Bipartite permutation graph の $L(2,1)$ ラベリング

荒木 徹

岩手大学工学部情報システム工学科
〒 020-8551 岩手県盛岡市上田 4-3-5

概要：グラフ G の $L(2,1)$ ラベリングとは，G の頂点集合から非負整数の集合 $\{0,1,\ldots,\lambda\}$ への写像 f であり，u,v が隣接していれば $|f(u)-f(v)|\geq 2$，u,v 間の距離が 2 なら $|f(u)-f(v)|\geq 1$ となるものである．グラフ G が $L(2,1)$ ラベリングを持つ最小の λ の値を $\lambda(G)$ と表す．この問題は，無線ネットワークのチャネル割り当て問題をモデル化したものである．

本論文では，bipartite permutation graph に対して $L(2,1)$ ラベリングを求める多項式時間アルゴリズムを示す．提案アルゴリズムのラベリングで用いられる最大のラベルは高々$\lambda(G)+1$ であり，最適に近いラベリングを計算する．

$L(2,1)$-labeling of bipartite permutation graphs

Toru Araki

Department of Computer and Information Sciences, Iwate University
Morioka, Iwate 020-8551
arakit@cis.iwate-u.ac.jp

Abstract: An $L(2,1)$-labeling of a graph G is an assignment f from vertices of G to the set of non-negative integers $\{0,1,\ldots,\lambda\}$ such that $|f(u)-f(v)|\geq 2$ if u and v are adjacent, and $|f(u)-f(v)|\geq 1$ if u and v are at distance 2 apart. The minimum value of λ for which G has $L(2,1)$-labeling is denoted by $\lambda(G)$. The $L(2,1)$-labeling problem is related to the channel assignment problem for wireless networks.

In this paper, we present a polynomial time algorithm for computing $L(2,1)$-labeling of a bipartite permutation graph G such that the largest label is at most $\lambda(G)+1$, which is a nearly optimal value.

1 Introduction

The *channel assignment problem* for wireless networks is to assign a channel to each radio transmitter so that close transmitters are received channels so as to avoid interference. This situation can be modeled by a graph whose vertices are the radio transmitters, and the adjacency indicate possible interference. The aim is to assign integers (corresponding to the channel) to the vertices such that adjacent vertices receive integers at least 2 apart, and nonadjacent vertices with a common neighbor receive distinct integers. This is called $L(2,1)$-labeling problem which is widely accepted model for the channel assignment problem. A formal definition is given as follows.

Definition 1.1. *An* $L(2,1)$-labeling *of* G *is an assignment* f *from* V *to the set of integers* $\{0,1,\ldots,\lambda\}$ *such that* $|f(u)-f(v)|\geq 2$ *if* $uv\in E$ *and* $|f(u)-f(v)|\geq 1$ *if* $\mathrm{dist}(u,v)=2$. *The minimum value of* λ *for which* G *has* $L(2,1)$*-labeling is denoted by* $\lambda(G)$.

The notion of $L(2,1)$-labeling has attracted a lot of attention for not only its motivation by the channel assignment problem, and also for its interesting graph theoretic properties. Griggs and Yeh [5] first considered this problem. There are many papers that study the problem for several graph classes (for example, see surveys [3, 11]). The complexity for deciding $\lambda(G)\leq k$ for fixed k is NP-complete [5], and for bipartite graphs and chordal graph are also NP-complete [1].

In this paper, we focus on the class of *bipartite permutation graphs* which is a permutation graph and bipartite graph. This class was investigated by Spinrad, Brandstädt, and Stewart [7]. Studies for the class are motivated by the fact that many NP-hard problems are efficiently solved in graphs of this class.

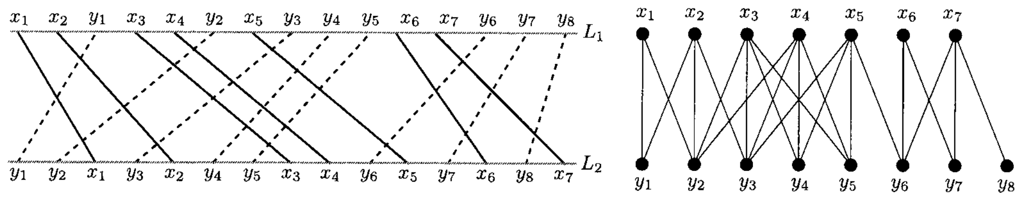

Figure 1: A bipartite permutation graph and the corresponding permutation diagram.

