Rankings of graphs

H.L. Bodlaender,[*] J.S. Deogun,[†] K. Jansen,[‡] T. Kloks,[§]
D. Kratsch,[¶] H. Müller[‖] and Zs. Tuza[**]

Abstract

A vertex (edge) coloring $c : V \to \{1, 2, \ldots, t\}$ ($c' : E \to \{1, 2, \ldots, t\}$) of a graph $G = (V, E)$ is a vertex (edge) *t-ranking* if for any two vertices (edges) of the same color every path between them contains a vertex (edge) of larger color. The *vertex ranking number* $\chi_r(G)$ (*edge ranking number* $\chi'_r(G)$) is the smallest value of t such that G has a vertex (edge) t-ranking. In this paper we study the algorithmic complexity of the VERTEX RANKING and EDGE RANKING problems. Among others it is shown that $\chi_r(G)$ can be computed in polynomial time when restricted to graphs with treewidth at most k for any fixed k. We characterize those graphs where the vertex ranking number χ_r and the chromatic number χ coincide on all induced subgraphs, show that $\chi_r(G) = \chi(G)$ implies $\chi(G) = \omega(G)$ (largest clique size) and give a formula for $\chi'_r(K_n)$.

1 Introduction

In this paper we consider vertex rankings and edge rankings of graphs. The vertex ranking problem, also called the *ordered coloring problem* [15], has received much attention lately because of the growing number of applications. There are applications in scheduling problems of assembly steps in manufacturing systems [19], e.g., edge ranking of trees can be used to model the parallel assembly of a product from its components in a quite natural manner [6, 12, 13, 14].

[*]Department of Computer Science, Utrecht University, P.O. Box 80.089, 3508 TB Utrecht, the Netherlands. Email: hansb@cs.ruu.nl. This author was partially supported by the ESPRIT Basic Research Actions of the EC under contract 7141 (project ALCOM II)

[†]Department of Computer Science and Engineering, University of Nebraska – Lincoln, Lincoln, NE 68588-0115, U.S.A. This author was partially supported by the Office of Naval Research under Grant No. N0014-91-J-1693

[‡]Institut für Informatik, TU München, 80290 München, Germany. On leave from Universität Trier, Germany

[§]Department of Mathematics and Computing Science, Eindhoven University of Technology, P.O.Box 513, 5600 MB Eindhoven, the Netherlands

[¶]IRISA, Campus Universitaire de Beaulieu, 35042 Rennes Cedex, France. On leave from Friedrich-Schiller-Universität, Jena. This author was partially supported by Deutsche Forschungsgemeinschaft under Kr 1371/1-1

[‖]Fakultät für Mathematik und Informatik, Friedrich-Schiller-Universität, Universitätshochhaus, 07740 Jena, Germany

[**]Computer and Automation Institute, Hungarian Academy of Sciences, H-1111 Budapest, Kende u. 13–17, Hungary. This author was partially supported by the "OTKA" Research Fund of the Hungarian Academy of Sciences, Grant No. 2569

Furthermore the problem of finding an optimal vertex ranking is equivalent to the problem of finding a minimum-height elimination tree of a graph [6, 7]. This measure is of importance for the parallel Cholesky factorization of matrices [3, 9, 18]. Yet other applications lie in the field of VLSI-layout [17, 26].

The VERTEX RANKING problem 'Given a graph G and a positive integer t, decide whether $\chi_r(G) \leq t$' is NP-complete even when restricted to cobipartite graphs since Pothen has shown that the equivalent minimum elimination tree height problem remains NP-complete on cobipartite graphs [20]. A short proof of the NP-completeness of VERTEX RANKING is given in Section 3. Much work has been done in finding optimal rankings of trees. For trees there is a linear-time algorithm finding an optimal vertex ranking [24]. For the closely related edge ranking problem on trees a $O(n^3)$ algorithm was given in [8]. Recently, Zhou and Nishizeki obtained an $O(n \log n)$ algorithm for optimally edge ranking trees [28] (see also [29]). Efficient vertex ranking algorithms for permutation, trapezoid, interval, circular-arc, circular permutation graphs, and cocomparability graphs of bounded dimension are presented in [7]. Moreover, the vertex ranking problem is trivial on split graphs and it is solvable in linear time on cographs [25].

In [15], typical graph theoretical questions, as they are known from the coloring theory of graphs, are investigated. This also leads to a $O(\sqrt{n})$ bound for the vertex ranking number of a planar graph and the authors describe a polynomial-time algorithm which finds a vertex ranking of a planar graph using only $O(\sqrt{n})$ colors. For graphs in general there is an approximation algorithm of performance ratio $O(\log^2 n)$ for the vertex ranking number [3, 16]. In [3] it is also shown that one plus the pathwidth of a graph is a lower bound for the vertex ranking number of the graph (hence a planar graph has pathwidth $O(\sqrt{n})$, which is also shown in [16] using different methods).

Our goal is to extend the known results in both the algorithmic and graph theoretic directions. The paper is organized as follows. In Section 2 the necessary notions and preliminary results are given. We study the algorithmic complexity of determining whether a graph G fulfills $\chi_r(G) \leq t$ and $\chi'_r(G) \leq t$, respectively, in Sections 3, 4, and 5. In Section 6 we characterize those graphs for which the vertex ranking number and the chromatic number coincide on every induced subgraph. Those graphs turn out to be precisely those containing no path and cycle on four vertices as an induced subgraph; hence, we obtain a characterization of the *trivially perfect graphs* [11] in terms of rankings. Moreover we show that $\chi(G) = \chi_r(G)$ implies that the chromatic number of G is equal to its largest clique size. In Section 7 we give a recurrence relation allowing us to compute the edge ranking number of a complete graph.

2 Preliminaries

We consider only finite, undirected and simple graphs $G = (V, E)$. Throughout the paper n denotes the cardinality of the vertex set V and m denotes that of the edge set E of the graph $G = (V, E)$. For graph-theoretic concepts, definitions and properties of graph classes not given here we refer to [4, 5, 11].

Let $G = (V, E)$ be a graph. A subset $U \subseteq V$ is *independent* if each pair of vertices $u, v \in U$ is nonadjacent. A graph $G = (V, E)$ is *bipartite* if there is a partition of V into two independent sets A and B. The *complement* of the graph $G = (V, E)$ is the graph $\overline{G}$ having vertex set V and edge set $\{\{v, w\} \mid v \neq w,\ \{v, w\} \notin E\}$. For $W \subseteq V$ we denote by $G[W]$ the subgraph of $G = (V, E)$ induced by the vertices of W, and for $X \subseteq E$ we write $G[X]$ for the graph (V, X) with vertex set V and edge set X.

