Edge ranking of graphs is hard

Tak Wah Lam\textsuperscript{a,}\textsuperscript{*}, Fung Ling Yue\textsuperscript{b}

\textsuperscript{a} Department of Computer Science, University of Hong Kong, Pokfulam Road, Hong Kong
\textsuperscript{b} Management Information Section, City University of Hong Kong, Kowloon, Hong Kong

Received 15 May 1997; received in revised form 22 December 1997; accepted 2 February 1998

Abstract

An edge ranking of a graph is a restricted coloring of the edges with integers. It requires 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. Recent research has revealed that the problem of finding an optimal edge ranking when restricted to trees admits a polynomial-time solution, yet the complexity of the problem for general graphs has remained open in the literature. In this paper, we prove that finding an optimal edge ranking of a graph is NP-hard. Also, we show that even finding a reasonably small edge ranking is infeasible in some cases. © 1998 Elsevier Science B.V. All rights reserved.

Keywords: NP-completeness; Graph labeling algorithms; Edge ranking; Computational complexity; Approximability

1. Introduction

Let \(G\) be an undirected graph. An \textit{edge ranking} of \(G\) is a labeling of its edges with integers such that every path between two edges with the same label \(i\) contains an intermediate edge with label \(j > i\). Obviously, an edge ranking of \(G\) is also an edge coloring of \(G\). We say that an edge ranking is \textit{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 [13, 17] of the graph. The problem of finding an optimal edge ranking was first studied by Iyer et al. [8] as they found the problem having an application in scheduling the assembly of multipart products.

Edge ranking has a vertex analogue: A \textit{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 studied intensively. In particular, Llewellyn et al. [11] and Pothen [14] independently proved that finding an optimal vertex ranking of graphs is NP-hard.

\* Corresponding author. E-mail: twlam@cs.hku.hk.

0166-218X/98/$19.00 © 1998 Elsevier Science B.V. All rights reserved.

\textit{P}II S0166-218X(98)00029-8
Fig. 1. An optimal edge ranking of a graph.

NP-hard. The problem remains NP-hard even when restricted to bipartite graphs [1]. On the other hand, Schäffer [15], improving the results of Iyer et al. [7] and Katchalski et al. [9], obtained a linear time algorithm for finding an optimal vertex ranking of a tree. There also exist polynomial time algorithms for different kinds of restricted graphs such as permutation graphs, interval graphs, and trapezoid graphs [4, 16]. Approximation algorithms for optimal vertex rankings can be found in [2, 9].

The complexity of finding an optimal edge ranking (Fig. 1) is relatively less understood. In the pioneering work of Iyer et al. [8], 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 vertex. Later, Zhou and Nishizeki [18, 19] showed that the running time can be improved to $O(n^2 \log \Delta)$, where $\Delta$ is the maximum degree. With respect to graphs, there is a general belief that finding an optimal ranking is NP-hard [1, 3, 19], yet there has not been any significant progress. In this paper, we settle the open problem by proving that, given a graph $G$ and an integer $t$, determining whether $G$ has an edge ranking using at most $t$ distinct labels is NP-complete.

There is 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 a relationship between edge-based and vertex-based graph problems also exists in the coloring problems – Vertex coloring was one of the first few problems known to be NP-complete [5, 12], yet it took another decade before the NP-completeness of edge coloring was revealed [6]. It is worth mentioning that the edge ranking problem does not resemble the edge coloring problem in various aspects. In particular, their “bounded” versions exhibit different complexity – Deciding whether a graph has an edge coloring with three colors is known to be NP-complete [6], while Bodlaender et al. [1] showed that the edge ranking problem when restricted to a fixed constant number of distinct labels can be solved in linear time.
With regard to the approximability of optimal edge ranking, Bodlaender et al. [2] showed that, unless \( P = NP \), no polynomial time algorithm \( X \) exists such that for any graph \( G \) with \( n \) vertices, \( X \) can compute a vertex ranking of \( G \) using at most \( n^{1-\varepsilon} \) labels in addition to the optimal, where \( \varepsilon \) is any constant greater than 0. In this paper, we present a similar non-approximability result for the edge ranking problem of graphs. We show that no polynomial time algorithm \( X \) exists such that for any graph \( G \) with \( m \) edges, \( X \) can compute an edge ranking of \( G \) using at most \( m^{1/2-\varepsilon} \) labels more than the optimal, where \( \varepsilon > 0 \).

We give the necessary notions in Section 2 and study the rank of a chain-like graph in Section 3. Then we prove the NP-completeness of edge ranking by first reducing the satisfiability problem to the edge ranking problem of multigraphs (see Section 4), and then transforming the latter to the edge ranking problem of simple graphs (see Section 5). Finally, we show in Section 6 that finding an edge ranking of a graph with \( m \) edges using \( o(m^{1/2}) \) labels more than the optimal is as difficult as computing an optimal solution, thus obtaining the above negative result on the approximability of optimal edge ranking.

2. Preliminary

Multigraphs and internal edge multiplicity: A multigraph is a graph in which a pair of vertices can be connected by one or more parallel edges. Since parallel edges between two vertices can form a path themselves, any edge ranking of a multigraph must assign different labels to all parallel edges between two vertices. Fig. 2 gives an optimal edge ranking of a multigraph. Let \( \psi \) be an edge ranking of a multigraph \( G \). We define the rank of \( \psi \), denoted \( \text{rank}(\psi) \), to be the number of distinct labels used by \( \psi \) and the rank of \( G \), denoted \( \text{rank}(G) \), to be the number of distinct labels used by an optimal edge ranking of \( G \).

