On List k-Coloring Convex Bipartite Graphs

Josep Díaz, Öznur Yaşar Diner, Maria Serna, and Oriol Serra

Abstract List k–Coloring (LI k-COL) is the decision problem asking if a given graph admits a proper coloring compatible with a given list assignment to its vertices with colors in $\{1, 2, \ldots, k\}$. The problem is known to be NP-hard even for $k = 3$ within the class of 3–regular planar bipartite graphs and for $k = 4$ within the class of chordal bipartite graphs. In 2015 Huang, Johnson and Paulusma asked for the complexity of LI 3-COL in the class of chordal bipartite graphs. In this paper, we give a partial answer to this question by showing that LI k-COL is polynomial in the class of convex bipartite graphs. We show first that biconvex bipartite graphs admit a multichain ordering, extending the classes of graphs where a polynomial algorithm of Enright, Stewart and Tardos (2014) can be applied to the problem. We provide a dynamic programming algorithm to solve the LI k-COL in the class of convex bipartite graphs. Finally, we show how our algorithm can be modified to solve the more general LI H-COL problem on convex bipartite graphs.

Keywords: List Coloring, Convex Bipartite, Biconvex bipartite graphs.

Josep Díaz
ALBCOM Research Group, CS Department, Universitat Politècnica de Catalunya, e-mail: diaz@cs.upc.edu

Öznur Yaşar Diner
Mathematics Department, Universitat Politècnica de Catalunya and Computer Engineering Department, Kadir Has University, Istanbul, e-mail: oznur.yasar@khas.edu.tr

Maria Serna
ALBCOM Research Group, CS Department, Universitat Politècnica de Catalunya, e-mail: mjserna@cs.upc.edu

Oriol Serra
Mathematics Department, Universitat Politècnica de Catalunya, e-mail: oriol.serra@upc.edu

1 Introduction

A *coloring* of a graph $G = (V,E)$ is a map $c : V \to \mathbb{N}$. A coloring is *proper* if no two adjacent vertices are assigned the same color. If there is a proper coloring of a graph that uses at most k colors, then we say that G is *k-colorable*, and that c is a *k-coloring* for G. The coloring problem COL asks for a given graph $G = (V,E)$, and a positive integer k, whether there is a k-coloring for G or not. When k is fixed, we have the k-COLORING problem.

A list assignment $L : V \to 2^{\mathbb{N}}$ is a map assigning a set of positive integers to each vertex of G. Given G and L, the List Coloring problem LICOL asks for the existence of a proper coloring c that obeys L, i.e., each vertex receives a color from its own list. If the answer is positive, G is said to be *L-colorable*. Variants of the problem are defined by bounding the total number of available colors or by bounding the list size. In LIST k-COLORING (LI k-COL), $L(v) \subseteq \{1,2,\ldots,k\}$ for each $v \in V$. Thus, there are k colors in total. On the other hand, in k-LIST COLORING (k-LICOL) each list L has size at most k. In this case, the total number of colors can be larger than k.

Precoloring Extension, PREXT, is a special case of LICOL and a generalization of COL. In PREXT all of the vertices in a subset W of V are previously colored; and the task is to extend this coloring to all of the vertices. If, in addition, the total number of colors is bounded, say by k, then it is called the k-Precoloring Extension, k-PREXT. k-COL is clearly a special case of k-PREXT, which in turn is a special case of LI k-COL. Refer to [16] for a chart summarizing these relationships.

For general graphs COL and its variants LICOL and PREXT are NP–complete; see [24, 14]. Most of their variants are NP-complete even when the parameter k is fixed for small values of k: k-COL, k-LICOL, LI k-COL and k-PREXT are NP-complete when $k \geq 3$ [29]; and they are polynomially solvable when $k \leq 2$ [13, 39].

