Joint Conference of 1996 International Computer Symposium
December 19~21, Kaohsiung, Taiwan, R.O.C.

The NP-completeness of Edge Ranking

T.W. Lam and F.L. Yue
Department of Computer Science
University of Hong Kong
Pokfulam Road, Hong Kong
Email address: {twlam, flyue}@cs.hku.hk

Abstract

An edge ranking of a graph G is a labeling of its edges with positive integers such that every path between two edges with the same label i contains an intermediate edge with label $j > i$. An edge ranking is optimal if it uses the least number of distinct labels among all possible edge rankings. Such a ranking corresponds to a minimum-height edge-separator tree of G. The problem of finding an optimal edge ranking has been studied intensively; recent development has shown that the problem when restricted to trees is not NP-hard and indeed admits a polynomial-time solution, yet the complexity of the problem for general graphs has remained open in the literature. In this paper we settle this open question and prove that finding an optimal edge ranking of a graph is NP-hard.

1 Introduction

Let G be an undirected graph. An *edge ranking* of G is a labeling of its edges with positive integers such that every path between two edges with the same label i contains an intermediate edge with label $j > i$. An edge ranking is *optimal* if it uses the least number of distinct labels among all possible edge rankings. Such a ranking corresponds to a minimum-height edge-separator tree of G. An example is given in Figure 1. The problem of finding an optimal edge ranking was first studied by Iyer, Ratiff, and Vijayan [8] as they found the problem having an application in scheduling the assembly of multipart products.

A closely related analogue of edge ranking is vertex ranking. A vertex ranking of a graph G is a labeling of its vertices such that every path between two vertices with the same label i contains an intermediate vertex with label $j > i$. The complexity of finding an optimal vertex ranking has been well studied. In particular, Pothen [9] showed that finding an optimal vertex ranking of graphs is NP-hard, while Schäffer [10], improving the work of Iyer, Ratiff, and Vijayan [7], obtained a linear time algorithm for finding an optimal node ranking of a tree. In the literature, there

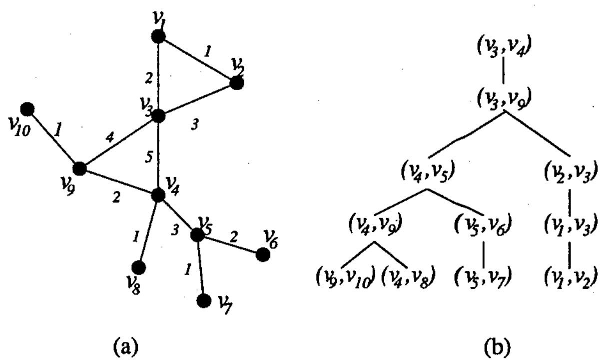


Figure 1: An optimal edge ranking of a graph and the corresponding edge-separator tree.

are also a number of polynomial time algorithms for finding approximately optimal vertex rankings of general graphs [2] and different kinds of restricted graphs such as permutation graphs [4].

The complexity of finding an optimal edge ranking has been relatively less understood. In the pioneer work of Iyer, Ratliff, and Vijayan [8], only an approximation algorithm was given for trees; whether finding an optimal edge ranking of a tree or a graph is in P or NP-hard was left undetermined. The open question for trees was eventually answered by de la Torre, Greenlaw, and Schäffer [3], who gave an $O(n^3 \log n)$ time algorithm for finding an optimal edge ranking of a tree, where n is the number of nodes. Later, Zhou and Nishizeki [11] showed that the running time can be improved to $O(n^2 \log \Delta)$, where Δ is the maximum degree. With respect to graphs, though there is a general belief that finding an optimal ranking is NP-hard [3, 1, 11], there has not been any significant progress in the literature. Recently, Bodlaendar *et al.* [1] found that the problem of deciding whether a graph G has an edge ranking using at most a fixed constant of distinct labels (independent of G) is solvable in linear time. In this paper we show that the general edge-ranking problem, which, given a graph G and an integer t, determines whether G has an edge ranking using at most t distinct labels, is NP-complete. Thus, finding an optimal ranking of a graph is NP-hard.

There is indeed a trivial reduction from the edge-ranking problem to the vertex-ranking problem [1], but the reverse has not been known. Proving the NP-completeness of edge ranking seems to be more difficult than that of the vertex analogue. Such relationship between edge-based and vertex-based graph problems is not new in the literature; a typical example is the edge coloring [6] versus vertex coloring [5]. In this paper we prove the NP-completeness of edge ranking by first reducing formula satisfiability to the edge-ranking problem of *multigraphs* (Section 4), and then transforming the latter to the edge-ranking problem of simple graphs (Section 5).

Given a formula F, we are going to construct a multigraph G such that F is satisfiable if and only if G has rank equal to a certain value. G is composed of a number of subgraphs, each corresponding to a variable or a clause of F. In Section 3, we discuss some basic rules of composing graphs that are useful in the construction.

Before we proceed to the NP-completeness proofs, we give a few more notations below and introduce some important and interesting properties of optimal edge rankings in Section 2.

The symbol ψ is used to denote an edge ranking of a graph. We define *rank*(ψ) (or the rank of ψ) to be the number of distinct labels used by ψ and *rank*(G) (or the rank of G) to be the number of distinct labels used by an optimal edge ranking of G.

A multigraph is a graph in which a pair of nodes can be connected by one or more parallel edges. Note that the parallel edges between two nodes can form a path themselves. Thus, all parallel edges between two nodes must be ranked with distinct labels. Figure 2 gives an optimal edge ranking of a multigraph.

