L(2,1)-labeling of Block Graphs

B. S. Panda* and Preeti Goel†
Computer Science and Application Group
Department of Mathematics
Indian Institute of Technology Delhi
Hauz Khas, New Delhi 110 016, India
E-mail: bspanda@maths.iitd.ac.in

Abstract

An $L(2,1)$-*labeling* of a graph $G(V, E)$ is a function f from its vertex set V to the set of nonnegative integers such that $|f(x) - f(y)| \geq 2$ if $xy \in E$ and $|f(x) - f(y)| \geq 1$ if x and y are at distance two apart. The span of an $L(2,1)$-labeling f is the maximum value of $f(x)$ over all vertices x of G. The $L(2,1)$-labeling number of a graph G, denoted as $\lambda(G)$, is the least integer k such that G has an $L(2,1)$-labeling of span k. Chang and Kuo [1996, SIAM J. Discrete Mathematics, Vol 9, No. 2, pp. 309 – 316] proved that $\lambda(G) \leq 2\Delta(G)$ and conjectured that $\lambda(G) \leq \Delta(G) + \omega(G)$, for a strongly chordal graph G, where $\Delta(G)$ and $\omega(G)$ are the maximum degree and maximum clique size of G, respectively. In this paper, we propose an algorithm for $L(2,1)$-labeling a block graph G with $\Delta(G)+\omega(G)+1$ colors. As block graphs are strongly chordal graphs, our result proves Chang and Kuo's conjecture for block graphs. We also obtain better bounds of $\lambda(G)$ for some special subclasses of block graphs. Finally, we investigate to find the exact value of $\lambda(G)$ of a block graph G.

Keywords: $L(2,1)$-labeling, Radio Coloring, Block Graphs, Graph Algorithms.

AMS Subject Classifications: 05C15, 05C35, 05C78, 05C85, 68R10.

1 Introduction

The radio frequency assignment problem is to assign frequencies to radio transmitters at different locations such that nearby transmitters are

*Corresponding Author
†This author was supported by Council of Scientific & Industrial Research, INDIA.

assigned frequencies without causing interference. This problem was introduced by Hale [12] and can be modeled as graph coloring problem where the vertices represent the transmitters and two vertices are adjacent if there is possible interference between the corresponding two transmitters.

$L(2,1)$-labeling problem was introduced by Griggs and Yeh [11], initially proposed by Roberts, as a variation of the frequency assignment problem. An $L(2,1)$-*labeling* of a graph $G(V, E)$ is a function f from its vertex set V to the set of nonnegative integers such that $|f(x) - f(y)| \geq 2$ if $xy \in E$ and $|f(x) - f(y)| \geq 1$ if x and y are distance two apart. The ***span*** of an $L(2,1)$-labeling f of G is $\max\{f(v)|v \in V(G)\}$. The $L(2,1)$-labeling number of G, denoted by $\lambda(G)$, is the smallest k such that G has an $L(2,1)$-labeling of span k.

Griggs and Yeh [11] proved that $\lambda(G) \leq \Delta^2 + 2\Delta$ for a graph G with maximum degree Δ. Chang and Kuo [7] later improved this bound to $\Delta^2 + \Delta$. It was further improved to $\Delta^2 + \Delta - 1$ in [15] and the current best bound that $\lambda(G) \leq \Delta^2 + \Delta - 2$ is due to Goncalves [10].

Griggs and Yeh [11] have proposed the following conjecture.

Conjecture 1. *[11] For any graph G with maximum degree $\Delta \geq 2$, $\lambda(G) \leq \Delta^2$.*

Havet et al. [13] have proved this conjecture asymptotically. However, the conjecture is still open in general and has been the motivating factor for considerable research on $L(2,1)$-labeling of graphs. This conjecture has been proved to be true for several special classes of graphs such as paths, cycles, wheels, complete k-partite graphs [11], trees [7, 11], cographs[7], regular tiling of the plane [2, 6], graphs of maximum degree two, chordal graphs, unit interval graphs [16], OSF-chordal, SF-chordal [7], outerplanar [3, 6], split graphs, permutation graphs [3], co-comparability graphs [5], and Hamiltonian cubic graphs [14]. For a comprehensive survey on this problem we refer to [4].

It was shown in [11] that the problems of determining $\lambda(G)$ of a graph G is NP-hard even for graphs with diameter two. It was further shown in [8] that deciding whether $\lambda(G) \leq k$ for any fixed integer $k \geq 4$ is NP-complete.

Chang and Kuo [7] proved that $\lambda(G) \leq 2\Delta$ for an OSF-chordal graph G (see Section 2 for definition). The class of OSF-chordal graphs properly contains strongly chordal graphs, directed path graphs, interval graphs, unit interval graphs, block graphs, and trees. They [7] conjectured that $\lambda(G) \leq \omega(G) + \Delta(G)$ for a strongly chordal graph G, where $\omega(G)$ is the clique number of G. However, their conjecture is still open.

In this paper, we show that $\lambda(G) \leq \omega(G) + \Delta(G)$ for a block graph G. This proves that the conjecture that $\lambda(G) \leq \omega(G) + \Delta(G)$ is true for a block graph which is an OSF graph.

The rest of the paper is organized as follows. Section 2 presents some definition and preliminary results. Section 3 presents a linear time algorithm to $L(2,1)$-label a block graph G using at most $\Delta(G)+\lambda(G)+1$ colors. This section also presents the complexity analysis of the algorithm and its proof of correctness. Section 4 presents improved bound for $\lambda(G)$ for some special subclasses of block graphs. Section 5 gives some properties of block graphs having $\lambda = \Delta + \omega$. Section 6 presents k-$L(2,1)$-labeling of block graphs. Finally, Section 7 concludes the paper with indication of some open problems.

2 Preliminaries

For a graph $G = (V, E)$, let $N_G(v) = \{u \in V | uv \in E\}$ and $N_G[v] = N_G(v) \cup \{v\}$ denote the *neighborhood* and the *closed neighborhood* of the vertex v, respectively. The *degree* of a vertex v is $|N_G(v)|$ and is denoted by $d_G(v)$. Let $\Delta(G)$ denote the maximum of the degrees of all vertices of G. When the context is clear, we can omit the index G. A subset of pairwise non-adjacent vertices is called an *independent set*, and a subset of pairwise adjacent vertices is called a *clique*. The size of the maximum size clique in G is known as the *clique number* and is denoted as $\omega(G)$. Let $d(u,v)$ denote the *distance* between the vertices u and v in $V(G)$. Let n and m denote the number of vertices, and number edges of G, respectively. Let $G[S]$, $S \subset V(G)$, denote the subgraph induced by G on S. A *proper k-coloring* of a graph G is an assignment of integers (*colors*) from the set $\{1, 2, \ldots, k\}$ to the vertices of G such that two vertices receive different colors whenever they are adjacent. The minimum k such that G admits a proper k-coloring is known as *chromatic number* of G and is denoted as $\chi(G)$. A graph is *chordal* if every cycle of length greater than three has a chord, i.e. an edge joining two non-consecutive vertices of the cycle. A *tree* T is a connected graph that has no cycles. A *rooted tree* T is a tree in which some vertex is distinguished as the *root*. The vertices of a rooted tree are also called nodes. Let x be a vertex in a rooted tree T with root r. The level of x, denoted by $l(x)$, is the length of the unique path from r to x in T. Any node y on the unique path from r to x is called an *ancestor* of x. If y is an ancestor of x, then x is called a *descendant* of y. If y is a descendant of x and xy is an edge, then x is called the *parent* of y and y is called a *child* of x. Clearly, root has no parent and leaf node has no child and all non-leaf nodes are the *internal nodes*.

A *k-sun* is a graph with vertex set $\{v_1, v_2, \ldots, v_k\} \cup \{x_1, x_2, \ldots, x_k\}$ such that $\{v_1, v_2, \ldots, v_k\}$ forms a clique and $\{x_1, x_2, \ldots, x_k\}$ forms an independent set and x_i is adjacent to v_i and v_{i+1}, for $1 \le i \le k-1$ and x_k is adjacent to v_k and v_1. A k-sun is called an *odd sun* if k is odd. If k is even, then a k-sun is called an *even sun.* A chordal graph which contains no k-sun with $k \geq 3$ as an induced subgraph is called as *SF-chordal* graph, where SF stands for sun-free. SF-chordal graphs are also called *strongly chordal graphs.* Odd-sun-free chordal graphs are called *OSF-chordal.* The bounds on $\lambda(G)$ for OSF-chordal graphs and strongly chordal graphs are due to Chang and Kuo [7] and are given below.

Theorem 2.1 ([7]). *$\lambda(G) \leq 2\Delta$ for any OSF-chordal graph G with maximum degree Δ.*

Theorem 2.2 ([7]). *$\lambda(G) \leq \Delta + 2\chi(G) - 2$ for any strongly chordal graph G with maximum degree Δ.*

Since every strongly chordal graph is OSF-chordal, the above two bounds hold for strongly chordal graphs. However, the above two bounds are not comparable. Chang and Kuo [7] have made the following conjecture.

Conjecture 2. *For any strongly chordal graph G with maximum degree $\Delta(G)$, $\lambda(G) \leq \Delta(G) + \chi(G)$.*

A vertex x of a connected graph G is called a *cut-vertex* of G if $G - x$ is disconnected. A maximal connected induced subgraph without a cut-vertex is called a ***block*** of G. A graph G is called a ***block graph*** if each block of G is a complete graph. The intersection of two distinct blocks can contain at most one vertex. Two blocks are called *adjacent blocks* if they have a common cut vertex of G. A block graph with one or more cut-vertices contains at least two blocks, each of which contains exactly one cut-vertex; we call such blocks *end blocks* [1]. The class of block graphs includes all trees and is a subclass of strongly chordal graphs. Therefore, by Theorems 2.1 and 2.2, $\lambda(G) \leq 2\Delta(G)$ and $\lambda(G) \leq \Delta(G) + 2\chi(G) - 2$ for a block graph G. Note that $\chi(G) = \omega(G)$ for a block graph G. So, $\lambda(G) \leq \Delta(G) + 2\omega(G) - 2$.

3 $L(2,1)$-labeling of block graphs

In this section, we present an $L(2,1)$-labeling algorithm of block graphs using at most $\Delta + \omega + 1$ colors.

Let $\alpha = v_1, v_2, \ldots, v_n$ be an ordering of $V(G)$. The following algorithm, known as Greedy-$L(2,1)$-labeling, finds always an $L(2,1)$-labeling f of G.