For example, algorithms for domination problems [6, 10], the path partition problem [8], and the longest path problem [9] were investigated. Books [2, 4] include surveys some algorithmic result for the class. Boldaender et al. [1] proved that $\lambda(G) \leq 5\Delta - 2$ for any permutation graph, where Δ is the maximum degree of G, and such labeling is calculated by a polynomial time algorithm that is greedy manner.

We consider the $L(2,1)$-labeling problem for bipartite permutation graphs. We present a polynomial time algorithm for computing a *nearly* optimal labeling. More precisely, the maximum value assigned for vertices is at most $\lambda(G) + 1$.

2 Preliminaries

Let $G = (V, E)$ be a graph with vertex set V and edge set E. The *neighborhood* of a vertex u is $N_G(u) = \{v \mid uv \in E\}$. The *degree* of a vertex u is $\deg u = |N_G(u)|$. The *distance* between two vertices u and v, denoted by $\text{dist}(u, v)$, is the length of shortest path between u and v. A graph $G = (V, E)$ is *bipartite* if V can be partitioned into two subsets X and Y such that every edge joins a vertex in X and another vertex in Y. A partition $X \cup Y$ of V is called *bipartition*. A bipartite graph with bipartition $X \cup Y$ is denoted by $G = (X, Y, E)$. A bipartite graph $G = (X, Y, E)$ is *complete* if each vertex in X is adjacent to every vertices in Y. For a bipartite graph, a subset of vertices is *biclique* if it induces a complete bipartite subgraph. The *biclique number* of a bipartite graph G is the number of vertices in a maximum biclique of G and it is denoted by $bc(G)$.

A graph $G = (V, E)$ with $V = \{v_1, v_2, \ldots, v_n\}$ is called a *permutation graph* if there is a permutation π over $\{1, 2, \ldots, n\}$ such that $v_i v_j \in E$ if and only if $(i - j)(\pi^{-1}(i) - \pi^{-1}(j)) < 0$. When a permutation graph is bipartite, it is said to be a *bipartite permutation graph*.

Intuitively, a permutation graph can be constructed from a permutation $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$ on $\{1, 2, \ldots, n\}$ in the following visual manner. Line up the numbers 1 to n horizontally on a line L_1. On the line below it, line up the corresponding permutation so that π_i is below i on a line L_2. Then connect each i and π_i^{-1} with a line segment which is corresponding to vertex v_i. The resulting diagram is referred to as a *permutation diagram*. In the permutation graph corresponding to π, two vertices v_i and v_j are adjacent if and only if the corresponding lines are crossing. An example of a bipartite permutation graph and the corresponding permutation diagram is shown in Fig. 1.

In the permutation diagram of a bipartite permutation graph $G = (X, Y, E)$, we can order line segments $x_1, x_2, \ldots, x_m$ in X from left to right (these are drawn by solid lines in Fig. 1). We also order vertices $y_1, y_2, \ldots, y_n$ in Y from left to right (these are dotted lines in Fig. 1). From now on, we suppose that vertices in $X = \{x_1, x_2, \ldots, x_m\}$ and $Y = \{y_1, y_2, \ldots, y_n\}$ are sorted such that the corresponding lines are arranged from left to right in the permutation diagram. It should be noted that Spinrad et al. [7] developed an $O(|V| + |E|)$ time algorithm for recognizing whether a given graph is a bipartite permutation graph and producing such orderings of the vertices if so.

A bipartite graph $G = (X, Y, E)$ is a *chain graph* if vertices can be ordered by inclusion: that is, there is an ordering of vertices $x_1, x_2, \ldots, x_m$ in X and $y_1, y_2, \ldots, y_n$ in Y such that $N_G(x_1) \subseteq N_G(x_2) \subseteq \cdots \subseteq N_G(x_m)$ and $N_G(y_n) \subseteq \cdots \subseteq N_G(y_2) \subseteq N_G(y_1)$. It is known that any chain graph is a bipartite permutation graph [9].