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 does not deserve a big label since it cannot be the intermediate edge of any path between two edges. The edge multiplicity of an edge \( e = (u,v) \) is the

Fig. 2. An optimal edge ranking of a multigraph.
number of parallel edges connecting \( u \) and \( v \). The internal edge multiplicity of \( G \) is defined to be the minimum edge multiplicity over all its internal edges.

To simplify our NP-completeness argument, we will focus on multigraphs that are connected and contain at least one internal edges throughout the paper.

**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 \( C' \subseteq C \) 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 vertices \( u \) and \( v \) become disconnected after the removal of \( C \).

Let \( \psi \) be an edge ranking of \( G \). Consider the process of removing edges from \( G \) in the decreasing order of the labels given by \( \psi \). The edge with the biggest label under \( \psi \) 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. The set of edges removed in this process is called the primitive separator of \( \psi \) and denoted by \( S \). \( S \) is a cut of \( G \). Removing \( S \) breaks \( G \) into two connected components, say, \( G_1 \) and \( G_2 \). \( G_1 \) and \( G_2 \) are each ranked with labels \( \leq \text{rank}(\psi) - |S| \).

**Minimal cut and normal form:** An edge ranking \( \psi \) is said to be in normal form if its primitive separator is a minimal cut of \( G \) and contains no terminal edges. In this paper, we only consider rankings in normal form for any ranking can be transformed into normal form, while preserving the rank. Details follow.

Suppose the primitive separator \( S \) of an edge ranking \( \psi \) contains a terminal edge. Let \( e \) be a terminal edge in \( S \). Since removing \( e \) disconnects \( G \) immediately, \( e \) must be the only terminal edge in \( S \) and receive the smallest label among all edges in \( S \). We reassign the label of \( e \) to one and increase the labels on all edges outside \( S \) by one. Denote \( \psi' \) as the resultant ranking. Note that \( \text{rank}(\psi') = \text{rank}(\psi) \) and \( e \) no longer lies in the primitive separator of \( \psi' \). If the primitive separator of \( \psi' \) still contains a terminal edge, we repeat the relabeling process until we obtain a ranking with a primitive
separator containing no terminal edge. Note that the label of a terminal edge $e$, once reset to one, may increase gradually but is always less than the labels of internal edges. As the internal edges themselves already form a cut, such a terminal edge cannot appear subsequently in any primitive separators. Thus, the relabeling process stops eventually.

Furthermore, if the primitive separator $S$ of an edge ranking $\psi$ is not a minimal cut of $G$, we can transform $\psi$ as follows. Let $S' \subset S$ be a minimal cut. We shuffle the labels on the edges of $S$ so that the edges of $S'$ receive the biggest labels. This results in a ranking that has $S'$ as the primitive separator and uses the same number of distinct labels as $\psi$ (see Fig. 3).

3. The rank of a chain

In this section, we study the rank of a special chain-like graph $G$ that is formed by connecting a sequence of multigraphs $X_1,X_2,X_3,\ldots$ together. In each $X_i$, an internal vertex $x_i$ is designated for connection purpose. Two consecutive graphs $X_i$ and $X_{i+1}$ are connected by parallel edges between the vertices $x_i$ and $x_{i+1}$. The following is a definition of a chain of length $2d$, where $d$ is any integer. For any integer $b \geq 1$, define $L([X_1,X_{2d}],b)$ to be a graph formed by connecting the graphs $L([X_1,X_{2d-1}],b+1)$ and $L([X_{2d-1},X_{2d}],b+1)$ with $b$ parallel edges between the vertices $x_{2d-1}$ and $x_{2d+1}$. If $d = 0$, the sequence consists of a single graph and we define $L([X_1],b)$ to be $X_1$ itself. Let $G = L([X_1,X_{2d}],b)$. Inside $G$, the multiplicity of any connecting edge, $(x_i,x_{i+1})$, is in the range $[b..b+d-1]$. Fig. 4 gives a chain composed of eight $X_i$'s.

We will choose each $X_i$ in such a way that its internal edge multiplicity is greater than the multiplicity of any connecting edge along the chain, i.e., at least $b+d$. The rest of this section studies two lemmas relating the ranks of the $X_i$'s to that of $G$. These lemmas provide the basis for reducing the satisfiability problem to the edge ranking problem.

**Lemma 1.** Let $G = L([X_1,X_{2d}],b)$ where the rank of each $X_i$ is equal to $k$. Then $\text{rank}(G) \leq db + \frac{1}{2}d(d-1) + k$.

