Information and Computation 206 (2008) 1288–1302

Contents lists available at ScienceDirect

Information and Computation

journal homepage: www.elsevier.com/locate/ic

Optimal vertex ranking of block graphs[☆]

Ruo-Wei Hung

Department of Computer Science and Information Engineering, Chaoyang University of Technology, Wufong, Taichung 41349, Taiwan

ARTICLE INFO

Article history:
Received 8 April 2007
Revised 3 August 2008
Available online 28 August 2008

Keywords:
Graph algorithms
Vertex ranking
Edge ranking
Trees
Block graphs

ABSTRACT

A vertex ranking of an undirected graph G is a labeling of the vertices of G with integers such that every path connecting two vertices with the same label i contains an intermediate vertex with label $j > i$. A vertex ranking of G is called optimal if it uses the minimum number of distinct labels among all possible vertex rankings. The problem of finding an optimal vertex ranking for general graphs is NP-hard, and NP-hard even for chordal graphs which form a superclass of block graphs. In this paper, we present the first polynomial algorithm which runs in $O(n^2 \log \Delta)$ time for finding an optimal vertex ranking of a block graph G, where n and Δ denote the number of vertices and the maximum degree of G, respectively.

1. Introduction

All graphs considered in this paper are finite and undirected, without loops or multiple edges. Let $G = (V,E)$ be a graph with vertex set V and edge set E. Throughout this paper, let m and n denote the numbers of edges and vertices of G, respectively, and let Δ denote the maximum degree of G. A *vertex ranking* of G is a labeling of vertices using positive integers $1, 2, \ldots, t$ such that all paths connecting two vertices with the same label i contain an intermediate vertex with label $j > i$. The integer labels $1, 2, \ldots, t$ are called *ranks*. The minimum number of ranks used for a vertex ranking of G is called the *vertex-ranking number* of G, and denoted by $\chi_r(G)$. We say that a vertex ranking is *optimal* if it uses $\chi_r(G)$ ranks. The vertex ranking problem, also called the *ordered coloring* problem [12], is to find an optimal vertex ranking of a graph. An edge ranking of G is a labeling of its edges satisfying an analogous condition, i.e., all paths between two edges with the same label i contain an intermediate edge with label $j > i$. The vertex ranking problem is equivalent to the problem of finding the *minimum-height elimination tree* of a graph [2,4,6]. And the edge ranking problem is equivalent to the problem of finding the *minimum-height edge-separator tree* of a graph [16,24]. The vertex ranking problem has applications in VLSI layout and in scheduling the parallel assembly of a complex multi-part product from its components [10,19,24]. For computing Cholesky factorizations of matrices in parallel, the vertex ranking problem plays an important role [6,17]. Finding an optimal edge ranking has an interesting application that schedules the assembly steps in manufacturing a complex multi-part product [11,19].

The vertex ranking problem is NP-hard for general graphs [18,20] and even for cobipartite, bipartite graphs [3] and chordal graphs [8]. However, polynomial-time algorithms exist for cographs [23], AT-free graphs [13], trapezoid graphs, permutation graphs, interval graphs and circular-arc graphs [7]. The edge ranking problem is NP-hard in general [14].

With respect to trees, Iyer et al. proposed an $O(n \log n)$-time algorithm to solve the vertex ranking problem [10]. Schäffer presented an $O(n)$-time algorithm for solving the same problem on trees [22]. In finding optimal edge ranking of trees, de la Torre et al. proposed the first polynomial-time algorithm which runs in $O(n^3 \log n)$ time [4]. There were some other

☆ The work was partially supported by the National Science Council of Taiwan (ROC) under Grant No. NSC94-2213-E-252-003.
E-mail address: rwhung@cyut.edu.tw

0890-5401/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.ic.2008.08.001

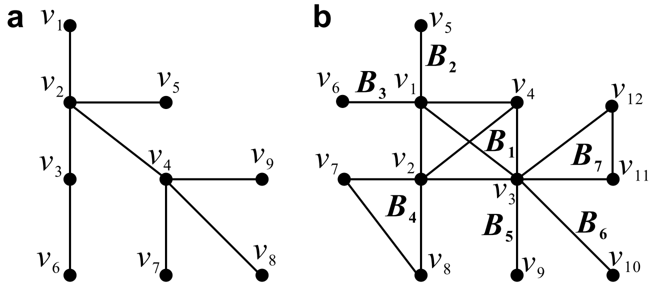

Fig. 1. (a) A tree, and (b) a block graph.

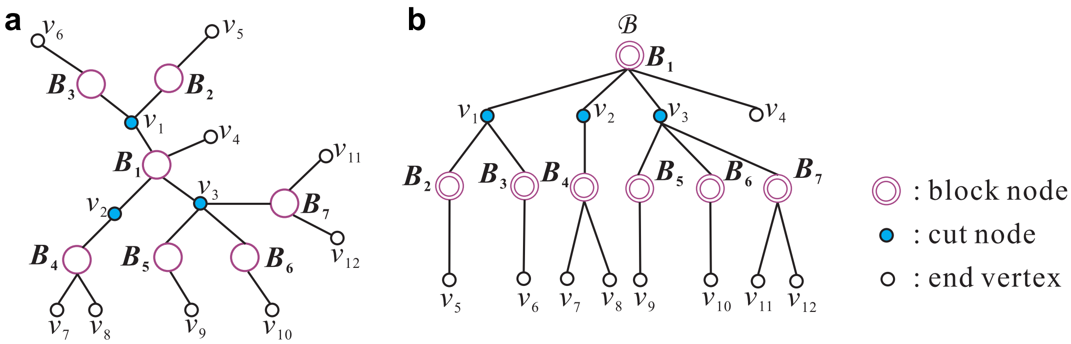

Fig. 2. (a) A representation tree constructed from a block graph shown in Fig. 1b, and (b) a block tree $T_{\mathcal{B}}$ with root $\mathcal{B} = B_1$ for (a).

papers in the literature on developing faster algorithms for the edge ranking problem on trees, though erroneous they are [5,27], culminating in the $O(n^2 \log \Delta)$-time algorithm of Zhou et al. [26]. Lam and Yue presented an $O(n)$-linear-time algorithm to solve the edge ranking problem of trees using a different approach [15]. Now, both the vertex ranking and edge ranking problems of trees can be solved in $O(n)$ time. Recently, Dereniowski and Nadolski proved that the vertex ranking problem on chordal graphs is NP-hard [8]. Note that block graphs form a superclass of trees and a subclass of chordal graphs.

In this paper, we present the first polynomial algorithm which runs in $O(n^2 \log \Delta)$ time to solve the vertex ranking problem on block graphs. Our idea is inspired by the algorithms given by Zhou et al. [26] for edge ranking of trees and given by Schäffer [22] for vertex ranking of trees. Note that the line graph of a tree is a block graph, but the reverse is not true. Hence, the algorithms for the edge ranking problem on trees can not be directly applied to solve the vertex ranking problem on block graphs.

2. Preliminaries

Let $G = (V,E)$ be a graph. The vertex and edge sets of G are denoted by $V(G)$ and $E(G)$, respectively. The union of two graphs G_1 and G_2, denoted by $G_1 + G_2$, is graph $(V(G_1) \cup V(G_2), E(G_1) \cup E(G_2))$. And let $G_1 - G_2$ denote the graph obtained from G_1 by deleting all vertices and edges of $(V(G_1) \cap V(G_2), E(G_1) \cap E(G_2))$. For any two sets X and Y, let $X - Y$ denote the set of elements of X that are not in Y.

Let G be a connected graph. A vertex v in G is called a *cut vertex* if the removal of v from G increases the number of connected components. In a connected graph, a *block* is a maximal connected subgraph without a cut vertex. A connected graph is a *block graph* if every block in it is a clique (complete graph). A vertex is a cut vertex in a block graph G if and only if it is the intersection of two or more blocks in G. If B_i and B_j are two distinct blocks in a block graph, then $B_i \cap B_j$ is empty or contains at most one vertex [1,9,21,25]. For example, Fig. 1b depicts a block graph and v_1 is a cut vertex which is the intersection of blocks B_1, B_2, B_3. On the other hand, Fig. 1a depicts a tree. We can find out that there are many similarities between them. In fact, trees are block graphs.

It follows from the above observations that we can construct a tree-like hierarchy, called *block tree*, from a block graph. Then, we can take the advantages of the ranking algorithms for trees while solving the vertex ranking problem on block graphs. With respect to the ranking algorithms on trees, researchers introduced the concept of *critical lists* [4,15,22,26]. In the following two subsections, we will define the *block tree* and the *critical list* of a block graph. These two structures are fundamentals and important in developing our efficient algorithm.

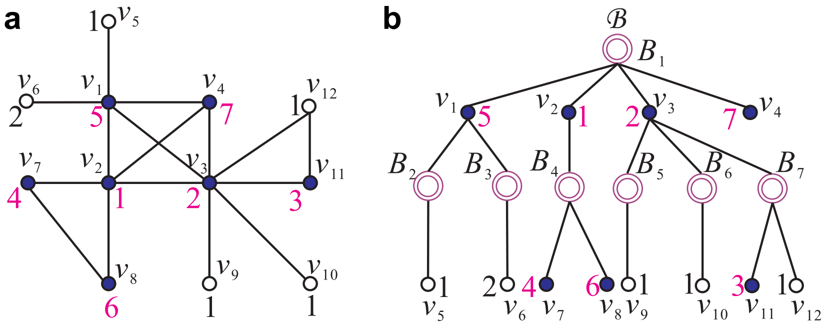

Fig. 3. A block graph and its block tree in Fig. 2b with a vertex ranking φ, where the visible vertices from B_1 are drawn by filled circles.

2.1. The block tree

Let $G = (V,E)$ be a block graph containing t blocks $B_1, B_2, \ldots, B_t$. The *representation tree* $T = (V_T,E_T)$ of G is constructed as follows: create t new nodes $B_1, B_2, \ldots, B_t$ standing for these t blocks in G. Let $B_T = \{B_1,B_2,\ldots,B_t\}$ and let $V_T = B_T \cup V$. The edge set E_T of T is defined as $\{(v_i,B_j)|v_i \in B_j$ in G for $1 \leqslant i \leqslant |V|$ and $1 \leqslant j \leqslant t\}$. For instance, given a block graph G shown in Fig. 1b, the representation tree T of G is shown in Fig. 2a.

While picking an arbitrary block node $\mathcal{B}$ of T as the root, we get a rooted tree with root $\mathcal{B}$. This rooted tree, denoted by $T_{\mathcal{B}} = (V^*,E^*)$, is called the *block tree* corresponding to block graph G. Fig. 2b depicts the block tree of the block graph shown in Fig. 1b. Note that rooting a representation tree suggests a natural way to decompose the computation. On the other hand, given a block graph $G = (V,E)$ the block tree can be constructed in $O(|V| + |E|)$ time by the depth first search [1].

We call an element of V^* a node of block tree $T_{\mathcal{B}}$ in general. The element of V^* is called a *block node* of $T_{\mathcal{B}}$ if it is in B_T; that is, it is not a vertex of V. A node is called an *end vertex* in $T_{\mathcal{B}}$ if it is in V and is not a cut vertex in G. The remnants of nodes in $T_{\mathcal{B}}$ are called *cut nodes*. Fig. 2b also reveals the types of nodes in $T_{\mathcal{B}}$.

Let G be a block graph and $T_{\mathcal{B}}$ be its corresponding block tree with root $\mathcal{B}$. The subtree of $T_{\mathcal{B}}$ rooted at node v is denoted by T_v, where v is either a cut node or a block node in $T_{\mathcal{B}}$. Let $G[T_v]$ denote the subgraph of $G = (V,E)$ induced by the set of vertices of V which are nodes in the subtree T_v of $T_{\mathcal{B}}$. For instance, $G[T_{B_4}] = (\{v_7,v_8\},\{(v_7,v_8)\})$ and $G[T_{v_1}] = (\{v_1,v_5,v_6\},\{(v_1,v_5),(v_1,v_6)\})$ in Fig. 2b.

Suppose that block node B has c children $v_1, v_2, \ldots, v_c$ in $T_{\mathcal{B}}$. Then we denote $T_B = \{B\} + T_{v_1} + T_{v_2} + \cdots + T_{v_c} + \{(v_i,B)|1 \leqslant i \leqslant c\}$. On the other hand, suppose that cut node v has b children $B_1, B_2, \ldots, B_b$ in $T_{\mathcal{B}}$. Then we denote $T_v = \{v\} + T_{B_1} + T_{B_2} + \cdots + T_{B_b} + \{(v,B_i)|1 \leqslant i \leqslant b\}$.

2.2. Critical lists and optimal vertex rankings

Let φ be a vertex ranking of a block graph G and let $T_{\mathcal{B}}$ be its corresponding block tree rooted at block node $\mathcal{B}$. The label of a vertex $v \in V(G)$ is denoted by $\varphi(v)$. For a vertex ranking φ of G and a subgraph G' of G, we denote by $\varphi_{|G'}$ a restriction of φ to $V(G')$. Let $\varphi' = \varphi_{|G'}$, then $\varphi'(v) = \varphi(v)$ for $v \in V(G')$.