Algorithm 1: Greedy-$L(2,1)$-labeling(α)

```
S = ∅;
foreach i = 1 to n do
    Let j be the smallest non-negative integer such that
    j ∉ ({f(v), f(v) − 1, f(v) + 1 | v ∈ N_G(v_i) ∩ S} ∪ {f(w) | w ∈ S and
    d(v_i, w) = 2});
    f(v_i) = j;
    S = S ∪ {v_i};
end
```

If α is taken to be an arbitrary ordering of $V(G)$ of a block graph G of maximum degree Δ, then the above greedy algorithm takes at most $3\Delta + \Delta(\Delta - 1) = \Delta^2 + 2\Delta + 1$ colors. This gives $\lambda(G) \leq \Delta^2 + 2\Delta$. So some special ordering needs to be considered to get better upper bound of $\lambda(G)$. Let $\alpha = u_1, u_2, \ldots, u_n$ be an ordering of $V(G)$ of a block graph G having maximum clique size ω such that u_i is a non-cut vertex of $G_i = G[\{u_i, u_{i+1}, \ldots, u_n\}]$. Note that u_i lies in exactly one block of G_i. Consider the ordering $\beta = v_1, v_2, \ldots, v_n$ such that $v_i = u_{n+1-i}, 1 \leq i \leq n$. For this ordering, the number of forbidden colors for v_i by the above greedy algorithm is $3d_{G_i}(v_i) + d_{G_i}(v_i)(\Delta - d_{G_i}(v_i)) \leq 3(\omega - 1) + (\omega - 1)(\Delta - 1) \leq \Delta(\omega - 1) + 2(\omega - 1)$. So, $\lambda(G) \leq \Delta(\omega - 1) + 2(\omega - 1)$. If G is not a tree, then $\omega(G) \geq 3$. So, $\Delta(\omega - 1) + 2(\omega - 1) \geq 2\Delta + 2\omega - 2$. So, Greedy-$L(2,1)$-labeling($\beta$) cannot give us $\lambda(G) \leq \Delta(G) + \omega(G)$ for an arbitrary block graph G.

Note that the complexity of algorithm Greedy-$L(2,1)$-labeling(α) is $O(\Delta(n+m))$ assuming that α can be computed in this time bound. So, to get an improved bound and a better time complexity, one needs to look for a different algorithm.

We next propose a linear time algorithm to $L(2,1)$-label a block graph using $\Delta(G) + \omega(G) + 1$ colors. The concept of cut-tree of a block graph, which is introduced below, is essential for this purpose.

Let $G = (V, E)$ be a block graph with h blocks, $B_1, B_2, \ldots, B_h$ and k cut vertices, $c_1, c_2, \ldots, c_k$. The ***cut-tree*** of G, denoted by $T^B = (V^B, E^B)$, is defined as $V^B = \{B_1, B_2, \ldots, B_h, c_1, c_2, \ldots, c_k\}$ and $E^B = \{B_i c_j | c_j \in B_i, 1 \leq i \leq h, 1 \leq j \leq k\}$. The cut-tree T^B of G can be made a rooted tree by selecting any block or a cut vertex as the root. A vertex in T^B, corresponding to a block of G, is called *block vertex*. The leaves of T^B are essentially the end blocks of G. Let r be a cut vertex of G having maximum degree. Since one of the cut vertex in a block graph has the maximum degree, $d(r) = \Delta(G)$. Let $T^B(r)$ be the rooted cut-tree of G rooted at r. A Block graph G and its cut-tree T^B is given in Figure 3.

The set of all blocks of a given block graph can be found in $O(n+m)$

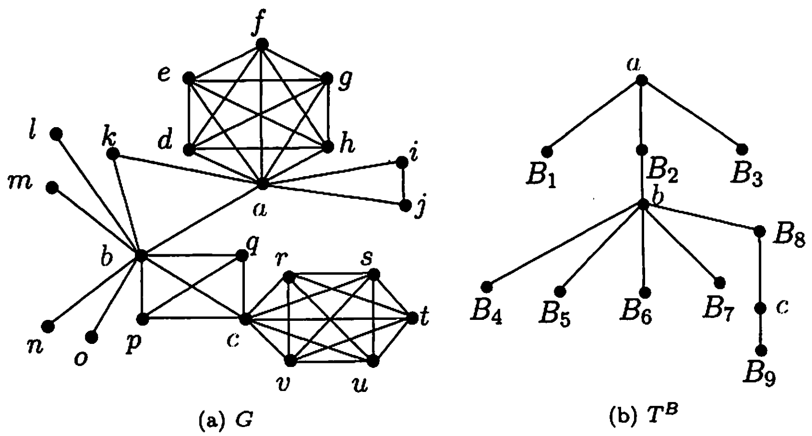


Figure 1: (a) a block graph G with 3 cut-vertices a, b, c and 9 blocks, $B_1 = \{a, d, e, f, g, h\}$, $B_2 = \{a, b, k\}$, $B_3 = \{a, i, j\}$, $B_4 = \{b, l\}$, $B_5 = \{b, m\}$ $B_6 = \{b, n\}$, $B_7 = \{b, o\}$, $B_8 = \{b, p, q, c\}$, $B_9 = \{c, r, s, t, u, v\}$,(b) cut-tree T^B of block graph G.

[9]. For each cut vertex v, the set of all blocks containing v can also be found easily in $O(n+m)$ time. Hence, the cut tree of a block graph can be constructed in $O(n+m)$ time.

Let T^B be the cut-tree, rooted at r, of a block graph G. Let B_c denote the set of all children of c in T^B. T^B is a tree rooted at r whose leaves are exactly the end blocks of G.

Let $L[c] = \{B^* | B^* = B \setminus \{c\},\ B \in B_c\}$ and $V(L[c]) = \bigcup_{B^* \in L[c]} B^*$. Let $P_B(c) = B \setminus \{c\}$, where B is the parent block of c in T^B. Let $P_c(B)$ denote the parent cut vertex of a block B. Let $\mathscr{F} = \{0, 1, 2, \ldots, \Delta + \omega\}$. Let $\mathscr{A}(c)$ denote set of *admissible labels* for the vertices of child blocks of a cut vertex c, $f(c)$ denote *label* assigned to a vertex c of G. $\mathscr{P}(c)$ denote set of labels assigned to the vertices of $P_B(c)$.

Next, we present an algorithm to $L(2,1)$-label a block graph. Our algorithm first constructs a rooted cut-tree rooted at a cut vertex r such that $d_G(r) = \Delta(G)$ of the given block graph G. It starts from the root and traverses towards the leaves of the cut-tree, which are the end block of G. Initially, all vertices of G are uncolored and $\mathscr{A}(c) = \mathscr{F}$ for each cut vertex c. Breadth First Search (BFS) is performed on T^B to get an ordering of the cut vertices and the blocks. Root is assigned color 0 and after that at every step the vertices of child blocks of a cut vertex c in $L[c]$ are labeled with a minimum admissible color from $\mathscr{A}(c)$. Before assigning a label to the vertices of B_c, the labels which are forbidden due to distance constraint with c and the colored neighbors of c, in $P_B(c)$ are removed from $\mathscr{A}(c)$. Once a label is assigned to a child vertex of c, it is removed from $\mathscr{A}(c)$ so that it is never repeated for the other child vertices. For any cut vertex c, its child blocks are first ordered in non increasing order of their cardinalities. First all blocks with cardinality at least $\omega/2$ are selected. At each step q vertices are selected for coloring one from each such block and each is colored with minimum label in $\mathscr{A}(c)$. These are then removed from $\mathscr{A}(c)$. That is, if there are q such blocks then, $v_1, v_2, \ldots, v_q$ will be selected such that $v_i \in B_i^*$ and will be assigned first q minimum labels from $\mathscr{A}(c)$ at every iteration. This is repeated till all the vertices of the block B_q^* are labeled, consequently B_q^* is removed from $L[c]$. The remaining blocks are then sorted again in the non increasing order of their cardinalities. Two blocks of maximum cardinality among the uncolored blocks are selected simultaneously for coloring. They are stored in temporary variables S_1 and S_2. Their vertices are labeled consecutively, by selecting one at a time from each block with the minimum label in $\mathscr{A}(c)$. Their labels are removed from $\mathscr{A}(c)$. Iteratively continue till either all the blocks in $L[v]$ are labeled or there is some block $B^* \in L[c]$ which still has some uncolored vertices. Finally, the vertices of B^* are labeled with the least color (say t) in $\mathscr{A}(c)$ and the labels t and $t+1$ are removed simultaneously from $\mathscr{A}(c)$.

The detail of the main algorithm, described as ColorBlockGraph(G), is shown in Figure 2, whereas the steps for labeling child blocks are given as a separate algorithm and is described as Color_Child_ Block($v, \mathscr{A}(v)$), shown in Figure 3. An $L(2,1)$-Labeling of the block graph G of Figure 1(a) generated by Algorithm ColorBlockGraph(G) is shown in Figure 4.

Next, we prove that Algorithm ColorBlockGraph(G) produces an $L(2,1)$-labeling and uses at most $\Delta + \omega + 1$ colors.

Theorem 3.1. *The labeling produced by Algorithm ColorBlockGraph(G) is an $L(2,1)$-labeling of a block graph G and uses at most $\Delta(G) + \omega(G) + 1$ colors. Furthermore, Algorithm ColorBlockGraph(G) takes $O(n+m)$ time.*

Proof. Let $G = (V, E)$ be a block graph. Let us first prove that the coloring f produced by the Algorithm ColorBlockGraph(G) is an $L(2,1)$-labeling.

Algorithm ColorBlockGraph(G) selects a maximum degree cut vertex r and computes cut tree T^B rooted at r. It then performs BFS on T^B to get an ordering of cut vertices and blocks in Line 2. Also, it computes $P_B(c)$ and $L[c]$ for each cut vertex c in Line 3. In Line 5, labels of all vertices are initialized to -1. Root r is colored first and hence is assigned color 0 and 0, 1 are removed from the admissible label set $\mathscr{A}(r)$ for the child blocks of r in Line 8. In Lines 9 and 15, vertices of the child blocks of a cut vertex v are labeled.