Definition 1 *Let $G = (V, E)$ be a graph and let t be a positive integer. A* (vertex) t-ranking*, called* ranking *for short if there is no ambiguity, is a coloring $c : V \to \{1, \ldots, t\}$ such that for every pair of vertices x and y with $c(x) = c(y)$ and for every path between x and y there is a vertex z on this path with $c(z) > c(x)$. The* vertex ranking number *of G, $\chi_r(G)$, is the smallest value t for which the graph G admits a t-ranking.*

By definition adjacent vertices have different colors in any t-ranking, thus any t-ranking is a proper t-coloring. Hence $\chi_r(G)$ is bounded below by the *chromatic number* $\chi(G)$. A vertex $\chi_r(G)$-ranking of G is said to be an *optimal (vertex) ranking* of G.

The edge ranking problem is closely related to the vertex ranking problem.

Definition 2 *Let $G = (V, E)$ be a graph and let t be a positive integer. An* edge t-ranking *is an edge coloring $c' : E \to \{1, \ldots, t\}$ such that for every pair of edges e and f with $c'(e) = c'(f)$ and for every path between e and f there is an edge g on this path with $c'(g) > c'(e)$. The* edge ranking number $\chi'_r(G)$ *is the smallest value of t such that G has an edge t-ranking.*

Remark 3 *There is a one-to-one correspondence between the edge t-rankings of a graph G and the vertex t-rankings of its line graph $L(G)$. Hence $\chi'_r(G) = \chi_r(L(G))$.*

An edge t-ranking of a graph G is a particular proper edge coloring of G. Hence $\chi'_r(G)$ is bounded below by the *chromatic index* $\chi'(G)$. An edge $\chi'_r(G)$-ranking of G is said to be an *optimal edge ranking* of G.

As shown in [7], the vertex ranking number of a connected graph is equal to its minimum elimination tree height plus one. Thus (vertex) separators and edge separators are a convenient tool for investigating rankings of graphs. A subset $S \subseteq V$ of a graph $G = (V, E)$ is said to be a *separator* if $G[V \setminus S]$ is disconnected. A subset $R \subseteq E$ of a graph $G = (V, E)$ is said to be an *edge separator* (or *edge cut*) if $G[E \setminus R]$ is disconnected.

In this paper we use the *separator tree* for studying vertex rankings. This concept is closely related to elimination trees (cf.[3, 7, 18]).

Definition 4 *Given a vertex t-ranking $c : V \to \{1, 2, \ldots, t\}$ of a connected graph $G = (V, E)$, we assign a rooted tree $T(c)$ to it by an inductive construction, such that a separator of a certain induced subgraph of G is assigned to each internal node of $T(c)$ and the vertices of each set assigned to a leaf of $T(c)$ have pairwise different colors:*

1. *If no color occurs more than once in G, then $T(c)$ consists of a single vertex r (called root), assigned to the vertex set of G.*

2. *Otherwise, let i be the largest color assigned to more than one vertex by c. Then $\{i+1, i+2, \ldots, t\}$ has to be a separator S of G. We create a root r of $T(c)$ and assign S to r. (The induced subgraph of G corresponding to the subtree of T rooted at r will be G itself.) Assuming that a separator tree $T_i(c)$ with root r_i has already been defined for each connected component G_i of the graph $G[V \setminus S]$, the children of r in $T(c)$ will be the vertices r_i and the subtree of $T(c)$ rooted at r_i will be $T_i(c)$.*

The rooted tree $T(c)$ is said to be a separator tree *of G.*

Notice that all vertices of G assigned to nodes of $T(c)$ on a path from a leaf to the root have different colors.

3 Unbounded ranking

It is still unknown whether the EDGE RANKING problem 'Given a graph G and a positive integer t, decide whether $\chi'_r(G) \leq t$' is NP-complete. Clearly, by Remark 3 this problem is equivalent to the VERTEX RANKING problem 'Given a graph G and a positive integer t, decide whether $\chi_r(G) \leq t$' when restricted to line graphs.

On the other hand, it is a consequence of the NP-completeness of the minimum elimination tree height problem shown by Pothen in [20] and the equivalence of this problem with the VERTEX RANKING problem [6, 7] that the latter is NP-complete even when restricted to graphs that are the complement of bipartite graphs, the so-called cobipartite graphs.

For reasons of self-containedness, we start with a short proof of the NP-completeness of VERTEX RANKING, when restricted to cobipartite graphs. The following problem, called BALANCED COMPLETE BIPARTITE SUBGRAPH (abbreviated BCBS) is NP-complete. This is problem [GT24] of [10].

> INSTANCE: A bipartite graph $G = (V, E)$ and a positive integer k.
>
> QUESTION: Are there two disjoint subsets $W_1, W_2 \subseteq V$ such that $|W_1| = |W_2| = k$ and such that $u \in W_1$, $v \in W_2$ implies that $\{u, v\} \in E$?

Theorem 5 VERTEX RANKING *is NP-complete for a cobipartite graphs.*

Proof: Clearly the problem is in NP. NP-hardness is shown by reduction from BCBS.

Let a bipartite graph $G = (V_1, V_2, E)$ and a positive integer k be given. Let $\overline{G}$ be the complement of G, thus $\overline{G}$ is a cobipartite graph.

We claim that G has a balanced complete bipartite subgraph with $2 \cdot k$ vertices if and only if $\overline{G}$ has a $(n \Leftrightarrow k)$-ranking.

Suppose we have sets $W_1 \subseteq V_1$, $W_2 \subseteq V_2$, such that $|W_1| = |W_2| = k$ and such that for all $u \in W_1$, $v \in W_2$: $\{u, v\} \in E$. We now construct a k-ranking of $\overline{G}$. Write $W_i = \{v_1^{(i)}, \ldots, v_k^{(i)}\}$ for $i \in \{1, 2\}$, write $V \setminus (W_1 \cup W_2) = \{v'_1, \ldots, v'_{n-2\cdot k}\}$. We define a vertex ranking c of G as follows:

$$\begin{aligned} c(v_j^{(1)}) = c(v_j^{(2)}) &= j \quad \text{for all } j,\ 1 \leq j \leq k. \\ c(v'_j) &= k + j \quad \text{for all } j,\ 1 \leq j \leq n \Leftrightarrow 2 \cdot k. \end{aligned}$$

One easily observes that c is a vertex $(n \Leftrightarrow k)$-ranking.

Next, let c be a $(n \Leftrightarrow k)$-ranking for $\overline{G}$. Since $\overline{G}$ is a cobipartite graph, for each color, there can be at most two vertices with that color, one lying in V_1 and the other in V_2. Therefore, we have k pairs $v_j^{(1)}$ and $v_j^{(2)}$ with $c(v_j^{(1)}) = c(v_j^{(2)})$ and we can assume that $W_1 = \{v_j^{(1)} | 1 \leq j \leq k\} \subseteq V_1$ and $W_2 = \{v_j^{(2)} | 1 \leq j \leq k\} \subseteq V_2$.