**Proof.** Each $X_i$ can be ranked with $k$ distinct labels. A sub-chain of length two, which takes the form of $L([X_{2i+1},X_{2i+2}],b+d-1)$, 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). A sub-chain of length four, which takes the form of $L([X_{4i+1},X_{4i+4}],b+d-1)$, can be

![Fig. 4. The graph $L([X_1,X_{2d}],b)$.](image-url)
ranked with \((b+d-2)+(b+d-1)+k\) distinct labels. In general, a sub-chain of length \(2^j\), where \(j = 0, 1, \ldots, d\), can be ranked with \(\sum_{\ell=1}^{j}(b+d-\ell)+k = jb+\sum_{\ell \leq j} (d-\ell)+k\) distinct labels. Thus, \(\text{rank}(G) \leq db + \frac{1}{2}d(d-1) + k\). \(\square\)

**Lemma 2.** Let \(G = \mathcal{L}([X_1, X_{2d}], b)\) where each \(X_i\) has internal edge multiplicity \(\geq b+d\) and rank equal to \(k\). Then \(\text{rank}(G) \geq db + \frac{1}{2}d(d-1) + k\). Moreover, if there exists some \(X_i\) with rank greater than \(k\) and the rest have rank equal to \(k\), \(\text{rank}(G) > db + \frac{1}{2}d(d-1) + k\).

**Proof.** \(G\) is a chain of length \(2^d\). For any \(j \in \{0, 1, \ldots, d\}\), define \(\mathcal{G}_j\) to be the set of consecutive sub-chains of \(G\) of length \(2^j\). For example, \(\mathcal{G}_0 = \{X_1, X_2,\ldots, X_{2d}\}\); \(\mathcal{G}_1 = \{\mathcal{L}([X_1, X_2], b+d-1), \mathcal{L}([X_3, X_4], b+d-1), \ldots, \mathcal{L}([X_{2d-1}, X_{2d}], b+d-1)\}\); \(\mathcal{G}_d = \{G\}\). For any integer \(j \geq 0\), define \(f(j) = jb + \sum_{\ell \leq j} (d-\ell)\). We claim that for any chain \(\mathcal{L} \in \mathcal{G}_j\) where \(j \in \{0, 1, \ldots, d\}\), \(\text{rank}(\mathcal{L}) \geq f(j) + k\); moreover, if \(\mathcal{L}\) contains some \(X_i\) with rank greater than \(k\), \(\text{rank}(\mathcal{L}) > f(j) + k\).

Lemma 2 is an immediate consequence of the claim. The proof of the claim is by induction on \(j\).

**Basis** \((j = 0)\). Any chain \(\mathcal{L} \in \mathcal{G}_0\) consists of a single graph \(X_i\). If \(\text{rank}(X_i) > k\) then \(\text{rank}(\mathcal{L}) > k\); if \(\text{rank}(X_i) = k\) then \(\text{rank}(\mathcal{L}) = k\).

**Induction.** Suppose the claim holds for any chain in \(\mathcal{G}_j\) where \(j \geq 0\). We consider the case for \(j+1\). Every chain \(\mathcal{L} \in \mathcal{G}_{j+1}\) is formed by joining two chains \(\mathcal{L}_1\) and \(\mathcal{L}_2\) in \(\mathcal{G}_j\) by \(b+d-(j+1)\) parallel edges. Let \(\psi\) be an optimal edge ranking of \(\mathcal{L}\) in normal form, and let \(S\) denote the primitive separator of \(\psi\). Note that \(\text{rank}(\mathcal{L}) = \text{rank}(\psi)\). There are two cases to consider.

**Case 1**—\(S\) contains solely the parallel edges between \(\mathcal{L}_1\) and \(\mathcal{L}_2\). In this case, \(\text{rank}(\psi) = |S| + \max\{\text{rank}(\mathcal{L}_1), \text{rank}(\mathcal{L}_2)\}\). If every \(X_i\) in \(\mathcal{L}\) has rank equal to \(k\), then, by the induction hypothesis, \(\max\{\text{rank}(\mathcal{L}_1), \text{rank}(\mathcal{L}_2)\} \geq f(j) + k\), and \(\text{rank}(\psi) \geq |S| + f(j) + k = b + d - (j+1) + f(j) + k = f(j+1) + k\). If \(\mathcal{L}\) contains an \(X_i\) with rank greater than \(k\), then either \(\mathcal{L}_1\) or \(\mathcal{L}_2\) contains \(X_i\). By the induction hypothesis, \(\max\{\text{rank}(\mathcal{L}_1), \text{rank}(\mathcal{L}_2)\} > f(j)+k\). Thus, \(\text{rank}(\psi) > |S| + f(j)+k = f(j+1)+k\).

**Case 2**—\(S\) contains some internal edge \(e\) of \(\mathcal{L}_1\) (or \(\mathcal{L}_2\)). Let \(E'\) be the set of all parallel edges joining the endpoints of \(e\). Since \(S\) is a minimal cut of \(\mathcal{L}\), \(S\) includes all edges in \(E'\). With respect to \(\psi\), the labels of \(E'\) are different from the labels of all other edges in \(\mathcal{L}\) and, in particular, those in \(\mathcal{L}_2\). Since \(\text{rank}(\mathcal{L}_2) \geq f(j)+k\), we have \(\text{rank}(\psi) \geq |E'| + f(j) + k\). The internal edge multiplicity of \(\mathcal{L}_1\) and \(\mathcal{L}_2\) are at least \(b + d - j\). So \(|E'| \geq b + d - j\) and \(\text{rank}(\psi) \geq b + d - j + f(j) + k > f(j+1) + k\).

We have completed the induction, thus proving Lemma 2. \(\square\)

**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.
Theorem 3. The edge ranking problem of multigraphs is NP-complete.