Let $x, y \in V(G)$ such that $d(x, y) = 1$, which implies x and y belong to same block. Let us assume that x is colored before y. If $x = r$, then

Algorithm 2: ColorBlockgraph(G)

1 Let $r \in V(G)$ be a cut vertex such that $d(r) = \Delta$. Compute the rooted cut-tree T^B of G rooted at r ;
2 compute the ordering $r = c_1, c_2 \dots, c_k$ of the cut vertices of G using BFS on T^B;
3 compute $P_B(c)$ and $L[c]$ for each cut vertex c of G;
4 set $\mathscr{F} := \{0, 1, \dots, \Delta + \omega\}$;
5 **foreach** $c \in V(G)$ **do**
6   initialize $f(v) := -1$;
7 **end**
8 initialize $f(c_1) := 0$ and $\mathscr{A}(c_1) := \mathscr{F} \setminus \{0, 1\}$;
9 Use Algorithm Color_Child_Block($c_1, \mathscr{A}(c_1)$) ;
10 **foreach** $i := 2$ **to** k **do**
11     $\mathscr{A}(c_i) := \mathscr{F} \setminus \{f(c_i) - 1, f(c_i), f(c_i) + 1\}$;
12   **foreach** $z \in P_B(c_i)$ **do**
13         $\mathscr{A}(c_i) := \mathscr{A}(c_i) \setminus f(z)$;
14   **end**
15   Use Algorithm Color_Child_Block($c_i, \mathscr{A}(c_i)$);
16 **end**

Figure 2: Algorithm for $L(2, 1)$-labeling of a block graph

$y \in B^*$ for some $B^* \in L[x]$. So, $f(x) = 0$ and y will be colored in Line 9. Since all the vertices of child block of x get labels from the set $\mathscr{A}(x)$, $f(y) > f(x) + 1$ as $\mathscr{A}(x)$ does not contain $0, 1$. Thus, $|f(x) - f(y)| \geq 2$ for this case.

So, let $x \neq r$. If x is a cut vertex, then either $y \in B^*$ such that $B^* \in L[x]$ or $x, y \in B_z$ for some cut vertex z, as we have assumed that x is colored before y. Let us first assume that $y \in B^*$ such that $B^* \in L[x]$. Then x will be labeled, before vertices of $L[x]$ are labeled, so y will be labeled in Line 15. But before that, $\{f(x) - 1, f(x), f(x) + 1\}$ will be deleted from admissible label set $\mathscr{A}(x)$ in Line 11, so that $|f(x) - f(y)| \geq 2$. If $x, y \in B^*$ where $B^* \in L[z]$ for some cut vertex z then x, y will be labeled by Algorithm Color_Child_Block($z, \mathscr{A}(z)$) and since we are assuming that x is colored before y, $f(y) > f(x)$. Suppose, x is labeled in Lines $2 - 7$ of Algorithm Color_Child_Block($z, \mathscr{A}(z)$). Once x is colored in Line 4, $f(x)$ is deleted from the set $\mathscr{A}(z)$ in Line 5 so that it is never assigned to any other child vertex of z. Next minimum color of the $\mathscr{A}(z)$ will be assigned to some vertex of some other child block of z other then B^* so that y will be assigned any label afterwards. Next, assume that x is labeled in Lines $10 - 28$. Note that x and y are not selected simultaneously and $f(x)$ is

Algorithm 3: Color_Child_Block($v, \mathscr{A}(v)$)

Compute an ordering $B_1^*, \ldots, B_{|L[v]|}^*$ of $L[v]$ such that $|B_i^*| \geq |B_{i+1}^*|$, $1 \leq i \leq |L[v]|$ and let index j be such that $|B_j^*| + 1 \geq \omega/2$ but $|B_{j+1}^*| + 1 < \omega/2$;
if $j \geq 2$ **then**
    while $|B_j^*| > 1$ **do** select $\{v_1, v_2, \ldots, v_j\}$, $v_i \in B_i^*$ and $B_i^* \in L[v]$;
    for $i = 1$ **to** j **do** Assign $f(v_i) := t$ such that $t := \min \mathscr{A}(v)$;
    update $\mathscr{A}(v) := \mathscr{A}(v) \setminus \{t\}$ and $B_i^* := B_i^* \setminus \{v_i\}$;
    if $B_i^* = \emptyset$ **then** $L[v] := L[v] \setminus B_i^*$;
end
if $|L[v]| = 1$ **then**
    set $S := B^*$ such that $B^* \in L[v]$;
else
    reorder $L[v]$, such that $|B_j^*| \geq |B_{j+1}^*|$, $1 \leq j \leq s-1$, $s = |L[v]|$;
    initialize $S_1 := B_1^*$, $S_2 := B_2^*$, $S := \emptyset$ and set $i := 3$;
    while $|L[v]| \geq 2$ **do** select $u \in S_1$, $w \in S_2$, assign $f(u) := t_1$, $f(w) := t_2$ such that $t_1, t_2 \in \mathscr{A}(v)$ are minimum, update $\mathscr{A}(v) := \mathscr{A}(v) \setminus \{t_1, t_2\}$, $S_1 := S_1 \setminus \{u\}$, $S_2 := S_2 \setminus \{w\}$;
    if $S_1 = \emptyset$ *and* $S_2 = \emptyset$ **then**
        $|L[v]| := |L[v]| - 2$;
        if $|L[v]| \geq 2$ **then** update $S_1 := B_i^*$, $S_2 := B_{i+1}^*$ and $i := i + 2$;
        if $|L[v]| := 1$ **then** update $S := B_i^*$;
    else
        if $S_1 = \emptyset$ **then**
            $|L[v]| := |L[v]| - 1$;
            if $|L[v]| = 1$ **then** remove $f(w) + 1$ from $\mathscr{A}(v)$ and update $S := S_2$;
            else $S_1 := B_i^*$ and $i := i + 1$;
        if $S_2 = \emptyset$ **then**
            $|L[v]| := |L[v]| - 1$;
            if $|L[v]| = 1$ **then** remove $f(u) + 1$ from $\mathscr{A}(v)$ and update $S := S_1$;
            else $S_2 := B_i^*$ and $i := i + 1$;
    end
end
if $|L[v]| = 1$ **then**
    while $S \neq \emptyset$ **do** select x from S and $t := \min\{\mathscr{A}(v)\}$, $f(x) := t$ and update $\mathscr{A}(v) := \mathscr{A}(v) \setminus \{t, t+1\}$, $S := S \setminus \{x\}$;
end

Figure 3: Algorithm for labeling the vertices of child blocks of a cut vertex v of a block graph

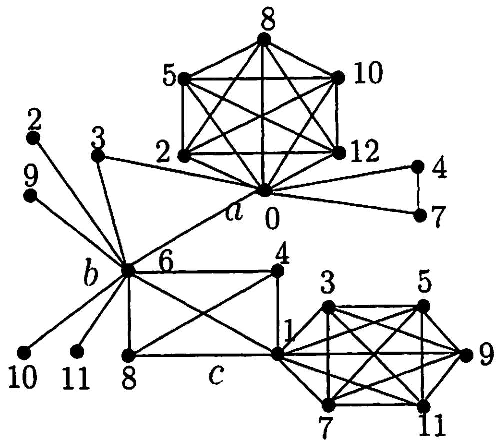


Figure 4: An $L(2,1)$-Labeling of block graph G of Figure 1(a) ($\Delta = 9, \omega = 6$) generated by Algorithm ColorBlockGraph with span=12.

deleted from the set $\mathscr{A}(z)$ in Line 13 so that it is never assigned to any other child vertex of z. Next min color of the $\mathscr{A}(z)$ will be assigned to some vertex of some other child block of z other then B^* so that y will be assigned any label afterwards. If $|L(z)|$ becomes 1 before x is colored, x and y will be colored in Lines $29-31$ and after assigning any color to x in Line 30, $f(x), f(x)+1$ will be deleted from $\mathscr{A}(z)$. Therefore, $|f(x)-f(y)| \geq 2$. Hence if $d(x,y)=1$ then $|f(x)-f(y)| \geq 2$.

Next, assume that $d(x,y)=2$. So, the blocks containing x and y will be have a common cut vertex, say z. Let us denote these blocks by $\mathcal{Q}_x$ and $\mathcal{Q}_y$ respectively. Since x is colored before y, either $\mathcal{Q}_x = P_B(z)$ or $\mathcal{Q}_x, \mathcal{Q}_y \in L[z]$. If $\mathcal{Q}_x = P_B(z)$, then y will be assigned color from $\mathscr{A}(z)$ in Line 15 of Algorithm ColorBlockgraph(G). But in Lines $12-14$, $f(w)$ (and so $f(x)$) will be deleted for all $w \in P_B(z) = \mathcal{Q}_x$. So, $f(y) \neq f(x)$. If $\mathcal{Q}_x, \mathcal{Q}_y \in L[z]$, then both will be labeled in Line 15 by Algorithm Color_Child_Block($z, \mathscr{A}(z)$). Since we sort $L[z]$ in Line 1 of Algorithm Color_Child_Block($z, \mathscr{A}(z)$), $|\mathcal{Q}_x| \geq |\mathcal{Q}_y|$ (since we are assuming x is colored before y). If x and y are selected simultaneously in Line 3 of Algorithm Color_Child_Block($z, \mathscr{A}(z)$), then $f(x)$ will be deleted from $\mathscr{A}(z)$ in Line 5, before y is labeled in next loop. Therefore, $f(y) \neq f(x)$. If x and y are selected in Lines $10-28$ of Algorithm Color_Child_Block($v, \mathscr{A}(v)$) then also, $f(x) \neq f(y)$ since we are selecting two labels from $\mathscr{A}(z)$ simultaneously. Therefore, $f(y) \neq f(x)$. Hence if $d(x,y)=2$ then $|f(x)-f(y)| \geq 1$.

Next, we prove that the maximum color number used is $\Delta(G)+\omega(G)$. Let x be a cut vertex. Then $d(x) = |P_B(x)| + \sum_{B^* \in L[x]} |B^*| \leq \Delta$ and

$\sum_{B^* \in L[x]} |B^*| = \Delta - |P_B(x)|$. Since vertices of the child blocks of x are labeled consecutively from the set $\mathscr{A}(x)$, (which do not contain the labels assigned to $P_B(x)$ and $f(x)-1, f(x), f(x)+1$) i.e. all the labels from the set $\mathscr{A}(x)$ are assigned to some vertex in B_x till there remains uncolored vertices in a single block, say $B' \in L[x]$. Let $B' = \max_{B^* \in L[x]} |B^*|$, uncolored vertices of B' will use at most $2(|B'|-1)$ colors. Thus the number of distinct labels used by the neighbors of x is at most $|P_B(x)| + \sum_{B^* \neq B' \in L[x]} |B^*| + 2(|B'|-2) \leq \Delta - |B'| + 2|B'| - 2 = \Delta + |B'| - 1$. Since $\{f(x)-1, f(x), f(x)+1\}$ are not used by any neighbor of x, the number of colors required for x and its neighbors is $\Delta + |B'| - 2 + 3$. Hence, the maximum color number assigned to x and its neighbors is at most $\Delta + \omega$.

Next, we show how Algorithm ColorBlockGraph(G) can be implemented in $O(n+m)$ time. For each cut vertex v, the blocks containing v can be sorted in non-increasing order of their sizes in $O(d(v))$ time using standard bucket sort algorithms as the sizes of the blocks are all integers between 1 and $\omega(G)$. Therefore, the total time taken for this for all cut vertices is at most $\sum_{v \in V} d(v) = O(n+m)$. It is easy to see that all other steps can be implemented in $O(n+m)$ time. Hence, Algorithm ColorBlockGraph(G) takes $O(n+m)$ time.