Now we show that the subgraph induced by the set $W_1 \cup W_2$ forms a balanced complete bipartite subgraph in G. To show this, we prove that each pair of vertices $u \in W_1$, $v \in W_2$ is not adjacent in $\overline{G}$. Suppose $v_i^{(1)}$ and $v_j^{(2)}$ are adjacent in $\overline{G}$. Then, the colors of these vertices must be different. Furthermore, assume w.l.o.g., that $c(v_i^{(1)}) < c(v_j^{(2)})$. Then we have a path $(v_j^{(1)}, v_i^{(1)}, v_j^{(2)})$ with $c(v_i^{(1)}) < c(v_j^{(1)}) = c(v_j^{(2)})$ contradicting the fact that c is a ranking. Hence the subgraph induced by $W_1 \cup W_2$ is indeed a balanced complete bipartite subgraph. This proves the claim, and the NP-completeness of VERTEX RANKING. □

We show that the analogous result holds for bipartite graphs as well.

Theorem 6 VERTEX RANKING *remains NP-complete for bipartite graphs.*

Proof: The transformation is from VERTEX RANKING for arbitrary graphs without isolated vertices. Given the graph G, we construct a graph $G' = (V', E')$. We take

$$V' = V \cup \{(e, i) \mid e \in E, 1 \leq i \leq t + 1\}$$

and

$$E' = \{\{v, (e, i)\} \mid v \in V, e \in E, 1 \leq i \leq t + 1 \text{ where } v \in e\}.$$

Clearly, the constructed graph G' is a bipartite graph. Now we show that G has a t-ranking if and only if G' has a $(t + 1)$-ranking.

Suppose G has a t-ranking $c : V \to \{1, \ldots, t\}$. We construct a coloring $\hat{c}$ for G' in the following way. For the vertices $v \in V$ we set $\hat{c}(v) = c(v) + 1$ and for the vertices $(e, i) \in V' \setminus V$ we set $\hat{c}((e, i)) = 1$. Clearly $\hat{c}$ is a $(t + 1)$-ranking of G'.

On the other hand, let $\hat{c} : V' \to \{1, \ldots, t+1\}$ be a $(t+1)$-ranking of G'. We show that $\hat{c}(v) > 1$ for every vertex $v \in V$. Suppose not and let v be a vertex of V with $\hat{c}(v) = 1$. Let $e = \{v, w\}$ be an edge incident to v in G. Hence v is adjacent to $(e, 1), (e, 2), \ldots, (e, t + 1)$ in G'. Then $\hat{c}(v) = 1$ implies $\hat{c}((e, i)) > 1$ for $i = 1, 2, \ldots, t + 1$. Since $\hat{c}$ is a $(t + 1)$-ranking, there are l, l' with $l \neq l'$ such that $\hat{c}((e,l)) = \hat{c}((e,l'))$, implying a path $(e,l) \Leftrightarrow v \Leftrightarrow (e,l')$ which contradicts the assumption that $\hat{c}$ is a ranking. This proves that $\hat{c}(v) > 1$ holds for every vertex $v \in V$. As a consequence, for each edge $e = \{u,v\} \in E$, there is a vertex $(e,i) \in V'$ with $\hat{c}((e,i)) < \min(\hat{c}(u), \hat{c}(v))$. Thus, changing $\hat{c}$ on $V' \setminus V$ to $\hat{c}((e,i)) = 1$ for all $(e,i) \in V'$, we obtain another $(t+1)$-ranking of G'. Now we define $c(v) = \hat{c}(v) \Leftrightarrow 1$ for every $v \in V$. The coloring c is a t-ranking of G since the existence of a path between two vertices v and w of G such that $c(v) = c(w)$ and all inner vertices have smaller colors implies the existence of a path from v to w in G' with $\hat{c}(v) = \hat{c}(w)$ and all inner vertices having smaller colors, contradicting the fact that $\hat{c}$ is a $(t+1)$-ranking of G'. □

4 Bounded ranking

We show that the 'bounded' ranking problems 'Given a graph G, decide whether $\chi_r(G) \leq t$ ($\chi'_r(G) \leq t$)' are solvable in linear time for any fixed t. This will be done by verifying that the corresponding graph classes are closed under certain operations.

Definition 7 *An* edge contraction *is an operation on a graph G replacing two adjacent vertices u and v of G by a vertex adjacent to all vertices that were adjacent to u or v. An* edge lift *is an operation on a graph G replacing two adjacent edges $\{v,w\}$ and $\{u,w\}$ of G by one edge $\{u,v\}$.*

Definition 8 *A graph H is a* minor *of the graph G if H can be obtained from G by a series of the following operations: vertex deletion, edge deletion, and edge contraction. A graph class $\mathcal{G}$ is* minor closed *if every minor H of every graph $G \in \mathcal{G}$ also belongs to $\mathcal{G}$.*

Lemma 9 *The class of graphs satisfying $\chi_r(G) \leq t$ is minor closed for any fixed t.*

Proof: Since vertex/edge deletion cannot create new paths between monochromatic pairs of vertices, we only have to show that edge contraction does not increase the ranking number. Let $G = (V,E)$ be a graph with $\chi_r(G) \leq t$, and assume $H = (V',E')$ is obtained from G by contracting the edge $\{u,v\} \in E$ into a new vertex $\widehat{uv}$. Suppose c is a t-ranking of G. We construct a coloring $\hat{c} : V' \to \{1,2,\ldots,t\}$ of H as follows.

$$\hat{c}(x) = \begin{cases} c(x) & \text{if } x \in V \setminus \{u,v\} \\ \max(c(u), c(v)) & \text{if } x = \widehat{uv} \end{cases}$$

Suppose $\hat{c}$ is not a t-ranking of H. Then there is a path $P : x_0 \Leftrightarrow x_1 \Leftrightarrow \cdots \Leftrightarrow x_s$, $s \geq 1$, of H such that $\hat{c}(x_0) = \hat{c}(x_s) > \hat{c}(x_i)$ for every $i \in \{1,2,\ldots,s \Leftrightarrow 1\}$. Since c is a t-ranking of G the vertex $\widehat{uv}$ must occur in the path. Depending on its neighbors in P we can 'decontract' $\widehat{uv}$ in the path P into u, v, $u \Leftrightarrow v$ or $v \Leftrightarrow u$ getting a path P' of G violating the ranking condition, in contradiction to the choice of c. □

Corollary 10 *For each fixed t, the class of graphs satisfying $\chi_r(G) \leq t$ can be recognized in linear time.*

Proof: In [1], using results from Robertson and Seymour [22, 23], it is shown that every minor closed class of graphs that does not contain all planar graphs, has a linear time recognition algorithm. The result now follows directly from Lemma 9. □

As regards edge rankings, a simple argument yields a much stronger assertion as follows.

Theorem 11 *For each fixed t, the class of connected graphs satisfying $\chi'_r(G) \leq t$ can be recognized in constant time.*

Proof: For any fixed t, there are only a finite number of connected graphs G with $\chi'_r(G) \leq t$, as necessary conditions are that the maximum degree of G is at most t, and the diameter of G is bounded by $2^t \Leftrightarrow 1$. □

Certainly, the above theorem immediately implies that the graphs G satisfying $\chi'_r(G) \leq t$ can be recognized in linear time, by inspecting the connected components separately. This result might have also been obtained via more involved methods, by using results of Robertson and Seymour on graph immersions [21]. Similarly, one can show that for fixed t and d, the class of connected graphs with $\chi_r(G) \leq t$ and maximum vertex degree d can be recognized in constant time.