Concerning the complexity of these problems in graph classes, COL is solvable in polynomial time for perfect graphs [18] whereas LICOL is NP-complete when restricted to perfect graphs and many of its subclasses, such as split graphs, bipartite graphs [28] and interval graphs [2]. On the other hand, LICOL is polynomially solvable for trees, complete graphs and graphs of bounded treewidth [23]. Refer to Tuza [38], and more recently to Paulusma [32] for related surveys.

For small values of k, Jansen and Scheffler [23] have shown that 3–LICOL is NP-complete when restricted to complete bipartite graphs and cographs, as observed in [15]. Kratochvíl and Tuza [27] showed that 3–LICOL is NP-complete even if each color appears in at most three lists, each vertex in the graph has degree at most three and the graph is planar. 3-PREXT is NP-complete even for 3–regular planar bipartite graphs and for planar bipartite graphs with maximum degree 4 [7].

For fixed $k \geq 3$, LI k-COL is polynomially solvable for P_5-free graphs [20]. Note that chordal bipartite graphs contain P_5-free graphs, but P_6 free graphs are incomparable with chordal bipartite graphs [37]. LI 3-COL is polynomial for P_6-free graphs [6] and for P_7-free graphs [3]. Computational complexity of LI 3-COL for P_8-free bipartite graphs is open [3]. Even the restricted case of LI 3-COL for P_8-free chordal bipartite graphs is open. Golovach et. al. [16] give a survey that summarizes the results for LI k-COL on H-free graphs in terms of the structure of H.

PREXT problem is solvable in linear time on P_5-free graphs; and it is NP-complete when restricted to P_6-free chordal bipartite graphs [22]. 3-PREXT is NP-complete even for planar bipartite graphs [26], even for those having maximum degree 4 [7]. Recall that PREXT generalizes *k*-PrExt and LI *k*-COL generalizes *k*-PREXT. But there is no direct relation between PREXT and LI *k*-COL [16].
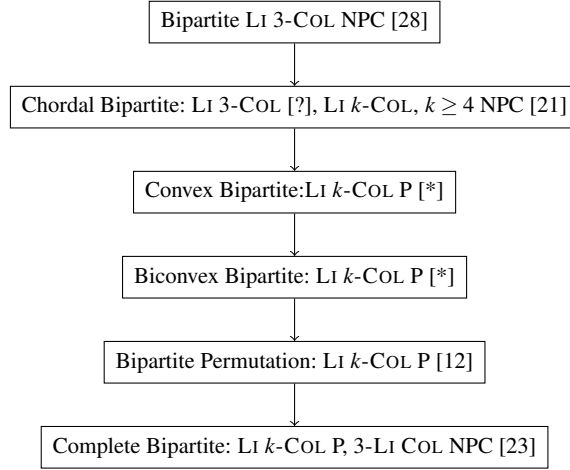
Coloring problems can be placed in the more general class of *H*–coloring problems. Given two graphs *G* and *H*, a function $f : V(G) \to V(H)$ such that $f(u)$ and $f(v)$ are adjacent in *H* whenever *u* and *v* are adjacent in *G* is called a graph homomorphism from *G* to *H*. For a fixed graph *H* and for an input *G*, the *H*-coloring problem, *H*-COL asks whether there is a *G* to *H* homomorphism. In the list *H*-coloring problem, LI *H*-COL, each vertex of the input graph *G* is associated with a list of vertices of *H*, and the question is whether a *G* to *H* homomorphism exists that maps each vertex to a member of its list. Observe that LI *H*-COL is a generalization of LI *k*-COL. The complexities of the *H*–coloring and list *H*–coloring problems for arbitrary input graphs are completely characterized in terms of the structure of *H*, see Nešetřil and Hell [19].