To prove Theorem 3, we show how to transform a Boolean formula $F$ to a multigraph $G$ and a value $t$ in polynomial time such that $F$ is satisfiable if and only if $\text{rank}(G) = t$.

Suppose $F$ consists of $n$ variables $\{x_1, x_2, \ldots, x_n\}$ and $\ell$ clauses $\{c_1, c_2, \ldots, c_{\ell}\}$ where $c_i = (\ell_{1,i} + \ell_{2,i} + \ell_{3,i})$. The definition of $G$ is based on several parameters defined by $F$. Let $d$ be the smallest integer such that $2^d > \max\{n, \ell\}$. Let $b = 3\ell + 1$ and $\varepsilon = b + d + 2$.

For each variable $x_i$, we construct a variable component $X_i$ with edge multiplicity $\varepsilon$ and two designated vertices $x_j$ and $\overline{x}_j$ as depicted in Fig. 5(a). The rank of $X_i$ is exactly $2\varepsilon$. If we attach up to $\varepsilon$ simple edges to either $x_j$ or $\overline{x}_j$, the resultant graph still has rank $2\varepsilon$. However, if edges are attached to both $x_j$ and $\overline{x}_j$, the resultant graph has rank $> 2\varepsilon$ (see Fig. 5(b) and (c)).

For each clause $c_i$, we construct a clause component $C_i$ consisting of two edges with multiplicity $\varepsilon$ and one edge with multiplicity $\varepsilon - 2$ joined at the vertex $c_i$ as depicted in Fig. 6(a). $C_i$ has rank exactly $2\varepsilon$. If two simple edges are attached to the vertex $c_i$, the rank of the resultant graph remains $2\varepsilon$. But attaching three or more edges to $c_i$ will cause the rank to exceed $2\varepsilon$ (see Fig. 6(b) and (c)).

The variable components and the clause components are respectively connected together to form two chain-like graphs: $X_1, X_2, \ldots, X_n$ are connected with $2^d - n$ dummy variable components $X_{n+1}, \ldots, X_{2^d}$ to form a chain $G_1 = \mathcal{L}([X_1, X_{2^d}], b)$; $C_1, C_2, \ldots, C_{\ell}$ 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
Fig. 7. A six-edge connector.

is at least $\varepsilon - 2 = b + d$. By Lemmas 1 and 2, we can easily derive the following bounds.

**Proposition 4.** $\text{rank}(G_1) = \text{rank}(G_2) = db + \frac{1}{2}d(d - 1) + 2a$.

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 Fig. 7, connecting the vertices of $G_1$ labeled with $\ell_{i,1}, \ell_{i,2}, \ell_{i,3}$ to the vertex $c_i$ of $G_2$. The six edges are denoted by $r_{i,1}, r_{i,2}, r_{i,3}, r'_{i,1}, r'_{i,2}, r'_{i,3}$. Fig. 8 shows an example of $G$.

We are now ready to prove the correctness of the reduction.

**Lemma 5.** $F$ is satisfiable if and only if $\text{rank}(G) \leq 3\ell + q$ distinct labels, where $q = db + \frac{1}{2}d(d - 1) + 2a$.

**Proof.** (The “only if” direction) Let $A$ be a satisfiable truth assignment for $F$. Without loss of generality, assume the first literal $\ell_{i,1}$ in each clause is true under $A$. A ranking $\psi$ of $G$ using $3\ell + q$ distinct labels can be constructed as follows. The primitive separator $S$ of $\psi$ consists of $3\ell$ edges, 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 $G_1$ and $G_2$, where $G_1$ contains $G_1$ and all the edges in $\cup_{i=1}^{\ell} \{r_{i,1}\}$, and $G_2$ consists of $G_2$ and the edges $\cup_{i=1}^{\ell} \{r'_{i,2}, r'_{i,3}\}$. It remains to prove that $G_1$ and $G_2$ can each be ranked using at most $q$ labels.

$G_1$ is still in the form of a chain. More precisely, $G_1 = G[\{X_j, \bar{X}_j, \ldots, X_{2\ell}, \bar{X}_{2\ell}\}, b]$ where each $X_j$ includes $X_j$ and the edges $r_{i,1}$'s attached to the vertices $x_j$ or $\bar{x}_j$. 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 $\bar{x}_j$. Since either $x_j$ or $\bar{x}_j$ is true under $A$, it is impossible to have edges attached to both $x_j$ and $\bar{x}_j$. There are at most $\ell' < \varepsilon$ edges attached to either $x_j$ or $\bar{x}_j$. The rank of each $X_j$ remains $2a$. By Lemma 1, we can rank $G_1$ using at most $db + \frac{d(d-1)}{2} + 2a$ distinct labels.

Similarly, $G_2 = G[\{C_1, C_2, \ldots, C_{2\ell}\}, b]$ where $C_j$ is formed by attaching two edges $r'_{i,2}, r'_{i,3}$ to the vertex $c_i$ in $C_j$. Again, the rank of each $C_j$ is $2a$. Thus, we can rank $G_2$ using $q$ distinct labels.