Definition 12 *A graph H is an* immersion *of the graph G if H can be obtained from G by a series of the following operations: vertex deletion, edge deletion and edge lift. A graph class $\mathcal{G}$ is* immersion closed *if every immersion H of a graph $G \in \mathcal{G}$ also belongs to $\mathcal{G}$.*

The proof of the following lemma is similar to the one of Lemma 9 and therefore omitted.

Lemma 13 *The class of graphs satisfying $\chi'_r(G) \leq t$ is immersion closed for any fixed t.*

Linear-time recognizability of the class of graphs satisfying $\chi'_r(G) \leq t$ now also follows from Lemma 13, the results of Robertson and Seymour, and the fact that graphs with $\chi'_r(G) \leq t$ have treewidth at most $2t + 2$.

5 Computing the vertex ranking number on graphs with bounded treewidth

In this section, we show that one can compute $\chi_r(G)$ of a graph G with treewidth at most k in polynomial time, for any fixed k. Such a graph is also called a partial k-tree. This result implies polynomial time computability of the vertex ranking number for any class of graphs with a uniform upper bound on the treewidth, e.g., outerplanar graphs, series-parallel graphs, Halin graphs.

The notion of treewidth has been introduced by Robertson and Seymour (see e.g., [22]).

Definition 14 *A* tree-decomposition *of a graph* $G = (V, E)$ *is a pair* $(\{X_i \mid i \in I\}, T = (I, F))$ *with* $X = \{X_i \mid i \in I\}$ *a collection of subsets of* V*, and* $T = (I, F)$ *a tree, such that*

- $\bigcup_{i \in I} X_i = V$
- *for all edges* $\{v, w\} \in E$ *there is an* $i \in I$ *with* $v, w \in X_i$
- *for all* $i, j, k \in I$*: if* j *is on the path from* i *to* k *in* T*, then* $X_i \cap X_k \subseteq X_j$*.*

The width of a tree-decomposition $(\{X_i \mid i \in I\}, T = (I, F))$ *is* $\max_{i \in I} |X_i| \Leftrightarrow 1$*. The treewidth of a graph* $G = (V, E)$ *is the minimum width over all tree-decompositions of* G*.*

We often abbreviate $(\{X_i \mid i \in I\}, T = (I, F))$ as (X, T). When the treewidth of $G = (V, E)$ is bounded by a constant k, one can find in $O(n)$ time a tree-decomposition (X, T) of width at most k, such that $I = O(n)$ and T is a rooted binary tree [1]. Denote the root of T as r. We say (X, T) is a rooted binary tree-decomposition.

Definition 15 *A* terminal graph *is a triple* (V, E, Z)*, with* (V, E) *an undirected graph, and* $Z \subseteq V$ *a subset of the vertices, called the* terminals.

To each node i of a rooted binary tree-decomposition (X, T) of graph $G = (V, E)$, we associate the terminal graph $G_i = (V_i, E_i, X_i)$, where $V_i = \bigcup\{X_j \mid j = i \text{ or } j \text{ is a descendant of } i\}$, and $E_i = \{\{v, w\} \in E \mid v, w \in V_i\}$. As shorthand notation we write $p(v, w, G, c, \alpha)$, iff there is a path in G from v to w with all internal vertices having colors, smaller than α under coloring c. If $p(v, w, G, c, \alpha)$, we denote with $P(v, w, G, c, \alpha)$ the set of paths in G from v to w with all internal vertices having colors (using color function c), smaller than α. In the following, suppose t is given.

Definition 16 *Let* $G = (V, E, Z)$ *be a terminal graph, and let* $c : V \rightarrow \{1, \ldots, t\}$ *be a vertex* t*-ranking of* (V, E)*. The* characteristic *of* c*,* $Y(c)$*, is the quadruple* $(c|_Z, f_1, f_2, f_3)$*, where*

- $c|_Z$ *is the function* c*, restricted to domain* Z*.*
- $f_1 : Z \times \{1, \ldots, t\} \rightarrow \{true, false\}$*, is defined by:* $f_1(v, i) = true$ *if and only if* $c(v) = i$ *or there is a vertex* $x \in V$ *with* $c(x) = i$ *and* $p(v, x, G, c, i)$*.*
- $f_2 : Z \times Z \times \{1, \ldots, t\} \rightarrow \{true, false\}$*, is defined by:* $f_2(v, w, i) = true$*, if and only if there is a vertex* $x \in V$ *with* $c(x) = i$ *and* $p(v, x, G, c, i)$ *and* $p(w, x, G, c, i)$*.*
- $f_3 : Z \times Z \rightarrow \{1, \ldots, t, \infty\}$ *is defined by:* $f_3(v, w)$ *is the smallest integer* t' *such that* $p(v, w, G, c, t')$*. If there is no path from* v *to* w *in* G*, then* $f_3(v, w) = \infty$*.*

Definition 17 *A set of characteristics S of vertex t-rankings of a terminal graph G is* a full set of characteristics of vertex t-rankings for G *(in short: a full set for G), if and only if for every vertex t-ranking c of G, $Y(c) \in S$.*

A set C of vertex t-rankings of a terminal graph G is an example set of vertex t-rankings for G *(in short: an example set for G), if and only if for every vertex t-ranking c of G, there is an $c' \in C$ with $Y(c) = Y(c')$, or, equivalently, the set of characteristics of the elements of C forms a full set of characteristics of vertex t-rankings for G.*

If $t = O(\log n)$, then a full set of characteristics of vertex t-rankings of $G = (V, E, Z)$ (with $|Z| \leq k+1$, k constant) has size polynomial in V: there are $O(log^{k+1}n)$ possible values for $c|_Z$, $2^{O((k+1)\log n)}$ possible values for f_1, $2^{O((k+1)^2 \log n)}$ possible values for f_2, and there are $O(\log^{\frac{1}{2}k(k+1)} n)$ possible values for f_3. The following lemma, given in [3], shows that we can ensure this property for graphs with treewidth at most k for fixed k.

Lemma 18 *If the treewidth of $G = (V, E)$ is at most k, then $\chi_r(G) = O(k \cdot \log n)$.*

Let (X, T) be a rooted binary tree-decomposition of G. Suppose $j \in I$ is a descendant of $i \in I$ in T. Suppose c is a vertex t-ranking of G_i. The restriction of c to G_j is the function $c|_{G_j} : V_j \to \{1, \ldots, t\}$, defined by $\forall v \in V_j : c_j(v) = c(v)$. Clearly, $c|_{G_j}$ is a vertex t-ranking of G_j. If c' is another vertex t-ranking of G_j, we define the function $R(c, c') : V_i \to \{1, \ldots, t\}$, by:

$$R(c, c')(v) = \begin{cases} c(v) & \text{if } v \in V_i \setminus V_j \\ c'(v) & \text{if } v \in V_j \end{cases}$$

Lemma 19 *Let (X, T) be a rooted binary tree-decomposition of $G = (V, E)$. Let j be a descendant of i. Let c be a vertex t-ranking of G_i, and c' be a vertex t-ranking of G_j. If $Y(c|_{G_j}) = Y(c')$, then $R(c, c')$ is a vertex t-ranking of G_i, and $Y(c) = Y(R(c, c'))$.*

Proof: For brevity, we write $c'' = R(c, c')$, $W_1 = (V_i \setminus V_j) \cup X_j$, $W_2 = V_j \setminus X_j$, and we write $Y(c|_{G_j}) = Y(c') = (c'|_{X_j}, f_1, f_2, f_3)$.