Although intensive research on this subject has been undertaken in the last two decades, there are still numerous open questions regarding computational complexities on LICOL and its variants when they are restricted to certain graph classes. Huang, Johnson and Paulusma [21] proved that LI 4-COL is NP-complete for P_8–free chordal bipartite graphs and 4–PREXT is NP-complete for P_{10}–free chordal bipartite graphs. They further pose the problem on the computational complexity of the LI 3-COL and 3-PREXT on chordal bipartite graphs. Here LI *k*-COL and *k*-PREXT on convex bipartite graphs, a proper subclass of chordal bipartite graphs, are studied, and a partial answer to this question is given. Figure 1 summarizes the related results.

A bipartite graph $G = (X \cup Y, E)$ is convex if it admits an ordering on one of the parts of the bipartition, say *X*, such that the neighbours of each vertex in *Y* are consecutive in this order. If both color classes admit such an ordering the graph is called biconvex bipartite (see Section 2 for formal definitions). Chordal bipartite graphs contain convex bipartite graphs properly. Convex bipartite graphs contain as a proper subclass biconvex bipartite graphs, which contain bipartite permutation graphs properly. More information on these classes can be found in Spinrad [37] and in Brandstädt, Le and Spinrad [4].

Enright, Stewart and Tardos [12] have shown that LI *k*-COL is solvable in polynomial time when restricted to graphs with all connected induced subgraphs having a multichain ordering. They apply this result to permutation graphs and interval graphs. Here, we show that connected biconvex graphs also admit a multichain ordering, implying a polynomial time algorithm for LI *k*-COL on this graph class.

From the point of view of parameterized complexity, treewidth can be computed in polynomial time on chordal bipartite graphs [25]. LI *k*-COL can be solved in polynomial time on chordal bipartite graphs with bounded treewidth [23, 10], which includes chordal bipartite graphs of bounded degree [30]. LI *k*-COL is polynomial for graphs of bounded cliquewidth [8]. Note that convex bipartite graph contains graphs with unbounded treewidth as well as graphs with unbounded cliquewidth.

> Bipartite LI 3-COL NPC [28]

> Chordal Bipartite: LI 3-COL [?], LI k-COL, $k \geq 4$ NPC [21]

> Convex Bipartite: LI k-COL P [*]

> Biconvex Bipartite: LI k-COL P [*]

> Bipartite Permutation: LI k-COL P [12]

> Complete Bipartite: LI k-COL P, 3-LI COL NPC [23]

Fig. 1 Chart for known complexities for LICOL and its variants for chordal bipartite graphs and its subclasses, for $k \geq 3$. The complexity results marked with [*] is the topic of this paper, while [?] stands for open cases. Results without reference are trivial. P stands for Polynomial and NPC for NP-complete.

The paper is organized as follows. In Section 2, we give the necessary definitions. In Section 3, we show that connected biconvex bipartite graphs admit multichain ordering. In Section 4, we show that LI k-COL is polynomially solvable when it is restricted to convex bipartite graphs. Then, we show how to extend this result to LI H-COL. For an extended version of this paper the reader may refer to [9].

2 Preliminaries

We consider finite simple graphs $G = (V, E)$. For terminology refer to Diestel [11]. An edge joining non adjacent vertices in the cycle, C_n, is called a *chord*. A graph G is *chordal* if every induced cycle of length $n \geq 4$ has a chord. *Chordal bipartite graphs* are bipartite graphs in which every induced $C_n, n \geq 6$ has a chord. This graph class is introduced by Golumbic and Gross [17]. Chordal bipartite graphs may contain induced C_4, so they do not constitute a subclass of chordal graphs but it is a proper subclass of bipartite graphs. Chordal bipartite graphs can be recognized in polynomial time [33].