(The “if” direction) Suppose $\text{rank}(G) \leq 3\ell + q$. Let $\psi$ be an optimal edge ranking of $G$ in normal form. Denote $H$ as the set of edges between $G_1$ and $G_2$ (i.e., edges due to the connectors). We will prove in Lemma 6 that the primitive separator $S$ of $\psi$ contains solely the edges of $H$ and the removal of $S$ from $G$ disconnects $G_1$ from $G_2$. Then, $S$, being a minimal cut of $G$, contains exactly $3\ell$ edges of $H$, namely, for each $i \in \{1, 2, \ldots, \ell\}$ and $k \in \{1, 2, 3\}$, either $r_{i,k}$ and $r'_{i,k}$. 


Fig. 8. The graph $G$ for the formula $F = (x + y + z)(x + y + w)(x + w + u)(y + z + w)$.
In this case, $b = 19, c = 24,$ and $q = 108$.

Suppose $S$ has been removed from $G$. Let $\hat{G}_1$ and $\hat{G}_2$ be the two connected components containing $G_1$ and $G_2$, respectively. Since $\text{rank}(G) \leq 3t + q$ and $S$ contains exactly $3t$ edges, the ranks of $\hat{G}_1$ and $\hat{G}_2$ are $\leq q$. Let $\hat{X}_j$ be the subgraph in $\hat{G}_1$ consisting of $X_j$ and the edges in $H - S$ attached to the vertices $x_j$ or $\bar{x}_j$ of $X_j$. Since $\text{rank}(X_j) = 2s$, we have $\text{rank}(\hat{X}_j) \geq 2s$. Based on the fact that $\text{rank}(\hat{G}_1) \leq db + \frac{1}{2}d(d - 1) + 2c$, we can apply Lemma 2 to deduce that there is no $\hat{X}_j$ with rank greater than $2s$, or equivalently, every $\hat{X}_j$ can be ranked using $2s$ distinct labels. Therefore, in each $\hat{X}_j$, the edges inherited from $H - S$ are attached to the vertex $x_j$ or the vertex $\bar{x}_j$, but not both.

A truth assignment for $F$ is given as follows: For each variable $x_j$, if the subgraph $\hat{X}_j$ gets at least one edge $r_{j,k}$ attached to the vertex $x_j$, we assign true to the variable $x_j$; otherwise, we assign false to $x_j$. Below, we explain why this truth assignment satisfies $F$. Let $\hat{C}_i$ be the subgraph in $\hat{G}_2$ including $C_i$ and the edges in $H - S$ attached to the vertex $c_i$. Again, since $\hat{G}_2$ can be ranked using $q = db + \frac{1}{2}d(d - 1) + 2c$ distinct
labels, each \( \tilde{C}_i \) can be ranked using 2\( \ell \) distinct labels. In each non-dummy \( \tilde{C}_i \), at least one of the edges in \( \{r_{i,1}', r_{i,2}', r_{i,3}'\} \) must be in the primitive separator \( S \) and is not attached to the vertex \( c_i \). Let \( r_{i,k}' \) be such an edge. Then \( r_{i,k}' \) is attached to a vertex labeled with \( \ell_{i,k} \) in \( G_1 \), and the literal \( \ell_{i,k} \) must be true. In other words, in every clause \( c_i = (\ell_{i,1} + \ell_{i,2} + \ell_{i,3}) \), at least one literal is true; thus, \( F \) is satisfiable. \( \square \)

Next, we prove the required properties of the primitive separator \( S \).

**Lemma 6.** The primitive separator \( S \) of \( \psi \) contains only the edges of \( H \) and the removal of \( S \) from \( G \) disconnects \( G_1 \) from \( G_2 \).

**Proof.** Suppose, by way of contradiction, that either \( S \not\subseteq H \) or \( S \subseteq H \) but the removal of \( S \) from \( G \) does not disconnect \( G_1 \) from \( G_2 \). Below, we prove that in either case, \( \psi \) must use more than \( 3\ell + q \) distinct labels. This contradicts the fact that \( \psi \) is optimal and \( \text{rank}(\psi) \leq 3\ell + q \).

**Case 1:** \( S \not\subseteq H \). Without loss of generality, suppose \( S \) contains an edge \( e \) of \( G_1 \). As \( \psi \) is in normal form, \( e \) is an internal edge. Let \( E' \) be the set of all parallel edges joining the endpoints of \( e \). Since the internal edge multiplicity of \( G_1 \) is \( b \), i.e., \( 3\ell + 1 \), we have \( |E'| \geq 3\ell + 1 \). On the other hand, \( E' \subseteq S \) (otherwise, \( S \) is not a minimal cut of \( G \)). With respect to \( \psi \), the labels of \( E' \) are all different from the labels of all other edges in \( G \). In conclusion, \( \text{rank}(\psi) \geq |E'| + \text{rank}(G_2) \geq 3\ell + 1 + db + \frac{1}{2}d(d - 1) + 2\varepsilon > 3\ell + q \).

**Case 2:** \( S \subseteq H \) but the removal of \( S \) from \( G \) does not disconnect \( G_1 \) from \( G_2 \). In this case, \( S \) contains exactly two edges connected to a vertex between \( G_1 \) and \( G_2 \). Suppose \( S \) is removed from \( G \). Let \( G' \) be the subgraph containing \( G_1 \) and \( G_2 \). Let \( \psi' \) be the ranking of \( G' \) inherited from \( \psi \). Assume that \( \psi' \) is in normal form (if it is not, transform it). Let \( S_1 \) be the primitive separator of \( \psi' \).