Definition 1. Let φ be a vertex ranking of a graph G. A vertex v of G is *visible* from vertex u with respect to φ if either $v = u$ or there exists a path from u to v such that all vertices (except v) in this path have ranks less than $\varphi(v)$. A label ℓ is said to be *visible* if there is a visible vertex labeled with ℓ.

Definition 2. Let φ be a vertex ranking of a block graph G and let $T_{\mathcal{B}}$ be its block tree rooted at block node $\mathcal{B}$. Let ω be either a cut node or a block node of $T_{\mathcal{B}}$. A vertex v of $G[T_\omega]$ is said to be *visible* from ω under φ if v is visible from ω under $\varphi_{|G[T_\omega]}$. A label ℓ is said to be *visible* if there is a visible vertex labeled with ℓ. Denote by $L_\omega(\varphi)$ the set of visible labels of $G[T_\omega]$ under φ.

According to the above definitions, the label of a vertex is visible from itself. Fig. 3 shows a vertex ranking φ of a block tree T_{B_1} shown in Fig. 2b, where $L_{B_1}(\varphi) = \{7,6,5,4,3,2,1\}$, $L_{v_3}(\varphi) = \{3,2\}$ and $L_{B_7}(\varphi) = \{3,1\}$.

The following lemma can be easily verified by definition.

Lemma 3. *A vertex labeling φ of a block graph $G[T_{\mathcal{B}}]$ is a vertex ranking of $G[T_{\mathcal{B}}]$ if and only if*

(1) *$\varphi_{|G[T_v]}$ is a vertex ranking of $G[T_v]$ for every child v of the root $\mathcal{B}$ of $T_{\mathcal{B}}$; and*
(2) *no more than one vertex of the same rank are visible from $\mathcal{B}$ under φ.*

By the above lemma, it is easy to verify that φ is a vertex ranking of block graph G if and only if it is a vertex ranking of block tree $T_{\mathcal{B}}$ corresponding to G. Thus, for solving the vertex ranking problem on block graphs, we can focus on finding an optimal vertex ranking of block tree $T_{\mathcal{B}}$ constructed from the input block graph G. Note that we only rank the vertices except block nodes in block tree $T_{\mathcal{B}}$ and the paths connecting two distinct vertices can pass through some block nodes. That is, for a valid vertex ranking of $T_{\mathcal{B}}$, every block node in $T_{\mathcal{B}}$ is assigned by dummy label '0'.

Suppose that p and q are two integers and $p \leqslant q$, then denote $\{p,p+1,\ldots,q\}$ and $\{p,p+1,\ldots,q-1\}$ by $[p,q]$ and $[p,q)$, respectively. If $p > q$, then $[p,q] = \emptyset$. We define the *lexicographical order* '$\prec$' on two sets of positive labels by examining the labels in decreasing order. For instance, $\{5,4,3,1\} \prec \{5,4,3,2\}$ and $\{4,3\} \prec \{5,4,3,1\}$. We write $L_1 \preceq L_2$ if $L_1 \prec L_2$ or $L_1 = L_2$. On the other hand, if one label is in two sets of positive labels then we call one *conflict* occurs in them; in other words, if two sets of positive labels are disjoint then no conflict appears in them.

Definition 4. Let ω be either a cut node or a block node of $T_{\mathcal{B}}$. A vertex ranking φ of $G[T_\omega]$ is *critical* if $L_\omega(\varphi) \preceq L_\omega(\psi)$ for any vertex ranking ψ of $G[T_\omega]$. The list of a critical vertex ranking of $G[T_\omega]$ is called the *critical list* of $G[T_\omega]$ and is denoted by $L^*(T_\omega)$, which is the set of visible labels of a critical vertex ranking of $G[T_\omega]$.

de la Torre et al. [4] and Zhou et al. [26] introduced the concept of *supercritical ranking* and *subcritical ranking*. We adapt the same notation but with different meanings.

Definition 5. Let B be a block node of $T_{\mathcal{B}}$. A vertex ranking φ of $G[T_B]$ is *supercritical* if the restriction $\varphi_{|G[T_{B'}]}$ is critical for every block node B' of T_B. A vertex ranking φ of $G[T_B]$ is *subcritical* if the restriction $\varphi_{|G[T_{B'}]}$ is critical for every block node B' and $B' \neq B$. The list of a supercritical vertex ranking φ of $G[T_B]$ is called the *supercritical list* of $G[T_B]$, and is also denoted by $L^*(T_B)$.

By the above definitions, a vertex ranking φ of $G[T_{\mathcal{B}}]$ is supercritical if and only if φ is subcritical and critical. And, a vertex ranking φ of $G[T_v]$ is critical if the restriction $\varphi_{|G[T_B]}$ is supercritical for every block node B and $L_v(\varphi)$ is critical, where v is a vertex in G and B is a child of v in $T_{\mathcal{B}}$.

It follows from the above definitions that every block tree has a supercritical vertex ranking because this is simply an optimal vertex ranking that has the lexicographically least supercritical list at the root. However, it does not follow that every block subtree has a supercritical vertex ranking. It is plausible that to obtain a list-optimal vertex ranking at the root, some subtree may require a non-optimal vertex ranking. In next section, we will prove that every block subtree does in fact have a supercritical vertex ranking and our algorithm always finds one.

3. An $O(n^2 \log \Delta)$-time algorithm

With respect to trees, researchers [4,10,11,15,22,26] considered the input tree to be rooted at an arbitrary node and use a bottom-up approach to compute an optimal (vertex or edge) ranking of the input tree. Note that the block tree is a rooted tree $T_{\mathcal{B}}$ with root block node $\mathcal{B}$. Hence, we can take the advantages of the (vertex and edge) ranking algorithms for trees while solving the vertex ranking problem on block graphs.

In this section, we will propose an $O(n^2 \log \Delta)$-time algorithm for ranking a block graph G based upon its block tree $T_{\mathcal{B}}$. Let $T_{\mathcal{B}}$ be the block tree of the input block graph. For simplicity, in the rest of the paper assume that for block node B in $T_{\mathcal{B}}$, vertices $v_1, v_2, \ldots, v_c$ are the children of B in $T_{\mathcal{B}}$, and for cut node υ in $T_{\mathcal{B}}$, block nodes $B_1, B_2, \ldots, B_b$ are the children of υ in $T_{\mathcal{B}}$.

We first sketch our algorithm as follows: The algorithm uses a bottom-up approach to compute a supercritical vertex ranking of $G[T_{\mathcal{B}}]$. For each end vertex v in $T_{\mathcal{B}}$, we construct its critical vertex ranking and the critical list $L^*(T_v) = \{1\}$ of $G[T_v]$. For each block node B in $T_{\mathcal{B}}$, by giving only vertices $v_1, v_2, \ldots, v_c$ new ranks, a supercritical vertex ranking and the supercritical list $L^*(T_B)$ of $G[T_B]$ can be obtained from critical vertex rankings of $G[T_{v_i}]$'s for $1 \leqslant i \leqslant c$. For each cut node υ in $T_{\mathcal{B}}$, suppose the supercritical vertex rankings of the subtrees $T_{B_1}, T_{B_2}, \ldots, T_{B_b}$, as well as their supercritical lists $L^*(T_{B_1}), L^*(T_{B_2}), \ldots, L^*(T_{B_b})$, have been computed. We focus on ranking vertex υ. Then, a critical vertex ranking and the critical list $L^*(T_\upsilon)$ of $G[T_\upsilon]$ can be computed from $L^*(T_{B_1}), L^*(T_{B_2}), \ldots, L^*(T_{B_b})$. By visiting the block tree $T_{\mathcal{B}}$ bottom-up, we can compute a supercritical list of $G[T_{\mathcal{B}}]$ and construct an optimal vertex ranking of $G[T_{\mathcal{B}}]$.

3.1. Ranking cut nodes of block trees

For a cut node υ in $T_{\mathcal{B}}$, assume the supercritical lists $L^*(T_{B_1}), L^*(T_{B_2}), \ldots, L^*(T_{B_b})$ of children of υ have been computed. The following lemma proves that by giving υ a label, a critical vertex ranking of $G[T_\upsilon]$ and a critical list $L^*(T_\upsilon)$ can be obtained from $L^*(T_{B_i})$'s for $1 \leqslant i \leqslant b$.

Lemma 6. *Let T_υ be a subtree of $T_{\mathcal{B}}$ rooted at cut node υ. Assume that $B_1, B_2, \ldots, B_b$ are children of υ in $T_{\mathcal{B}}$ and φ is a vertex ranking that is supercritical for each subtree T_{B_i}, $1 \leqslant i \leqslant b$, but does not yet assign a label to υ. Then, assigning a label $\varphi(\upsilon)$ to υ makes φ into a critical vertex ranking of $G[T_\upsilon]$.*

Proof. Let τ be the integer $\max\{l|l \in L^*(T_{B_i})$ for $1 \leqslant i \leqslant b\}$. Let α be the largest rank occurring on more than one supercritical list or 0 if the lists have no conflicts. Let β be the smallest integer strictly larger than α that does not occur on any supercritical list $L^*(T_{B_i})$. Let $\varphi_i = \varphi_{|G[T_{B_i}]}$; that is, φ_i is a restriction of φ to $G[T_{B_i}]$. Then,

$$\varphi(u) = \begin{cases} \varphi_i(u), & \text{if } u \in G[T_{B_i}] \text{ for } 1 \leqslant i \leqslant b; \\ \beta, & \text{if } u = \upsilon. \end{cases}$$

We will prove that φ is a critical ranking of $G[T_\upsilon]$ and $L^*(T_\upsilon) = \cup_{1\leqslant i\leqslant b}\{l|l \in L^*(T_{B_i})$ and $l > \beta\} \cup \{\beta\}$.

First, we prove that φ is a valid vertex ranking of $G[T_\upsilon]$. We first consider that $\beta = \tau + 1$. Let $\varphi_i(x) = \varphi_j(y)$ for $i \neq j$. Suppose that $\varphi_i(x)$ and $\varphi_j(y)$ are in $L^*(T_{B_i})$ and $L^*(T_{B_j})$, respectively. Then, they become invisible from υ while $\varphi(\upsilon) = \tau + 1$. Hence, φ is a valid vertex ranking of $G[T_\upsilon]$. On the other hand, suppose that $\beta < \tau$. Now, we consider that $\varphi(x) = \varphi(y) = \alpha' \ (\leqslant \alpha)$ for x, $y \neq \upsilon$ and $x, y \in G[T_\upsilon]$. If both x and y are in the same subtree T_{B_i} of T_υ for some i, then there exists one vertex $z \in G[T_{B_i}]$ such that z is on the path connecting x and y and $\varphi(z) > \alpha'$, both x and y are invisible from B_i under φ_i, and, hence, $\varphi(\upsilon) = \beta$ makes φ into a valid ranking of $G[T_\upsilon]$. Suppose that $x \in G[T_{B_i}]$ and $y \in G[T_{B_j}]$ for $i \neq j$. Since $\alpha' \leqslant \alpha < \beta$ and all paths connecting x and y pass through υ in $G[T_\upsilon]$, α' becomes invisible in $L_\upsilon(\varphi)$. Thus, there exists no conflict in $L_\upsilon(\varphi)$. This implies that φ is a valid vertex ranking of $G[T_\upsilon]$. By the above arguments, φ is a valid vertex ranking of $G[T_\upsilon]$ while $\varphi(\upsilon) = \beta$.