A bipartite graph is represented by $G = (X \cup Y, E)$, where X and Y form a bipartition of the vertex set into stable sets. An ordering of the vertices X in a bipartite graph $G = (X \cup Y, E)$ has the *adjacency property* (or the ordering is said to be *convex*) and G is said to have *convexity with respect to X* if, for each vertex $v \in Y$, $N(v)$ consists of vertices which are consecutive in the ordering of X. We say that an ordering of the vertices X in a bipartite graph $G = (X \cup Y, E)$ has the *enclosure*
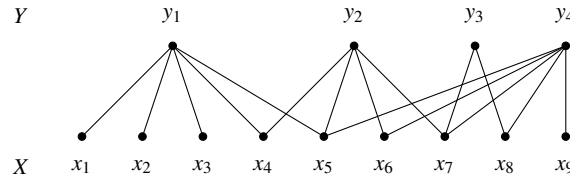
**Fig. 2** A convex bipartite graph which is not biconvex.
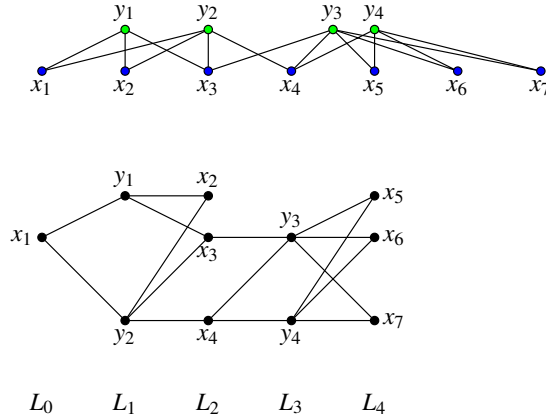
Fig. 3 A convex bipartite graph and its associated distance layers from x_1.

property if for every pair of vertices $u,v \in Y$ such that $N(u) \subseteq N(v)$, the vertices in $N(v) \backslash N(u)$ occur consecutively in the ordering of X.

  Convex bipartite graphs are bipartite graphs $G = (X \cup Y, E)$ that have the adjacency property on one of the partite sets and *biconvex bipartite graphs* have the adjacency property on both partite sets X and Y. Fig. 2 shows a graph that is convex but not biconvex. *Bipartite permutation graphs* are biconvex bipartite graphs in which one of the partite sets obeys both the adjacency and the enclosure properties. There are linear time recognition algorithms for these classes [36, 31].

  A *chain graph* [40] is a bipartite graph that contains no induced $2K_2$ (a graph formed by two independent edges). The following characterization from [12] is equivalent: a connected bipartite graph with bipartite sets X and Y is a chain graph if and only if for any two vertices $y_1, y_2 \in Y$ we have $N(y_1) \subseteq N(y_2)$ or $N(y_2) \subseteq N(y_1)$. If the vertices in X are ordered with respect to their degrees starting from the highest degree, then for any $y \in Y$, the vertices in $N(y)$ will be consecutive in the ordering on X and, if the graph is connected, there is always a vertex $y \in Y$ so that $N(y)$ includes the first vertex in X. In particular, chain graphs are a proper subclass of convex bipartite graphs.
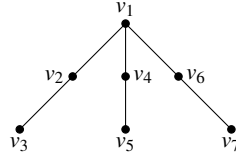
Fig. 4 Subdivision of $K_{1,3}$.

3 List k-Coloring on Biconvex Graphs

Enright, Stewart and Tardos [12] show that LI k-COL, as well as the general LI H-COL, is solvable in polynomial time when restricted to graphs with all connected induced subgraphs having a multichain ordering. They apply this result to permutation graphs and interval graphs. Here, we show that connected biconvex graphs also admit a multichain ordering.

The *distance layers* of a connected graph $G = (V,E)$ from a vertex v_0 are $L_0, L_1, ..., L_z$, where $L_0 = \{v_0\}$ and, for $i > 0$, L_i consists of the vertices at distance i from v_0 and z is the largest integer for which this set is non-empty (see Figure 3 for an example). These layers form a *multi-chain ordering* [5] of G if, for every two consecutive layers L_i and L_{i+1}, the edges connecting these two layers form a chain graph (not necessarily the layers themselves). All connected bipartite permutation graphs [5] and interval graphs [12] admit multichain orderings. Observe that, for the graph given in Fig. 3, the distance layers from x_1 provide a multichain ordering.