- If \( S_1 \) is not a subset of \( H \), we can use the argument in Case 1 to show that \( \text{rank}(\psi') > 3\ell + q \). Then, \( \text{rank}(\psi) = |S| + \text{rank}(\psi') > 2 + 3\ell + q \).
- If \( S_1 \) is a subset of \( H \) and removing \( S_1 \) from \( G' \) disconnects \( G_1 \) from \( G_2 \), \( S_1 \) contains at least \( 3\ell - 1 \) edges and \( \psi' \) uses at least \( |S_1| + \text{rank}(G_1) \geq 3\ell - 1 + q \) distinct labels. It follows that \( \text{rank}(\psi) = |S| + \text{rank}(\psi') \geq 2 + 3\ell - 1 + q > 3\ell + q \).
- Suppose \( S_1 \) is a subset of \( H \) and removing \( S_1 \) from \( H \) still leaves a subgraph connecting \( G_1 \) and \( G_2 \). Then, \( S_1 \), like \( S \), contains two edges only. We can repeat the argument above to further extract cuts \( S_2, \ldots, S_e \) from \( H \) until either \( S_e \not\subseteq H \) or removing \( S_1 \cup S_2 \cup \ldots \cup S_e \) from \( G' \) disconnects \( G_1 \) and \( G_2 \). In either case, we can again argue that \( \text{rank}(\psi) > 3\ell + q \). \( \square \)

5. Transformation to simple graphs

The edge ranking problem of simple graphs is obviously in NP. In what follows, we give a polynomial time reduction from the edge-ranking problem of multigraphs to that of simple graphs, thus showing the latter is NP-complete.
Theorem 7. The edge ranking problem of simple graphs is NP-complete.

The way we reduce a multigraph to a simple graph makes use of a technique called the clique graph transformation. Let $G = (V, E)$ be a multigraph. Suppose $V = \{v_1, v_2, \ldots, v_n\}$ and $E = \{e_1, e_2, \ldots, e_m\}$. The clique graph transformation of $G$ is a simple graph $\Pi(G) = (V', E')$ formed 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 $e_r = (v_i, v_j) \in E$, putting an edge between the vertices $v_{i, r}$ and $v_{j, r}$ in $G'$. Fig. 9 gives an example. Below, we show that the rank of $G$ is bounded by an integer $t$ if and only if the rank of $\Pi(G)$ is bounded by a function of $t$.

Lemma 8. For any multigraph $G$ with $m$ edges, $\text{rank}(\Pi(G)) = \text{rank}(G) + \alpha$ where $\alpha$ is the rank of a clique of $m + 2$ vertices.

Proof. (Part I: $\text{rank}(\Pi(G)) \leq \text{rank}(G) + \alpha$) Let $\psi$ and $\varphi$ be optimal edge rankings of $G$ and a clique of $m + 2$ vertices, respectively. Note that $\text{rank}(\psi) = \text{rank}(G)$ and $\text{rank}(\varphi) = \alpha$. Construct an edge ranking $\eta$ for $\Pi(G)$ such that, every clique $K_i$ in $\Pi(G)$ is labeled according to $\varphi$ and every inter-clique edge, say, $(v_{i, r}, v_{j, r})$, receives a label equal to the label of $(v_i, v_j)$ of $G$ under $\psi$ plus $\alpha$. Clearly, $\text{rank}(\eta) = \text{rank}(\psi) + \text{rank}(\varphi)$. Thus, $\text{rank}(\Pi(G)) \leq \text{rank}(G) + \alpha$.

(Part II: $\text{rank}(\Pi(G)) \geq \text{rank}(G) + \alpha$) Let $H$ be any vertex induced and connected subgraph of $G$. Define $\Pi(G)|_H$ to be the vertex induced subgraph of $\Pi(G)$ comprising all the cliques $K_i$ where $v_i$ is in $H$. We will prove, by induction on the number of vertices in $H$, that $\text{rank}(\Pi(G)|_H) \geq \text{rank}(H) + \alpha$. It follows that $\text{rank}(\Pi(G)) \geq \text{rank}(G) + \alpha$. Below, we denote by $h$ the number of vertices in $H$.

Basis ($h = 1$). $H$ consists of only one vertex and has rank zero. The graph $\Pi(G)|_H$ is exactly a clique of $m + 2$ vertices and has rank $\alpha$. So $\text{rank}(\Pi(G)|_H) = \alpha = \text{rank}(H) + \alpha$.

Induction ($h > 1$). Let $H$ be a subgraph of $G$ with $h$ vertices. Let $\eta$ be an optimal edge ranking of $\Pi(G)|_H$, of which the primitive separator $S$ forms a minimal cut of $\Pi(G)|_H$.