Next, we will prove that φ is a critical ranking of $G[T_\upsilon]$. Let $\widehat{L_i} = \{l|l \in L^*(T_{B_i})$ and $l > \beta\}$ for $1 \leqslant i \leqslant b$ and let $L = \cup_{1\leqslant i\leqslant b}\widehat{L_i} \cup \{\beta\}$. We will show that L is critical. Let ψ be a vertex ranking of $G[T_\upsilon]$ that is different from φ. Let $\psi(\upsilon) = \beta_\psi$. Let $\widehat{L'_i} = \{l|l \in L_{B_i}(\psi)$ and $l > \beta_\psi\}$ for $1 \leqslant i \leqslant b$ and let $L' = \cup_{1\leqslant i\leqslant b}\widehat{L'_i} \cup \{\beta_\psi\}$. By definition, $L_\upsilon(\psi) = L'$. Let α_ψ be the largest rank that occurs in $L_{B_i}(\psi)$'s at least twice; that is, $\alpha_\psi = \max\{l|l \in L_{B_i}(\psi) \cap L_{B_j}(\psi), i \neq j\}$. Consider the following two cases:

Case 1: $\alpha > \alpha_\psi$. Let L_1 and L_2 denote $\cup_{1\leqslant i\leqslant b}\{l|l \in L_{B_i}(\varphi)$ and $l > \alpha\}$ and $\cup_{1\leqslant i\leqslant b}\{l|l \in L_{B_i}(\psi)$ and $l > \alpha\}$, respectively. Since $L_{B_i}(\varphi)$ is the supercritical list of $G[T_{B_i}]$, $\{l|l \in L_{B_i}(\varphi)$ and $l > \alpha\} \preceq \{l|l \in L_{B_i}(\psi)$ and $l > \alpha\}$ for any i. Hence, $L_1 \preceq L_2$ and $[\alpha+1, \beta-1] \subseteq L_1$, but $\beta \notin L_1$. Let $L_1^\beta = \{l|l \in L_1$ and $l \geqslant \beta\}$ and $L_2^\beta = \{l|l \in L_2$ and $l \geqslant \beta\}$. Since $[\alpha+1, \beta-1] \subseteq L_1$, $L_2 \cap [\alpha+1, \beta-1] \preceq L_1 \cap [\alpha+1, \beta-1] = [\alpha+1, \beta-1]$. If $L_2^\beta \prec L_1^\beta$, then $L_2 = L_2^\beta \cup (L_2 \cap [\alpha+1, \beta-1]) \prec L_1^\beta \cup [\alpha+1, \beta-1] = L_1$, a contradiction occurs. Thus, $L_1^\beta \preceq L_2^\beta$. Consider that $L_1^\beta = L_2^\beta$. Let $\gamma = \max\{l|l \geqslant \alpha, l \in L_{B_i}(\varphi)$, and $l \notin L_{B_i}(\psi)$ for $1 \leqslant i \leqslant b\}$. Since $\alpha_\psi < \alpha$, $\gamma \geqslant \alpha$ and γ exists (for example α). Let $\gamma \in L_{B_s}(\varphi)$ but $\gamma \notin L_{B_s}(\psi)$ for some s. Since $L_{B_s}(\varphi) \preceq L_{B_s}(\psi)$, there exists a rank $\delta > \gamma$ such that $\delta \in L_{B_s}(\psi)$ but $\delta \notin L_{B_s}(\varphi)$. Clearly, $\delta > \alpha$. Since $L_1^\beta = L_2^\beta$ and $[\alpha+1, \beta-1] \subseteq L_1$, $\delta \in L_1$. In other words, $\delta \in L_{B_t}(\varphi)$, $t \neq s$, and $\delta \notin L_{B_t}(\psi)$. This contradicts the definition of γ. Thus, $L_1^\beta \neq L_2^\beta$. Now, we consider $L_1^\beta \prec L_2^\beta$. Then, there exists one rank $w = \max\{l|l > \beta, l \in L_2^\beta$, and $l \notin L_1^\beta\}$. By definition, $\beta_\psi \neq w$, $w \in L_2^\beta$, and $w \notin L_1^\beta$. Let ℓ be a rank larger than w. If $\ell \in L_1^\beta$ but $\ell \notin L_2^\beta$, then there exists one rank x such that $x > \ell$, $x \in L_2^\beta$ but $x \notin L_1^\beta$, and it contradicts the definition of w. Hence, $\{l|l > w$ and $l \in L_1^\beta\} = \{l|l > w$ and $l \in L_2^\beta\}$. If $\beta_\psi > w$, then $\beta_\psi \notin L_1^\beta$ and, hence, $L = L_1^\beta \cup \{\beta\} \prec \{l|l \in L_2^\beta, l > \beta_\psi\} \cup \{\beta_\psi\} = L'$. If $\beta_\psi < w$, then $L = L_1^\beta \cup \{\beta\} \prec \{l|l \in L_2^\beta, l > \beta_\psi\} \cup \{\beta_\psi\} = L'$ since $w \in L_2^\beta$, $w \notin L_1^\beta$, and $\{l|l > w$ and $l \in L_1^\beta\} = \{l|l > w$ and $l \in L_2^\beta\}$. In either case, $L \prec L'$.

Case 2: $\alpha \leqslant \alpha_\psi$. In this case, $\alpha \leqslant \alpha_\psi < \beta_\psi$. Let L_3 and L_4 denote $\cup_{1\leqslant i\leqslant b}\{l|l \in L_{B_i}(\varphi)$ and $l \geqslant \beta_\psi\}$ and $\cup_{1\leqslant i\leqslant b}\{l|l \in L_{B_i}(\psi)$ and $l \geqslant \beta_\psi\}$, respectively. Since $L_{B_i}(\varphi)$ is the supercritical list of $G[T_{B_i}]$, $\{l|l \in L_{B_i}(\varphi)$ and $l \geqslant \beta_\psi\} \preceq \{l|l \in L_{B_i}(\psi)$ and $l \geqslant \beta_\psi\}$ for $1 \leqslant i \leqslant b$. Hence, $L_3 \preceq L_4$. By definition, $\beta_\psi \notin L_4$. Then, we consider three cases of $\beta_\psi = \beta$, $\beta_\psi > \beta$, or $\beta_\psi < \beta$. We first consider the case of $\beta_\psi = \beta$. Since $L_3 \preceq L_4$, $L = L_3 \cup \{\beta\}$, and $L' = L_4 \cup \{\beta_\psi\}$, we get that $L \preceq L'$. Next, we consider that $\beta_\psi > \beta$. Suppose $\beta_\psi \notin L_3$. It is clear that $L_3 \prec L_4 \cup \{\beta_\psi\} = L'$. Thus, $L_3 \cup [\beta, \beta_\psi) \prec L_4 \cup \{\beta_\psi\}$. Since $L \preceq L_3 \cup [\beta, \beta_\psi)$ and $L' = L_4 \cup \{\beta_\psi\}$, we obtain that $L \prec L'$. Now, suppose $\beta_\psi \in L_3$. Then, there exists one label γ such that $\gamma > \beta_\psi$, $\gamma \in L_4$ but $\gamma \notin L_3$, and $\{l|l \in L_3$ and $l > \gamma\} = \{l|l \in L_4$ and $l > \gamma\}$. Thus, $L \preceq L_3 \cup [\beta, \beta_\psi] \prec L_4 \cup \{\beta_\psi\} = L'$. Finally, we consider the case of $\beta_\psi < \beta$. Since $L_3 \preceq L_4$, $\beta_\psi \in L_3$, and $\beta_\psi \notin L_4$, there exists one label γ such that $\gamma > \beta_\psi$, $\gamma \in L_4$ but $\gamma \notin L_3$, and $\{l|l \in L_3$ and $l > \gamma\} = \{l|l \in L_4$ and $l > \gamma\}$. If $\beta > \gamma$, it contradicts that any rank in $[\alpha, \beta)$ is used in φ since $\gamma \notin L_3$. Hence, $\beta \leqslant \gamma$. Since $\{l|l \in L_3$ and $l > \gamma\} = \{l|l \in L_4$ and $l > \gamma\}$, $\gamma \in L_4$ but $\gamma \notin L_3$, and $\gamma \geqslant \beta > \beta_\psi$, we get that $L = \{l|l \in L_3$ and $l > \beta\} \cup \{\beta\} \prec L_4 \cup \{\beta_\psi\} = L'$.

It follows from the above cases that $L \preceq L'$ and, hence, $L_\upsilon(\varphi) \preceq L_\upsilon(\psi)$ for any vertex ranking ψ of $G[T_\upsilon]$. Thus, φ is a critical ranking of $G[T_\upsilon]$ and L is the critical list of $G[T_\upsilon]$.

Let n_υ and n_i, $1 \leqslant i \leqslant b$, be the numbers of vertices in $G[T_\upsilon]$ and $G[T_{B_i}]$, respectively. Then, $n_\upsilon = 1 + \sum_{i=1}^{b} n_i$. It follows from Lemma 6 that ranking a cut node υ and constructing its critical list $L^*(T_\upsilon)$ of $G[T_\upsilon]$ from $L^*(T_{B_i})$'s can be done in $O(n_\upsilon)$ time, since all lists are searched once [22]. Since the number of cut nodes in $T_\mathcal{B}$ is bounded in $O(n)$, ranking all cut nodes takes $O(n^2)$ time in total. Therefore, we have the following theorem:

Theorem 7. *Let G be a block graph with n vertices and let $T_\mathcal{B}$ be its corresponding block tree. The cut nodes of G can be optimally ranked in $O(n^2)$ time if the supercritical lists of their children have been computed.*

3.2. Processing block nodes of block trees

In the rest of this paper, we focus on processing block nodes of a block tree $T_\mathcal{B}$. Assume a block node B of $T_\mathcal{B}$ is visited and vertices $v_1, v_2, \ldots, v_c$ are children of B in $T_\mathcal{B}$. Suppose that their associated critical lists are $L^*(T_{v_1}), L^*(T_{v_2}), \ldots, L^*(T_{v_c})$

corresponding to the critical vertex rankings $\varphi_1, \varphi_2, \ldots, \varphi_c$ of $G[T_{v_1}], G[T_{v_2}], \ldots, G[T_{v_c}]$, respectively, are given. The following lemma shows that we don't need to relabel any vertex v for $v \in V(G[T_{v_i}]) - \{v_i\}$, $1 \leqslant i \leqslant c$, while constructing a supercritical vertex ranking of $G[T_B]$. Thus, by giving only vertices $v_1, v_2, \ldots, v_c$ new ranks, a supercritical vertex ranking and the supercritical list $L^*(T_B)$ of $G[T_B]$ can be obtained from φ_i's for $1 \leqslant i \leqslant c$.

Lemma 8. *Let B be a block node of $T_{\mathcal{B}}$ with children $v_1, v_2, \ldots, v_c$. Assume that $\varphi_1, \varphi_2, \ldots, \varphi_c$ are critical vertex rankings of $G[T_{v_1}]$, $G[T_{v_2}], \ldots, G[T_{v_c}]$, respectively. Then, $G[T_B]$ has a supercritical vertex ranking φ such that $\varphi(v) = \varphi_i(v)$ for $v \in V(G[T_{v_i}]) - \{v_i\}$ and $1 \leqslant i \leqslant c$.*

Proof. Let $r_{\max}$ be the integer $\max\{l | l \in L^*(T_{v_i})$ for $1 \leqslant i \leqslant c\}$. Let ψ be a vertex ranking of $G[T_B]$ such that

$$\psi(v) = \begin{cases} r_{\max} + i, & \text{if } v = v_i \text{ for } 1 \leqslant i \leqslant c; \\ \varphi_i(v), & \text{if } v \in V(G[T_{v_i}]) - \{v_i\} \text{ for } 1 \leqslant i \leqslant c. \end{cases}$$

It is easy to see that ψ is a subcritical vertex ranking of $G[T_B]$ extended from φ_i's for $1 \leqslant i \leqslant c$. Thus, there exist subcritical vertex rankings of $G[T_B]$ that are extensions of $\varphi_1, \varphi_2, \ldots, \varphi_c$ by giving only $v_1, v_2, \ldots, v_c$ new ranks. Assume that φ is a subcritical vertex ranking of $G[T_B]$ such that $L_B(\varphi)$ is the lexicographically least list among those of all subcritical vertex rankings of $G[T_B]$ extended from $\varphi_1, \varphi_2, \ldots, \varphi_c$ by giving only $v_1, v_2, \ldots, v_c$ new ranks. We will show that φ is critical and hence supercritical.

Suppose that η is a vertex ranking of $G[T_B]$. Let $\eta_i = \eta_{|G[T_{v_i}]}$, $1 \leqslant i \leqslant c$. Since $L_{v_i}(\varphi_i)(= L^*(T_{v_i}))$ is the critical list of $G[T_{v_i}]$, $L_{v_i}(\varphi_i) \preceq L_{v_i}(\eta_i)$ for $1 \leqslant i \leqslant c$. Let r_i, $1 \leqslant i \leqslant c$, be the largest integer such that $r_i \in L_{v_i}(\eta_i)$ but $r_i \notin L_{v_i}(\varphi_i)$ or 0 if $L_{v_i}(\eta_i) = L_{v_i}(\varphi_i)$. For simplicity, denote $\varphi_i(v_i)$ by ℓ_i, for $i = 1$ to c. Then, we define a vertex ranking $\widehat{\varphi}$ of $G[T_B]$ extended from $\varphi_1, \varphi_2, \ldots, \varphi_c$ as follows:

$$\widehat{\varphi}(v) = \begin{cases} \max\{\ell_i, r_i\}, & \text{if } v = v_i \text{ for } 1 \leqslant i \leqslant c; \\ \varphi_i(v), & \text{if } v \in V(G[T_{v_i}]) - \{v_i\} \text{ for } 1 \leqslant i \leqslant c. \end{cases}$$

Since $L_{v_i}(\varphi_i)$ is the critical list of $G[T_{v_i}]$, $L_{v_i}(\varphi_i) \preceq L_{v_i}(\eta_i)$ for $1 \leqslant i \leqslant c$. By definition, $r_i = 0$ if $L_{v_i}(\eta_i) = L_{v_i}(\varphi_i)$; and $r_i = \max\{l | l \in L_{v_i}(\eta_i)$ but $l \notin L_{v_i}(\varphi_i)\}$ otherwise. Consider that $r_i \neq 0$. Let r be an integer that is greater than r_i. Suppose $r \in L_{v_i}(\varphi_i)$ but $r \notin L_{v_i}(\eta_i)$. Since $L_{v_i}(\varphi_i) \prec L_{v_i}(\eta_i)$, there exists one integer $\widetilde{r}$ such that $\widetilde{r} > r$, $\widetilde{r} \in L_{v_i}(\eta_i)$ but $\widetilde{r} \notin L_{v_i}(\varphi_i)$. This contradicts the definition of r_i. Thus, $\{l | l \in L_{v_i}(\varphi_i)$ and $l > r_i\} = \{l | l \in L_{v_i}(\eta_i)$ and $l > r_i\}$.