Lemma 2.1 (Uehara, Valiente [9]). *Let $G = (X, Y, E)$ be a connected chain graph with $N_G(x_1) \subseteq N_G(x_2) \subseteq \cdots \subseteq N_G(x_m)$ and $N_G(y_n) \subseteq \cdots \subseteq N_G(y_2) \subseteq N_G(y_1)$. Then, it has a corresponding permutation diagram such that (1) $x_1 < x_2 < \cdots < x_m < y_1 < y_2 < \cdots < y_n$ on L_1, and (2) $y_1 < x_1$ and*

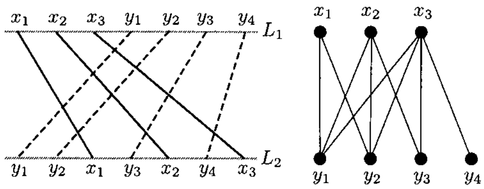

Figure 2: A chain graph and the corresponding permutation diagram.

$y_n < x_m$ on L_2. Conversely, if a graph G has a corresponding permutation diagram satisfying conditions (1) and (2), then it is a connected chain graph.

Figure 2 shows an example of chain graph and the corresponding permutation diagram.

3 Labeling of chain graphs

In this section, we show that an optimal $L(2,1)$-labeling of chain graph can be solved in linear time. For simplicity, we may assume that the given graph is connected. The following is easily obtained.

Lemma 3.1. *For a complete bipartite graph $G = (X, Y, E)$, $\lambda(G) = |X| + |Y|$.*

It is obvious that $\lambda(G) \geq \lambda(H)$ if H is a subgraph of G. Hence we obtain a lower bound of $\lambda(G)$ for any bipartite graph G from Lemma 3.1.

Corollary 3.2. *$\lambda(G) \geq bc(G)$ for any bipartite graph G.*

Theorem 3.3. *Let $G = (X, Y, E)$ be a connected chain graph such that $N_G(x_1) \subseteq N_G(x_2) \subseteq \cdots \subseteq N_G(x_m)$ and $N_G(y_n) \subseteq \cdots \subseteq N_G(y_2) \subseteq N_G(y_1)$. Define a labeling cl of vertices such that*

$$\begin{aligned} \mathsf{cl}(x_i) &= bc(G) - m + i, \text{ for } 1 \leq i \leq m, \\ \mathsf{cl}(y_j) &= j - 1, \text{ for } 1 \leq j \leq n. \end{aligned}$$

Then cl is an optimal $L(2,1)$-labeling of G. The labeling cl satisfies the inequality $2 \leq \mathsf{cl}(x_i) - \mathsf{cl}(y_j) \leq bc(G)$ for $x_iy_j \in E$. Moreover, $\mathsf{cl}(x_i) - \mathsf{cl}(y_j) = bc(G)$ for $x_iy_j \in E$ if and only if $i = m$ and $j = 0$.

Proof. Since every vertex in X (or Y) receives distinct labels, every pair of vertices distance two apart have distinct labels.

Then we show that $\mathsf{cl}(x_i) - \mathsf{cl}(y_j) \geq 2$ if $x_iy_j \in E$. Suppose to the contrary that $\mathsf{cl}(x_i) - \mathsf{cl}(y_j) \leq 1$. Then $(k - m + i) - (j - 1) \leq 1$, where $k = bc(G)$. Hence $k \leq m - i + j$. On the other hand, the set of vertices $\{x_i, x_{i+1}, \ldots, x_m\} \cup \{y_1, y_2, \ldots, y_j\}$ induces a biclique. Thus we obtain $k \geq (m - i + 1) + j$. This contradicts the inequality $k \leq m - i + j$.