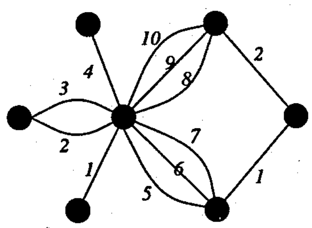


Figure 2: An optimal edge ranking of a multigraph.

2 Preliminary

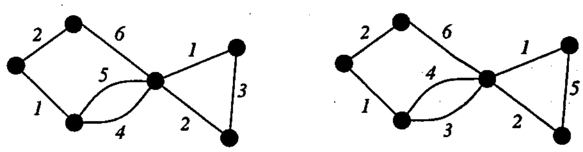
This section shows that, among all optimal edge rankings of a graph G, there is one satisfying some desirable properties. Such a ranking is said to be in a normal form.

Minimal cut and primitive separator: Let $G = (V, E)$ be a multigraph. For any $C \subseteq E$, C is an *edge cut* of G if the removal of C from G disconnects G. An edge cut C of G is said to be *minimal* if the removal of any subset $C' \subset C$ does not disconnect G. For any minimal cut C of G, the removal of C disconnects G into exactly two connected components. Also, if C contains an edge (u, v), all the parallel edges connecting u and v belong to C and the nodes u and v become disconnected after the removal of C.

Let ψ be an edge ranking of G. Consider the process of removing edges from G in the decreasing order of the labels given by ψ. Recall that the edge with the biggest label under ψ is unique. After this edge is removed, G either remains connected or is disconnected into two components. In the former case, the edge with the second largest label is unique. We remove it from G and so on until G becomes disconnected. Let t be the label of the last edge removed from G. We define the *primitive separator* of ψ to be the set of edges that have labels $\geq t$ under ψ. The removal of the primitive separator from G disconnects it into exactly two connected components. All edges not in the primitive separator have labels less than t.

Fact: Let ψ be an optimal edge ranking of a graph G. Let S be the primitive separator of ψ and let G_1 and G_2 be the connected components after removing S from G. Then both G_1 and G_2 can be ranked using at most *rank*(G) $- |S|$ distinct labels.

We say that ψ satisfies the *minimal cut property* if its primitive separator S forms a minimal cut in G and the restrictions of ψ to the two connected components resulting from the removal of C from G also satisfy the minimal cut property.



(a) (b)

Figure 3: The edge ranking in (a) satisfies the minimal cut property but the one in (b) does not.

Lemma 1 Any multigraph G has an optimal edge ranking satisfying the minimal cut property.

Proof: The following algorithm shows that, given an edge ranking ψ of G, we can rearrange the labels of the edges such that the resultant ranking satisfies the minimal cut property.

Let S be the primitive separator of ψ and let $S' \subseteq S$ be a minimal cut of G. We rearrange the labels on the edges of S such that S' receives the biggest labels. Note that such rearrangement still preserves the requirement of a ranking. Removing S' from G leaves exactly two connected components G'_1 and G'_2. If G'_1 or G'_2 consists of one or more edges, we proceed to rearrange the labels on their edges recursively. Let ψ' be the resultant ranking of G. As the process above never introduces new labels, $rank(\psi') = rank(\psi)$. If ψ is optimal then ψ' is optimal, too. □

Terminal edges and internal edges: In a multigraph G, we refer to the *degree* of a vertex as the number of edges incident to it (instead of the number of adjacent vertices). We call an edge a *terminal edge* if one of its endpoints has degree one. Other edges are called internal edges. Intuitively, a terminal edge cannot be the intermidate edge of any path between two edges and does not deserve a big label. The following lemma shows the existence of an optimal ranking that does not include any terminal edge in its primitive separator.

Lemma 2 For any graph G containing at least one internal edge, G has an optimal edge ranking ψ satisfying the minimal cut property such that the primitive separator of ψ contains no terminal edges.

Proof: Let ψ be an optimal edge ranking of G satisfying the minimal cut property. Suppose the primitive separator S of ψ contains a terminal edge $\hat{e}$. Since the removal of $\hat{e}$ disconnects its unit degree endpoint from G, S contains $\hat{e}$ as its only edge and $\hat{e}$ gets the biggest label in ψ. In this case, we construct another optimal edge ranking ψ' for G from ψ such that $\hat{e}$ gets a label 1 as follows:

$$\psi'(e) = \begin{cases} 1 & \text{if } e = \hat{e} \\ \psi(e) + 1 & \text{otherwise.} \end{cases}$$

ψ' uses the same number of distinct labels as ψ and $\hat{e}$ no longer lies in the primitive separator. Also, ψ satisfies the minimal cut property. If the primitive separator of ψ' does not contains any terminal edge then we are done. Otherwise, we can repeat the process until we get an optimal edge ranking of G with an internal edge getting the maximum label. □

More Definitions: An optimal edge ranking ψ is said to be in a *normal form* if it satisfies the minimal cut property and its primitive separator contains no terminal edges.

Given a multigraph G, the *edge multiplicity* of an edge $e = (u, v)$ is the number of parallel edges connecting u and v in G. The *internal edge multiplicity* of G is defined to be the minimum edge multiplicity over all its internal edges.

3 Composition of graphs

In this section, we describe a way to compose graphs together. The resultant graph will have a natural lower bound on its rank. The most interesting property of such a composition is that if the resultant graph can be ranked tightly (i.e. meeting its lower bound), the individual constituent graphs can be ranked tightly, too. We first examine the way of composing two smaller graphs. Then we study a generalization to compose a sequence of graphs in Section 3.1.