We start with proving two claims.

Claim 20 *For all $v, w \in W_1$ and all $t' \leq t$: $p(v, w, G_i, c, t') \Leftrightarrow p(v, w, G_i, c'', t')$.*

Proof: Let $v, w \in W_1$, and suppose we have a path $p \in P(v, w, G_i, c, t')$. We consider those parts of the path p that are part of G_j: write $p = (p_0, p'_0, p_1, p'_1, \ldots, p_{r-1}, p'_{r-1}, p_r)$, such that each p_α $(0 \leq \alpha \leq r)$ is a path with all vertices in W_1, and each p'_α $(0 \leq \alpha \leq r \Leftrightarrow 1)$ is a path in G_j. (Each path is a collection of successive edges, i.e., the last vertex of a path is the first vertex of the next path.) Write v_α for the first vertex on path p'_α and w_α for the last vertex on path p'_α $(0 \leq \alpha \leq r \Leftrightarrow 1)$. Note that $p'_\alpha \in P(v_\alpha, w_\alpha, G_j, c, t')$, hence $f_3(v_\alpha, w_\alpha) \leq t'$. We now have that there also exists a path $p''_\alpha \in P(v, w, G_j, c', t')$. (In words: there exists a path from v to w in G_j such that all colors of internal vertices are smaller than t', using coloring c (or, equivalently $c|_{G_j}$). As $c|_{G_j}$ and c' have the same characteristics, there also exists such a path using color function c'.) Now, the path formed by the sequence $(p_0, p''_0, p_1, p''_1, \ldots, p_{r-1}, p''_{r-1}, p_r)$ is a path in G_i between v and w with all colors of internal vertices smaller than t', hence $p(v, w, G_i, c'', t')$. This shows: $p(v, w, G_i, c, t') \Rightarrow p(v, w, G_i, c'', t')$. $p(v, w, G_i, c, t') \Leftarrow p(v, w, G_i, c'', t')$ can be shown in the same way. □

Claim 21 *For all $v \in W_1$, t': there exists a vertex $w \in V_i$ ($w \in V_j$) with $p(v, w, G_i, c, t')$ and $c(w) = t'$, if and only if there exists a vertex $w' \in V_i$ ($w' \in V_j$) with $p(v, w', G_i, c'', t')$, and $c''(w') = t'$.*

Proof: Let $w \in V_i$ with $p(v, w, G_i, c, t')$ and $c(w) = t'$. If $w \in W_1$, then, by claim 20, we have $p(v, w, G_i, c'', t')$. Otherwise, let x be the last vertex on a path $p \in P(v, w, G_i, c, t')$ that belongs to W_1. Write $p = (p', p'')$, where x is the last vertex of p' and the first vertex of p''. $p' \in P(v, x, G_i, c, t')$, hence there exists a path $q' \in P(v, x, G_i, c'', t')$. $p'' \in P(x, w, G_j, c|_{G_j}, t')$, hence $f_1(x, t') =$ true. Using equality of the characteristics of $c|_{G_j}$ and c', we have that there exists a vertex $w' \in V_j$ with $c'(w) = t' = c''(w)$ and a path $q'' \in P(x, w', G_j, c', t')$. Now (q', q'') is a path from v to w' in G_i with all internal vertices of color (under color function c'') smaller that t', hence $p(v, w', G_i, c'', t')$. The reverse implication of the claim can be shown in a similar way. □

We now show that c'' is a vertex t-ranking, or, equivalently, that for all $v, w \in V_i$, if $c''(v) = c''(w)$, then $\neg p(v, w, G_i, c'', c''(v))$. Let $v, w \in V_i$ with $c''(v) = c''(w) = t', v \neq w$ be given. We consider four cases:

1. $v, w \in W_1$. If $p(v, w, G_i, c'', t')$, then by Claim 20, $p(v, w, G_i, c, t')$, and $c(v) = c''(v) = t'$, $c(w) = c''(w) = t'$, hence c is not a vertex ranking, contradiction.

2. $v \in W_1$, $w \in W_2$. If $p(v, w, G_i, c'', t')$, then by Claim 21, there exists a $w' \in V_i$ with $p(v, w', G_i, c, t')$ and $c(w') = c(v)$, hence again c is not a vertex ranking, contradiction.

3. $w \in W_1$, $v \in W_2$. Similar to Case 2.

4. $v, w \in W_2$. Let $p \in P(v, w, G_i, c'', t')$. If all vertices on p belong to W_2, then p is a path in G_j, and hence c' was not a vertex ranking of G_j, contradiction. So, there exist vertices on p that belong to W_1.

   Let x be the first vertex on p that belongs to W_1. Then $p = (p_1, p_2)$, with $p_1 \in P(v, x, G_j, c'', t')$ and $p_2 \in P(x, w, G_i, c'', t')$. By Claim 21, there must exist vertices $v', w' \in V_j$ with $c(v') = c(w') = t'$ and paths q_1, q_2, with $q_1 \in P(v', x, G_j, c, t')$, $q_2 \in P(x, w', G_i, c, t')$. The path $q = (q_1, q_2)$ is a path from v' to w' with all internal vertices of color (with color function c) less than t'. Hence c is not a vertex ranking, contradiction.

It remains to show that $Y(c) = Y(c'')$. Clearly, $c|_{X_i} = c''|_{X_i}$. Suppose $Y(c) = (c|_{X_i}, g_1, g_2, g_3)$, and $Y(c'') = (c|_{X_i}, g'_1, g'_2, g'_3)$. It follows directly from Claim 21 that $g_1 = g'_1$.