Since $\lambda(G) \geq bc(G)$ and $\max_{v \in X \cup Y} \mathsf{cl}(v) = \mathsf{cl}(x_m) = bc(G)$, the labeling f is an optimal $L(2,1)$-labeling. □

Lemma 3.4. $bc(G) = \max_{1 \leq j \leq n}\{j + \deg y_j\}$ *for a chain graph G.*

Proof. This can be derived easily from the fact that $N_G(x_1) \subseteq \cdots \subseteq N_G(x_m)$ and $N_G(y_n) \subseteq \cdots \subseteq N_G(y_1)$. □

We present an algorithm for computing the biclique number and an optimal labeling for a chain graph in Algorithm 1 and 2. Clearly, this algorithm runs in linear time.

Theorem 3.5. *An optimal $L(2,1)$-labeling of a chain graph can be computed in $O(N)$ time, where N is the number of vertices.*

An example of the $L(2,1)$-labeling cl obtained by LABELING_CHAIN(G) is illustrated in Fig. 3. The chain graph G with $|X| = 7$ and $|Y| = 6$ has the biclique number $bc(G) = \max_{y_j \in Y}\{j + \deg y_j\} = 3 + \deg y_3 = 9$ (in fact, the set $\{x_2, \ldots, x_7\} \cup \{y_1, y_2, y_3\}$ forms the maximum biclique).

Algorithm 1: BICLIQUE_CHAIN(G)

Input: a chain graph $G = (X, Y, E)$ with $X = \{x_1, \ldots, x_m\}$ and $Y = \{y_1, \ldots, y_n\}$.
Output: the biclique number $bc(G)$
$bc \leftarrow 0$;
for $j \leftarrow 1$ **to** n **do**
  if $bc < j + \deg y_j$ **then** $bc \leftarrow j + \deg y_j$
end
return bc

Algorithm 2: LABELING_CHAIN(G)

Input: a chain graph $G = (X, Y, E)$ with $X = \{x_1, \ldots, x_m\}$ and $Y = \{y_1, \ldots, y_n\}$.
Output: an $L(2,1)$-labeling cl of G
$bc \leftarrow$ BICLIQUE_CHAIN(G) ; /* the biclique number of G */
foreach $x_i \in X$ **do** $\mathsf{cl}(x_i) \leftarrow bc - m + i$;
foreach $y_j \in Y$ **do** $\mathsf{cl}(y_j) \leftarrow j - 1$;
return cl

4 Labeling of bipartite permutation graphs

In this section, we present a polynomial time algorithm for calculating an $L(2,1)$-labeling f for a bipartite permutation graph G so that $\max f(v) \leq bc(G) + 1$.

Definition 4.1. *Let $G = (X, Y, E)$ be a bipartite permutation graph. For $y_j \in Y$, let G_j be the subgraph of G induced by $X_j \cup Y_j$, where*

$$\begin{aligned} X_j &= N_G(y_j) = \{x_i, x_{i+1}, \ldots, x_k\}, \text{ and} \\ Y_j &= \{y_l \mid x_k y_l \in E \text{ and } l \geq j\}. \end{aligned}$$

Lemma 4.2. *G_j is a chain graph such that $N_{G_j}(x_i) \subseteq N_{G_j}(x_{i+1}) \subseteq \cdots \subseteq N_{G_j}(x_k)$ and $N_{G_j}(y_l) \subseteq \cdots \subseteq N_{G_j}(y_{j+1}) \subseteq N_{G_j}(y_j)$, where y_l is the maximum neighbor of x_k.*

Proof. It is easy to see that the vertices in G_j are arranged such that $x_i < x_{i+1} < \cdots < x_k < y_j < y_{j+1} < \cdots < y_l$ on L_1 of the permutation diagram, and $y_j < x_i$ and $y_l < x_k$ on L_2. By Lemma 2.1, the lemma holds. □

4.1 Algorithm

The outline of our algorithm for a bipartite permutation graph G is as follows:

1. Visit $y_r \in Y$ starting from $r = 1$ to $r = n$ consecutively.
   (a) Construct a chain graph G_r and calculate a labeling cl of G_r by LABELING_CHAIN.
   (b) Determine the $L(2,1)$-labeling label(v) of vertices v in G_r by *adjusting* cl to already assigned labels of G.

