ranking number of its supergraph which means that $\chi_r(P) \leq \chi_r(P') \leq \chi_r(T)$. \square

Given an interval graph I , define $v_{left} \in V(I)$ ($v_{right} \in V(I)$) so that for each $v \in V(I)$ it holds $l_{v_{left}} \leq l_v$ ($r_v \leq r_{v_{right}}$, respectively). We have the following

Lemma 5 *If T is a minimum vertex ranking spanning tree of I and P is a shortest path connecting v_{left} and v_{right} in I then $\chi_r(P) = \chi_r(T)$.*

Proof: Lemma 4 implies that $\chi_r(P) \leq \chi_r(T)$. In order to prove the reverse inequality we construct a spanning tree T' such that $\chi_r(T') = \chi_r(P)$. Initially let $T' = P$. Let v be any vertex in $V(I) \setminus V(P)$. Since I is a proper interval graph, v is adjacent in I to at least two vertices x, y of P . This is true, because if v has only one neighbor x in P then it means that either $I_v \subseteq I_x$ or $l_v < l_x, x = v_{left}$ or $r_v > r_x, x = v_{right}$. All those situations lead to a contradiction (with the definition of proper interval graphs or with the definition of v_{left} or v_{right}). Let $l_x < l_y$ and we may without loss of generality assume that x and y are adjacent, because if this is not the case then there exists a vertex $z \in V(P)$ such that $l_x < l_z < l_y$ and $\{z, y\} \in E(P)$, which means that $r_z > l_y$. Since v is adjacent to both x and y , we have $(r_x, l_y) \subseteq I_v$ and the fact $(r_x, l_y) \cap I_z \neq \emptyset$ implies that z is adjacent to v in I and we can use z as x . Let c be an optimal vertex ranking of P . We have $c(x) \neq c(y)$. Assume that $c(x) > c(y)$ (the case when $c(x) < c(y)$ is analogous). Add the vertex v and the edge $\{v, x\}$ to T' and let $c(v) = 1$. Clearly, $c(x) > 1$ which means that c is a valid vertex ranking of the new tree T' . This completes the proof, because v was selected arbitrary. \square

The vertex ranking algorithm for proper interval graphs is as follows:

Step 1: find the vertices v_{left} and v_{right} ;

Step 2: compute a shortest path P connecting v_{left} and v_{right} and let $T := P$;

Step 3: find an optimal vertex ranking c of P ;

Step 4: for each $v \in V(I) \setminus V(P)$ find a vertex $x \in V(P)$ such that $c(x) > 1$, $\{x, v\} \in E(I)$ and execute $V(T) := V(T) \cup \{v\}$, $E(T) := E(T) \cup \{\{v, x\}\}$;

Step 5: return T ;

Now we discuss the time complexity of the second step of the algorithm. Without loss of generality we can assume that for each $v \in V(I)$, $l_v, r_v \in \{1, \dots, 2n\}$, where $n = |V(I)|$, see e.g. [4]. This means that we can sort the vertices of I by the values of l_v in $O(n)$ time. Assume that $V(I) = \{v_1, \dots, v_n\}$, where $l_{v_i} < l_{v_{i+1}}$, $i = 1, \dots, n - 1$. Authors in [3] gave an efficient algorithm for computing shortest paths in interval graphs. Assuming that the vertices of an interval graph are sorted according to the values of l_v and r_v , they designed a data structure which in $\Theta(1)$ time gives the lengths of the path between two given vertices and in $O(l)$ time computes this path, where l is the length of the path. The above data structure can be computed in linear time $O(|V(I)|)$.