We first show that $\widehat{\varphi}$ is a valid vertex ranking of $G[T_B]$. It is sufficient to prove that no label is visible for two different vertices under $\widehat{\varphi}$. Assume that vertex x is visible from B under $\widehat{\varphi}$ and $x \in G[T_{v_i}]$. If $r_i = 0$, then $\widehat{\varphi}(x) \geqslant \widehat{\varphi}(v_i) = \ell_i$ and $L_{v_i}(\eta_i) = L_{v_i}(\varphi_i)$; otherwise, $\widehat{\varphi}(x) \geqslant r_i \geqslant \eta_i(v_i)$. Suppose $r_i \neq 0$. Since $\widehat{\varphi}(x)$ is visible from B under $\widehat{\varphi}$, $\{l | l \in L_{v_i}(\varphi_i)$ and $l > r_i\} = \{l | l \in L_{v_i}(\eta_i)$ and $l > r_i\}$, and $\widehat{\varphi}(x) \geqslant r_i \geqslant \eta_i(v_i)$, we get that $\widehat{\varphi}(x)$ is also visible from B under η. This implies that if both $\widehat{\varphi}(x)$ and $\widehat{\varphi}(y)$ are visible from B under $\widehat{\varphi}$, then they are also visible from B under η. Hence, if $\widehat{\varphi}(x) = \widehat{\varphi}(y)$ and both of them are visible from B under $\widehat{\varphi}$, then there exist two labels $\widehat{\varphi}(x)$ and $\widehat{\varphi}(y)$ which are visible from B under η, and, hence, it contradicts that η is a valid vertex ranking of $G[T_B]$. Therefore, there exist no distinct vertices with the same rank which are visible from B under $\widehat{\varphi}$; that is, $\widehat{\varphi}$ is a valid vertex ranking of $G[T_B]$.

Let $\widehat{\varphi}_i = \widehat{\varphi}_{|G[T_{v_i}]}$. By definition, $\widehat{\varphi}$ is a subcritical vertex ranking of $G[T_B]$ extended from φ_i's by giving only v_i, $1 \leqslant i \leqslant c$, rank $\max\{\ell_i, r_i\}$. By definition, $L_B(\varphi) \preceq L_B(\widehat{\varphi})$. We claim that $L_B(\widehat{\varphi}) \preceq L_B(\eta)$. Since "$\preceq$" is transitive, we get $L_B(\varphi) \preceq L_B(\eta)$, as desired.

In the following, we will prove the above claim that $L_B(\widehat{\varphi}) \preceq L_B(\eta)$. Since $\widehat{\varphi}$ is a valid vertex ranking of $G[T_B]$, lists $L_{v_i}(\widehat{\varphi}_i)$'s are pairwise disjoint. Thus, we can only prove that $L_{v_i}(\widehat{\varphi}_i) \preceq L_{v_i}(\eta_i)$ for $1 \leqslant i \leqslant c$. By definition, $L_{v_i}(\widehat{\varphi}_i) = \{\widehat{\varphi}(v_i)\} \cup \{l | l \in L_{v_i}(\varphi_i)$ and $l > \widehat{\varphi}(v_i)\}$ and $L_{v_i}(\eta_i) = \{\eta(v_i)\} \cup \{l | l \in L_{v_i}(\eta_i)$ and $l > \eta(v_i)\}$. By definition of r_i, we have
(i) if $r_i = 0$, then $L_{v_i}(\varphi_i) = L_{v_i}(\eta_i)$; and
(ii)if $r_i \neq 0$, then $L_{v_i}(\varphi_i) \prec L_{v_i}(\eta_i)$ and $r_i = \max\{l | l \in L_{v_i}(\eta_i)$ but $l \notin L_{v_i}(\varphi_i)\}$.

By the above cases, assume that $r_i \neq 0$. Then, $\{l | l \in L_{v_i}(\varphi_i)$ and $l > r_i\} = \{l | l \in L_{v_i}(\eta_i)$ and $l > r_i\}$. The rank $\widehat{\varphi}(v_i)$ depends on the relative values of ℓ_i and r_i. There are two cases:

Case 1: $\widehat{\varphi}(v_i) = \ell_i$. In this case, $\ell_i > r_i \geqslant \eta(v_i)$. Since $\ell_i \in L_{v_i}(\varphi_i)$, $\ell_i > r_i$, and $\{l | l \in L_{v_i}(\varphi_i)$ and $l > r_i\} = \{l | l \in L_{v_i}(\eta_i)$ and $l > r_i\}$, we get that $\ell_i \in L_{v_i}(\eta_i)$. Thus, $L_{v_i}(\widehat{\varphi}_i) = \{l | l \in L_{v_i}(\varphi_i)$ and $l \geqslant \ell_i\} = \{l | l \in L_{v_i}(\eta_i)$ and $l \geqslant \ell_i\} \prec \{l | l \in L_{v_i}(\eta_i)$ and $l \geqslant \ell_i\} \cup \{\eta(v_i)\} \preceq L_{v_i}(\eta_i)$. That is, $L_{v_i}(\widehat{\varphi}_i) \prec L_{v_i}(\eta_i)$.

Case 2: $\widehat{\varphi}(v_i) = r_i$. In this case, $r_i \geqslant \ell_i$ and $r_i \geqslant \eta(v_i)$. Then, $L_{v_i}(\widehat{\varphi}_i) = \{r_i\} \cup \{l | l \in L_{v_i}(\varphi_i)$ and $l > r_i\}$, $L_{v_i}(\eta_i) = \{\eta(v_i)\} \cup \{l | l \in L_{v_i}(\eta_i)$ and $l > \eta(v_i)\}$, and $r_i \in L_{v_i}(\eta_i)$. Since $\{l | l \in L_{v_i}(\varphi_i)$ and $l > r_i\} = \{l | l \in L_{v_i}(\eta_i)$ and $l > r_i\}$, $r_i \geqslant \eta(v_i)$, and $r_i \in L_{v_i}(\eta_i)$, we obtain that $L_{v_i}(\widehat{\varphi}_i) = \{r_i\} \cup \{l | l \in L_{v_i}(\varphi_i)$ and $l > r_i\} = \{r_i\} \cup \{l | l \in L_{v_i}(\eta_i)$ and $l > r_i\} = \{l | l \in L_{v_i}(\eta_i)$ and $l \geqslant r_i\} \preceq \{l | l \in L_{v_i}(\eta_i)$ and $l \geqslant \eta(v_i)\} = L_{v_i}(\eta_i)$. Thus, $L_{v_i}(\widehat{\varphi}_i) \preceq L_{v_i}(\eta_i)$.

Let $v_1, v_2, \ldots, v_c$ be children of a block node B in $T_{\mathcal{B}}$. The vertex v_i, $1 \leqslant i \leqslant c$, is referred to as *branch i* of T_B. The set of vertices emanating (down) from a block node B is denoted by V_B, i.e., $V_B = \{v_1, v_2, \ldots, v_c\}$. By Lemma 8, we can only label the branches of T_B to obtain a supercritical vertex ranking of $G[T_B]$ while the critical rankings of $G[T_{v_1}], G[T_{v_2}], \ldots, G[T_{v_c}]$, as well as their critical lists $L^*(T_{v_1}), L^*(T_{v_2}), \ldots, L^*(T_{v_c})$, have been computed. To identify a labeling of V_B, we often list the labels and their associated vertices as ordered pairs, so that (ℓ_i, v_i) means that label ℓ_i is assigned to vertex v_i. We usually sort a labeling by the order of its labels (ranks) but not by vertex names. In all of the proofs where the order matters, it is convenient to have (ℓ_c,v_c) correspond to the branch with the largest label and (ℓ_1,v_1) correspond to the branch with the smallest label. This means that v_i is the branch with i-th smallest label.

In the rest of this section, we assume that B is a block node in $T_{\mathcal{B}}$ with branch set $V_B = \{v_1,v_2,\ldots,v_c\}$, φ is a partial vertex ranking of $G[T_B]$ on $G[T_{v_1}], G[T_{v_2}], \ldots, G[T_{v_c}]$; that is, φ labels all vertices in $G[T_B]$ except those of V_B, and the critical rankings $\varphi_1, \varphi_2, \ldots, \varphi_c$ of $G[T_{v_1}], G[T_{v_2}], \ldots, G[T_{v_c}]$, respectively, as well as their critical lists $L^*(T_{v_1}), L^*(T_{v_2}), \ldots, L^*(T_{v_c})$, have been computed by φ.

Let $L = \{(\ell_1,v_1),(\ell_2,v_2),\ldots,(\ell_c,v_c)\}$ be an arbitrary labeling of V_B. Define $\widehat{L_i}$ to be the set $\{l | l \in L^*(T_{v_i})$ and $l > \ell_i\}$ for $1 \leqslant i \leqslant c$. Obviously, $\widehat{L_i}$ includes the labels of $L^*(T_{v_i})$ that are still visible while assigning label ℓ_i to v_i . By Lemma 8, we do not need to relabel any vertex in $V(G[T_{v_i}]) - \{v_i\}$ for $1 \leqslant i \leqslant c$, and we can obtain a supercritical (hence optimal) vertex ranking of $G[T_B]$. We can easily verify that $\ell_i \geqslant \varphi_i(v_i)$ for any i if L is a valid labeling and any vertex in $V(G[T_{v_i}]) - \{v_i\}$ is not relabeled. Then, we define the *valid labeling* of V_B as follows:

Definition 9. $L = \{(\ell_1,v_1),(\ell_2,v_2),\ldots,(\ell_c,v_c)\}$ is said to be a *valid labeling* of V_B if for any branch i, the following conditions hold: (1) $\ell_i \geqslant \varphi_i(v_i)$, (2) $\ell_i \notin L^*(T_{v_i}) - \{\varphi_i(v_i)\}$, and (3) $(\widehat{L_i} \cup \{\ell_i\}) \cap (\widehat{L_j} \cup \{\ell_j\}) = \emptyset$ for any branch $j \neq i$.

It is easily verified that a valid labeling together with φ forms a subcritical vertex ranking of $G[T_B]$. A valid labeling L of V_B is called *optimal* if $L_B(L \cup \varphi)$ forms a supercritical list of $G[T_B]$.

Lemma 10. *Let $L = \{(\ell_c,v_c),\ldots,(\ell_2,v_2),(\ell_1,v_1)\}$ be a valid labeling of V_B such that $\ell_c > \cdots > \ell_2 > \ell_1$. Then, there is no optimal labeling of V_B in which the largest label used is greater than ℓ_c.*

Proof. Let $\widetilde{L} = \{\ell_1,\ell_2,\ldots,\ell_c\}$. Consider the visible list $L' = \cup_{1\leqslant i\leqslant c}\widehat{L_i} \cup \widetilde{L}$ that will be passed up the tree as a result of the labeling L. It starts with a (possible empty) prefix of ranks $r_1, r_2, \ldots, r_t$ that are all greater than ℓ_c. Without loss of generality, assume that $r_1 < r_2 < \cdots < r_t$. And set r_0 equal to ℓ_c.

Assume by contradiction that there exists an optimal labeling M of V_B in which the highest label, h, used is greater than ℓ_c. Suppose $r_k \leqslant h \leqslant r_{k+1}$ for $k \geqslant 0$. Then, the ranks $r_{k+1}, r_{k+2}, \ldots, r_t$ will be in $L_B(M \cup \varphi)$. Since L is valid and each of the r_i's is visible, each r_i is on $\widehat{L_j}$ of exactly one child of B. Thus, if $h = r_k$ or $h = r_{k+1}$, then the label h will also be in $\widehat{L_i}$, $1 \leqslant i \leqslant c$; that is, M is an invalid labeling. Thus, $r_k < h < r_{k+1}$. Then, $L_B(M \cup \varphi)$ will start with $h, r_{k+1}, \ldots, r_{t-1}, r_t$, which is lexicographically greater than $L_B(L \cup \varphi)$ that starts with $r_k, r_{k+1}, \ldots, r_{t-1}, r_t$. This contradicts that M is an optimal labeling of V_B. Thus, there exists no optimal labeling of V_B in which the highest label used is greater than ℓ_c.