Let G_1 and G_2 be two connected multigraphs. We construct a bigger multigraph G by connecting G_1 and G_2 with another multigraph H where $V(H) = U_1 \cup U_2 \cup U_3$ for some $U_1 \subseteq V(G_1)$, $U_2 \subseteq V(G_2)$ and $U_3 \cap (V(G_1) \cup V(G_2)) = \emptyset$, and $E(H) \subseteq (U_1 \times U_2) \cup (U_1 \times U_3) \cup (U_3 \times U_2)$. For any $C \subseteq E(H)$, C is said to be a *total cut* of H if the removal of C from H disconnects all the vertices in U_1 from all the vertices in U_2. Figure 4 illustrates a cut and a total cut. Let f_H denote the size of the smallest total cut of H.

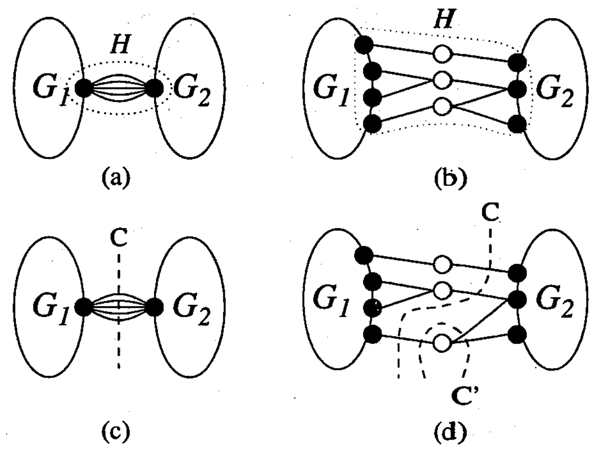


Figure 4: (a) and (b) give examples of joining G_1 and G_2 with different H's. (c) and (d) illustrate cuts of H. C is a total cut of H but C' is not.

In the reduction argument to be shown later, the graphs G_1, G_2 and H involved are not chosen arbitrarily. We know a lower bound on the ranks of G_1 and G_2 and their internal edge multiplicities are big enough to exceed the value of f_H. The next lemma shows that, with such assumptions, the graph G also has a non-trivial lower bound on its ranks.

Lemma 3 Let G be a multigraph formed by connecting two multigraphs G_1 and G_2 with another multigraph H, where G_1 and G_2 have ranks $\geq k$ and internal edge multiplicity $\geq f_H + 1$. Then $rank(G) \geq f_H + k$.

Proof: We prove Lemma 3 by induction on the number of edges m in H.

Base Case, $m = 1$: Obviously, $f_H = 1$. Suppose on the contrary that there exists a ranking ψ of G using no more than k distinct labels. Let e be the edge in G getting the maximum label under ψ. Note that e is unique. W.L.O.G., we assume e is not in G_1. Then ψ induces a ranking of G_1 using at most $k-1$ distinct labels. This contradicts the fact that the rank of G_1 is at least k.

Induction Step: Suppose the lemma holds whenever H has less than m edges. Consider the case when H has m edges. Let ψ be an optimal edge ranking of G in a normal form, and let S denote the primitive separator of ψ. We show that $rank(\psi) \geq f_H + k$ in each of the following cases:

Case 1—$S \not\subseteq E(H)$: Suppose S contains an internal edge e of G_1. Let E' be the set of all parallel edges joining the endpoints of e. Since G_1 has internal edge multiplicity greater than f_H, $|E'| \geq f_H + 1$. On the other hand, $E' \subseteq S$ (otherwise, S is not a minimal cut of G); hence, with respect to ψ, all other edges in G have a label different from those of E'. In particular, G_2 receives at least k distinct labels. Thus, $rank(\psi) \geq |E'| + k > f_H + k$.

Case 2—$S \subseteq E(H)$ and S is a total cut of H: Note that $|S| \geq f_H$. Also, G_1 contains at least k distinct labels different from those in S under ψ. Therefore, ψ uses at least $|S| + k \geq f_H + k$ distinct labels.

Case 3—$S \subseteq E(H)$ and S is not a total cut of H: Suppose we remove S from G, G_1 and G_2 are still connected through some proper subgraph H' of H. Let G' denote the graph formed by connecting G_1 and G_2 with H'. Then,

$$\begin{aligned} rank(\psi) &\geq |S| + rank(G') \\ &\geq |S| + f_{H'} + k \quad \text{(by induction hypothesis)} \\ &> k + f_H \end{aligned}$$

The last inequality follows because a total cut of H' plus S forms a total cut of H but contains at least one redundant edge in S. We have completed the induction and proved Lemma 3. □

Next, we show a useful observation when G can actually be ranked using $f_H + k$ distinct labels.

Lemma 4 Let G_1, G_2, H and G be multigraphs defined as in Lemma 3. Suppose $rank(G) = f_H + k$. Then there exists an optimal edge ranking ψ of G such that the primitive separator of ψ is a minimum total cut of H.

Proof: Let ψ be an optimal edge ranking of G in a normal form. Its primitive separator S cannot contain any edges of G_1 or G_2 (otherwise, we can use the argument in Case 1 of Lemma 3 to prove that ψ uses more than $f_H + k$ distinct labels and a contradiction occurs). So $S \subseteq E(H)$

Suppose on the contrary that S is not a total cut of H. Then removing S from H still leaves a subgraph H' that can connect G_1 and G_2. Let G' denote the resultant graph. Therefore,

$$\begin{aligned} rank(\psi) &\geq |S| + rank(G') \\ &\geq |S| + f_{H'} + k \qquad \text{(by Lemma 3)} \\ &> f_H + k. \end{aligned}$$

A contradictions occurs. We conclude that S is a total cut of H.