□

Theorem 3.2. $\Delta(G) + 1 \leq \lambda(G) \leq \Delta(G) + \omega(G)$ *for a block graph* G. *Furthermore, for every* $j, 0 \leq j \leq \omega - 2$, *there is a block graph* G *with* $\lambda(G) = (\Delta(G) + \omega(G) - 1) - j$.

Proof. It is known that $\lambda(G) \geq \Delta(G)+1$ for any graph G [11]. By Theorem 3.1, $\lambda(G) \leq \Delta(G) + \omega(G)$ for a block graph G. Hence, $\Delta(G) + 1 \leq \lambda(G) \leq \Delta(G) + \omega(G)$ for a block graph G.

Let $\omega \geq 2$ be any integer. Consider the graph $G_j = G(\omega, j) = (V_j E_j)$, where j is an integer such that $0 \leq j \leq \omega - 2$, $V_j = \{x_1, x_2, \ldots, x_\omega\} \cup \{y_1, y_2, \ldots, y_j\}$ and $E_j = \{x_s x_t | 1 \leq s < t \leq \omega\} \cup \{x_1 y_i, 1 \leq i \leq j\}$. The graph $G_4 = G(6, 4)$ is illustrated in Figure 5.

It is easy to see that $\omega(G_j) = \omega$ and $\Delta(G_j) = \omega + j - 1$. Let $G = G_j[\{x_1, x_2, \ldots, x_\omega\}]$. It is easy to see that our algorithm uses $2\omega - 1$ colors when applied to G_j. In fact, an $L(2,1)$-labeling f produced by our algorithm when applied to G_j is as follows: $f(x_i) = 2(i-1), 1 \leq i \leq \omega$ and $f(y_s) = 2(s+1) - 1, 1 \leq s \leq j$. So, $\lambda(G) \leq 2\omega - 2 = \omega + (\omega + j - 1) - 1 - j = (\omega(G_j) + \Delta(G_j) - 1) - j$. Next, we show that $\lambda(G_j) \geq (\Delta(G_j) + \omega(G_j) - 1) - j$. Clearly, G is a complete graph on ω vertices. So, $2\omega - 2 = \lambda(G) \geq \lambda(G_j)$. Hence, $\lambda(G_j) = (\Delta(G_j) + \omega(G_j) - 1) - j$.

□

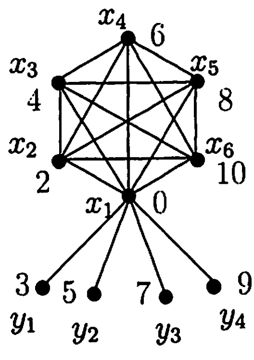


Figure 5: The graph $G_4 = G(6,4)$

4 Special Block Graphs

We now give a class of block graphs achieving lower upper bounds for λ. A block graph G is called ***block-regular*** if all the blocks of G are of same size. The ***block degree*** of a cut vertex v is denoted by $d_B(v)$ and is the number of blocks containing v.

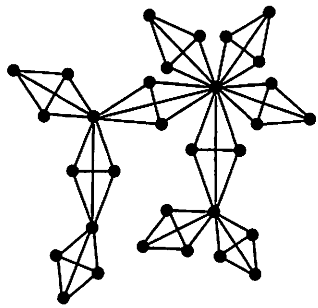
Theorem 4.1. *The following are true for a block-regular graph G.*

(*i*) *If $d_B(v) = 2$ for each cut vertex v of G, then $\lambda(G) \leq \Delta(G) + 4$.*

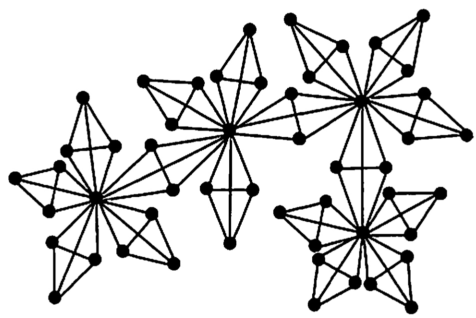
(*ii*) *If $d_B(v) \geq 3$ for some cut vertex v of G, then $\lambda(G) \leq \Delta(G) + 2$.*

Proof of (i) : Let G be a block-regular graph such that for any cut vertex $v \in G, d_B(v) = 2$. We prove the theorem by presenting an $L(2,1)$-labeling f of G with span at most $\Delta(G) + 4$.

Consider the cut-tree T^B of G rooted at an end block, say B_1. Let r be the only cut vertex of G contained in B_1. So $d(r) = \Delta(G)$. Let $B_1, B_2, \ldots, B_k$ be the ordering of the blocks of G obtained by performing BFS on T^B. Clearly, B_2 is a child of r in T^B. Assign color 0 to r, i.e. $f(r) = 0$, and color the vertices of $B_1^* = B_1 \setminus \{r\}$ and $B_2^* = B_2 \setminus \{r\}$ alternatively with smallest even and odd admissible colors. Let $|B_1| = s$, $B_1^* = \{x_1, x_2, \ldots, x_{s-1}\}$ and $B_2 = \{y_1, y_2, \ldots, y_{s-1}\}$. So, $f(x_i) = 2i$ and $f(y_i) = 2i + 1$, $1 \leq i \leq s - 1$. Assume that all the vertices of the blocks $B_1, B_2, \ldots, B_{i-1}$ have been colored. Consider the next block B_i. Let $P(B_i) = v_i$, $P_B(v_i) = B_j$, and $P(B_j) = v_j$. Note that $v_i, v_j \in B_j$. Note that $f(v_i)$ and $f(v_j)$ are of different parity, i.e., if one is odd then the other is even. If $f(v_i)$ is even, then color the vertices of B_i^* with odd colors starting with smallest admissible odd color. Suppose $f(v_i) = 2k$ and $f(v_j) = 2l + 1$. So, the odd colors forbidden for the vertices of B_i^* are $2k - 1, 2k + 1$ and



(a) G



(b) G'

Figure 6: (a) A block regular graph G, (b) block regular graph G' constructed from G.

$2l+1$ as $v_iv_j \in E$. Since there are $r-1$ vertices in B_i^*, the vertices of B_i^* can be colored with the colors from the set $\{2i+1, 0 \leq i \leq 2(r+1)+1\} \setminus \{2k-1, 2k+1, 2l+1\}$ which has exactly $r-1$ odd colors. Similarly, if $f(v_i)$ is odd and $f(v_j)$ is even, then color the vertices of B_i^* with even colors starting with smallest admissible even color. Suppose $f(v_i) = 2k+1$ and $f(v_j) = 2l$, the even colors forbidden for the vertices of B_i^* are $2k, 2k+2$ and $2l$. Since there are $r-1$ vertices in B_i^*, the vertices of B_i^* can be colored with the colors from the set $\{2i, 1 \leq i \leq 2(r+1)\} \setminus \{2k, 2k+2, 2l\}$ which has exactly $r-1$ even colors. Note that this coloring satisfies the property that if v_i and v_j are cut vertices such that $P_c(P_B(v_i)) = v_j$ in T^B, then $f(v_i)$ and $f(v_j)$ are of different parity. Also, if B is not the root block, then the colors of all vertices except $P_c(B)$ will be of same parity. Repeating this till all the vertices of all the blocks are colored. It is easy to see that the coloring f is an $L(2,1)$-labeling of G. The number of colors needed is $2(r+1)+1 = (2(r-1)+4)+1) = \Delta(G)+4+1$. Hence, $\lambda(G) \leq \Delta(G)+4$. $\square$

Proof of (ii) : Suppose G is a block regular such that $d_B(w) \geq 3$ for some cut vertex $w \in G$. Let $|B| = s$ for each block B of G. Let v be the cut vertex contained in maximum number of blocks of G and $d_B(v) = r$. Then, $r \geq 3$. Let $\{c_1, c_2, \ldots, c_k\}$ be the set of all cut vertices of G such that $2 \leq d_B(c_i) \leq r-1$. We construct the graph G' from G as follows. For each $i, 1 \leq i \leq k$, add $r - d_B(c_i)$ blocks of size s each having the common cut vertex c_i. Figure 6 illustrates the construction of G' from G.

Color G' by Algorithm ColorBlockGraph(G). Since G is block regular, at every iteration in Line 1 of Algorithm Color_Child_Block($u, \mathscr{A}(u)$) all the child blocks of a cut vertex u will be selected i.e. $j = |L[u]|$ for all cut vertex u and Lines 14 – 35 will never be executed. Let $|L[u]| = q$. Then, $v_1, v_2, \ldots, v_q$, will be selected such that $v_i \in B_i^*$, $B_i^* \in L[v]$ and will be assigned first q minimum label from $\mathscr{A}(u)$ to each v_i and this will be repeated $|L[v]|$ times. So, $|\mathscr{A}(u)| = q(s-1)$. Since s colors which are assigned to the vertices of $P_B(u)$ cannot be in $\mathscr{A}(u)$ and $f(u)-1$, $f(u)+1$ are also cannot be in $\mathscr{A}(u)$ therefore $\mathscr{A}(u) \subset \{0, 1 \ldots, (q+1)(s-1)+3\}$. Since $(q+1)(s-1)+3 = (\Delta(G')+2)+1$, $\lambda(G') \leq \Delta(G')+2$. Furthermore, G is a subgraph of G' implies $\lambda(G) \leq \lambda(G') \leq \Delta(G')+2$. Since $\Delta(G') = \Delta(G)$, hence $\lambda(G) \leq \Delta(G) + 2$. □

Theorem 4.2. *Let G be a block graph with $\omega \leq \lceil \frac{\Delta}{2} \rceil$ then $\lambda(G) \leq \Delta + \frac{\omega}{2} + 1$.*

Proof. Let G be a block graph with $\omega \leq \lceil \frac{\Delta}{2} \rceil$. Let T^B be the cut-tree of G rooted at a cut vertex u such that $d(u) = \Delta(G)$. Let v be a cut vertex and $L[v] = \{B_1^*, B_2^*, \ldots, B_s^*\}$ such that $|B_j^*| \geq |B_{j+1}^*|$ for $1 \leq j \leq |L[v]|$ and let f be a labeling given by Algorithm ColorBlockGraph(G). Let index j be such that $|B_j^* + 1| \geq \omega/2$ but $|B_{j+1}^* + 1| < \omega/2$. Let $j > 1$. Then the vertices of $B_1^*, B_2^*, \ldots, B_j^*$ will be labeled first, successively and will be assigned consecutive labels from $\mathscr{A}(v)$, till all the vertices of B_j^* are labeled. The uncolored vertices of remaining blocks will be colored with the colors from $\mathscr{A}(v)$ by taking two blocks at a time. Finally, only one block, say B_q^*, remains with some uncolored vertices. The uncolored vertices of B_q^* will be assigned labels with a gap of at least 1 from $\mathscr{A}(v)$. If $q \leq j$, then there will at least $\omega/2$ vertices in B_q^* that are already colored. If $q > j$ then $|B_q^*| \leq \omega/2$. Thus, in any case there will at most $\frac{\omega}{2}$ vertices that are uncolored and v has at least $\Delta - |B^*| \geq \Delta - \omega/2$ colored neighbors those are colored before we label remaining uncolored vertices of B^*. Hence the number of labels required for v and its neighbors is at most $\Delta(G) - \omega/2 + 3 + 1 + 2(\frac{\omega}{2} - 2) \leq \Delta(G) + \frac{\omega}{2} + 1$. □

5 Extremal Block Graphs

We have already seen in Theorem 3.2 that $\lambda(G) \leq \Delta(G) + \omega(G)$ for a block graph. We call a block graph G ***extremal block graph*** if $\lambda(G) = \Delta(G) + \omega(G)$.