Lemma 11. *Let $L = \{(\ell_c,v_c),\ldots,(\ell_2,v_2),(\ell_1,v_1)\}$ be an optimal labeling of V_B such that $\ell_c > \cdots > \ell_2 > \ell_1$. Let $S_k = \{v_k,v_{k-1},\ldots,v_1\}$ for $1 \leqslant k \leqslant c$, where S_k is a set comprising a suffix of the branch set when sorted by labels in L. Then, the labeling L restricted to the branch set S_k is optimal for $G[T_B - \cup_{k+1\leqslant i\leqslant c}T_{v_i}]$.*

Proof. Let L_{S_k} be a restriction of L to S_k. Since $L \cup \varphi$ is a valid vertex ranking of $G[T_B]$, $L_{S_k} \cup \varphi$ is also a valid vertex ranking of $G[T_B - \cup_{k+1\leqslant i\leqslant c}T_{v_i}]$. By Lemma 10, there is no optimal labeling of S_k that uses labels larger than ℓ_k since L_{S_k} is a valid labeling of S_k. Suppose that M is a valid labeling of S_k which is better than L_{S_k}. We can replace L_{S_k} with M and get a better overall labeling of V_B. This contradicts the assumption that L is optimal. Thus, L_{S_k} is an optimal labeling of S_k.

For simplicity, we denote $\{l | l \in L^*(T_{v_i})$ and $l \leqslant x\}$ by $L_{i|x}$. Then, we define the *greedy-cover labeling* given in [4] as follows:

Definition 12. Let $L = \{(\ell_c,v_c),\ldots,(\ell_2,v_2),(\ell_1,v_1)\}$ be a valid labeling of V_B. A branch i is said to satisfy the *greedy-cover* (abbreviated as *gc*) property if for any branch j with $\ell_j < \ell_i$, $L_{i|\ell_i} \succeq L_{j|\ell_i}$. L is called a *gc labeling* if every branch satisfies the gc property.

Intuitively, a gc labeling assigns the largest label to a branch so as to cover the lexicographically biggest set of labels. For instance, the labelings shown in Fig. 4b and c are gc labelings. The following lemma implies that there is always an optimal labeling that is also gc labeling, but the reverse may not be true.

Lemma 13. *Let $L = \{(\ell_c,v_c),\ldots,(\ell_2,v_2),(\ell_1,v_1)\}$ be a valid labeling of V_B such that $\ell_c > \cdots > \ell_2 > \ell_1$. Then, there is a valid gc labeling whose largest label is also ℓ_c. Moreover, if L is optimal, then the corresponding gc labeling can be made optimal too.*

Proof. We will prove the lemma by induction on the number of branch vertices. Initially, any labeling on a single branch vertex is a gc labeling. Assume that the lemma is true if there are strictly fewer than k branch vertices to be labeled for $k \leqslant c$. That is, there exists a gc labeling on i branches that is optimal for $i < k$. We will show that the lemma is true when the number of branches is k. Let $L = \{(\ell_1,v_1),(\ell_2,v_2),\ldots,(\ell_{k-1},v_{k-1}),(\ell_k,v_k)\}$ be a valid labeling on branch set $V_B = \{v_1,v_2,\ldots,v_{k-1},v_k\}$ such that $\ell_1 < \ell_2 < \cdots < \ell_{k-1} < \ell_k$.

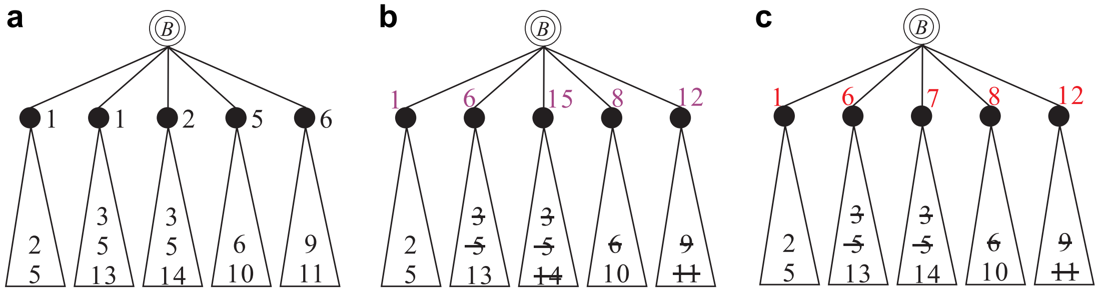

Fig. 4. (a) An initial status of T_B, (b) a gc labeling, and (c) an optimal gc labeling.

Consider that for every $j < k$, $L_{k|\ell_k} \succeq L_{j|\ell_k}$. Then, assigning ℓ_k to v_k makes a valid labeling of V_B. By the induction hypothesis, we can find a gc labeling of $V_B - \{v_k\}$ using no label greater than ℓ_{k-1}, and combining with the labeling (ℓ_k,v_k) makes a gc labeling for the entire branch set of V_B with the largest label ℓ_k. Furthermore, if L is optimal, then the labeling $L - \{(\ell_k,v_k)\}$ is optimal on $V_B - \{v_k\}$ by Lemma 11. By the induction hypothesis, we can find an optimal gc labeling of $V_B - \{v_k\}$, and combining it with (ℓ_k,v_k) gives an optimal gc labeling of V_B.

Next, we consider that there exists a branch $\jmath$ such that $L_{\jmath|\ell_k} \succ L_{k|\ell_k}$. Choose the $\jmath$ that lexicographically maximizes $L_{\jmath|\ell_k}$. We construct a modification of the labeling L that is still valid and as required by the definition of gc labeling, assigns its maximum label ℓ_k to $v_\jmath$. We can then use the induction hypothesis, exactly as above, to fill out a gc labeling for $V_B - \{v_\jmath\}$. Since $L_{\jmath|\ell_k} \succ L_{k|\ell_k}$, there exists a largest integer γ such that $\gamma \in L_{\jmath|\ell_k}$ but $\gamma \notin L_{k|\ell_k}$. There are two cases:

Case 1: $\gamma < \ell_\jmath$. In this case, we modify L to L' by swapping the labels $\ell_\jmath$ and ℓ_k, so that $v_\jmath$ gets label ℓ_k and v_k gets label $\ell_\jmath$; all the other labels stay the same. Observe that by the definition of γ, each integer in the interval between γ and ℓ_k is on both $L^*(T_{v_\jmath})$ and $L^*(T_{v_k})$ or on neither of them. Since $\gamma < \ell_\jmath$ and $\gamma < \ell_k$, every integer less than γ is invisible on both lists in L and L'. Thus, the visible labels from the union of $L^*(T_{v_\jmath})$ and $L^*(T_{v_k})$ are the same in L and L'. Thus, swapping the labels $\ell_\jmath$ and ℓ_k has no impact on the other labels.

Case 2: $\gamma \geqslant \ell_\jmath$. In this case, we modify L to L'' by labeling $v_\jmath$ with ℓ_k and labeling v_k with γ; all the other labels stay the same. Since $\gamma \geqslant \ell_\jmath$ and L is a valid labeling, γ cannot occur as a visible label on $\widehat{L}_i$, $i \neq \jmath$, or as a label in L. When $v_\jmath$ gets label ℓ_k, γ becomes invisible. Since in L'' we rank $v_\jmath$ with ℓ_k greater than γ, we can now reuse γ to rank v_k and still has a valid labeling. By the induction hypothesis, we can find a gc labeling of $V_B - \{v_\jmath\}$, and combining it with $(\ell_k,v_\jmath)$ gives a gc labeling of V_B since $L_{\jmath|\ell_k} \succ L_{i|\ell_k}$ for $v_i \in V_B - \{v_\jmath\}$. On the other hand, suppose that L is optimal. As shown in Case 1, the definition of γ ensures that the integers in the interval between γ and ℓ_k that are left visible in L are the same as the integers left visible in L''. All integers less than γ on $L^*(T_{v_\jmath})$ and $L^*(T_{v_k})$ are invisible in L'', but they might be visible on $L^*(T_{v_\jmath})$ and $L^*(T_{v_k})$ in L. So in terms of optimality, L'' is at least as good as L. Hence, L'' is also optimal since L is optimal. Let $\widetilde{L}'' = L'' - \{(\ell_k,v_\jmath)\}$. By Lemma 11, $\widetilde{L}''$ is an optimal labeling of $V_B - \{v_\jmath\}$. By the induction hypothesis, we can find a gc labeling of $V_B - \{v_\jmath\}$ that is optimal, and combining it with $(\ell_k,v_\jmath)$ gives an optimal gc labeling of V_B.

It follows immediately from the above lemma that we have the following corollary:

Corollary 14. *There is always an optimal labeling of V_B that is also gc labeling.*

Based on the above corollary, we can restrict our search for labelings to the class of gc labelings. The following lemma gives us which gc labeling we are searching for.

Lemma 15. *Among all valid gc labelings of $V_B = \{v_1,v_2,\ldots,v_{c-1},v_c\}$, the labeling that has the lexicographically smallest list of labels is optimal.*

Proof. Consider two distinct valid gc labelings of V_B as follows:

$$L = \{(\ell_c,v_c),(\ell_{c-1},v_{c-1}),\ldots,(\ell_1,v_1)\},$$

$$L' = \{(\ell'_c,v'_c),(\ell'_{c-1},v'_{c-1}),\ldots,(\ell'_1,v'_1)\},$$

that are both sorted in decreasing order of labels; that is, $\ell_c > \ell_{c-1} > \cdots > \ell_1$ and $\ell'_c > \ell'_{c-1} > \cdots > \ell'_1$.

Without loss of generality, assume that L has a lexicographically smaller label list than L'; and in particular, that $\jmath$ is the highest index at which $\ell_\jmath$ is less than $\ell'_\jmath$. All the labels larger than $\ell_\jmath$ agree that $\ell_c = \ell'_c, \ell_{c-1} = \ell'_{c-1}, \ldots, \ell_{\jmath+1} = \ell'_{\jmath+1}$. Now consider the definition of gc labeling. Given that the largest label is *fixed* at ℓ_c, the choice of which branch vertex gets that label is deterministic. Since both L and L' are gc labelings, we can easy to see that $v_c = v'_c$, $v_{c-1} = v'_{c-1}$, ..., $v_{\jmath+1} = v'_{\jmath+1}$.

Let $\widetilde{L} = \cup_{J+1 \leqslant i \leqslant c}\{\{l \in L^*(T_{v_i}) | l > \ell_i\} \cup \{\ell_i\}\}$ and $\widetilde{L}' = \cup_{J+1 \leqslant i \leqslant c}\{\{l \in L^*(T_{v_i}) | l > \ell'_i\} \cup \{\ell'_i\}\}$. Since $\ell'_J > \ell_J$ and $\ell_i = \ell'_i$ for $J+1 \leqslant i \leqslant c$, we get that $\{l \in \widetilde{L} | l > \ell'_J\} = \{l \in \widetilde{L}' | l > \ell'_J\}$; that is, each such integer is either on both $\widetilde{L}$ and $\widetilde{L}'$ or on neither of them. On the other hand, every label in $L^*(T_{v_i})$, $1 \leqslant i \leqslant J$, larger than ℓ'_J will be on both of $L_B(L \cup \varphi)$ and $L_B(L' \cup \varphi)$ since $\ell'_J > \ell_J > \ell_{J-1} > \cdots > \ell_1$. Furthermore, since ℓ'_J is used as a label in L', it will be in $L_B(L' \cup \varphi)$ and it cannot be in any visible list $\widehat{L}_i$ for $1 \leqslant i \leqslant J$. So the integer ℓ'_j is *not* in $L_B(L \cup \varphi)$ induced by the labeling L, and, hence, L is a better labeling than L'.

Since L and L' are arbitrary gc labelings, the above argument shows that any gc labeling that is not the lexicographically smallest will not be optimal. By Corollary 14, there is a gc labeling that is optimal. Therefore, the optimal gc labeling must be the gc labeling with the lexicographically smallest label list.

Based on the above lemma, our algorithm will search for a gc labeling with a lexicographically smallest label list. The good news is that given a fixed label list, there is at most one gc labeling using that list and it can be found in polynomial time. The bad news is that there appear, at first glance, to be exponentially many label lists to consider. We need a search strategy to narrow down the exponential search space in polynomial time. Our strategy is based on the idea of Zhou et al. [26]. We will pin down the label list one value at a time from largest to smallest. That is, for a given prefix of label list $\ell_c, \ell_{c-1}, \ldots, \ell_{J+1}$, we will determine in polynomial time whether there is a gc labeling whose label list starts with this prefix.

3.3. The algorithm

In this subsection, we will propose an algorithm to find an optimal gc labeling of branch set V_B. We first define the following notation:

Definition 16. For a vertex ranking ψ of block graph $G[T_B]$, define M_ψ to be equal to $\max\{\psi(v_i) | 1 \leqslant i \leqslant c\}$, where $v_1, v_2, \ldots, v_c$ are branch vertices of T_B.