If S is not a minimum total cut of H then S contains more than f_H edges and G_1 (and G_2) is ranked using less than k labels under ψ. Contradiction occurs again and S must be a minimum total cut of H. □

Corollary 5 Let G_1, G_2, H and G be multigraphs defined as in Lemma 3. Suppose $rank(G) = f_H + k$. Then G_1 and G_2 can each be ranked using k distinct labels.

3.1 Restricted generalization

In this subsection, we restrict our attention to the simplest kind of connection graphs H where $V(H) = \{u_1, u_2\}$ for some $u_1 \in V(G_1)$ and $u_2 \in V(G_2)$ and $E(H)$ contains parallel edges between u_1 and u_2. We generalize the composition of two graphs to construct a chain-like graph G that is composed of a sequence of graphs $G_1, G_2, \cdots, G_{2^d}$ for some $d \geq 1$. As mentioned before, we will assume all the graphs G_i have a known lower bound on their ranks. Then we can obtain a lower bound for G, and more importantly, prove that if G can be ranked tightly, each individual graph G_i can be ranked tightly, too.

For each G_i, we choose a particular internal node u_i for connection purpose. Inside G, two consecutive graphs G_i, G_{i+1} are simply connected by b or more parallel edges between the two designated nodes u_i and u_{i+1}, where b is some fixed integer. The following is a recursive definition of the way we compose a sequence of graphs to form a chain: For any integers $d \geq 0$ and $b \geq 1$, define $\mathcal{L}([G_1..G_{2^d}], b)$ to be a graph formed by connecting the graphs $\mathcal{L}([G_1..G_{2^{d-1}}], b+1)$ and $\mathcal{L}([G_{2^{d-1}+1}..G_{2^d}], b+1)$ with b parallel edges between the nodes $u_{2^{d-1}}$ and $u_{2^{d-1}+1}$. If $d = 0$, the sequence consists of one single graph and we define $\mathcal{L}([G_1], b)$ to be G_1 itself. Let G be the graph

$\mathcal{L}([G_1..G_{2^d}], b)$. Inside G, the multiplicity of the edge (u_i, u_{i+1}) is in the range $[b..b+d-1]$; thus, G_i and G_{i+1} are connected by a graph with a total cut of size at most $b+d-1$.

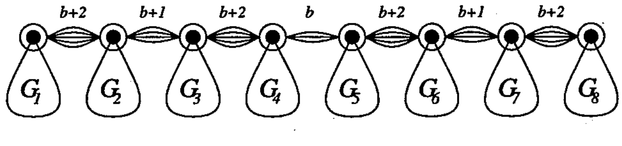


Figure 5: The graph $G([G_1..G_8], b)$.

The recursive definition of the chain $\mathcal{L}([G_1..G_{2^d}], b)$ involves defining chains of length 2^j for $j = 0, 1, \cdots, d$. Let $\mathcal{G}_j$ denote the set of chains of length 2^j, i.e.,

$$\mathcal{G}_j = \{\mathcal{L}([G_{(i-1)2^j+1}..G_{i2^j}], b+d-j) \mid 1 \le i \le 2^{d-j}\}.$$

Note that a chain in $\mathcal{G}_j$, say, $\mathcal{L}([G_{(i-1)2^j+1}..G_{i2^j}], b+d-j)$, is formed by joining two chains in $\mathcal{G}_{j-1}$, namely $\mathcal{L}([G_{(i-1)2^j+1}..G_{(2i-1)2^{j-1}}], b+d-(j-1))$ and $\mathcal{L}([G_{(2i-1)2^{j-1}+1}..G_{i2^j}], b+d-(j-1))$, with $b+d-j$ parallel edges.

Obviously, if each G_i can be ranked with k distinct labels then each chain of length 2 in $\mathcal{G}_1$ can be ranked with $(b+d-1)+k$ distinct labels (the highest $b+d-1$ labels are put on the $b+d-1$ parallel connection edges). Similarly, each chain in $\mathcal{G}_2$ can be ranked with $(b+d-2)+(b+d-1)+k$ distinct labels. In general, for each $j = 0, 1, \cdots, d$, every chain in $\mathcal{G}_j$ can be ranked with $T(j) = \sum_{\ell=1}^{j}(b+d-\ell)+k = jb+\sum_{1\le\ell\le j}(d-\ell)+k$ distinct labels. So $\mathcal{L}([G_1..G_{2^d}], b)$ can be ranked with $T(d) = db + \frac{d(d-1)}{2} + k$ distinct labels.

In many cases we may only know the lower bound on the rank of each G_i. The following lemma derives a lower bound of $rank(G)$.

Lemma 6 Let $G = \mathcal{L}([G_1..G_{2^d}], b)$, where each G_i is a connected multigraph with rank $\ge k$ and the internal edge multiplicity of each G_i is at least $b+d$. Then $rank(G) \ge db + \frac{d(d-1)}{2} + k$.

Proof: We prove by induction on $j = 0, 1, \cdots, d$ that the ranks of each chain in $\mathcal{G}_j$ is at least $T(j) = jb+\sum_{1\le l\le j}(d-l)+k$. In particular, when $j = d$, we obtain the result that $rank(G) \ge T(d) = db+\frac{d(d-1)}{2}+k$.

Base case, $j = 0$: We are dealing with sequences each consisting of one graph G_i. Recall that G_i is assumed to have rank $\ge k$. Thus $\mathcal{L}([G_i], b+d)$ also has a rank $\ge k$.