Recall that a subdivision of a graph G is the graph $G' = subd(G)$ obtained from G by replacing each edge by a path of length two. Thus $|E(G')| = 2|E(G)|$ and $|V(G')| = |V(G)| + |E(G)|$.

Lemma 1. *If G is a biconvex graph, then G does not contain $subd(K_{1,3})$ as an induced subgraph.*

Proof. Let G be a biconvex graph and let $H = subd(K_{1,3})$. Let v_1 be the vertex of degree 3 in H, v_2, v_4 and v_6 be the vertices in $N(v_1)$ and v_3, v_5 and v_7 the vertices of degree 1 so that v_i is adjacent to v_{i+1} for $i = 2, 4, 6$, see Fig. 4. We observe that there is no ordering of $\{v_1, ..., v_7\}$ in which the three sets $N(v_2) = \{v_1, v_3\}$, $N(v_4) = \{v_1, v_5\}$ and $N(v_6) = \{v_1, v_7\}$ become consecutive. Therefore, a bipartite graph which contains H as an induced subgraph does not admit a biconvex ordering.

Proposition 1. *Every connected biconvex graph admits a multichain ordering.*

Proof. To see that biconvex graphs admit a multichain ordering, we use the notion of biconvex straight ordering introduced by Abbas and Stewart [1]. Let $G = (X, Y, E)$ be a bipartite graph with a linear ordering $\leq$ defined on $X \cup Y$. Two edges $xy, x'y' \in E$, where $x, x' \in X$ and $y, y' \in Y$, are said to *cross* if $x < x'$ and $y > y'$. If xy and $x'y'$ cross, we call (x, y') and (x', y) the *corresponding straight pairs*. An ordering on $X \cup Y$ is

a *straight ordering* if, for each pair $xy, x'y'$ of crossing edges, at least one of the corresponding straight pairs, (x, y') or (x', y), is an edge of the graph [1].

Let $G = (X, Y, E)$ be a connected biconvex graph. It follows from [1, Theorem 11] that G admits a biconvex straight ordering, say $v_0, v_1, \ldots, v_n$ of $X \cup Y$. Let $L_0 = \{v_0\}, L_1, \ldots, L_m$ be the distance layers of G from v_0. Since the graph G is connected, $V = L_0 \cup L_1 \cup \cdots \cup L_m$. Let us show that these layers form a multi–chain ordering.

The first layers L_0 and L_1 trivially form a multi–chain ordering. Let $L_1 = \{v_{i_1}, \cdots, v_{i_\ell}\}$, where the vertices are listed according to the ordering. When $\ell = 1$, L_1, L_2 trivially form a chain graph. When $\ell > 1$, since the ordering is straight, all the edges joining v_{i_1} with vertices in L_2 cross with the edge $v_0 v_{i_2}$. As, v_{i_2} is not connected to v_0, the other straight pair (v_{i_1}, v_{i_2}) should be an edge in G. Therefore, $N(v_{i_1}) \subseteq N(v_{i_2})$. By iterating the same argument, we see that $N(v_{i_1}) \subseteq \cdots \subseteq N(v_{i_\ell})$. Thus, the layers L_0, L_1, L_2 form a multi–chain ordering. By a similar discussion, it can be shown that L_0, L_1, L_2, L_3 form a multichain ordering.

Suppose that $m > 3$. Let $i > 3$ be the largest subscript such that $L_0, L_1, \ldots, L_i$ form a multichain ordering. Suppose for a contradiction that $i < m$. Thus, the bipartite graph induced by the layers L_i, L_{i+1} contain an induced copy of $2K_2$, say with edges $uv, u'v'$, $u, u' \in L_i$ and $v, v' \in L_{i+1}$. As the ordering is straight, we may assume $u < u'$ and $v < v'$. We consider two cases:

Case 1: $N(u) \cap N(u') \cap L_{i-1} \neq \emptyset$. Let $w \in N(u) \cap N(u') \cap L_{i-1}$ and consider predecessors $w' \in L_{i-2}, w'' \in L_{i-3}$ of w. Then the subgraph induced by w, w', w'', u, u', v, v' is isomorphic to a subdivision H of $K_{1,3}$, contradicting Lemma 1.