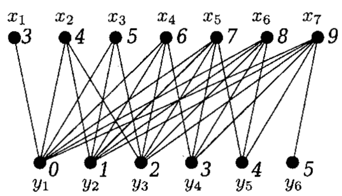

Figure 3: A chain graph G with $bc(G) = 9$ and its optimal $L(2,1)$-labeling.

Algorithm 3: LABELING_BIPARTITE_PERMUTATION(G)

Input: a bipartite permutation graph $G = (X, Y, E)$.
Output: an $L(2,1)$-labeling f of G

1 **foreach** $v \in X \cup Y$ **do** $\mathsf{label}(v) \leftarrow$ undef;
2 $x_{\max} \leftarrow x_0$; /* the maximum vertex in X s.t. $\mathsf{label}(x) \neq$ undef */
3 $y_{\max} \leftarrow y_0$; /* the maximum vertex in Y s.t. $\mathsf{label}(y) \neq$ undef */
4 $r \leftarrow 1$;
5 **while** $r \leq n$ **do**
6   **if** $\max N_G(y_r) > x_{\max}$ **then**
7   Construct the chain graph G_r ; /* Definition 4.1 */
8     $\mathsf{cl} \leftarrow$ LABELING_CHAIN(G_r);
9   **if** $r = 1$ **then** $s \leftarrow 0$;
10   **else** $s \leftarrow \max\{0, \mathsf{label}(x_{\max}) - \mathsf{cl}(x_{\max}), \mathsf{label}(y_{\max}) - \mathsf{cl}(y_{\max})\}$;
    /* $X_r \cup Y_r$ is the bipartition of G_r */
11   **foreach** $x \in X_r$ **do**
12   **if** $\mathsf{label}(x) =$ undef **then** $\mathsf{label}(x) \leftarrow \mathsf{cl}(x) + s$
13   **if** $s = 0$ *or* $s = \mathsf{label}(y_{\max}) - \mathsf{cl}(y_{\max})$ **then**
14   **foreach** $y \in Y_r$ **do**
15   **if** $\mathsf{label}(y) =$ undef **then** $\mathsf{label}(y) \leftarrow \mathsf{cl}(y) + s$
16   **else**
      /* $\mathsf{label}(x_{\max}) - \mathsf{cl}(x_{\max}) > \mathsf{label}(y_{\max}) - \mathsf{cl}(y_{\max})$ */
17   **foreach** $y \in Y_r$ **do** $\mathsf{label}(y) \leftarrow \mathsf{cl}(y) + s$
18     $x_{\max} \leftarrow \max N_{G_r}(y_r)$;
19     $y_{\max} \leftarrow \max N_{G_r}(x_{\max})$;
20   $r \leftarrow r + 1$
21 **foreach** $v \in X \cup Y$ **do** $f(v) \leftarrow \mathsf{label}(v) \bmod (bc(G) + 2)$;
22 **return** f

2. After the assignment label are determined for all vertices, calculate $f = \mathsf{label}(v) \bmod (bc(G) + 2)$, and output the resulting label assignment f.

The detail of the algorithm is described in Algorithm 3.

4.2 Example of our algorithm

We present Figs. 4–8 as an example of our labeling algorithm for the bipartite permutation graph G of Fig. 1.

1. In Fig. 4, the chain graph G_1 and its labeling cl are calculated. Then label for vertices in G_1 is defined.

2. The chain graph G_2 and its labeling cl are obtained as in Fig. 5. In this case, $s = \max\{0, \mathsf{label}(x_2) - \mathsf{cl}(x_2), \mathsf{label}(y_3) - \mathsf{cl}(y_3)\} = \max\{0, 4-4, 2-1\} = 1$. Thus $\mathsf{label}(v) = \mathsf{cl}(v) + 1$ for $v \in \{x_3, x_4, y_4, y_5\}$.

3. The chain graph G_3 and its labeling cl are obtained as in Fig. 6. In this case, $s = \max\{0, \mathsf{label}(x_4) - \mathsf{cl}(x_4), \mathsf{label}(y_5) - \mathsf{cl}(y_5)\} = \max\{0, 7-5, 4-2\} = 2$. Thus $\mathsf{label}(v) = \mathsf{cl}(v) + 2$ for $v \in \{x_5, y_6\}$.