Consider $v, w \in X_i$, $t' \in \{1, \ldots, t\}$. Suppose $g_2(v, w, t') =$ true. Let $x \in V_i$ be the vertex with $c(x) = t'$ and $p(v, x, G_i, c, t')$ and $p(w, x, G_i, c, t')$. If $x \in W_1$, then, by Lemma 20, $p(v, x, G_i, c'', t')$ and $p(w, x, G_i, c'', t')$, hence $g'_2(v, w, t') =$ true. If $x \in W_2$, Let $p_1 \in P(v, x, G_i, c, t')$, and let $p_2 \in P(w, x, G_i, c, t')$. We can write $p_1 = (p_{11}, p_{12})$ with $p_{11} \in P(v, y, G_i, c, t')$, $p_{12} \in P(y, x, G_j, c, t')$ and $y \in X_j$. (Let y be the last vertex in X_j on p_1.) Similarly, we can write $p_2 = (p_{21}, p_{22})$ with $p_{21} \in P(w, z, G_i, c, t')$, $p_{22} \in P(z, x, G_j, c, t')$ and $z \in X_j$. This implies that $f_2(y, z, t')$ is true. Hence, there is a vertex x' with paths $p'_{12} \in P(y, x', G_j, c'', t')$ and $p'_{22} \in P(z, x', G_j, c'', t')$. Also, by Lemma 20, we have paths $p'_{11} \in P(v, z, G_i, c'', t')$ and $p'_{21} \in P(w, z, G_i, c'', t')$. Now, using path (p'_{11}, p'_{12}) from v to x' and path (p'_{21}, p'_{22}) from w to x', it follows that $g'_2(v, w, t')$ is true. So $g_2(v, w, t') \Rightarrow g'_2(v, w, t')$. An almost identical argument shows $g'_2(v, w, t') \Rightarrow g_2(v, w, t')$, hence $g_2 = g'_2$.

Finally, it follows directly from Claim 20 that $g_3 = g'_3$. □

We now describe our algorithm. After a rooted binary tree-decomposition (X, T) of $G = (V, E)$ has been found (in linear time [1]), the algorithm computes a full set and an example set for every node $i \in I$, in a bottom-up order. Clearly, when we have a full set for the root node of T, we can determine whether G has a vertex t-ranking, as we only have to check whether the full set of the root is non-empty. If so, any element of the example set of the root node gives us a vertex t-ranking of G.

It remains to show that we can compute for any node $i \in I$ a full set and an example set, given a full set and an example set for each of the children of $i \in I$. This is straightforward for the case that i is a leaf node: enumerate all functions $c : X_i \to \{1, \ldots, t\}$; for each such function c, test whether it is a vertex t-ranking of G_i, and if so, put c in the example set, and $Y(c)$ in the full set of characteristics.

Next suppose $i \in I$ has two children j_1 and j_2. (If i has one child j_1, then we can add another child j_2, which is a leaf in T and has $X_{j_2} = X_i$.) Suppose we have example sets Q_1, Q_2 for G_{j_1} and G_{j_2}. We compute a full set S and an example set Q for G_i in the following way:

Initially, we take S and Q to be empty.

For each triple (c_1, c_2, c_3), where c_1 is an element of Q_1, c_2 is an element of Q_2, and c_3 is an arbitrary function $c_3 : X_i \setminus (X_{j_1} \cup X_{j_2}) \to \{1, \ldots, t\}$, do the following:

- Check whether for all $v \in X_{j_1} \cap X_{j_2}$, $c_1(v) = c_2(v)$. If not, skip the following steps and proceed with the next triple.
- Compute the function $c : X_i \to \{1, \ldots, t\}$, defined as follows:

$$c(v) = \begin{cases} c_1(v) & \text{if } v \in V_{j_1} \\ c_2(v) & \text{if } v \in V_{j_2} \\ c_3(v) & \text{if } v \in X_i \setminus (X_{j_1} \cup X_{j_2}) \end{cases}$$

- Check whether c is a vertex t-ranking of G_i. If not, skip the following steps and proceed with the next triple.

- Compute $Y(c)$.
- If $Y(c) \notin S$, then put $Y(c)$ in S and put c in Q.

We claim that the resulting sets S and Q form a full set and an example set for G_i. Consider an arbitrary vertex t-ranking c' of G_i. Let $c_1 \in Q_1$ be the vertex t-ranking of Y_{j_1} that has the same characteristic as $c'|_{G_{j_1}}$. By definition of example set, c_1 must exist. Similarly, let $c_2 \in Q_2$ fulfill $Y(c_2) = Y(c'|_{G_{j_2}})$. Let $c_3 : X_i \setminus (X_{j_1} \cup X_{j_2}) \to \{1, \ldots, t\}$ be defined by $c_3(v) = c(v)$ for all $v \in X_i \setminus (X_{j_1} \cup X_{j_2})$. When the algorithm processes the triple (c_1, c_2, c_3), the first test will hold. Suppose c is the function, computed in the second test. Now note that $c = R(R(c', c_1), c_2)$. Hence, by Lemma 19, c is a vertex t-ranking and has the same characteristic as c'. Hence, S will contain $Y(c)$, and Q will contain a vertex t-ranking of G_i with the same characteristic as c and c'.

As the size of a full set, and hence of an example set for graphs G_i, $i \in I$ is polynomial, it follows that the computation of a full set and example set from these sets associated with the children of the node, can be done in polynomial time. (There are a polynomial number of triples (c_1, c_2, c_3). For each triple, the computation given above costs polynomial time.) As there are a linear number of nodes of the tree-decomposition, computing whether there exists a vertex t-ranking costs polynomial time (assuming $t = O(\log n)$.) By testing for each applicable value of t (see Lemma 18) for the existence of vertex t-rankings of G, we obtain the following result:

Theorem 22 *For any fixed k, there exists a polynomial time algorithm, that determines the vertex ranking number of graphs G with treewidth at most k, and finds an optimal vertex ranking of G.*

6 The equality $\chi_r = \chi$

In this section we consider questions related to the equality of the chromatic number and the vertex ranking number of graphs.

Theorem 23 *If $\chi_r(G) = \chi(G)$ holds for a graph G, then G also satisfies $\chi(G) = \omega(G)$.*

Proof: Suppose that $G = (V, E)$ has a vertex t-ranking $c : V \to \{1, 2, \ldots, t\}$ with $t = \chi(G)$. We are going to consider the separator tree $T(c)$ of this t-ranking. Recall that $T(c)$ is a rooted tree and that every internal node of $T(c)$ is assigned to a subset of the vertex set of G which is a separator of the corresponding subgraph of G, namely more than one component arises when all subsets on the path from the node to the root are deleted from the graph. Furthermore, all vertices assigned to the nodes of a path from a leaf to the root of $T(c)$ have pairwise different colors.

The goal of the following recoloring procedure is to show that either $\chi(G) = \omega(G)$ or we can recolor G to obtain a proper coloring with a smaller number of colors. However, the latter contradicts the choice of the $\chi(G)$-ranking c.

We label the nodes of the tree $T(c)$ according to the following marking rules:

1. Mark a node s of $T(c)$ if the union $U(s)$ of all vertex sets assigned to all nodes on the path from s to the root is *not* a clique in G.

2. Also, mark a leaf l of $T(c)$ if the union $U(l)$ of all vertex sets assigned to all nodes on the path from l to the root is a clique in G, but $|U(l)| < t$.

Case 1: There is an unmarked leaf l.
We have $|U(l)| = t$ and $U(l)$ is a clique. Hence, $\omega(G) = \chi(G)$.

Case 2: There is no unmarked leaf.
We will show that this would enable us to recolor G saving one color, contradicting the choice of c.