Lemma 17. *Let ψ_1 and ψ_2 be two subcritical vertex rankings of $G[T_B]$. If $M_{\psi_1} < M_{\psi_2}$, then $L_B(\psi_1) \prec L_B(\psi_2)$.*

Proof. Since ψ_1 and ψ_2 are two subcritical rankings of $G[T_B]$, we have the following equations:

$$L_B(\psi_1) = \cup_{1 \leqslant i \leqslant c}\{l \in L^*(T_{v_i}) | l > \psi_1(v_i)\} \cup \{\psi_1(v_1), \psi_1(v_2), \ldots, \psi_1(v_c)\}, \tag{1}$$

$$L_B(\psi_2) = \cup_{1 \leqslant i \leqslant c}\{l \in L^*(T_{v_i}) | l > \psi_2(v_i)\} \cup \{\psi_2(v_1), \psi_2(v_2), \ldots, \psi_2(v_c)\}. \tag{2}$$

By Eqs. (1)–(2) and $M_{\psi_1} < M_{\psi_2}$, we have

$$L_B(\psi_1) - [1, M_{\psi_2}] = \cup_{1 \leqslant i \leqslant c} L^*(T_{v_i}) - [1, M_{\psi_2}],$$

$$L_B(\psi_2) - [1, M_{\psi_2}] = \cup_{1 \leqslant i \leqslant c} L^*(T_{v_i}) - [1, M_{\psi_2}].$$

Hence, $L_B(\psi_1) - [1, M_{\psi_2}] = L_B(\psi_2) - [1, M_{\psi_2}]$.
Since ψ_2 is a valid ranking and $M_{\psi_2} \notin \cup_{1 \leqslant i \leqslant c} L^*(T_{v_i})$, we get that $M_{\psi_2} \notin L_B(\psi_1)$ and $M_{\psi_2} \in L_B(\psi_2)$. Therefore, $L_B(\psi_1) \prec L_B(\psi_2)$.

It follows immediately from Lemma 17 that the following corollary holds:

Corollary 18. *The following two statements hold*:

(1) *if ψ is supercritical and $\widehat{\psi}$ is subcritical of $G[T_B]$, then $M_\psi \leqslant M_{\widehat{\psi}}$;*

(2) *every supercritical vertex ranking ψ of $G[T_B]$ has the same value M_ψ of $G[T_B]$.*

We denote by β_{sup} the same value M_ψ for all supercritical vertex rankings ψ of $G[T_B]$, and call β_{sup} the *super rank* of $G[T_B]$. Then, Corollary 18 immediately implies the following result.

Corollary 19. *The super rank β_{sup} of $G[T_B]$ is equal to the minimum integer β for which $G[T_B]$ has a subcritical vertex ranking $\widehat{\psi}$ with $M_{\widehat{\psi}} = \beta$.*

We will later give Algorithm SuperRank to find the super rank β_{sup} of $G[T_B]$ using Corollary 19.

Let β be a positive integer and κ be a branch of T_B such that $L_{\kappa|\beta}$ is the lexicographically largest among all $L_{i|\beta}$'s, $1 \leqslant i \leqslant c$; that is, if v_κ is labeled by β then the list of ranks in $L^*(T_{v_\kappa})$ covered by β will be lexicographically largest (v_κ satisfies the gc property). Then, Lemma 13 implies the following result.

Lemma 20. *Let β be a positive integer and let v_κ be a branch vertex of T_B. Assume that $G[T_B]$ has a subcritical vertex ranking $\widehat{\psi}$ with $M_{\widehat{\psi}} = \beta$. Then, $G[T_B]$ has a subcritical vertex ranking ψ such that $\psi(v_\kappa) = M_\psi = \beta$, v_κ satisfies the gc property, and $L_B(\psi) \preceq L_B(\widehat{\psi})$.*

The following corollary is an immediate consequence of Lemma 20.

Corollary 21. *$G[T_B]$ has a supercritical vertex ranking ψ such that $\psi(v_\kappa) = \beta_{sup}$, where v_κ is a branch vertex of T_B for which $L_{\kappa|\beta_{sup}}$ is the lexicographically largest among $L_{i|\beta_{sup}}$'s for $1 \leqslant i \leqslant c$.*

By the above corollary, we can decide the branch vertex v_κ labeled by β_{sup} if β_{sup} is given; that is, $L_{\kappa|\beta_{sup}} \succeq L_{i|\beta_{sup}}$ for $i \neq \kappa$. The following lemma will show that the supercritical vertex ranking ψ of $G[T_B]$ can be extended by any supercritical vertex ranking ψ' of $G[T_B - T_{v_\kappa}]$. Recall that φ_i is a critical vertex ranking of $G[T_{v_i}]$ for $1 \leqslant i \leqslant c$ and φ is a partial vertex ranking of $G[T_B]$ that labels all vertices of $G[T_B]$ except those of branch set V_B.

Lemma 22. *Assume that ψ is a supercritical vertex ranking of $G[T_B]$. Let v_κ be the branch vertex of T_B such that $\psi(v_\kappa) = \beta_{sup}$ and v_κ satisfies the gc property, and let $T' = T_B - T_{v_\kappa}$. Then, the following statements hold:*

(1) *the vertex ranking $\psi' = \psi_{|G[T']}$ is supercritical of $G[T']$;*

(2) *any supercritical vertex ranking ψ' of $G[T']$ can be extended to a supercritical vertex ranking ψ of $G[T_B]$ as follows:*

$$\psi(v) = \begin{cases} \beta_{sup}, & \text{if } v = v_\kappa; \\ \psi'(v), & \text{if } v \in \{v_1, v_2, \ldots, v_c\} - \{v_\kappa\}; \\ \varphi(v), & \text{if } v \in V(G[T_{v_i}]) - \{v_i\} \text{ for } 1 \leqslant i \leqslant c. \end{cases}$$

Proof. By Lemma 11, Statement (1) immediately holds. By Lemma 8 and Statement (1), Statement (2) holds.

Using Lemma 15, Corollary 21, and Statement (2) of Lemma 22, it is easy to verify that the following algorithm named GC correctly decides the ranks of $v_1, v_2, \ldots, v_c$ if Algorithm SuperRank presented later correctly finds the super rank of $G[T_B]$.

Algorithm GC
Input: The branch set $V_B = \{v_1, v_2, \ldots, v_c\}$ of T_B and the critical lists $L^*(T_{v_1}), L^*(T_{v_2}), \ldots, L^*(T_{v_c})$ of $G[T_{v_1}], G[T_{v_1}], \ldots, G[T_{v_c}]$, respectively.
Output: $L_{gc} = \{(\ell_1, v_1), (\ell_2, v_2), \ldots, (\ell_c, v_c)\}$, the *optimal greedy cover labeling* of V_B and $L^*(T_B)$, the supercritical list of $G[T_B]$.
Method:

1. $T' \leftarrow T_B$; $L_{gc} \leftarrow \emptyset$; $L^*(T_B) \leftarrow \emptyset$;
2. **for** $i = c$ **downto** 1 **do**
3. let $v_1, v_2, \ldots, v_i$ be the children of B in T';
4. find the super rank β_{sup} of T' by Algorithm SuperRank presented later;
5. find a branch κ, $1 \leqslant \kappa \leqslant i$, such that $L_{\kappa|\beta_{sup}}$ is the lexicographically largest and $\beta_{sup} \geqslant \varphi_\kappa(v_\kappa)$;
6. label v_κ with β_{sup};
7. $L_{gc} \leftarrow L_{gc} \cup \{(\beta_{sup}, v_\kappa)\}$;
8. $L^*(T_B) \leftarrow L^*(T_B) \cup \{l \in L^*(T_{v_\kappa}) | l > \beta_{sup}\} \cup \{\beta_{sup}\}$;
9. $T' \leftarrow T' - T_{v_\kappa}$;
10. **Output** L_{gc} and $L^*(T_B)$.

Obviously, line 1 of Algorithm GC can be done in $O(1)$ time. Lines 3–9 are iterated c times. One execution of lines 3 and 5–9 can be done in $O(n)$ time, where n is the number of vertices of $G[T_B]$. Therefore, Algorithm GC runs in $O(cn \log c)$ time if Algorithm SuperRank takes $O(n \log c)$ time. In the following, we will give Algorithm SuperRank for finding β_{sup} of a block tree in $O(n \log c)$ time.

For simplicity, we define some natation as follows:

Definition 23. Let L be a list of labels and let l be an integer. Define $count(L,l)$ as follows:

$$count(L,l) = \begin{cases} 1, & \text{if } l \in L; \\ 0, & \text{otherwise.} \end{cases}$$

Definition 24. Define the *maximum conflict* $\alpha_{\max}$ on $L^*(T_{v_i})$'s, $1 \leqslant i \leqslant c$, as follows: $\alpha_{\max} = 0$ if $L^*(T_{v_i}) \cap L^*(T_{v_j}) = \emptyset$ for $i \neq j$; and $\alpha_{\max} = \max\{l | l \in L^*(T_{v_i}) \cap L^*(T_{v_j})$ for $i \neq j$ and $1 \leqslant i,j \leqslant c\}$ otherwise.

By Corollary 19, in order to find β_{sup}, we need to inspect the existence of a subcritical vertex ranking ψ with $M_\psi = \beta$ for a given integer β. A necessary and sufficient condition for the existence will be given in Lemma 27. Before giving Lemma 27, we first show the following two lemmas.

Lemma 25. *Assume that $G[T_B]$ has a subcritical vertex ranking ψ. Let β be any integer such that $\beta \geqslant M_\psi$ and $\beta \notin \cup_{1\leqslant i\leqslant c} L^*(T_{v_i})$. Then, $G[T_B]$ has a subcritical vertex ranking η with $M_\eta = \beta$.*

Proof. Let $M_\psi = \psi(v_\jmath)$ for some $\jmath$, $1 \leqslant \jmath \leqslant c$. We modify ψ to η by labeling $v_\jmath$ with β; all the other labels stay the same. Since $\Sigma_{1\leqslant i\leqslant c} count(L^*(T_{v_i}),\beta) = 0$ and $\beta \geqslant M_\psi$, we get that $count(L_B(\eta),l) \leqslant count(L_B(\psi),l) \leqslant 1$ for $l \neq \beta$ and $count(L_B(\eta),\beta) = \Sigma_{1\leqslant i\leqslant c} count(L^*(T_{v_i}),\beta) + 1 = 1$. That is, $M_\eta = \beta$ and η is a required subcritical vertex ranking of $G[T_B]$.

Lemma 26. *Assume that $\alpha_{\max} = 0$. Then, $G[T_B]$ has a supercritical vertex ranking φ that is the union of $\varphi_1, \varphi_2, \ldots, \varphi_c$ and $\beta_{sup} = \max\{\varphi_i(v_i) | 1 \leqslant i \leqslant c\}$.*

Proof. Let φ be the union of $\varphi_1, \varphi_2, \ldots, \varphi_c$. Since $\alpha_{\max} = 0$, $L^*(T_{v_i}) \cap L^*(T_{v_j}) = \emptyset$ for $i \neq j$. Thus, φ is a valid vertex ranking of $G[T_B]$. Since $L^*(T_{v_i})$ is the critical list of $G[T_{v_i}]$ for $1 \leqslant i \leqslant c$, $L^*(T_{v_i}) \preceq L_{v_i}(\eta)$ for any vertex ranking η of $G[T_B]$. Thus, $L_B(\varphi) = \cup_{1\leqslant i\leqslant c} L^*(T_{v_i}) \preceq \cup_{1\leqslant i\leqslant c} L_{v_i}(\eta) = L_B(\eta)$. That is, φ is a supercritical vertex ranking of $G[T_B]$. By definition, $\beta_{sup} = M_\varphi = \max\{\varphi_i(v_i) | 1 \leqslant i \leqslant c\}$.

Lemma 27. *Assume that $\alpha_{\max} > 0$, β is a positive integer, and that v_κ is a branch vertex of T_B. Then, $G[T_B]$ has a subcritical vertex ranking ψ such that $\psi(v_\kappa) = M_\psi = \beta$ if and only if the following conditions hold:*

(1) $\alpha_{\max} < \beta$;

(2) either $(\varphi_\kappa(v_\kappa) \neq \beta$ *and* $\sum_{i=1}^{c} count(L^*(T_{v_i}),\beta) = 0)$ *or* $(\varphi_\kappa(v_\kappa) = \beta$ *and* $\sum_{i=1; i\neq\kappa}^{c} count(L^*(T_{v_i}), \beta) = 0)$; *and*

(3) $G[T'] = G[T_B - T_{v_\kappa}]$ *has a subcritical vertex ranking* ψ' *with* $M_{\psi'} < \beta$.

Proof. *Only if* part: Let ψ be a subcritical vertex ranking of $G[T_B]$ such that $\psi(v_\kappa) = M_\psi = \beta$. We will prove Conditions (1)–(3) hold. Since $\alpha_{\max} > 0$, we have that $c \geqslant 2$ and there exist two branches s,t such that $L^*(T_{v_s}) \cap L^*(T_{v_t}) \neq \emptyset$.