4. The chain graph G_6 and its labeling cl are obtained as in Fig. 7. In this case, $s = \max\{0, \mathsf{label}(x_5) - \mathsf{cl}(x_5), \mathsf{label}(y_6) - \mathsf{cl}(y_6)\} = \max\{0, 8-2, 5-0\} = 6$. Since $s = 6 = \mathsf{label}(x_5) - \mathsf{cl}(x_5) > \mathsf{label}(y_6) - \mathsf{cl}(y_6)\}$, $\mathsf{label}(v) = \mathsf{cl}(v) + 6$ for $v \in \{x_6, x_7, y_6, y_7, y_8\}$ (line 17 of Algorithm 3).

5. Finally, $L(2,1)$-labeling f of G is obtained by $f(v) = \mathsf{label}(v) \bmod 8$ as shown in Fig. 8.

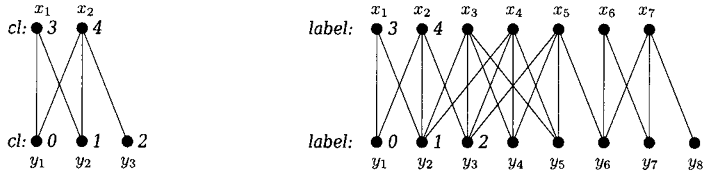

Figure 4: The chain graph G_1 and its labeling cl (left), and the labeling label of G (right). In this case, $s = 0$.

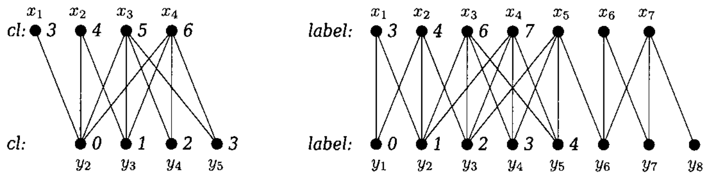

Figure 5: G_2 and its labeling cl, and label of G ($s = 1$).

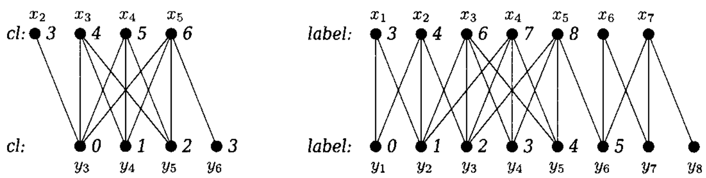

Figure 6: G_3 and its labeling cl, and label of G ($s = 2$).

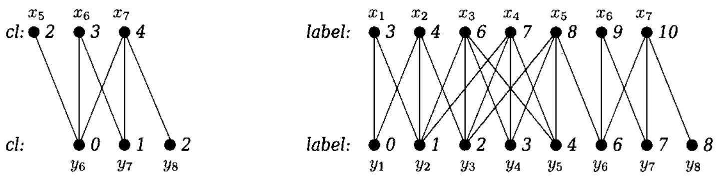

Figure 7: G_6 and its labeling cl, and label of G ($s = 6$).

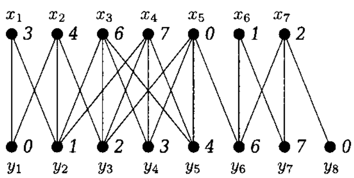

Figure 8: The $L(2,1)$-labeling of G which is obtained from label and $bc(G) + 2 = 8$.

Note that, in each step, the biclique number $bc(G_r)$ is equal to the value of $\mathsf{label}(x) - \mathsf{label}(y_r)$, where x is the maximum neighbor of y_r. For example, in Fig. 5, $bc(G_2) = 6 = \mathsf{label}(x_4) - \mathsf{label}(y_2)$ holds.

It also should be noted that if the condition $s = \mathsf{label}(x_{\max}) - \mathsf{cl}(x_{\max}) > \mathsf{label}(y_{\max}) - \mathsf{cl}(y_{\max})$ holds (line 17), the labeling of nodes of Y in G_r are increased. For example, $\mathsf{label}(y_6) = 5$ in Fig. 6, then it is increased to 6 in Fig. 7 because the situation $s = \mathsf{label}(x_5) - \mathsf{cl}(x_5) > \mathsf{label}(y_6) - \mathsf{cl}(y_6)$ occurs.