First of all, we will prove that $S$ contains inter-clique edges only. For the purpose of contradiction, suppose $S$ contains an edge $e$ in some clique $K_i$ of $\Pi(G)|_H$. As
$S$ is a minimal cut, the removal of $S$ from $\Pi(G)_H$ disconnects the endpoints of $e$. Since these two endpoints are joined by $m + 1$ edge disjoint paths inside $K_i$, $S$ must contain $m + 1$ edges inside $K_i$. Each of these edges gets a distinct label. Consider any clique $K_j$ of $\Pi(G)_H$ where $j \neq i$. The labels in $K_j$ are all different from those on the $m + 1$ edges just mentioned. Note that $K_j$ requires at least $x$ distinct labels. Therefore, there are at least $m + 1 + x$ distinct labels on the edges of $\Pi(G)_H$, or equivalently, $\text{rank}(\Pi(G)_H) \geq m + 1 + x$. To observe the contradiction, we need to recall the result in Part I that $\text{rank}(\Pi(G)) \leq \text{rank}(G) + x$, which implies $\text{rank}(\eta) \leq m + x$. Therefore, $S$ cannot contain any edge inside a clique.

If we remove $S$ from $\Pi(G)_H$, all edges inside the cliques remain and $\Pi(G)_H$ is decomposed into two connected components in the form of $\Pi(G)_H'$ and $\Pi(G)_H''$, for some vertex induced subgraphs $H'$ and $H''$ of $H$. $H$ has $h$ vertices and $H'$ and $H''$ each must have less than $h$ vertices. We have

$$
\text{rank}(\Pi(G)_H) = \text{rank}(\eta) = |S| + \max\{\text{rank}(\Pi(G)_H'), \text{rank}(\Pi(G)_H'')\}
\geq |S| + \max\{\text{rank}(H') + x, \text{rank}(H'') + x\}
\quad \text{(by induction hypothesis)}
\geq \text{rank}(H) + x.
$$

Combining the results in Parts I and II, we have proved Lemma 8. □

Given any multigraph $G$ and integer $t > 0$, let $G'$ be the clique transformation of $G$. Suppose $G$ has $m$ edges. Then by Lemma 8, $\text{rank}(G) \leq t$ if and only if $\text{rank}(G') \leq t + x$ where $x$ is the rank of the clique with $m+2$ vertices. Bodlaender et al. [1] have showed that the rank of a clique of $n$ vertices is $\frac{1}{2}(n^2 + g(n))$ where $g(n)$ is defined recursively as follows: $g(1) = 1$, $g(2n) = g(n)$ and $g(2n + 1) = g(n + 1) + n$. From this formula, 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. The reduction described above can be computed in polynomial time. This completes the proof of Theorem 7.

6. Approximability

In this section, we show that computing an edge ranking of a graph with $m$ edges within an additive error of $m^{1-\varepsilon}$ for any $\varepsilon > 0$ is as difficult as finding an optimal solution of the graph. To ease our discussion, we define the following notion.

Let $G$ be any simple graph (which contains zero or more internal edges). For any integer $k \geq 2$, define $G^K$ to be a multigraph with the same vertex set as $G$. For every edge $e$ in $G$, there are $K$ parallel edges connecting the endpoints of $e$ in $G^K$. Note that $G^K$ does not contain any terminal edge. See Fig. 10 for an example.

**Lemma 9.** $\text{rank}(G^K) = K \cdot \text{rank}(G)$. 
Proof. (** ) Let $\psi$ be an optimal edge ranking of $G$. We construct a ranking of $G^K$ using $K \cdot \text{rank}(G)$ labels as follows: For each edge $e$ in $G$, the $K$ parallel edges connecting the endpoints of $e$ in $G^K$ are assigned with the labels $(\psi(e) - 1)K + 1, (\psi(e) - 1)K + 2, \ldots, (\psi(e) - 1)K + K$. So $\text{rank}(G^K) \leq K \cdot \text{rank}(G)$.

(\geq ) We prove by induction on the number of edges $m$ in $G$ that $\text{rank}(G) \leq \frac{1}{K} \cdot \text{rank}(G^K)$.

Basis ($m = 1$). In this case, $\text{rank}(G) = 1$ and $\text{rank}(G^K) = K$. The lemma thus follows.

Induction ($m > 1$). Suppose $G$ contains $m > 1$ edges. Let $\psi$ be an optimal edge ranking of $G^K$ such that its primitive separator $S$ forms a minimal cut of $G^K$. Then, for every edge $(u, v)$ in $S$, all $K$ parallel edges connecting $u$ and $v$ in $G^K$ must be in $S$. Removing $S$ from $G^K$ decomposes it into two connected subgraphs which take the form of $G^K_1$ and $G^K_2$, for some subgraphs $G_1$ and $G_2$ of $G$.

Let $S'$ be the set of edges $(u, v)$ in $G$ such that $(u, v)$ is in $S$. $|S'| = \frac{1}{K} |S|$. We construct an edge ranking $\psi'$ of $G$ as follows. The primitive separator of $\psi'$ contains all the edges in $S'$. Removing $S'$ from $G$ disconnects $G$ into two connected components that are exactly $G_1$ and $G_2$, respectively. By the induction hypothesis, $G_1$ and $G_2$ can be ranked using $\left(\frac{1}{K}\right)\text{rank}(G^K_1)$ and $\left(\frac{1}{K}\right)\text{rank}(G^K_2)$ distinct labels, respectively. Thus,

$$
\text{rank}(\psi') \leq \frac{1}{K} |S'| + \max\left(\frac{1}{K} \text{rank}(G^K_1), \frac{1}{K} \text{rank}(G^K_2)\right)$$

$$
= \frac{1}{K} \left(|S'| + \max(\text{rank}(G^K_1), \text{rank}(G^K_2))\right)$$

$$
= \frac{1}{K} \text{rank}(\psi) = \frac{1}{K} \text{rank}(G^K).
$$

Therefore, $\text{rank}(G) \leq \text{rank}(\psi') \leq \frac{1}{K} \text{rank}(G^K)$. The induction proof is completed.