Induction Step: Consider $j \ge 1$. Recall that every chain $\mathcal{L}$ in $\mathcal{G}_j$ is formed by joining two chains $\mathcal{L}_1$ and $\mathcal{L}_2$ in $\mathcal{G}_{j-1}$ by $b+d-j$ parallel edges. Denote by H the graph comprising the $b+d-j$ parallel edges between $\mathcal{L}_1$ and $\mathcal{L}_2$. Then $f_H = b+d-j$.

We would like to apply Lemma 3 to deduce the lower bound for $\mathcal{L}$, which requires two preconditions. **(i)** By the induction hypothesis, the ranks of $\mathcal{L}_1$ and $\mathcal{L}_2$ are at least $T(j-1)$. **(ii)** Every G_i is assumed to have internal edge multiplicity $\ge b+d > f_H$. Moreover, the edges connecting two consecutive G_i's in $\mathcal{L}_1$ and $\mathcal{L}_2$ have multiplicity $\ge b+d-(j-1) > f_H$. Thus, by Lemma 3, $rank(G) \ge b+d-j+T(j-1) = T(j)$. □

Lemma 7 Let $G = \mathcal{L}([G_1..G_{2^d}], b)$, where each G_i is a connected multigraph with rank $\ge k$ and the internal edge multiplicity of each G_i is at least $b+d$. If $rank(G) = db+\frac{d(d-1)}{2}+k$ then each G_i can be ranked using k distinct labels.

Proof: We use backward induction on $j = d, d-1, \cdots, 0$ to prove that every chain in $\mathcal{G}_j$ can be ranked using $T(j) = jb+\sum_{1\le l\le j}(d-l)+k$ distinct labels. Thus, when $j = 0$, it implies that each individual G_i can be ranked using k distinct labels.

The base case where $j = d$ is trivial. The induction step is based on Corollary 5. Consider any $j < d$. Suppose every chain in $\mathcal{G}_{j+1}$ can be ranked using $T(j+1)$ distinct labels. For any chain $\mathcal{L}_1$ in $\mathcal{G}_j$, $\mathcal{L}_1$ is joined with another chain $\mathcal{L}_2$ in $\mathcal{G}_j$ by $b+d-(j+1)$ parallel edges to form a chain $\mathcal{L}$ in $\mathcal{G}_{j+1}$. Denote by H the graph comprising the $b+d-(j+1)$ parallel edges between $\mathcal{L}_1$ and $\mathcal{L}_2$. Then $f_H = b+d-(j+1)$. By supposition, $\mathcal{L}$ has rank $\le T(j+1) = b+d-(j+1)+T(j) = f_H+T(j)$

Both $\mathcal{L}_1$ and $\mathcal{L}_2$ have internal edge multiplicity $\ge (b+d-j) > f_H$. By Lemma 6, the rank of both $\mathcal{L}_1$ and $\mathcal{L}_2$ are at least $T(j)$. So by Corollary 5, both $\mathcal{L}_1$ and $\mathcal{L}_2$ can be ranked using no more than $T(j)$ distinct labels. □

4 Reduction from satisfiability

This section gives a reduction from the satisfiability problem (in particular, 3CNF-SAT) [5] to the edge ranking problem of multigraphs, thus proving the latter is NP-complete.

Let F be a Boolean formula with n variables $\{x_1, x_2, \ldots, x_n\}$ and ℓ clauses $\{c_1, c_2, \ldots, c_\ell\}$ where $c_i = (\ell_{i,1} + \ell_{i,2} + \ell_{i,3})$. Let d be the smallest integer such that $2^d \ge \max\{n, \ell\}$. Let $b = 3\ell+1$ and $\epsilon = b+d+2$.

For each variable x_j, we construct a variable component X_j with edge multiplicity ϵ and two designated nodes x_j and $\overline{x_j}$ as depicted in Figure 6(a).

Fact: The rank of X_j is exactly 2ϵ. If we attach up to ϵ simple edges to either x_j or $\overline{x_j}$, the resultant graph

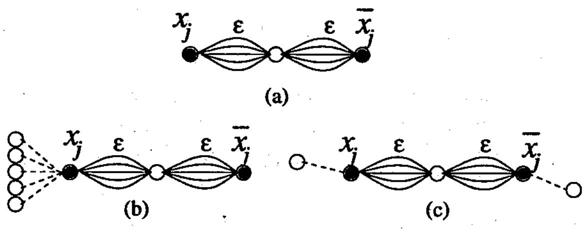


Figure 6: A variable component.

still has rank 2ϵ. However, if edges are attached to both x_j and $\overline{x_j}$, the resultant graph has rank $> 2\epsilon$ (see Figure 6(b) and (c)).

For each clause c_i, we construct a clause component C_i consisting of two edges with multiplicity ϵ and one edge with multiplicity $\epsilon - 2$ joined at the node c_i as depicted in Figure 7(a).

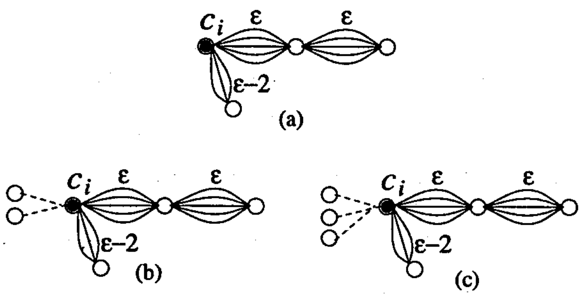


Figure 7: A clause component.

Fact: C_i has rank exactly 2ϵ. The rank of the graph obtained by attaching two simple edges to the node c_i remains 2ϵ. But the one formed by attaching three or more edges to c_i has rank $> 2\epsilon$ (see Figure 7(b) and (c)).