4.3 Correctness

The labeling label calculated in the algorithm is an $L(2,1)$-labeling of G, which is guaranteed by the following two lemmas.

Lemma 4.3. $2 \leq \mathsf{label}(x_i) - \mathsf{label}(y_j) \leq bc(G)$ *if* $x_i y_j \in E$.

Sketch of proof. If an edge $x_i y_j$ is in G_r, then $\mathsf{cl}(x_i) - \mathsf{cl}(y_j) \leq bc(G_r)$ by Lemma 3.3, where cl is the labeling of G_r. Since $\mathsf{label}(x_i) - \mathsf{label}(y_j) \leq \mathsf{cl}(x_i) - \mathsf{cl}(y_j)$, the inequality $\mathsf{label}(x_i) - \mathsf{label}(y_j) \leq bc(G)$ holds.

So we should show that $\mathsf{label}(x_i) - \mathsf{label}(y_j) \geq 2$. This condition would be violated only when the following situation occurs:

(i) $\mathsf{label}(y_j)$ is increased in line 17 for some chain graph G_r, and

(ii) The labels of vertices of X_r in G_r are not consecutive numbers.

An example of non-consecutive labels is G_3 in Fig. 6. The vertices of X_3, x_2, x_3, x_4 and x_5, have labels 4,6,7 and 8, respectively, in G, which are not consecutive numbers. Furthermore, vertex x_2 is adjacent to y_3 and $\mathsf{label}(x_2) - \mathsf{label}(y_3) = 2$. Thus, if $\mathsf{label}(y_3)$ would be increased after processing G_3, then $\mathsf{label}(x_2) - \mathsf{label}(y_3) < 2$.

However, we can show that the above situation (i) and (ii) does not occur simultaneously. The detailed proof of this will be presented in the full version of this paper. □

Lemma 4.4. *The labeling* label *satisfies the following inequalities:*

1. $1 \leq \mathsf{label}(x_k) - \mathsf{label}(x_i) \leq bc(G) - 2$ *if* $\mathrm{dist}(x_i, x_k) = 2$ *and* $1 \leq i < k \leq m$.
2. $1 \leq \mathsf{label}(y_l) - \mathsf{label}(y_j) \leq bc(G) - 2$ *if* $\mathrm{dist}(y_j, y_l) = 2$ *and* $1 \leq j < l \leq n$.

Proof. Suppose that $\mathrm{dist}(x_i, x_k) = 2$ and $i < k$. Clearly $\mathsf{label}(x_i) < \mathsf{label}(x_k)$. Let y be a common neighbor of x_i and x_k, and $q = \mathsf{label}(y)$. By Lemma 4.3, we have $\mathsf{label}(x_k) \leq q + bc(G)$ and $\mathsf{label}(x_i) \geq q + 2$. Hence $\mathsf{label}(x_k) - \mathsf{label}(x_i) \leq bc(G) - 2$.

Similarly, we suppose that $\mathrm{dist}(y_j, y_l) = 2$ and $j < l$. Clearly $\mathsf{label}(y_j) < \mathsf{label}(y_l)$. Let x be a common neighbor of y_j and y_l, and $p = \mathsf{label}(x)$. By Lemma 4.3, we have $\mathsf{label}(y_l) \leq p - 2$ and $\mathsf{label}(y_j) \geq p - bc(G)$. Hence $\mathsf{label}(y_l) - \mathsf{label}(y_j) \leq bc(G) - 2$. □

Theorem 4.5. *The labeling f calculated by Algorithm 3 is an $L(2,1)$-labeling of G, and* $\max_{v \in X \cup Y} f(v) \leq bc(G) + 1$. *This algorithm runs in $O(|V| + |E|)$ time.*

Proof. Since $f(v) = \mathsf{label}(v) \bmod (bc(G) + 2)$, the inequality $\max_{v \in X \cup Y} f(v) \leq bc(G) + 1$ holds.