Consider the tree $T = (V, E)$, where $V(T) = \{1, 2 \ldots, n\}$, $n = 3k + 2, k \geq 1$, and $E(T) = \{12, 23\} \cup \{1x\ x \in S_1\} \cup \{2y|y \in S_2\} \cup \{3z|z \in S_3\}$, where $S_1 \cup S_2 \cup S_3 = V - \{1, 2, 3\}$, $S_i \cap S_j = \emptyset$, for $i \neq j$, and $|S_1| = |S_3|$ and $|S_2| = |S_1 - 1|$. It is known that $\Delta(T) + 1 \leq \lambda(T) \leq \Delta(T) + 2$ for a tree [11]. Also it is known that $N[v]$ contains at most two $\Delta(G)$-degree vertices for all $v \in V(G)$ for a graph G with $\lambda(G) = \Delta(G)+1$ [11]. As $N[2]$, the closed neighborhood of vertex 2, contains three $\Delta(T)$-degree vertices, $\lambda(T) = \Delta(T) + 2 = \Delta(T) + \omega(T)$. Hence, T is an extremal block graph. However, $\omega(T) = 2$. Therefore, it is interesting to find extremal block graphs of larger clique sizes.

Let $minblock(G)$ denote the cardinality of the minimum size block of G. The following two lemmas give upper bound on $minblock(G)$ of an extremal block graph.

Lemma 5.1. *Let G be a block graph such that $d_B(v) = 2$ for each cut vertex v of G. If $minblock(G) \geq 5$, then $\lambda(G) \leq \Delta(G) + \omega(G) - 1$.*

Proof. Let G be a block graph such that $minblock(G) \geq 5$ and $d_B(v) = 2$ for each cut vertex v of G. Add non-cut vertices to increase the size of each block of G to $\omega(G)$ to obtain the block graph G'. Now $\Delta(G') \leq \Delta(G) + \omega(G) - 5$. By Theorem 4.1 (*i*), $\lambda(G') \leq \Delta(G') + 4$. But, $\lambda(G) \leq \lambda(G')$, since G is a subgraph of G'. Therefore, $\lambda(G) \leq \Delta(G')+4 \leq \Delta(G)+\omega(G)-5+4 \leq \Delta(G) + \omega(G) - 1$. □

Lemma 5.2. *Let G be a block graph such that $d_B(v) \geq 3$ for some cut vertex v of G. If $minblock(G) \geq 3$, then $\lambda(G) \leq \Delta(G) + \omega(G) - 1$.*

Proof. Let G be a block graph such that $minblock(G) \geq 3$ and $d_B(v) \geq 3$ for some cut vertex v of G. Add non-cut vertices to increase the size of each block of G to $\omega(G)$ to obtain the block graph G'. Now, $\Delta(G') \leq \Delta(G) + \omega(G) - 3$ and by Theorem 4.1 (*ii*), $\lambda(G') \leq \Delta(G') + 2$. But $\lambda(G) \leq \lambda(G')$ since G is a subgraph of G'. Therefore, $\lambda(G) \leq \Delta(G')+2 \leq \Delta(G)+\omega(G) - 3 + 2 \leq \Delta(G) + \omega(G) - 1$. □

In view of the above two lemmas, an extremal block graph G must have $minblock(G) \leq 4$. Furthermore, if $d_B(v) = 2$ for every cut vertex v of an extremal block graph G, then $minblock(G) \leq 4$. Similarly, if $d_B(v) > 2$ for some cut vertex v of an extremal block graph G then $minblock(G) = 2$.

A block B of a block graph G is called an internal block of type 1 if all of its vertices are a cut vertex of G. A block is called an **internal block of**

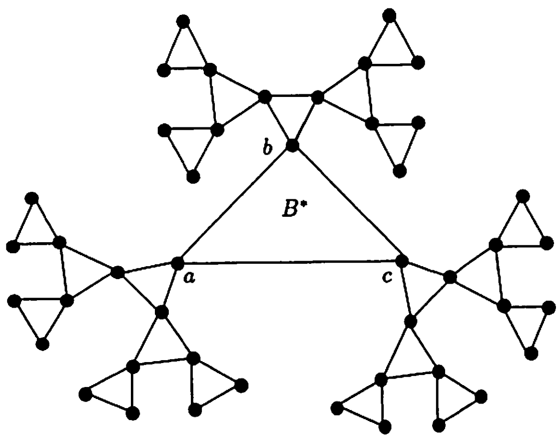


Figure 7: Block regular graph $G[3,2]$.

type i, $i \geq 2$ if all of its vertices are cut vertices of G and all of its adjacent blocks are of type j, where $j \leq i-1$. So, if a block B is an internal block of type i, then it is an internal block of type j for each $j, 1 \leq j \leq i$. The block B^* of the graph $G[3,2]$ given in Figure 7 is an internal block of type 3.

Let f be any $L(2,1)$-labeling of G. For a block $B = \{x_1, x_2, x_3\}$, $f(B) = \{f(x_1), f(x_2), f(x_3)\}$ denotes set of colors of its vertices assigned by f.

Next we give example of a class of extremal block graphs having block size three.

Theorem 5.3. *$\lambda(G[3,2]) = \Delta(G[3,2]) + \omega(G[3,2])$ for the graph $G[3,2]$ given in Figure 7.*

Before proving this theorem we will prove some important lemmas.

Lemma 5.4. *Let B and B' be the blocks containing a cut vertex v of $G[3,2]$ and let f be any 6-$L(2,1)$-labeling of $G[3,2]$.*

1. *If $f(v) = 1$ then $\{f(B), f(B')\} = \{\{1,3,5\},\{1,4,6\}\}$.*
2. *If $f(v) = 2$ then $\{f(B), f(B')\} = \{\{0,2,5\},\{2,4,6\}\}$.*
3. *If $f(v) = 3$ then $\{f(B), f(B')\} = \{\{0,3,5\},\{1,3,6\}\}$ or $\{f(B), f(B')\} = \{\{0,3,6\},\{1,3,5\}\}$.*
4. *If $f(v) = 4$ then $\{f(B), f(B')\} = \{\{0,2,4\},\{1,4,6\}\}$.*
5. *If $f(v) = 5$ then $\{f(B), f(B')\} = \{\{0,2,5\},\{1,3,5\}\}$.*

Proof. Let f be any 6-$L(2,1)$-labeling of $G[3,2]$. Let B and B' be the blocks containing a cut vertex v of G. If $f(v) = 1$ then $\{f(B), f(B')\} = \{\{1,3,5\},\{1,4,6\}\}$. So, (1) is true. If $f(v) = 2$ then $\{f(B), f(B')\} = \{\{0,2,5\},\{2,4,6\}\}$. So, (2) is true. The proofs of (3) to (5) are similar and hence are omitted.

□

Lemma 5.5. *Let f be a 6-$L(2,1)$-labeling of $G[3,2]$. If $f(x_1) = 0$ and $f(x_2) = 6$ for an internal block $B = \{x_1, x_2, x_3\}$ of $G[3,2]$, then $f(x_3) = 3$.*

Proof. Since $f(x_1) = 0$ and $f(x_2) = 6$, $f(x_3) \in \{2,3,4\}$. Since B is an internal block, x_3 is a cut vertex. So, by Lemma 5.4, $f(x_3) \notin \{2,4\}$. Therefore, $f(x_3) = 3$. □

Lemma 5.6. *Let f be a 6-$L(2,1)$-labeling of $G[3,2]$. If $B = \{x_1, x_2, x_3\}$ is an internal block of $G[3,2]$, then $f(B) \notin \{\{0,2,4\},\{0,2,6\}, \{0,3,5\},\{1,3,6\}, \{0,4,6\},\{2,4,6\}\}$*

Proof. By Lemma 5.5 $f(B) \notin \{\{0,2,6\},\{0,4,6\}\}$. If $f(B) = \{0,2,4\}$, then wlg assume that $f(x_1) = 2$. Since B is an internal block, there is a block B' such that $B' = \{x_1, y_1, y_2\}$. Since $f(y_i) \notin \{0,1,2,3,4\}, i = 1,2$, the admissible colors for y_1 and y_2 are 5 and 6. However, $\{f(y_1), f(y_2)\}$ cannot be $\{5,6\}$. So, $f(B) \notin \{0,2,4\}$.

Next assume that $f(B) = \{0,3,5\}$. Wlg, $f(x_1) = 5$. Since B is an internal block, there is a block B' such that $B' = \{x_1, y_1, y_2\}$. Now, $f(y_i) \notin \{0,3,4,5,6\}, i = 1,2$. So, the admissible colors for y_1 and y_2 are 1 and 2. However, $\{f(y_1), f(y_2)\}$ cannot be $\{1,2\}$. So, $f(B) \notin \{0,3,5\}$.

Next assume that $f(B) = \{1,3,6\}$. Wlg, $f(x_1) = 1$. Since B is an internal block, there is a block B' such that $B' = \{x_1, y_1, y_2\}$. Now, $f(y_i) \notin \{0,1,2,3,6\}, i = 1,2$. So, the admissible colors for y_1 and y_2 are 4 and 5. However, $\{f(y_1), f(y_2)\}$ cannot be $\{4,5\}$. So, $f(B) \notin \{1,3,6\}$.