We connect $X_1, X_2, \ldots, X_n$ with $2^d - n$ dummy variable components $X_{n+1}, \ldots, X_{2^d}$ to form a chain, $G_1 = \mathcal{L}([X_1..X_{2^d}], b)$. Also, $C_1, C_2, \ldots, C_\ell$ are connected with $2^d - \ell$ dummy clause components $C_{\ell+1}, C_{\ell+2}, \ldots, C_{2^d}$ to form another chain, $G_2 = \mathcal{L}([C_1, \ldots, C_{2^d}], b)$. Inside each X_j or C_i, the internal edge multiplicity is at least $\epsilon - 2 = b + d$. As shown in Section 3.1, the rank of G_1 and G_2 is $db + \frac{d(d-1)}{2} + 2\epsilon$. Let t denote $db + \frac{d(d-1)}{2} + 2\epsilon$.

Finally, we connect G_1 and G_2 to form a multigraph G as follows: for each clause $c_i = (\ell_{i,1} + \ell_{i,2} + \ell_{i,3})$, we create a six-edge connector, as depicted in Figure 8, connecting the nodes of G_1 labeled with $\ell_{i,1}, \ell_{i,2}, \ell_{i,3}$ to the node c_i of G_2. The six edges are denoted by $r_{i,1}, r_{i,2}, r_{i,3}, r'_{i,1}, r'_{i,2}, r'_{i,3}$. Let H be the graph comprising all of the connectors between G_1 and G_2, and let G be the graph formed by connecting G_1 and G_2 with H. Figure 9 shows an example of constructing such a graph.

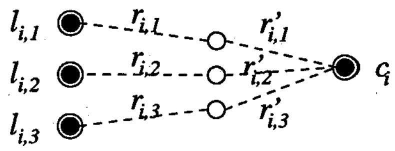


Figure 8: A six-edge connector.

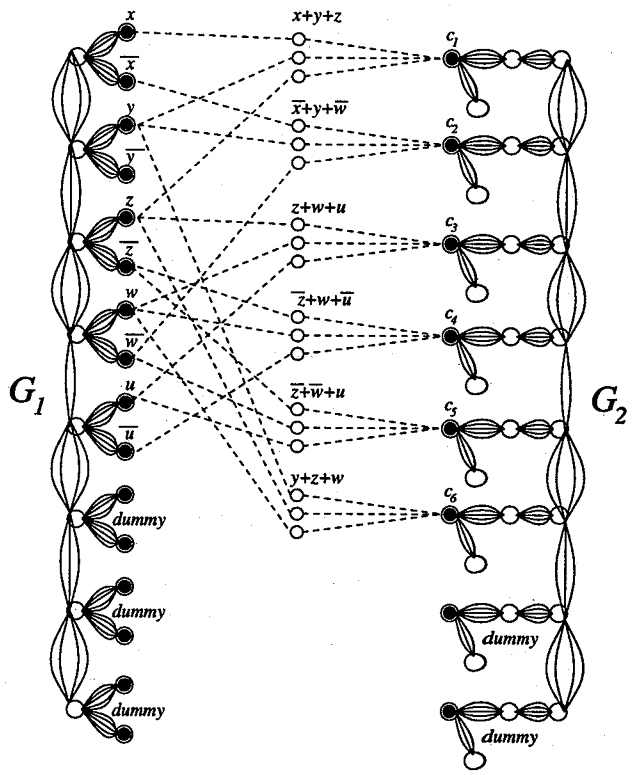


Figure 9: The graph G for the formula $F = (x + y + z)(\overline{x}+y+\overline{w})(z+w+u)(\overline{z}+w+\overline{u})(\overline{z}+\overline{w}+u)(y+z+w)$, where $b = 19, \epsilon = 24$, and $t = 108$.

The size of a minimum total cut of H is 3ℓ. The rank of G_1 or G_2 is at least t, and their internal edge multiplicity is at least $b > 3\ell$. By Lemma 3, *rank*$(G) \geq 3\ell + t$.

Lemma 8 *F is satisfiable if and only if rank$(G) = 3\ell + t$, where $t = db + \frac{d(d-1)}{2} + 2\epsilon$.*

($\Rightarrow$) Let A be a satisfiable truth assignment for F. W.L.O.G., assume the first literal $\ell_{i,1}$ in each clause is true under A. A ranking ψ of G using $3\ell + t$ distinct labels is constructed as follows: The primitive separator S of ψ consists of 3ℓ edges from H, namely

$\cup_{i=1}^{\ell}\{r_{i,2}, r_{i,3}, r'_{i,1}\}$. The removal of S from G decomposes G into two connected components $\widehat{G_1}$ and $\widehat{G_2}$, where $\widehat{G_1}$ contains G_1 and all the edges in $\cup_{i=1}^{\ell}\{r_{i,1}\}$, and $\widehat{G_2}$ consists of G_2 and the edges $\cup_{i=1}^{\ell}\{r'_{i,2}, r'_{i,3}\}$.

$\widehat{G_1}$ is still in the form of a chain. More precisely, $\widehat{G_1} = \mathcal{L}([\widehat{X_1}, \widehat{X_2}, \ldots, \widehat{X_{2^d}}], b)$ where each $\widehat{X_j}$ includes X_j and the edges $r_{i,1}$'s attached to the nodes x_j or $\overline{x_j}$. Note that an edge $r_{i,1}$ is attached to x_j if and only if x_j $(= l_{i,1})$ is true under A, and similarly for $\overline{x_j}$. Since either x_j or $\overline{x_j}$ is true under A, it is impossible to have edges attached to both vertices. Also, there are at most $\ell < \epsilon$ edges attached to either x_j or $\overline{x_j}$. The rank of each $\widehat{X_j}$ remains 2ϵ. Thus, we can rank $\widehat{G_1}$ using $t = db + \frac{d(d-1)}{2} + 2\epsilon$ distinct labels.