Let $xy \in E$, where $x \in X$ and $y \in Y$. Then, by Lemma 4.3, $2 \leq \mathsf{label}(x) - \mathsf{label}(y) \leq bc(G)$. Since $f(x) = \mathsf{label}(x) \bmod (bc(G) + 2)$ and $f(y) = \mathsf{label}(y) \bmod (bc(G) + 2)$, the value of $|f(x) - f(y)|$ cannot be 0 or 1.

If $\mathrm{dist}(x_i, x_k) = 2$, then $\mathsf{label}(x_k) - \mathsf{label}(x_i) \leq bc(G) - 2$ by Lemma 4.4. Hence $|f(x_k) - f(x_i)| \geq 1$. Similarly, we can show that $|f(y_l) - f(y_j)| \geq 1$ if $\mathrm{dist}(y_j, y_l) = 2$.

If the degree of y_j and $\max N_G(y_j)$ are d_1 and d_2, respectively, then the chain graph G_j has at most $d_1 + d_2$ vertices. Hence, cl and label of vertices in G_j are calculated in $O(d_1 + d_2) = O(\Delta)$ time, where Δ is the maximum degree of G. Since the number of chain graphs constructed in our algorithm is $O(|V|)$, the total running time of the algorithm is $O(|V| + \Delta|V|) = O(|V| + |E|)$. □

Corollary 4.6. *Any bipartite permutation graph G satisfies $\lambda(G) \leq bc(G) + 1$.*

5 Conclusion

In this paper, we investigated the $L(2,1)$-labeling problem for bipartite permutation graphs. We showed that an optimal $L(2,1)$-labeling of a chain graph, a special class of bipartite permutation graphs, can be computed in linear time. We also present a linear time algorithm for computing $L(2,1)$-labeling of a bipartite permutation graph such that the maximum label is at most $bc(G)+1$. Since $\lambda(G) \geq bc(G)$ for any bipartite graph G, our algorithm computes a nearly optimal solution.

Acknowledgement

This work was partly supported by Japan Society for the Promotion of Science, Grant-in-Aid for Young Scientists (B) (no. 19700001).

References

[1] H.L. Bodlaender, T. Kloks, R.B. Tan, and J. van Leeuwen, "Approximations for λ-coloring of graphs," *The Computer Journal*, vol. 47, no. 2, pp. 193–204, 2004.

[2] A. Brandstädt, V.B. Le, and J.P. Spinrad, *Graph classes: A survey*, ser. SIAM Monographs on Discrete Mathematics and Applications. SIAM, 1999.

[3] T. Calamoneri, "The $L(h,k)$-labelling problem: A survey and annotated bibliography," *The Computer Journal*, vol. 49, no. 5, pp. 585–608, 2006.

[4] M.C. Golumbic, *Algorithmic graph theory and perfect graphs*, 2nd ed. Elsevier B.V., 2004.

[5] J.R. Griggs and R.K. Yeh, "Labelling graphs with a condition at distance 2," *SIAM Journal on Discrete Math.*, vol. 5, no. 4, pp. 586–595, 1992.

[6] C.L. Lu and C.Y. Tang, "Solving the weighted efficient edge domination problem on bipartite permutation graphs," *Discrete Applied Math.*, vol. 87, pp. 203–211, 1998.

[7] J. Spinrad, A. Brandstädt, and L. Stewart, "Bipartite permutation graphs," *Discrete Applied Math.*, vol. 18, no. 3, pp. 279–292, 1987.

[8] G. Steiner, "On the k-path partition of graphs," *Theoretical Computer Science*, vol. 290, pp. 2147–2155, 2003.

[9] R. Uehara and G. Valiente, "Linear structure of bipartite permutation graphs and the longest path problem," *Information Processing Letters*, vol. 103, pp. 71–77, 2007.

[10] G. Xu, L. Kang, and E. Shan, "Acyclic domination on bipartite permutation graphs," *Information Processing Letters*, vol. 99, pp. 139–144, 2006.

[11] R.K. Yeh, "A survey on labeling graphs with a condition at distance two," *Discrete Math.*, vol. 306, pp. 1217–1231, 2006.