Next assume that $f(B) = \{2,4,6\}$. Wlg, $f(x_1) = 4$. Since B is an internal block, there is a block B' such that $B' = \{x_1, y_1, y_2\}$. Now, $f(y_i) \notin \{2,3,4,5,6\}, i = 1,2$. So, the admissible colors for y_1 and y_2 are 0 and 1. However, $\{f(y_1), f(y_2)\}$ cannot be $\{0,1\}$. So, $f(B) \notin \{2,4,6\}$.

Therefore, $f(B) \notin \{\{0,2,4\},\{0,2,6\},\{0,3,5\},\{1,3,6\},\{0,4,6\},\{2,4,6\}\}$

□

Lemma 5.7. *Let f be a 6-$L(2,1)$-labeling of $G[3,2]$. If $B = \{x_1, x_2, x_3\}$ is an internal block of type 2 of $G[3,2]$, then $f(B) \notin \{\{0,2,5\},\{1,4,6\}\}$.*

Proof. If possible, suppose $f(B) = \{0,2,5\}$. Without loss of generality, $f(x_1) = 2$. Since, B is an internal block of type 2, there is an internal block $B' = \{x_1, y_1, y_2\}$ of $G[3,2]$ such that $B \neq B'$. Thus, by Lemma 5.4, $f(B') = \{2,4,6\}$. Since B' is an internal block, by Lemma 5.6, $f(B') \neq$

$\{2,4,6\}$. Therefore $f(B) \neq \{0,2,5\}$. If possible, suppose $f(B) = \{1,4,6\}$. Without loss of generality, $f(x_1) = 4$. Since, B is an internal block of type 2, there is an internal block $B' = \{x_1, y_1, y_2\}$ of $G[3,2]$ such that $B \neq B'$. Thus, by Lemma 5.4, $f(B') = \{0,2,4\}$. Since B' is an internal block, by Lemma 5.6, $f(B') \neq \{0,2,4\}$. Therefore $f(B) \neq \{1,4,6\}$. Hence $f(B) \notin \{\{0,2,5\},\{1,4,6\}\}$. □

Lemma 5.8. *Let f be a 6-$L(2,1)$-labeling of $G[3,2]$. If $B = \{x_1, x_2, x_3\}$ is an internal block of type 3 of $G[3,2]$, then $f(B) \notin \{\{0,3,6\},\{1,3,5\}\}$.*

Proof. If possible, suppose $f(B) = \{0,3,6\}$. Without loss of generality, $f(x_1) = 0$. Since, B is an internal block of type 3, there is an internal block $B' = \{x_1, y_1, y_2\}$ of type 2 of $G[3,2]$ such that $B \neq B'$. Thus, by Lemma 5.4, $f(B') \in \{\{0,2,4\},\{0,2,5\}\}$. Since B' is an internal block of type 2, by Lemma 5.6 and Lemma 5.7, $f(B') \notin \{\{0,2,4\},\{0,2,5\}\}$. Therefore $f(B) \neq \{0,3,6\}$.

If possible, suppose $f(B) = \{1,3,5\}$. Without loss of generality, $f(x_1) = 5$. Since, B is an internal block of type 3, there is an internal block $B' = \{x_1, y_1, y_2\}$ of type 2 of $G[3,2]$ such that $B \neq B'$. Thus, by Lemma 5.4, $f(B') = \{0,2,5\}$. Since B' is an internal block of type 2, by Lemma 5.7, $f(B') \neq \{2,4,6\}$. So $f(B) \neq \{1,3,5\}$. Hence $f(B) \notin \{\{0,3,6\},\{1,3,5\}\}$. □

Proof of Theorem 5.3. Since $\Delta(G[3,2]) = 4$ and $\omega(G[3,2]) = 3$, $\lambda(G[3,2]) \leq \Delta(G[3,2]) + \omega(G[3,2]) = 4+3 = 7$. Now, we will prove that $\lambda(G[3,2]) > 6$. We will prove it by contradiction.

If possible, let f be any 6-$L(2,1)$-coloring of $G[3,2]$. Consider B^*, the internal block of type 3 of G. Since, B^* is a block, $f(B^*) \in \{\{0,2,4\},\{0,2,5\}, \{0,2,6\},\{0,3,5\},\{0,3,6\},\{0,4,6\},\{1,3,5\},\{1,3,6\},\{1,4,6\},\{2,4,6\}\}$. Since B^* is an internal block of type i for each $i \in \{1,2,3\}$, by Lemmas 5.6, 5.7 and 5.8, $f(B^*) \notin \{\{0,2,4\},\{0,2,6\},\{0,3,5\},\{1,3,6\},\{1,4,6\},\{0,4,6\}, \{2,4,6\},\{0,2,5\},\{1,3,5\},\{0,3,6\}\}$. This contradicts that f is a 6-$L(2,1)$-labeling of $G[3,2]$. So, $\lambda(G[3,2]) \geq 7$ and hence $\lambda(G[3,2]) = \Delta(G[3,2]) + 3$. □

Since $\lambda(G) \geq \lambda(H)$ if H is a subgraph of G, we have the following corollary.

Corollary 5.9. *If a block graph G with $\omega(G) = 3$ and $\Delta(G) = 4$ contains $G[3,2]$ of Figure 7 as a subgraph, then $\lambda(G) = \Delta(G) + \omega(G)$.*

6 k-Labeling of Block Graphs

In this section, we study the problem of $L(2,1)$-labeling a block graph G with a given number of colors. We show that it is related to finding certain matching in a certain Hyper graph.

Let T^B be the cut-tree of the block graph G rooted at an end block $\mathcal{Q}$. We call $\mathcal{Q}$, the ***root block*** and cut vertex $u \in \mathcal{Q}$ (u is the child of $\mathcal{Q}$), the ***root cut vertex*** of G, respectively. Every vertex w not belonging to $\mathcal{Q}$ has a parent cut vertex denoted by $P_c(w)$ and has a parent block B where $B = P_B(P_c(w))$. Let $T^B(v)$ be the subgraph of T^B rooted at the cut vertex v. Let $\mathscr{B}_G(v)$ be the subgraph of G corresponding to $T^B(v)$. Let the parent block of v be $P_B(v) = B$. The tree $T^B(v, P_B(v))$ is obtained by adding a vertex to $T^B(v)$ and joining it to v only. The subgraph $\mathscr{B}^*{}_G(v)$ of G corresponding to $T^B(v, P_B(v))$ is obtained by attaching B to $\mathscr{B}_G(v)$ at v. Note that, $\mathscr{B}^*{}_G(u) = G$. $T^B(v)$ and $T^B(v, P_B(v))$ for a cut tree T^B of a block graph G together with $\mathscr{B}_G(v)$, $\mathscr{B}^*{}_G(v)$ are depicted in Figure 8.

Let v be a cut vertex of G. Let $B_1, B_2, \ldots, B_s$ be the children of v in T^B. Let $B = \{w_j^i | 1 \le j \le s_i\}$ where $s_i = |B_i|$, $P_B(v) = \{v_1', v_2', \ldots, v_p'\}$, $v_p' = P_c(v)$, and $p = |P_B(v)|$. So, $v = v_p' = w_1^i$ for $1 \le i \le s$. Define,

$\mathscr{T}[\mathscr{B}_G(v)] = \{(a_1, a_2, \ldots, a_{q-1}, b, a_{q+1} \ldots, a_p) |$ there is a k-L$(2,1)$-labeling f of $\mathscr{B}^*{}_G(v)$ with $f(v) = b$ and $f(v_i') = a_i$, $\forall\, v_i' \in P_B(v)\}$.

Then, $\lambda(G) \le k$ if and only if $\mathscr{T}[\mathscr{B}_G(u)] \ne \phi$.

Definition 6.1. An s-tuple $(A)_{i=1}^s \equiv (A_1, A_2, \ldots, A_s)$ is called a system of distinct representative (SDR) for a system of sets $\mathscr{A}_{i=1}^s \equiv (\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_s)$ if $A_i \in \mathscr{A}_i$ for $1 \le i \le s$ and $A_i \cap A_j = \emptyset$ for each i and j with $i \ne j$.

The following theorem gives a recursive structure of $\mathscr{T}[\mathscr{B}_G(v)]$.

Theorem 6.2. $\mathscr{T}[\mathscr{B}_G(v)] = \Big\{(a_1, a_2, \ldots, a_{q-1}, b, a_{q+1} \ldots, a_p) | 0 \le a_\alpha \le k,$ $0 \le b \le k$, $|a_\alpha - b| \ge 2$, $|a_\alpha - a_\beta|_{\alpha \ne \beta} \ge 2$, $1 \le \alpha, \beta \le p$ *and* $(\mathscr{A})_{i=1}^s$ *has an SDR, where* $\mathscr{A}_i \equiv \{A_i | A_i = (c_2^i, c_3^i, \ldots, c_{s_i}^i)$, $c_j^i \ne a_\alpha$, $2 \le j \le s_i$, $1 \le \alpha \le p$ *and* $(b, c_2^i, \ldots, c_{s_i}^i) \in \mathscr{T}[\mathscr{B}_G(w_j^i)]$ *for every cut vertex* $w_j^i \in B_i \Big\}$

Proof. Let $\mathscr{T}$ denote the set on the right hand side of the equality in the theorem. Suppose $(a_1, a_2, \ldots, a_{q-1}, b, a_{q+1} \ldots, a_p) \in \mathscr{T}[\mathscr{B}_G(v)]$. So, there is a k-$L(2,1)$-labeling f of $\mathscr{B}_G(v)$ with $f(v) = b$ and $f(v_i') = a_i$ for all $1 \le i \le p$. Obviously, $0 \le a_\alpha \le k$, $0 \le b \le k$, $|a_\alpha - b| \ge 2$, $|a_\alpha - a_\beta|_{\alpha \ne \beta} \ge 2$, $1 \le \alpha, \beta \le p$. Let f_{ij} be the restriction of f on $\mathscr{B}^*{}_G(w_j^i)$. Then $w_j'^i = w_j^i$ for $j \le 2 \le s_i$ and $1 \le i \le s$. Now f_{ij} is a k-$L(2,1)$-labeling of $\mathscr{B}^*{}_G(w_j^i)$ with $f_{ij}(w_t'^i) = f(w_t^i)$, $f_{ij}(w_1^i) = f(v) = b$, $f(w_t^i) \ne a_\alpha$, $1 \le \alpha \le p$ for all $2 \le t \le$