Similarly, $\widehat{G_2} = G([\widehat{C_1}, \widehat{C_2}, \ldots, \widehat{C_{2^d}}], b)$ where $\widehat{C_i}$ is formed by attaching two edges $r'_{i,2}, r'_{i,3}$ to the node c_i in C_i. Again, the rank of each $\widehat{C_i}$ is 2ϵ. Thus, we can rank $\widehat{G_2}$ using t distinct labels.

($\Leftarrow$) Suppose the rank of G is $3\ell + t$. By Lemma 4, G has an optimal edge ranking ψ such that its primitive separator S forms a minimum total cut of H. That is, S contains exactly 3ℓ edges from H and the removal of S from G disconnects G_1 from G_2. Then for each $i \in \{1, 2, \cdots, \ell\}$ and $k \in \{1, 2, 3\}$, S contains exactly one of the edges $r_{i,k}$ and $r'_{i,k}$.

Suppose S has been removed from G. Let $\widehat{G_1}$ and $\widehat{G_2}$ be the two connected components containing G_1 and G_2, respectively. Since $rank(G) = 3\ell + t$ and S contains exactly 3ℓ edges, the ranks of $\widehat{G_1}$ and $\widehat{G_2}$ are exactly t.

Let $\widehat{X_j}$ be the subgraph in $\widehat{G_1}$ consisting of X_j and the edges in $H - S$ attached to the nodes x_j and $\overline{x_j}$ of X_j. Since $rank(X_j) = 2\epsilon$, we have $rank(\widehat{X_j}) \geq 2\epsilon$. Moreover, $rank(\widehat{G_1}) = t = db + \frac{d(d-1)}{2} + 2\epsilon$. By Lemma 7, each $\widehat{X_j}$ can be ranked using 2ϵ distinct labels. Therefore, in each $\widehat{X_j}$, the edges inherited from $H - S$ can attach to the node x_j or the node $\overline{x_j}$, but not both.

A truth assignment for F is given as follows: For each variable x_j, if the subgraph $\widehat{X_j}$ gets at least one edge $r_{i,k}$ attached to the node x_j, the variable x_j in F is assigned the value true. Otherwise, x_j is given the value false.

Below, we explain why this truth assingment satisfies F. Let $\widehat{C_i}$ be the subgraph in $\widehat{G_2}$ including C_i and the edges in $H - S$ attached to the node c_i. Since $\widehat{G_2}$ can be ranked using $t = db + \frac{d(d-1)}{2} + 2\epsilon$ distinct labels, by Lemma 7, each $\widehat{C_i}$ has rank exactly 2ϵ. In each non-dummy $\widehat{C_i}$, at least one of the edges in $\{r'_{i,1}, r'_{i,2}, r'_{i,3}\}$ must be in the primitive separator S and not attached to the node c_i. Suppose $r'_{i,k} \in S$. Then $r_{i,k}$ is attached to a node labeled with $\ell_{i,k}$ in G_1, and the literal $\ell_{i,k}$ must have assigned true. In other words, in every clause $c_i = (\ell_{i,1} + \ell_{i,2} + \ell_{i,3})$ of F, at least one literal is getting a value true.

Theorem 9 The edge-ranking problem of multigraphs is NP-complete.

Proof: The edge-ranking problem of multigraphs is in NP because, given any multigraph G and any integer $t \geq 0$, we can guess a ranking ψ of G nondeterministically and verify the validity and optimality of ψ deterministically in polynomial time.

With the construction given at the beginning of this section, we have shown that finding an optimal edge ranking for a multigraph is NP-hard (Lemma 8). So the edge-ranking problem is NP-complete. □

5 Transformation to simple graphs

The edge-ranking problem of simple graphs is obviously in NP. In what follows, we prove that the edge-ranking problem for simple graphs is NP-hard, thus showing the latter is NP-complete.

Theorem 10 The edge-ranking problem of simple graphs is NP-complete.

The following describes a polynomial time reduction from the edge-ranking problem of multigraphs to that of simple graphs. For any graph $G = (V, E)$ that possibly contains parallel edges, let $V = \{v_1, v_2, \ldots, v_n\}$ and $E = \{e_1, e_2, \ldots, e_m\}$. Construct a simple graph $G' = (V', E')$ by replacing each vertex v_i in G with a clique K_i of $(m+2)$ vertices $\{v_{i,1}, v_{i,2}, \ldots, v_{i,m+2}\}$ and for each edge $e_\ell = (v_i, v_j) \in E$, putting an edge between the nodes $v_{i,\ell}$ and $v_{j,\ell}$ in G'. Formally, $V' = \{v_{i,\ell} \mid 1 \leq i \leq n, 1 \leq \ell \leq m+2\}$ and $E' = \{(v_{i,\ell}, v_{i,\ell'}) \mid 1 \leq i \leq n, 1 \leq \ell, \ell' \leq m+2\} \cup \{(v_{i,\ell}, v_{j,\ell}) \mid e_\ell = (v_i, v_j) \in E\}$.

Claim: $rank(G') = rank(G) + k$ where k is the rank of a clique of $m+2$ vertices.

Proof: ($\leq$) Let ψ and φ be the optimal edge rankings for G and a clique of $m+2$ vertices respectively. Construct an edge ranking η for G' such that, for each edge e in K_i, $\eta(e) = \varphi(e)$ and, for each $e_\ell = (v_i, v_j) \in E$, $\eta((v_{i,\ell}, v_{j,\ell})) = \psi(e_\ell) + k$. Then $rank(\eta) = rank(\psi) + k$. So $rank(G') \leq rank(G) + k$.