Assume by contradiction that $\beta \leqslant \alpha_{\max}$. Then, we have that $count(L_B(\psi),\alpha_{\max}) = \sum_{i=1}^{c} count(L_{v_i}(\psi),\alpha_{\max}) \geqslant 2$. This contradicts that ψ is a valid vertex ranking of $G[T_B]$. Thus, Condition (1) holds.

Next, we prove that Condition (2) holds. Assume by contradiction that $\varphi_\kappa(v_\kappa) \neq \beta$ and $\sum_{i=1}^{c} count(L^*(T_{v_i}),\beta) \geqslant 1$. Then, $\beta \in L^*(T_{v_\jmath})$ for some $\jmath \neq \kappa$. Since $\psi(v_\kappa) = M_\psi = \beta$, we get that $count(L_B(\psi),\beta) = \sum_{i=1}^{c} count(L_{v_i}(\psi),\beta) = count(L_{v_\kappa}(\psi),\beta) + \sum_{i=1; i\neq\kappa}^{c} count(L_{v_i}(\psi),\beta) \geqslant 2$. This contradicts ψ is a valid vertex ranking of $G[T_B]$. Thus, $\varphi_\kappa(v_\kappa) = \beta$ or $\sum_{i=1}^{c} count(L^*(T_{v_i}),\beta) = 0$. On the other hand, suppose that $\varphi_\kappa(v_\kappa) = M_\psi = \beta$. Then, $count(L^*(T_{v_\kappa}),\beta) = 1$ and $count(L_B(\psi),\beta) = \sum_{i=1}^{c} count(L^*(T_{v_i}),\beta) = count(L^*(T_{v_\kappa}),\beta) + \sum_{i=1; i\neq\kappa}^{c} count(L^*(T_{v_i}),\beta) \leqslant 1$. Thus, $\sum_{i=1; i\neq\kappa}^{c} count(L^*(T_{v_i}),\beta) = 0$. By the above arguments, Condition (2) immediately holds.

By the definition of subcritical rankings, $\psi_{|G[T']} = \psi'$ is a subcritical ranking of $G[T']$. Moreover, $M_{\psi'} < M_\psi = \beta$. Therefore, Condition (3) holds.

If part:

Suppose that Conditions (1)–(3) hold. Let ψ' be a subcritical vertex ranking of $G[T']$ with $M_{\psi'} < \beta$. Define ψ to be a vertex ranking of $G[T_B]$ as follows:

$$\psi(v) = \begin{cases} \beta, & \text{if } v = v_\kappa; \\ \varphi_\kappa(v), & \text{if } v \in V(G[T_{v_\kappa}]) - \{v_\kappa\}; \\ \psi'(v), & \text{otherwise.} \end{cases}$$

Let $\widetilde{L}_\kappa = \{\beta\} \cup \{l \in L^*(T_{v_\kappa}) | l > \beta\}$ and $L' = \cup_{1\leqslant i\leqslant c; i\neq\kappa}(\{l \in L_{v_i}(\psi) | l > \psi(v_i)\} \cup \{\psi(v_i)\})$. Then, $\psi(v_\kappa) = M_\psi = \beta$ and

$$L_B(\psi) = L' \cup \widetilde{L}_\kappa. \tag{3}$$

Now, we will prove that for every $l \geqslant 1$ the following equation is satisfied:

$$count(L_B(\psi),l) \leqslant 1. \tag{4}$$

That is, ψ is a valid subcritical vertex ranking of $G[T_B]$. If $l \notin \widetilde{L}_\kappa$, then by Eq. (3), $count(L_B(\psi),l) = count(L',l) \leqslant 1$. If $l \in \widetilde{L}_\kappa - \{\beta\}$, then by Eq. (3), $l > \beta > \alpha_{\max}$ and $count(L_B(\psi),l) = \sum_{i=1}^{c} count(L_{v_i}(\psi),l) = count(L^*(T_{v_\kappa}),l) \leqslant 1$. On the other hand, $count(L_B(\psi),\beta) = \sum_{i=1}^{c} count(L^*(T_{v_i}),\beta) + 1 = 1$ if $\sum_{i=1}^{c} count(L^*(T_{v_i}),\beta) = 0$; and $count(L_B(\psi),\beta) = \sum_{i=1; i\neq\kappa}^{c} count(L^*(T_{v_i}),\beta) + count(L^*(T_{v_\kappa}),\beta) = 1$ if $\sum_{i=1; i\neq\kappa}^{c} count(L^*(T_{v_i}),\beta) = 0$. By the above arguments, Eq. (4) is true. Hence, ψ is a valid vertex ranking of $G[T_B]$ and is subcritical.

By Lemma 26, if $\alpha_{\max} = 0$, then a supercritical vertex ranking φ of $G[T_B]$ can be obtained by uniting $\varphi_1, \varphi_2, \ldots, \varphi_c$ and $\beta_{sup} = \max\{\varphi_i(v_i) | 1 \leqslant i \leqslant c\}$. By Lemma 27, if $\alpha_{\max} \geqslant 1$, then $\beta_{sup} > \alpha_{\max}$. Let $\beta_0 = \max\{\varphi_1(v_1), \varphi_2(v_2), \ldots, \varphi_c(v_c)\}$ and $\beta_1 < \beta_2 < \cdots < \beta_{c-1}$ be the smallest $c-1$ integers that are greater than $\alpha_{\max}$ and appear in none of the lists $L^*(T_{v_i})$ for $1 \leqslant i \leqslant c$. Then, we have the following two lemmas.

Lemma 28. *The block graph $G[T_B]$ has a subcritical vertex ranking η with $M_\eta \leqslant \max\{\beta_0, \beta_{c-1}\}$, and $G[T_B]$ has no subcritical vertex ranking ψ with $M_\psi < \min\{\beta_0, \beta_1\}$.*

Proof. Let v_κ be the branch vertex of T_B such that $\varphi_\kappa(v_\kappa) = \beta_0$. Let η be defined as follows:

$$\eta(v) = \begin{cases} \varphi_i(v), & \text{if } v \in V(G[T_{v_i}]) - \{v_i\} \text{ for } 1 \leqslant i \leqslant c; \\ \beta_0, & \text{if } v = v_\kappa; \\ \beta_i, & \text{if } v = v_i \text{ for } i \neq \kappa. \end{cases}$$

Note that η assigns β_i, $1 \leqslant i \leqslant c-1$, to an arbitrary branch vertex except v_κ. Then, $count(L_B(\eta), \alpha_{\max}) \leqslant 1$ and $count(L_B(\eta), l) \leqslant 1$ for $l \in [1, n]$, where n is the number of vertices of $G[T_B]$. That is, the vertex ranking η of $G[T_B]$ is valid and subcritical. In addition, $M_\eta = \beta_{c-1}$ if $\beta_0 < \beta_{c-1}$ and $M_\eta = \beta_0$ if $\beta_{c-1} < \beta_0$.

Let ψ be any subcritical vertex ranking of $G[T_B]$. By Lemmas 26 and 27, we get that (1) if $\alpha_{\max} = 0$, then $M_\psi = \beta_0$; and (2) if $\alpha_{\max} > 0$, then $M_\psi > \alpha_{\max}$ and $\sum_{i=1}^{c} count(L^*(T_{v_i}), M_\psi) = 0$. By definition of β_1, $\beta_1 \leqslant M_\psi$. Thus, $M_\psi \geqslant \min\{\beta_0, \beta_1\}$.

Corollary 18 and Lemmas 25–28 immediately imply the following lemma.

Lemma 29. *The following two statements hold*:

(1) $\beta_{sup} \in \{\beta_0\} \cup \{\beta_1, \beta_2, \ldots, \beta_{c-1}\}$; *and*
(2) *let γ be an integer such that $1 \leqslant \gamma \leqslant c-1$, then $\beta_{sup} \in \{\beta_0\} \cup \{\beta_1, \beta_2, \ldots, \beta_\gamma\}$ if and only if $G[T_B]$ has a subcritical ranking η with $M_\eta = \beta_\gamma$.*

If $G[T_B]$ has a subcritical vertex ranking η with $M_\eta = \beta$, then $\beta \in \{\beta_0, \beta_1, \ldots, \beta_{c-1}\}$. By Lemmas 26 and 27, we can easily derive the following recursive procedure, named Check, to determine whether or not $G[T_B]$ has a subcritical vertex ranking η with $M_\eta = \beta$ by inspecting Conditions (1)–(3) of Lemma 27.

Procedure Check(T_B, β)
Input: T_B, $L^*(T_{v_i})$'s for $1 \leqslant i \leqslant c$, and β.
Output: True, if $G[T_B]$ has a subcritical vertex ranking η with $M_\eta = \beta$; and False, otherwise.
Method:

1. **if** the root B of $G[T_B]$ has exactly one child, **then return** true;
2. let $v_1, v_2, \ldots, v_c$ be the children of B in T_B;
3. **if** $\sum_{i=1}^{c} count(L^*(T_{v_i}), l) \leqslant 1$ for $1 \leqslant l \leqslant |V(G[T_B])|$, **then return** true;
4. $\alpha_{\max} \leftarrow \max\{l | l \in L^*(T_{v_i}) \cap L^*(T_{v_j}) \text{ for } i \neq j\}$;
5. **if** $\alpha_{\max} \geqslant \beta$, **then return** false;
6. let κ be the branch of T_B for which $L_{\kappa|\beta}$ is the lexicographically largest among all $L_{i|\beta}$'s for $1 \leqslant i \leqslant c$;
7. **if** ($\varphi_\kappa(v_\kappa) \neq \beta$ and $\sum_{i=1}^{c} count(L^*(T_{v_i}), \beta) \neq 0$) or ($\varphi_\kappa(v_\kappa) = \beta$ and $\sum_{i=1; i\neq\kappa}^{c} count(L^*(T_{v_i}), \beta) \neq 0$), **then return** false;
8. $T'_B \leftarrow T_B - T_{v_\kappa}$;
9. let β' be the largest integer such that $\beta' < \beta$ and $\beta' \notin \cup_{1 \leqslant i \leqslant c; i \neq \kappa} L^*(T_{v_i})$;
10. call Check(T'_B, β');

By Corollary 19 and Lemmas 26–27, β_{sup} is the smallest integer satisfying Conditions (1)–(3) of Lemma 27. Therefore, we have the following algorithm called SuperRank to find β_{sup} of $G[T_B]$.

Algorithm SuperRank
Input: T_B and $L^*(T_{v_i})$'s for $1 \leqslant i \leqslant c$.
Output: β_{sup}, the super rank of $G[T_B]$.
Method:

1. $\mathcal{K} \leftarrow \{\beta_0, \beta_1, \ldots, \beta_{c-1}\}$;
2. choose the smallest integer $\beta \in \mathcal{K}$ satisfying the subcritical conditions in Lemma 27 by calling Procedure Check;
3. $\beta_{sup} \leftarrow \beta$;
4. **Output** β_{sup}.

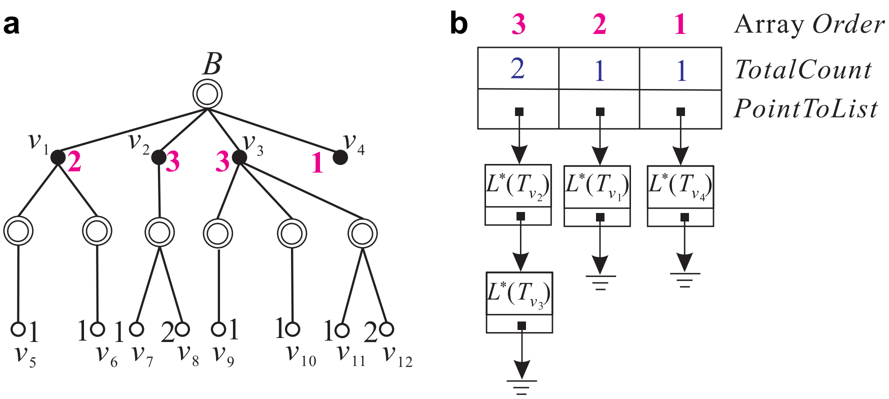

Fig. 5. (a) A subcritical vertex ranking φ of $G[T_B]$, and (b) the data structure of array *Order* under φ [26].

Zhou et al. [26] proposed an efficient approach for searching an index κ such that $L^*(T_{v_\kappa}) \cap [1,\beta]$ is the lexicographically largest among all $L^*(T_{v_i}) \cap [1,\beta]$ for $1 \leqslant i \leqslant c$. They define a data structure called *Order* to support their method. The array *Order* consists of records, each of which contains two items of data: *TotalCount* and *PointerToList*. The length ω of *Order* is the largest integer in $\cup_{1 \leqslant i \leqslant c} L^*(T_{v_i})$. For each integer l, $1 \leqslant l \leqslant \omega$, the item $TotalCount[l] = \sum_{i=1}^{c} count(L^*(T_{v_i}),l)$. The item $PointToList[l]$ stores the lists $L^*(T_{v_i})$ for $l \in L^*(T_{v_i})$. For example, Fig. 5b illustrates the data structure *Order* of the subcritical vertex ranking φ on Fig. 5a ranking all nodes except the branch vertices of T_B. Then, they use the technique of radix sorting to sort $L^*(T_{v_i})$'s in array *Order*, and it can be done in $O(n)$ time [1]. Then all $L^*(T_{v_\kappa})$'s can be decreasingly picked from array *Order* in $O(n)$ time [26]. We then have the following lemma:

Lemma 30 ([26]). Procedure Check(T_B,β) takes $O(n)$ time, where n is the number of vertices in $G[T_B]$.