Since every leaf of $T(c)$ is marked, every path from a leaf to the root consists of marked nodes eventually followed by unmarked nodes. Consequently, there is a collection of marked branches of $T(c)$, i.e., subtrees of $T(c)$ induced by one node and all its descendants for which all nodes are marked and the father of the highest node of each branch is unmarked or the highest node is the root of $T(c)$ itself.

If the root of $T(c)$ is marked then we have exactly one marked branch, namely $T(c)$ itself. Then, by definition, the separator S assigned to the root is *not* a clique. However, none of its colors is used by the ranking for vertices in $V \setminus S$. Simply, any coloring of the separator S with fewer than $|\,S\,|$ colors will produce a coloring of G with fewer than $\chi(G)$ colors; contradiction.

If the root is unmarked, then we have to work with a collection of b marked branches, $b > 1$. Notice that all color-1 vertices of G are assigned to leaves of $T(c)$ and that any leaf of $T(c)$ belongs to some marked branch B. We are going to recolor the graph G by recoloring the marked branches one by one such that the new coloring of G does not use color 1. Let us consider a marked branch B. Let h be its highest node in $T(c)$, and $S(h)$ the set assigned to h. Since h is marked but the root is unmarked, there must exist a vertex x of $S(h)$ and a vertex y belonging to $U(h)$ which are nonadjacent. Then $c(x) \neq c(y)$ since all vertices of $U(h)$ have pairwise different colors.

Assume $c(x) = 1$ or $c(y) = 1$. Then h is a leaf of $T(c)$. Hence, x and y, respectively, is the only color-1 vertex of G assigned to a node of B. We simply recolor x and y with $\max(c(x), c(y))$.

Finally consider the case $c(x) \neq 1$ and $c(y) \neq 1$. All color-1 vertices in the subgraph of G corresponding to B are recolored with $c(x)$ and x is recolored with $c(y)$. By the construction of $T(c)$, this does not influence other parts of the graph, since they are separated by vertex sets with higher colors.

Having done this operation in every marked branch, eventually we get a new color assignment of G which is still a proper coloring (though usually not a ranking). Since all leaves of $T(c)$ are marked, and no internal node of $T(c)$ contains color-1 vertices, color 1 is eliminated from G, contradicting the assumption $\chi_r(G) = \chi(G)$. Consequently, Case 2 cannot occur, implying $\chi(G) = \omega(G)$. This completes the proof. □

Clearly, $\chi_r(G) = \chi(G)$ does not imply that G is a perfect graph. (Trivial counterexamples are of the form $G = G' \cup K_{\chi_r(G')}$ where G' is an arbitrary imperfect graph.) On the other hand, if we require the equality on all induced subgraphs, then we remain with a relatively small class of graphs that is also called 'trivially perfect' in the literature (cf. [11]).

Theorem 24 *A graph $G = (V, E)$ satisfies $\chi_r(G[A]) = \chi(G[A])$ for every $A \subseteq V$ if and only if neither P_4 nor C_4 is an induced subgraph of G.*

Proof: The condition is necessary since $\chi_r(P_4) = \chi_r(C_4) = 3$ and $\chi(P_4) = \chi(C_4) = 2$.

Now let G be a P_4-free and C_4-free graph. The graphs with no induced P_4 and C_4 are precisely those in which every connected induced subgraph H contains a dominating vertex w, i.e., w is adjacent to all vertices of H [27]. Hence, the following efficient algorithm produces an optimal ranking in such graphs: If $H = (V', E')$ is connected, then we assign the color $\omega(H)$ to a dominating vertex w. Clearly, $\chi(H[V' \setminus \{w\}]) = \omega(H[V' \setminus \{w\}]) = \omega(H) \Leftrightarrow 1$, and it is easily seen that $\chi_r(H[V' \setminus \{w\}]) = \chi_r(H) \Leftrightarrow 1$ also holds; thus, induction can be applied. On the other hand, if H is disconnected, then an optimal ranking can be generated in each of its components separately. □

7 Edge rankings of complete graphs

While obviously $\chi_r(K_n) = n$, it is not easy to give a closed formula for the edge ranking number of the complete graph. The most convenient way to determine $\chi'_r(K_n)$ seems to introduce a function $g(n)$ by the rules

$$g(1) = \Leftrightarrow 1,$$
$$g(2n) = g(n),$$
$$g(2n+1) = g(n+1) + n.$$

In terms of this $g(n)$, the following statement can be proved.

Theorem 25 *For every positive integer n,*

$$\chi'_r(K_n) = \frac{n^2 + g(n)}{3}.$$

Proof: The assertion is obviously true for $n = 1, 2, 3$. For larger values of n we are going to apply induction.

Similarly to vertex t-rankings, the following property holds for every edge t-ranking of a graph $G = (V, E)$: if i is the largest color occurring more than once, then the edges with colors $i+1, i+2, \ldots, t$ form an edge separator of G. Moreover, doing an appropriate relabeling of these colors $i+1, i+2, \ldots, t$ we get a new edge t-ranking of G with the property that there is a color $j > i$ such that all edges with colors $j, j+1, \ldots, t$ form an edge separator of G which is minimal under inclusion.

We have to show that the best way to choose this edge separator R with respect to an edge ranking in a complete graph is by making the two components of $G[E \setminus R]$ as equal-sized as possible. Let us consider a K_n, $n \geq 4$. Let n_1 and n_2 be the numbers of vertices in the components, hence $n_1 + n_2 = n$ and the corresponding edge separator has size $n_1 n_2$. Every edge ranking starting with this separator has at least

$$n_1 n_2 + \max\{\chi'_r(K_{n_1}), \chi'_r(K_{n_2})\} = n_1 n_2 + \chi'_r(K_{\max\{n_1,n_2\}})$$

colors, and there is indeed one using exactly that many colors. Defining $a_1 := \min(n_1, n_2)$ and repeating the same argument for $n' := n - a_1$, and so on, we eventually get a sequence of positive integers $a_1, \ldots, a_s$, for some s, such that $\sum_{i=1}^{s} a_i = n$ and

$$a_i \leq \sum_{i<j\leq s} a_j \quad \text{for all } i, \quad 1 \leq i < s. \tag{1}$$

Notice that at least the last two terms of any such sequence are equal to 1. It is easy to see that the number of colors of any edge ranking represented by $a_1, \ldots, a_s$ is equal to $\sum_{1\leq i<j\leq s} a_i a_j$, consequently

$$\chi'_r(K_n) = \min \sum_{1\leq i<j\leq s} a_i a_j = \binom{n}{2} - \max \sum_{i=1}^{s} \binom{a_i}{2},$$

subject to the condition (1). Since a decreasing sort of the sequence maintains (1) we may assume $a_1 \geq a_2 \geq \ldots \geq a_s$. Thus, for each value of n, $\min \sum_{1\leq i<j\leq s} a_i a_j$ is attained precisely by the unique sequence satisfying $a_i = \lfloor \frac{1}{2} \sum_{i\leq j\leq s} a_j \rfloor$ for all i, $1 \leq i < s$. In particular, we obtain

$$\chi'_r(K_n) = \chi'_r(K_{\lceil n/2 \rceil}) + \lfloor n/2 \rfloor \lceil n/2 \rceil.$$

Applying this recursion, it is not difficult to verify that, indeed, $\chi'_r(K_n)$ can be written in the form $\frac{1}{3}(n^2 + g(n))$, where $g(n)$ is the function defined above. □

Observing that $g(2^n) = -1$ for all $n \geq 1$, we obtain the following interesting result.