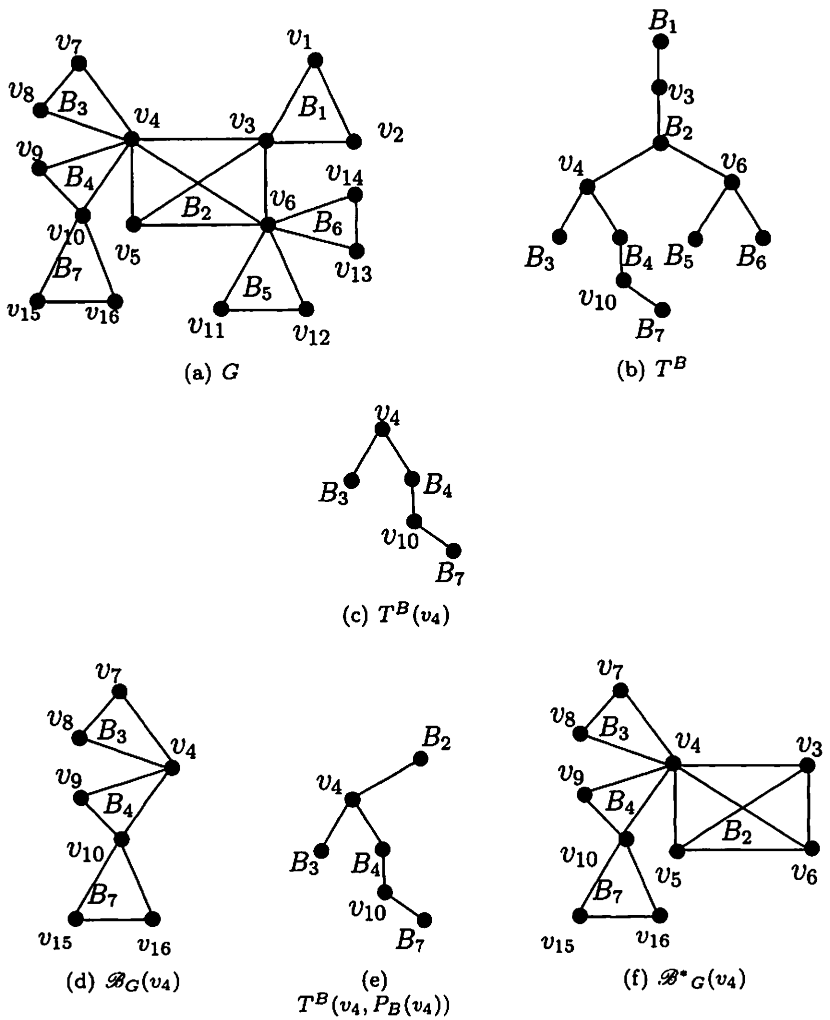


Figure 8: (a) a block graph G with 7 blocks, (b) the cut tree T^B of G rooted at end block B_1, (c) the subgraph $T^B(v_4)$ of T^B rooted at v_4, (d) the subgraph $\mathscr{B}_G(v_4)$ of G corresponding to $T^B(v_4)$,(e) the subgraph $T^B(v_4, P_B(v_4))$ of T^B rooted at B_2 obtained from $T^B(v_4)$ and (f) the subgraph $\mathscr{B}^*{}_G(v_4)$ of G corresponding to $T^B(v_4, P_B(v_4))$

s_i and $1 \leq i \leq s$ i,.e., $(b, f(w_2^i), \ldots, f(w_{j-1}^i), f(w_j^i), f(w_{j+1}^i), \ldots, f(w_{s_i}^i)) \in \mathscr{T}[\mathscr{B}_G(w_j^i)]$ and $(f(w_2^i), \ldots, f(w_{s_i}^i)) \in \mathscr{A}_i$. Thus it is an SDR of $(\mathscr{A})_{i=1}^s$. This proves that $\mathscr{T}[\mathscr{B}_G(v)] \subseteq \mathscr{T}$.

On the other hand, suppose $(a_1, a_2, \ldots, a_{q-1}, b, a_{q+1} \ldots, a_p) \in \mathscr{T}$. Then $0 \leq a_\alpha \leq k$, $0 \leq b \leq k$, $|a_\alpha - b| \geq 2$, $|a_\alpha - a_\beta|_{\alpha \neq \beta} \geq 2$, $1 \leq \alpha, \beta \leq p$ and $(\mathscr{A})_{i=1}^s$ has an SDR. Let f_{ij} be the k-$L(2,1)$-labeling of $\mathscr{B}_G(w_j^i)$ such that $f_{ij}(w_j^i) = c_j^i$ where $c_j^i \in A_i$ for $j \leq 2 \leq s_i$ and $1 \leq i \leq s$. Now, $w_t'^i = w_t^i$ and $w_1'^i = v$ for $2 \leq t \leq s_i$ and $1 \leq i \leq s \Rightarrow f_{ij}(w_1'^i) = b$. Consider the labeling f of $\mathscr{B}_G(v)$. Now $f(w_t^i) = f_{ij}(w_t'^i) = f_{ij}(w_t^i)$, $f(x) = f_{ij}(x)$ for $x \in \mathscr{B}_G(w_j^i)$ and $f(v_i') = a_i$ such that $v_i' \in P_B(v)$, $i \neq q$ it is easy to see that f is a $k - L(2,1)$labeling of $\mathscr{B}_G(v)$ with $f(v_i) = a_i$, $v_i' \in P_B(v)$ and $f(v) = b$. Therefore, $(a_1, a_2, \ldots, a_{q-1}, b, a_{q+1} \ldots, a_p) \in \mathscr{T}[\mathscr{B}_G(v)]$. This proves that $\mathscr{T} \subseteq \mathscr{T}[\mathscr{B}_G(v)]$.

□

Next, we propose an algorithm to decide whether a block graph G has a k-$L(2,1)$-labeling. We calculate $\mathscr{B}_G(v)$ for all cut vertices v of the block graph G. The algorithm first construct a cut tree T^B of G rooted at an end block. It visits all cut vertices from leaf level and works towards the root cut vertex, calculating the colors admissible for the vertices of each block by the set $\mathscr{T}[\mathscr{B}_G(v)]$. At any stage, if $\mathscr{B}_G(x) = \emptyset$ for some cut vertex x, then G does not admit an $L(2,1)$-labeling with k colors. For every end block B^* at the leaf level of cut-tree T^B, we select the cut vertex w in B^* and calculate $\mathscr{T}[\mathscr{B}_G(w)]$ as follows:

$$\begin{aligned}\mathscr{T}[\mathscr{B}_G(w)] = \; & \Big\{(a_1, a_2, \ldots, a_{q-1}, b, a_{q+1} \ldots, a_p) : 0 \leq a_i \leq k,\ 0 \leq b \leq k, \\ & |a_i - b| \geq 2, |a_i - a_j|_{i \neq j} \geq 2,\ 1 \leq i, j \leq p\Big\}, \text{ where } p = |B|.\end{aligned}$$

For any cut vertex v, let $B_1, B_2, \ldots, B_s$ be its children. We use $\mathscr{T}[\mathscr{B}_G(w_j^i)]$ to calculate $\mathscr{T}[\mathscr{B}_G(v)]$ for every cut vertex $w_j^i \in B_i$. Let h_i denotes the size of the child block B_i i.e., $h_i = |B_i|$ for $1 \leq i \leq s$. For any $(a_1, a_2, \ldots, a_q, b, a_{q+1} \ldots, a_p)$ with $0 \leq a_i \leq k$, $0 \leq b \leq k$, $|a_i - b| \geq 2$ and $|a_i - a_j|_{i \neq j} \geq 2$ for $1 \leq i, j \leq p$, we check if $(a_1, a_2, \ldots, a_q, b, a_{q+1} \ldots, a_p) \in \mathscr{T}[\mathscr{B}_G(v)]$ by the following method.

Construct a hyper graph graph $\mathscr{H} = (X \cup Y, \mathscr{E})$ such that

$$X = \{x_j^i : 1 \leq j \leq h_i, 1 \leq i \leq s\},\ Y = \{0, 1, \ldots, k\},$$

and

$$\begin{aligned}\mathscr{E} = \; & \Big\{\{x_1^i, \ldots x_{h_i}^i, c_1, \ldots, c_{h_i}\} | c_j \neq a_\alpha, 1 \leq \alpha \leq p, 1 \leq j \leq h_i, 1 \leq i \leq s, \\ & \mathit{and}\ \ (b, c_1, c_2, \ldots, c_{h_i}) \in \mathscr{T}[\mathscr{B}_G(w_j^i)]\Big\}\end{aligned}$$

Theorem 6.3. *Let $B_1, B_2, \ldots, B_s$ be the children of a cut vertex v in T^B and $h_i = |B_i|$ for $1 \leq i \leq s$. Let $\mathscr{H} = (X \cup Y, \mathscr{E})$ be the hyper graph graph such that $X = \{x^i_j : 1 \leq j \leq h_i, 1 \leq i \leq s\}$, $Y = \{0, 1, \ldots, k\}$, and $\mathscr{E} = \Big\{(x^i_1, \ldots x^i_{h_i}, c_1, \ldots, c_{h_i}) | c_j \neq a_\alpha, 1 \leq \alpha \leq p,\ 1 \leq j \leq h_i,\ 1 \leq i \leq s$, and $(b, c_1, c_2, \ldots, c_{h_i}) \in \mathscr{T}[\mathscr{B}_G(w^i_j)]\Big\}$. Then, $(a_1, a_2, \ldots, a_q, b, a_{q+1} \ldots, a_p) \in \mathscr{T}[\mathscr{B}_G(v)]$ if and only if there is a matching $\mathscr{M}$ in $\mathscr{H}$ that covers all the vertices of X.*

Proof. Let $\mathscr{M}$ be a maximum matching in $\mathscr{H}$. $|X| = \sum_{i=1}^{s} h_i$ and $|Y| = k+1$. The vertices in X corresponds to vertices of the children of v and vertices of Y are the colors and every edge of the hyper graph $\mathscr{H}$ will contain equal number of vertices from X and from Y. By construction, a hyper edge corresponds a coloring of a child block. Since $\mathscr{M}$ is a matching and for any pair of hyper edges $e_1, e_2 \in \mathscr{M}$, $e_1 \cap e_2 = \emptyset$. If $\mathscr{M}$ covers X then the hyper edges in $\mathscr{M}$ will correspond to a labeling of child blocks. Also, by construction $c_j \neq a_\alpha, 1 \leq \alpha \leq p, 1 \leq j \leq h_i, 1 \leq i \leq s$, and $(b, c_1, c_2, \ldots, c_{h_i}) \in \mathscr{T}[\mathscr{B}_G(w^i_j)]$. Therefore, $(a_1, a_2, \ldots, a_q, b, a_{q+1} \ldots, a_p) \in \mathscr{T}[\mathscr{B}_G(v)]$.