($\geq$) In what follows, we will construct an edge ranking of G from an optimal edge ranking of G' using at most $rank(G') - k$ distinct labels.

Let $N(K_i)$ be the set of inter-clique edges of G' with one of its endpoints in K_i. We show that if η is an optimal edge ranking of G' satisfying the minimal cut property then, for each K_i in G, every edge in K_i

gets a label smaller than all the labels on the edges in $N(K_i)$.

Let η be an optimal edge ranking of G' satisfying the minimal cut property. As shown before, $rank(G')$ is at most $rank(G) + k$. Also, $rank(G)$ is bounded by m, the number of edges in G. Thus, $rank(\eta) \leq m+k$. Suppose on the contrary that there exists an edge e of some K_i such that $\eta(e) \geq \eta(e')$ for some $e' \in N(K_i)$. Let e_0 be the one with the biggest label. Assume e_0 is in K_i and $\eta(e_0) > \eta(e'_0)$ where e'_0 connects K_i to another clique K_j. Note that e'_0 and the edges in K_i and K_j all receive labels not exceeding $\eta(e_0)$. Consider the graph formed by removing all the edges with label $> \eta(e_0)$ from G'. Obviously, e'_0 and all edges in K_i and K_j are not removed, and they appear in the same connected component, say, Q. Let η_Q be the labeling of Q inherited from η. Below, we show that η_Q uses at least $m+1+k$ distinct labels contradicting the fact that $rank(\eta) \leq m+k$.

Because η satisfies the minimal cut property, η_Q also satisfies this property.[1] We also observe that e_0 is the edge in Q having the biggest label. The primitive separator S_Q of η_Q must contain e_0. By the minimal cut property of η_Q, the removal of S_Q from Q must disconnect the endpoints of e_0. Since these two nodes are joined by $m+1$ edge disjoint paths in K_i, S_Q must contain least $m+1$ edges in K_i, each of which gets a distinct label. All other edges in Q have labels different from those on these $m+1$ edges. In particular, K_j contains at least k distinct labels different from them. So there are at least $m+1+k$ different labels in Q and a contradiction occurs. We conclude that for each K_i in G, every edge in K_i gets a label smaller than all the labels on the edges in $N(K_i)$. for each $1 \leq i \leq n$.

We are now ready to construct a ranking for G. With respect to an optimal edge ranking η of G', every K_i contains at least k distinct labels and each inter-clique edge has label $> k$ under η. Let ψ be an edge ranking of G such that for each $e_\ell = (v_i, v_j) \in E$, $\psi(e_\ell) = \eta((v_{i,\ell}, v_{v,\ell})) - k$. Therefore, $rank(\psi) \leq rank(\eta) - k$ and $rank(G) \leq rank(G') - k$. □

Bodlander *et al.* have given a close formula for the ranks of cliques in [1], from which the rank of a clique of $m+2$ vertices can be determined in $O(\log m)$ time. Note that the graph G' contains $O(nm)$ vertices and $O(nm^2)$ edges. So the reduction described above can be computed in polynomial time. This completes the proof of Theorem 10.

[1] In general, we can prove by induction on $\ell = rank(\eta), rank(\eta) - 1, \cdots, 1$ that if we delete all the edges of G' with labels $\geq \ell$, the labeling of each connected component inherited from η still satisfies the minimal cut property.

References

[1] H.L. Bodlaender, J.S. Deogun, K. Jansen, T. Kloks, D. Kratsch, H. Müller, and Zs. Tuza, Rankings of graphs, In *Proc. of International Workshop on Graph-Theoretic Concepts in Computer Science*, 1994, pages 292-304.

[2] H.L. Bodlaender, J.R. Gilbert, H. Hafsteinsson, and T. Kloks, Approximating treewidth, pathwidth, frontsize, and shortest elimination tree, *Journal of Algorithms*, 18 (1995), pages 238-255.

[3] P. de la Torre, R. Greenlaw, and A.A. Schäffer, Optimal Edge Ranking of Trees in Polynomial Time, In *Proc. of the 4th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 138-144, Austin, Texas, 1993.

[4] J.S. Deogun, T. Kloks, D. Kratsch, and H. Müller, On vertex ranking for permutation and other graphs, In *Proceedings of the 11th Annual Symposium on Theoretical Aspects of Computer Science*, Lecture Notes in Computer Science 775, Springer-Verlag, Berlin, 1994, pages 747-758.

[5] M.R. Garey and D.S. Johnson, Computers and Intractability, Freeman, 1979.

[6] I. Holyer, The NP-completeness of edge-coloring, *SIAM Journal of Computing*, 10:4 (1981), pages 718-720.

[7] A.V. Iyer, H.D. Ratliff, and G. Vijayan, Optimal node ranking of trees, *Information Processing Letters*, 28 (1988), pages 225-229.

[8] A.V. Iyer, H.D. Ratliff, and G. Vijayan, On an edge ranking problem of trees and graphs, *Discrete Applied Mathematics*, 30 (1991), pages 43-52.

[9] A. Pothen, The Complexity of Optimal Elimination Trees, *Technical Report CS-88-13*, The Pennsylvania State University, 1988.

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

[11] X. Zhou and T. Nishizeki, Finding optimal edge-rankings of trees, Technical reports 9501, Dept. of Inf. Eng., Tohoku Univ., 1995 (a preliminary version appeared in *Proc. of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, 1995, pages 122-131).