Corollary 26

$$\chi'_r(K_{2^n}) = \frac{4^n - 1}{3}.$$

8 Conclusions

We studied algorithmic and graph-theoretic properties of rankings of graphs. For many special classes of graphs, the algorithmic complexity of VERTEX RANKING is now known. However the algorithmic complexity of VERTEX RANKING when restricted to chordal graphs or circle graphs is still unknown. Furthermore it is not even known whether the EDGE RANKING problem is NP-complete.

We started a graph-theoretic study of vertex ranking and edge ranking as a particular kind of proper (vertex) coloring and proper edge coloring, respectively. Much research has to be done in this direction. It is of particular interest which of the well-known problems in the theory of vertex colorings and edge colorings are also worth studying for vertex rankings and edge rankings.

References

[1] H.L. BODLAENDER. A linear time algorithm for finding tree-decompositions of small treewidth. *Proceedings of the 25th Annual ACM Symposium on Theory of Computing*, ACM Press, New York, 1993, pp. 226–234.

[2] H.L. BODLAENDER. A tourist guide through treewidth. *Acta Cybernetica* **11** (1993), 1–23.

[3] H.L. BODLAENDER, J.R. GILBERT, H. HAFSTEINSSON, T. KLOKS. Approximating treewidth, pathwidth and minimum elimination tree height. *Proceedings of the 17th International Workshop on Graph-Theoretic Concepts in Computer Science WG'91*, Springer-Verlag, Lecture Notes in Computer Science 570, 1992, pp. 1–12. To appear in *Journal of Algorithms.*

[4] J.A. BONDY, U.S.R. MURTY. *Graph Theory with Applications.* American Elsevier, New York, 1976.

[5] A. BRANDSTÄDT. Special graph classes – a survey. *Schriftenreihe des FB Mathematik*, Universität Duisburg, SM–DU–199, 1991.

[6] P. DE LA TORRE, R. GREENLAW, A.A. SCHÄFFER. Optimal edge ranking of trees in polynomial time. *Proceedings of the 4th Annual ACM-SIAM Symposium on Discrete Algorithms*, Austin, Texas, 1993, pp. 138–144.

[7] J.S. DEOGUN, T. KLOKS, D. KRATSCH, H. MÜLLER. On vertex ranking for permutation and other graphs. *Proceedings of the 11th Annual Symposium on Theoretical Aspects of Computer Science*, P. Enjalbert, E.W. Mayr, K.W. Wagner, (eds.), Lecture Notes in Computer Science 775, Springer-Verlag, Berlin, 1994, 747–758.

[8] J.S. DEOGUN, Y. PENG. Edge ranking of trees. *Congressus Numerantium* **79** (1990), 19–28.

[9] I.S. DUFF, J.K. REID. The multifrontal solution of indefinite sparse symmetric linear equations. *ACM Transactions on Mathematical Software* **9** (1983), 302–325.

[10] M.R. GAREY, D.S. JOHNSON. *Computers and Intractability: A Guide to the Theory of NP-completeness.* W.H. Freeman and Company, New York, 1979.

[11] M.C. GOLUMBIC. *Algorithmic Graph Theory and Perfect Graphs.* Academic Press, New York, 1980.

[12] A.V. IYER, H.D. RATLIFF, G. VIJAYAN. Optimal node ranking of trees. *Information Processing Letters* **28** (1988), 225–229.

[13] A.V. IYER, H.D. RATLIFF, G. VIJAYAN. Parallel assembly of modular products—an analysis. Technical Report 88-06, Georgia Institute of Technology, 1988.

[14] A.V. IYER, H.D. RATLIFF, G. VIJAYAN. On an edge ranking problem of trees and graphs. *Discrete Applied Mathematics* **30** (1991), 43–52.

[15] M. KATCHALSKI, W. MC CUAIG, S. SEAGER. Ordered colourings. Manuscript, University of Waterloo, 1988.

[16] T. KLOKS. *Treewidth.* Ph. D. Thesis, Utrecht University, The Netherlands, 1993.

[17] C.E. LEISERSON. Area efficient graph layouts for VLSI. *Proceedings of the 21st Annual IEEE Symposium on Foundations of Computer Science*, 1980, pp. 270–281.

[18] J.W.H. LIU. The role of elimination trees in sparse factorization. *SIAM Journal of Matrix Analysis and Applications* **11** (1990), 134–172.

[19] J. NEVINS, D. WHITNEY. (eds.) *Concurrent Design of Products and Processes*. McGraw-Hill, New York, 1989.

[20] A. POTHEN. The complexity of optimal elimination trees. Technical Report CS-88-13, Pennsylvania State University, U.S.A., 1988.

[21] N. ROBERTSON, P. D. SEYMOUR. Graph minors. IV. Tree-width and well-quasi-ordering. *Journal on Combinatorial Theory* Series B **48** (1990), 227–254.

[22] N. ROBERTSON, P. D. SEYMOUR. Graph minors. V. Excluding a planar graph. *Journal on Combinatorial Theory* Series B **41** (1986), 92–114.

[23] N. ROBERTSON, P. D. SEYMOUR. Graph minors. XIII. The disjoint paths problem. Manuscript, 1986.

[24] A.A. SCHÄFFER. Optimal node ranking of trees in linear time. *Information Processing Letters* **33** (1989/90), 91–96.

[25] P. SCHEFFLER. Node ranking and searching on graphs (Abstract). in: U. Faigle and C. Hoede, (eds.), *3rd Twente Workshop on Graphs and Combinatorial Optimization*, Memorandum No.1132, Faculty of Applied Mathematics, University of Twente, The Netherlands, 1993.

[26] A. SEN, H. DENG, S. GUHA. On a graph partition problem with application to VLSI layout. *Information Processing Letters* **43** (1992), 87–94.

[27] E.S. WOLK. The comparability graph of a tree. *Proceedings of the American Mathematical Society* **3** (1962), 789–795.

[28] X. ZHOU and T. NISHIZEKI. Finding optimal edge-rankings of trees. To appear in *Proceedings of the 6th Annual ACM-SIAM Symposium on Discrete Algorithms*, 1995.

[29] X. ZHOU and T. NISHIZEKI. An efficient algorithm for edge-ranking trees. In J. van Leeuwen, editor, *Proceedings 2nd Annual European Symposium on Algorithms ESA'94*, pages 118–129. Springer Verlag, Lecture Notes in Computer Science, vol. 855, 1994.