Definition. Let $AA$ be a polynomial time approximation algorithm for computing an edge ranking of a graph. For any graph $G$, denote by $\text{rank}_{AA}(G)$ the number of distinct labels used by the edge ranking computed by $AA$ on the graph $G$.

Theorem 10. If $P \neq \text{NP}$, then no polynomial time approximation algorithm $AA$ exists such that for any graph $G$ with $m$ edges, $\text{rank}_{AA}(G) - \text{rank}(G) \leq m^{(\frac{1}{2})-\epsilon}$, where $\epsilon$ is a positive constant.
The way we prove this impossibility result is as follows: Suppose on the contrary that such an approximation algorithm AA exists, we show that the edge ranking problem of simple graphs can be solved in polynomial time using AA. This contradicts the NP-completeness result obtained in Section 5. More precisely, given any simple graph $G$ and integer $t \geq 0$, we show how to construct another graph $G'$ and an integer $t'$ such that $\text{rank}(G) \leq t$ if and only if $\text{rank}_{\text{AA}}(G') < t'$.

Recall that $\text{rank}_{\text{AA}}(G')$ can be any value between $\text{rank}(G')$ and $\text{rank}(G') + m^{\frac{1}{2}-\epsilon}$, where $\epsilon$ is a positive constant. $G'$ is constructed in such a way that the possible values of $\text{rank}_{\text{AA}}(G')$ corresponding to the cases where $\text{rank}(G) \leq t$ and $\text{rank}(G) > t$ do not overlap, thus we can use AA to determine whether $\text{rank}(G) \leq t$ or not. In our construction of $G'$, we first construct the multigraph $G^K$ for some big enough integer $K$ (the value of $K$ will be specified later). We then transform this multigraph to a simple graph via the clique graph transformation, giving the graph $G'$. Let $n$ and $m$ be the number of vertices and edges in $G$, respectively. Then $G^K$ has $mK$ edges. By Lemma 8, $\text{rank}(G') = \text{rank}(G^{K+1}) + \alpha$, where $\alpha$ is the rank of a clique of $mK + 2$ vertices. Fig. 11 summarizes the construction of $G'$.

The value of $K$ is chosen to have a value strictly bigger than $(m')^{\frac{1}{2}-\epsilon}$ where $m'$ is the number of edges in the graph $G'$. Since $m' = mK + n\left(\frac{mK + 2}{2}\right)$, $m' \leq 4nm^2K^2$.

Note that $K > (4nm^2K^2)^{\frac{1}{2}-\epsilon}$ if and only if $K > (4nm^2)^{1-2\epsilon/4\epsilon}$. The value of $K$ is chosen to be the smallest integer bigger than $(4nm^2)^{1-2\epsilon/4\epsilon}$. Then the possible values of $\text{rank}_{\text{AA}}(G')$ corresponding to the cases $\text{rank}(G) \leq t$ and $\text{rank}(G) > t$ do not overlap (see Lemma 11 below).

**Lemma 11.** $\text{rank}(G) \leq t$ if and only if $\text{rank}_{\text{AA}}(G') < \alpha + Kt + K$.

**Proof.** If $\text{rank}(G) \geq t + 1$, then $\text{rank}_{\text{AA}}(G') \geq \text{rank}(G') \geq \alpha + K(t + 1) = \alpha + Kt + K$.

If $\text{rank}(G) \leq t$, then $\text{rank}(G') \leq \alpha + Kt$. By assumption, AA computes an edge ranking of $G'$ using at most $(m')^{\frac{1}{2}-\epsilon}$ labels more than the optimal, where by the choice of $K$, $(m')^{\frac{1}{2}-\epsilon} < K$. So $\text{rank}_{\text{AA}}(G') < \text{rank}(G') + K \leq \alpha + Kt + K$. \qed

Lemma 11 implies that computing an edge ranking of a graph with $m$ edges using at most $(m')^{\frac{1}{2}-\epsilon}$ distinct labels more than the optimal is no easier than computing the optimal edge ranking, thus proving the former being NP-hard. This completes the proof of Theorem 10.
7. Remarks

The vertex ranking and edge ranking problems are naturally defined in the context of undirected graphs. Very recently, Kratochvíl and Tuza [10] have also studied a directed variant of vertex ranking and proved that deciding whether a directed graph has a vertex ranking using at most a constant number of distinct labels is NP-complete. The complexity of directed edge ranking is left open, however.

Based on the algorithms in [1, 2], we can derive a polynomial-time approximation algorithm for the edge ranking problem such that the number of distinct labels used is $O(\log^2 m)$ times of the optimal. The idea is quite simple. We first transform the edge ranking problem to the vertex ranking problem via the reduction given in [1], and then solve the latter with the approximation algorithm for finding a vertex ranking given by Bodlaender et al. [2]. It is interesting to know whether an approximation algorithm with a constant approximation ratio exists or not.

References