Corollary 1 ([3]) *The shortest path between the vertices v_{left} and v_{right} can be computed in linear time.*

Theorem 2 *There exists a linear time algorithm solving the MVRST problem for proper interval graphs.*

Proof: By Lemma 5, the algorithm is optimal. We show that it has a linear running time. Step 1 of our algorithm can be performed in constant time, because $v_{left} = v_1$ and $v_{right} = v_n$ (the vertices are sorted as mentioned above). Using Corollary 1 we have that the second step can be done in linear time. Step 3 can be done in $O(|V(P)|) = O(n)$ time [16]. If v_i and v_{i+j} are two vertices in $V(I) \setminus V(P)$ and $\{v_i, v_l\} \in E(I)$, $c(v_l) > 1$ where $v_l \in V(P)$ then there exists a vertex $v_{l+k} \in V(P)$, $k \geq 0$ such that $c(v_{l+k}) > 1$ and $\{v_{i+j}, v_{l+k}\} \in E(I)$. Thus, to perform the last step of the algorithm in linear time we iterate over the vertices in $V(I) \setminus V(P)$ according to increasing values of their indices and we iterate over the vertices in $V(P)$, also according to increasing values of the indices. From the above argument it follows that we do not need to backtrack, so the fourth step of the algorithm can be done in linear time. \square

References

- [1] A.S. Arefin, M.A. Kashem, NP-Completeness of the minimum edge-ranking spanning tree problem on series-parallel graphs, *Proc. 10th International Conference on Computer and Information Technology*, 2008 (ICIT 2007) 1-4.
- [2] K.S. Booth, G.S. Lueker, Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms, *J. Comput. System Sci.* **13** (1976) 335-379.

- [3] D.Z. Chen, D. Lee, R. Sridhar, C.N. Sekharan, Solving the all-pair shortest path query problem on interval and circular-arc graphs, *Networks* **31** (1998) 249-257.
- [4] J.S. Deogun, T. Kloks, D. Kratsch, H. Müller, On the vertex ranking problem for trapezoid, circular-arc and other graphs, *Discrete Appl. Math.* **98** (1999) 39-63.
- [5] D. Dereniowski, Edge ranking and searching in partial orders, *Discrete Appl. Math.* **156** (2008) 2493-2500.
- [6] D. Dereniowski, M. Kubale, Efficient parallel query processing by graph ranking, *Fundamenta Informaticae* **69** (2006) 273-285.
- [7] D. Dereniowski, A. Nadolski, Vertex rankings of chordal graphs and weighted trees, *Inform. Process. Letters* **98** (2006) 96-100.
- [8] S.-Y. Hsieh, On vertex ranking of a starlike graph, *Inform. Process. Letters* **82** (2002) 131-135.
- [9] M.A. Kashem, M.A. Haque, M.R. Uddin, S.A. Sharmin, An algorithm for finding minimum edge-ranking spanning tree of series-parallel graphs, *Proc. of the 9th International Conference on Computer and Information Technology (ICCIT 2006)* 108-112.
- [10] K. Makino, Y. Uno, T. Ibaraki, Minimum edge ranking spanning trees of threshold graphs, *Lecture Notes in Comp. Sci.* **2518** (2002) 59-77.
- [11] K. Makino, Y. Uno, T. Ibaraki, On minimum edge ranking spanning trees, *Journal of Algorithms* **38** (2001) 411-437.
- [12] K. Miyata, S. Masuyama, S. Nakayama, L. Zhao, Np-hardness proof and an approximation algorithm for the minimum vertex ranking spanning tree problem, *Discrete Appl. Math.*, **154** (2006) 2402-2410.
- [13] S. Nakayama, S. Masuyama, An algorithm for solving the minimum vertex ranking spanning tree problem on interval graphs, *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* **E86-A** (2003) 1019-1026.
- [14] S. Nakayama, S. Masuyama, A polynomial time algorithm for obtaining a minimum vertex ranking spanning tree in outerplanar graphs, *IEICE Transactions on Information and Systems* **E89-D** (2006) 2357-2363.
- [15] A. Pothen, The complexity of optimal elimination trees, *Tech. Report CS-88-13*, The Pennsylvania State University, 1988.
- [16] A.A. Schäffer, Optimal node ranking of trees in linear time, *Infor. Process. Lett.* **33** (1989) 91-96.