We can use the binary search technique to find the smallest integer β_{sup}. Hence, Procedure Check is called at most $\log c$ times. By Lemma 30, Algorithm SuperRank runs in $O(n \log c)$ time. Since the block node B has c children, we totally need to find β_{sup} c times. Thus, we label the branches of T_B in $O(cn \log c)$ time. Note that $c \leqslant \Delta$, where Δ represents the maximum degree of the input block graph $G[T_\mathcal{B}]$. Therefore, an optimal vertex ranking (supercritical ranking) of $G[T_\mathcal{B}]$ can be found in $O((c_1 + c_2 + \cdots + c_k)n \log \Delta)$ time, where k is the number of block nodes in $T_\mathcal{B}$ and c_i, $1 \leqslant i \leqslant k$, is the number of children of block node B_i in $G[T_\mathcal{B}]$. Obviously, $c_1 + c_2 + \cdots + c_k = O(n)$. Thus, processing all block nodes in $G[T_\mathcal{B}]$ needs $O(n^2 \log \Delta)$ time. By the result and Theorem 7, we conclude the following theorem:

Theorem 31. *Given a block graph with n vertices and maximum degree Δ, the vertex ranking problem can be solved in $O(n^2 \log \Delta)$ time.*

The following example shows how to construct an optimal gc labeling by Algorithm GC.

Example. Given an initial subcritical ranking of T_B shown in Fig. 6a. In Fig. 6a, $\mathcal{K} = \{6,7,8,12,15\}$. Using binary search strategy, we first pick β to be 8. Then, call procedure Check$(T_B, 8)$. In Check$(T_B, 8)$, $|V_B| = 5$, $\alpha_{\max} = 6$, $\kappa = 1$ (branch 1 has the largest $L_{i|8}$) and $\beta' = 7$. Then, $T_B = T_B - T_{v_1}$ and call Check$(T_B, 7)$. In Check$(T_B, 7)$, $|V_B| = 4$, $\alpha_{\max} = 5$, $\kappa = 2$, and $\beta' = 4$. Then, $T_B = T_B - T_{v_2}$ and call Check$(T_B, 4)$. In Check$(T_B, 4)$, $|V_B| = 3$ and $\alpha_{\max} = 5$. Since $\alpha_{\max} > 4 = \beta$, Check$(T_B, 4)$ returns false. Hence, $\beta_{sup} \neq 8$. We then select β to be 12. Following the above checking, we can find Check$(T_B, 12)$ returns true. Therefore, $\beta_{sup} = 12$. Consequently, branch 2 gets label 12 since $\ell_2 \leqslant 12$ and $L_{2|12}$ is the lexicographically largest among all $L_{i|12}$'s. Then, $L_{gc} = \{(12,v_2)\}$, $L^*(T_B) = \{12\}$, and $T' = T_B - T_{v_2}$. The resultant labeling is shown in Fig. 6b. Continue to find the other super ranks in Fig. 6c–e, we finally get an optimal gc labeling $L_{gc} = \{(8,v_1),(12,v_2),(1,v_3),(6,v_4),(7,v_5)\}$ and $L^*(T_B) = \{1,2,5,6,7,8,10,12,13,14\}$ shown in Fig. 6f.

4. Concluding remarks

In this paper, we present an $O(n^2 \log \Delta)$-time algorithm to solve the vertex ranking problem on block graphs. We use the tree structure, called block tree $T_\mathcal{B}$, of a block graph to investigate the problem. We traverse $T_\mathcal{B}$ in a bottom-up manner. When the cut node is visited, we assign a new rank to it. If one block node B is visited, we apply the edge ranking algorithm of trees in [26] to rank the branches of T_B and obtain a supercritical vertex ranking of $G[T_B]$. It is interesting to see if the structure of block tree can be applied to the other problems on block graphs. On the other hand, whether there exists an efficient

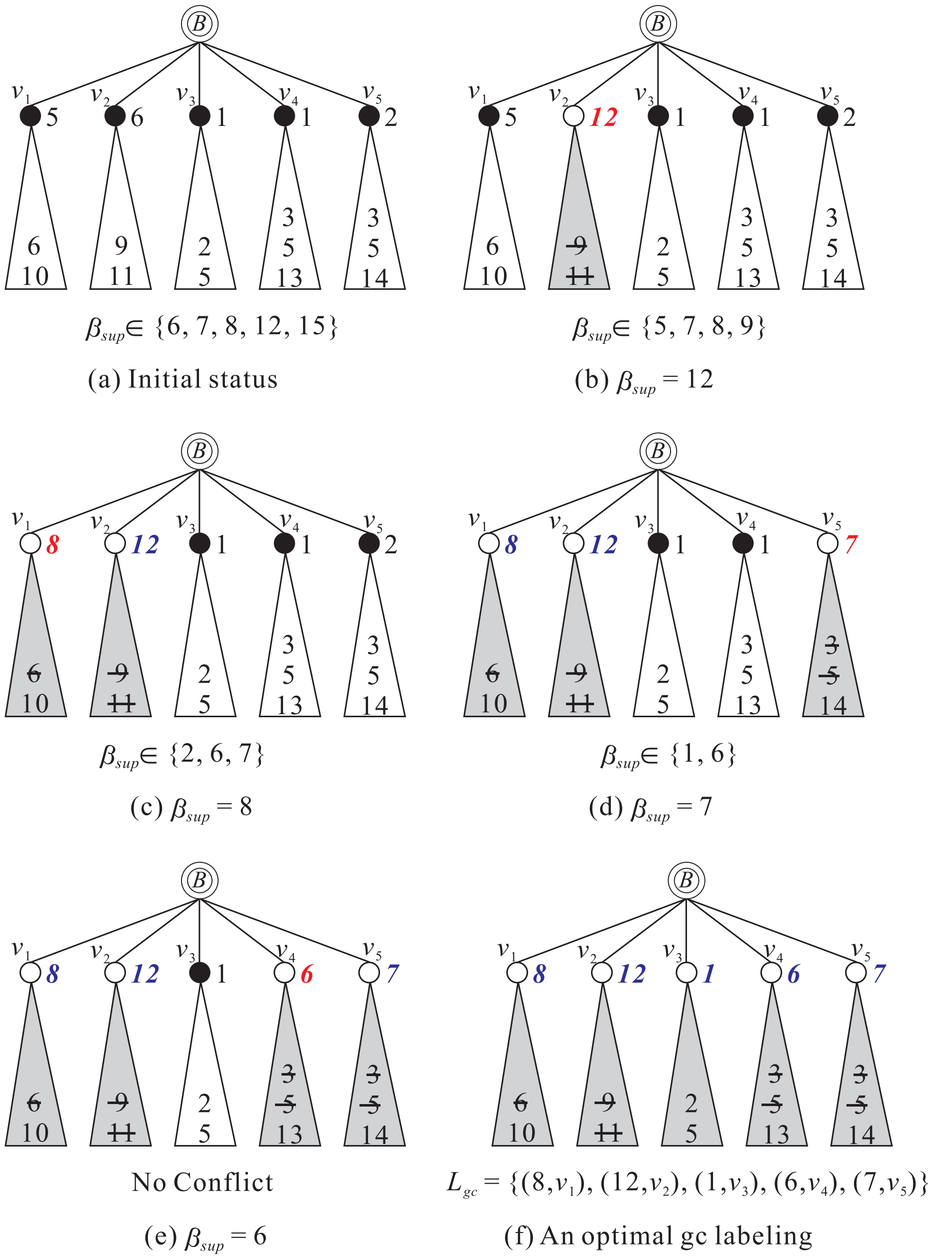

Fig. 6. (a) Initial subcritical vertex ranking, (b) the first super rank is 12, (c) the second super rank is 8, (d) the third super rank is 7, (e) the fourth super rank is 6, and (f) an optimal gc labeling $L_{gc} = \{(8,v_1),(12,v_2),(1,v_3),(6,v_4),(7,v_5)\}$ and the supercritical list $L^*(T_B) = \{1,2,5,6,7,8,10,12,13,14\}$.

algorithm whose time-complexity is better than $O(n^2 \log \Delta)$ for solving the vertex ranking problem on block graphs remains open.

Acknowledgments

The author would thank anonymous referees for many useful comments and suggestions which have improved the presentation of this paper.

References

[1] A.V. Aho, J.E. Hopcroft, J.D. Ullman, The Design and Analysis of Computer Algorithms, Addison-Wesley, Reading, MA, 1974.
[2] B. Aspvall, P. Heggernes, Finding minimum height elimination trees for interval graphs in polinomial time, BIT 34 (1994) 484–509.

[3] H.L. Bodlaender, J.S. Deogun, K. Jansen, T. Kloks, D. Kratsch, H. Müller, Z. Tuza, Rankings of graphs, SIAM J. Discrete Math. 11 (1998) 168–181.
[4] P. De La Torre, R. Greenlaw, A.A. Schäffer, Optimal edge ranking of trees in polynomial time, Algorithmica 13 (1995) 592–618.
[5] J.S. Deogun, Y. Peng, Edge ranking of trees, Congr. Numer. 79 (1990) 19–28.
[6] J.S. Deogun, T. Kloks, D. Kratsch, H. Müller, On vertex ranking for permutation and other graphs, Lecture Notes in Computer Science, Springer-Verlag, Berlin, 1994, pp. 747–758.
[7] J.S. Deogun, T. Kloks, D. Kratsch, H. Müller, On vertex ranking for trapezoid, circular-arc and other graphs, Discrete Appl. Math. 98 (1999) 39–63.
[8] D. Dereniowski, A. Nadolski, Vertex rankings of chordal graphs and weighted trees, Inform. Process. Lett. 98 (2006) 96–100.
[9] L.T. Quoc Hung, M.M. Sysło, M.L. Weaver, D.B. West, Bandwidth and density for block graphs, Discrete Math. 189 (1998) 163–176.
[10] A.V. Iyer, H.D. Ratliff, G. Vijayan, Optimal node ranking of trees, Inform. Process. Lett. 28 (1988) 225–229.
[11] A.V. Iyer, H.D. Ratliff, G. Vijayan, On an edge ranking problem of trees and graphs, Discrete Appl. Math. 30 (1991) 43–52.
[12] M. Katchalski, W. Mccuaig, S. Seager, Ordered colourings, Discrete Math. 142 (1995) 141–154.
[13] T. Kloks, H. Müller, C.K. Wong, Vertex ranking of asteroidal triple-free graphs, Inform. Process. Lett. 68 (1998) 201–206.
[14] T.W. Lam, F.L. Yue, Edge ranking of graphs is hard, Discrete Appl. Math. 85 (1998) 71–86.
[15] T.W. Lam, F.L. Yue, Optimal edge ranking of trees in linear time, Algorithmica 30 (2001) 12–33.
[16] C.E. Leiserson, Area efficient graph layouts for VLSI, in: Proceedings of the 21st Annual IEEE Symposium on Foundations of Computer Science (FOCS'80), 1980, pp. 270–281.
[17] J.W.H. Liu, The role of elimination trees in sparse factorization, SIAM J. Matrix Anal. Appl. 11 (1990) 134–172.
[18] D.C. Llewellyn, C. Tovey, M. Trick, Local optimization on graphs, Discrete Appl. Math. 23 (1989) 157–178.
[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, The Pennsylvania State University, 1988.
[21] J.W.E. Robert, Complexity and block graphs, Congr. Numer. 33 (1981) 129–142.
[22] A.A. Schäffer, Optimal node ranking of trees in linear time, Inform. Process. Lett. 33 (1989/1990) 91–96.
[23] P. Scheffler, Node ranking and searching on graphs (Abstract), in: U. Faigle, C. Hoede (Eds.), in: The 3rd Twente Workshop on Graphs and Combinatorial Optimization, Memorandum No. 1132, Faculty of Applied Mathematics, University of Twente, The Netherlands, 1993.
[24] J.D. Ullman, Computational Aspects of VLSI, Computer Science Press, Rockville, MD, 1984.
[25] P.K. Wong, Optimal path cover problem on block graphs, Theoret. Comput. Sci. 225 (1999) 163–169.
[26] X. Zhou, M.A. Kashem, T. Nishizeki, Generalized edge-rankings of trees, IEICE Trans. Fund. Electron., Commun. Comput. Sci. 81-A-2 (1998) 310–320.
[27] X. Zhou, T. Nishizeki, Finding optimal edge-rankings of trees, in: Proceedings of the 6th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'95), 1995, pp. 122–131.