Let $(a_1, a_2, \ldots, a_q, b, a_{q+1} \ldots, a_p) \in \mathscr{T}[\mathscr{B}_G(v)]$. So, , $0 \leq a_\alpha \leq k$, $0 \leq b \leq k$, $|a_\alpha - b| \geq 2$, $|a_\alpha - a_\beta|_{\alpha \neq \beta} \geq 2$, $1 \leq \alpha, \beta \leq p$. By Theorem 6.2 for every cut vertex $w^i_j \in B_i$ there is $(b, c^i_2, \ldots, c^i_{s_i}) \in \mathscr{T}[\mathscr{B}_G(w^i_j)]$ such that $\{(c^i_2, c^i_3, \ldots, c^i_{s_i}),\ c^i_j \neq a_\alpha,\ 2 \leq j \leq s_i,\ 1 \leq \alpha \leq p\}$ is an SDR of $(\mathscr{A})^s_{i=1}$. Take $\mathscr{M} = \{e_i | e_i = (x^i_1, \ldots x^i_{h_i}, c^i_2, \ldots, c^i_{s_i}), 1 \leq i \leq s\}$. Since $h_i = s_i - 1$, hence $\mathscr{M}$ is a matching that covers X.

□

For any $(a_1, a_2, \ldots, a_q, b, a_{q+1} \ldots, a_p)$ with $0 \leq a_i \leq k$, $0 \leq b \leq k$, $|a_i - b| \geq 2$ and $|a_i - a_j|_{i \neq j} \geq 2$ for $1 \leq i, j \leq p$, $(a_1, a_2, \ldots, a_q, b, a_{q+1} \ldots, a_p) \in \mathscr{T}[\mathscr{B}_G(v)]$ iff Theorem 6.3 holds for the cut vertex v.

Example: We illustrate the above algorithm with an example. Consider the block graph G shown in Figure 9. G has only one cut vertex v and 3 blocks B_1, B_2, B_3. We will check whether $\lambda(G) \leq 7$.

For each $i, 1 \leq i \leq 4$, $\mathscr{T}[\mathscr{B}_G(w_i)] = \{(0,2,4)$, $(0,2,5)$, $(0,2,6)$, $(0,2,7)$, $(0,3,5)$, $(0,3,6)$, $(0,3,7)$, $(0,4,6)$, $(0,4,7)$, $(0,5,7)$, $(1,3,5)$, $(1,3,6)$, $(1,3,7)$, $(1,4,6)$, $(1,4,7)$, $(1,5,7)$, $(1,6,3)$, $(1,6,4)$, $(1,7,3)$, $(1,7,5)$, $(2,0,4)$, $(2,0,5)$, $(2,0,6)$, $(2,0,7)$, $(2,4,0)$, $(2,4,6)$, $(2,4,7)$, $(2,5,0)$, $(2,5,7)$, $(2,6,0)$, $(2,6,4)$, $(2,7,0)$, $(2,7,4)$, $(2,7,5)$, $(3,0,5)$, $(3,0,6)$, $(3,0,7)$, $(3,1,0)$, $(3,1,5)$, $(3,1,6)$, $(3,1,7)$, $(3,5,0)$, $(3,5,7)$, $(3,6,0)$, $(3,6,1)$, $(3,7,0)$, $(3,7,1)$, $(3,7,5)$, $(4,0,2)$, $(4,0,6)$, $(4,0,7)$, $(4,1,6)$, $(4,1,7)$, $(4,2,0)$, $(4,2,6)$, $(4,2,7)$, $(4,6,0)$, $(4,6,1)$, $(4,6,2)$, $(4,7,0)$, $(4,7,1)$, $(4,7,2)$, $(5,0,2)$, $(5,0,3)$, $(5,0,7)$, $(5,1,3)$, $(5,1,7)$, $(5,2,0)$, $(5,2,7)$, $(5,3,0)$, $(5,3,1)$, $(5,3,7)$, $(5,7,0)$, $(5,7,1)$, $(5,7,2)$, $(5,7,3)$, $(6,0,2)$, $(6,0,3)$, $(6,0,4)$, $(6,1,3)$, $(6,1,4)$, $(6,2,0)$, $(6,2,4)$, $(6,3,0)$, $(6,3,1)$,

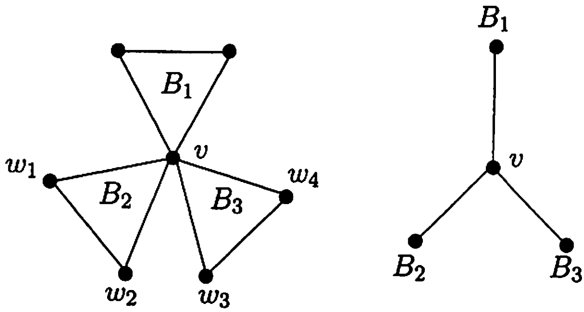


Figure 9: *a*) A block graph G with 3 blocks, *b*) the cut tree T^B of G rooted at end block B_1

$(6,4,0), (6,4,1), (6,4,2), (7,0,2), (7,0,3), (7,0,4), (7,0,5), (7,1,3), (7,1,4), (7,1,5), (7,2,0), (7,2,4), (7,2,5), (7,3,0), (7,3,1), (7,3,5), (7,4,0), (7,4,1), (7,4,2)\}$.
The next step is to compute $\mathscr{T}[\mathscr{B}_G(v)]$.

To check any color set (c_1, c_2, c_3) to be in $\mathscr{T}[\mathscr{B}_G(v)]$ construct hypergraph $\mathscr{H} = (X, Y, \mathscr{E})$ where $X := \{x_1^1, x_2^1, x_1^2, x_2^2\}$, $Y := \{0, 1, \ldots, 7\}$. For example to check $(4,0,6) \in \mathscr{T}[\mathscr{B}_G(v)]$, take $\mathscr{E} := \{(x_1^1, x_2^1, 2, 5), (x_1^1, x_2^1, 2, 7), (x_1^1, x_2^1, 3, 5), (x_1^1, x_2^1, 3, 7), (x_1^1, x_2^1, 5, 7), (x_1^2, x_2^2, 2, 5), (x_1^2, x_2^2, 2, 7), (x_1^2, x_2^2, 3, 5), (x_1^2, x_2^2, 3, 7), (x_1^2, x_2^2, 5, 7)\}$. $M = \{(x_1^1, x_2^1, 2, 5), (x_1^2, x_2^2, 3, 7)\}$ is a matching that covers X. So, $(4,0,6) \in \mathscr{T}[\mathscr{B}_G(v)]$.

To check $(0,2,6) \in \mathscr{T}[\mathscr{B}_G(v)]$, take $\mathscr{E} := \{(x_1^1, x_2^1, 4, 7), (x_1^1, x_2^1, 5, 7), (x_1^1, x_2^1, 7, 4), (x_1^2, x_2^2, 4, 7), (x_1^2, x_2^2, 5, 7), (x_1^2, x_2^2, 7, 4)\}$. Clearly, in this case there is no matching that covers X. Hence, $(0,2,6) \notin \mathscr{T}[\mathscr{B}_G(v)]$.
In this manner we can compute other members of $\mathscr{T}[\mathscr{B}_G(v)]$.

The complexity of the above algorithm depends on the complexity of the testing the conditions of Theorem 6.3. However, this complexity is not known.

7 Conclusion

In this paper, we proposed an $L(2,1)$-labeling of block graphs to establish that $\lambda(G) \leq \Delta(G) + \omega(G)$ for a block graph G. We also established lower upper bounds on λ for some special block graphs. We study the structure of extremal block graphs, i.e. block graphs for which $\lambda(G) = \Delta(G) + \omega(G)$. We also constructed a class of extremal block graphs. Finally, we proposed an algorithm to find $\lambda(G)$ of a block graph G. The complexity of this algorithm depends on the complexity of the problem of finding a certain matching in an appropriate hyper graph. Our study leaves the following problems open.

- Characterize extremal block graphs.
- Find a polynomial time algorithm to compute $\lambda(G)$ for a block graph G.

Acknowledgement

The authors are grateful to the anonymous referee for the constructive suggestions leading to improvements in the presentation of the paper.

References

[1] M. Behzad, G. Chartrand, and L. Lesniak-Foster. *Graphs and digraphs.* Wadsworth International Group, Belmont, CA, 1979.

[2] A. A. Bertossi, C. M. Pinotti, and R. B. Tan, *Efficient use of radio spectrum in wireless networks with channel separation between close stations*, DIALM '00: Proceedings of the 4th international workshop on Discrete algorithms and methods for mobile computing and communications (New York, NY, USA), ACM, (2000), pp. 18-27.

[3] H. L. Bodlaender, T. Kloks, R. B. Tan, and J. V. Leeuwen, *Approximations of λ-colorings of graphs*, Comput. J. **47** (2004), pp. 193–204.

[4] T. Calamoneri, *The $L(h, k)$-labelling problem: A survey and annotated bibliography*, Comput. J. **49** (2006), pp. 585–608.

[5] T. Calamoneri, S. Caminiti, S. Olariu, and R. Petreschi, *On the $L(h, k)$-labeling of co-comparability graphs and circular-arc graphs,* Networks, 53 (2009), pp. 27–34.

[6] T.Calamoneri and R. Petreschi, *$L(h, 1)$-labeling subclasses of planar graphs,* Journal of Parallel and Distibuted Computing, 64 (2004), pp. 414-426.

[7] G. J. Chang and D. Kuo, *The $L(2, 1)$-labeling problem on graphs*, SIAM J. Discrete Math. **9** (1996), pp. 309-316.

[8] J. Fiala, T. Kloks, and J. Kratochvil, *Fixed parameter complexity of λ-labelings*, Discrete Appl. Math. **113** (2001), pp. 59-72.

[9] M. C. Golumbic, Algorithmic Graph Theory and Perfect Graphs, 2nd Edition, Annals of Discrete Mathematics 57, Elsevier, 2004.

[10] D. Gonçalves, *On the L(p,1)-labelling of graphs*, Discrete Math. **308** (2008), pp. 1405 - 1414.

[11] J. R. Griggs and R. K. Yeh. *Labelling graphs with a condition at distance 2.* SIAM J. Discrete Math. **5** (1992), pp. 586–595.

[12] W. K. Hale, *Frequency assignment: theory and applications*, Proc. IEEE **68**, (1980), pp. 1497-1514.

[13] F. Havet, B. Reed, and J. S. Sereni, $L(2,1)$-labeling of Graphs, Proc. 19th annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2008), SIAM, 2008, 621-630.

[14] J.-H. Kang, $L(2,1)$*-Labeling of Hamiltonian graphs with Maximum Degree* 3, SIAM J. Discrete Math. **22** (2008), pp. 213-230 .

[15] D. Král' and R. Skrekovski, *A theorem about the channel assignment problem*, SIAM J. Discrete Math. **16** (2003), pp. 426–437.

[16] D. Sakai, *Labeling chordal graphs: Distance two condition*, SIAM J. Discret. Math. **7** (1994), pp. 133–140.