Case 2: $N(u) \cap N(u') \cap L_{i-1} = \emptyset$. Let $w \in N(u) \cap L_{i-1}$ and $w' \in N(u') \cap L_{i-1}$ be some predecessors of u and u' in the previous layer. Observe that the two edges $wu, w'u'$ induce a $2K_2$ in the subgraph induced by $L_{i-1} \cup L_i$ contradicting the choice of i.

Let us state the main result by Enright, Stewart and Tardos in [12] explicitly:

Theorem 1. *[12] Let H be a fixed graph.* LI H-COL *is polynomial- time solvable for input graphs G satisfying that every connected induced subgraph of G admits a multichain ordering.*

Proposition 1 and Theorem 1 give us the main result in this section.

Theorem 2. *For any H,* LI H-COL *is solvable in polynomial time when restricted to biconvex graphs.*

As LI k-COL is a particular case of LI H-COL and LI k-COL generalizes k-PREXT, we have the following corollary.

Corollary 1. LI k-COL *and k-PREXT are solvable in polynomial time when restricted to biconvex graphs.*

Concerning the running time of the algorithms, it is shown in Abbas and Stewart [1] that a biconvex straight ordering of a biconvex bipartite graph can be found in linear time on the number of vertices of the graph. On the other hand, the algorithm
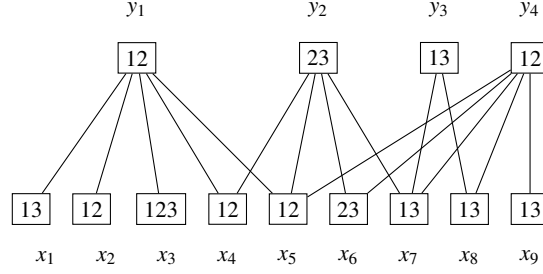
Fig. 5 A list assignment for the convex bipartite graph given in Figure 2. Labels inside vertex indicate the list of colors, from $\{1,2,3\}$, associated to the node.

in [12] is shown to run in time $O(n^{k^2-3k+4})$ time when a multichain ordering in decreasing ordering of degrees is given. Observe that to get such ordering, we only have to reorder the elements in the layers provided by the straight ordering, therefore it can be obtained in linear time. Altogether, it gives an upper bound $O(n^{k^2-3k+4})$ on the complexity of LI k-COL in the class of biconvex graphs.

4 List k-Coloring of Convex Bipartite Graphs

Let $G = (X \cup Y, E)$ be a connected bipartite graph that is convex with respect to X. Let $X = \{x_1, \ldots, x_n\}$ be a convex ordering of X, that is, for each $y \in Y$ there are two positive integers $a_y \leq b_y$ such that $N(y) = \{x_i \mid a_y \leq i \leq b_y\}$.

Consider the set of integers $A = \{a_y \mid y \in Y\}$ and $B = \{b_y \mid y \in Y\}$. For the graph given in Fig. 5, $A = \{1, 4, 5, 7\}$ and $B = \{5, 7, 8, 9\}$.

We use the set B to direct the dynamic programming algorithm and the elements in A to determine the relevant information to be kept for the next step. Assume that $B = \{b_1, \ldots, b_\beta\}$ are sorted so that $b_1 < b_2 < \cdots < b_\beta$. By connectivity of G, we have $b_\beta = n$. For each $1 \leq j \leq \beta$, let $X_j = \{x_i \in X \mid i \leq b_j\}$, $Y_j = \{y \in Y \mid b_y \leq b_j\}$, and $Z_j = \{y \in Y \mid a_y \leq b_j < b_y\}$. Define $G_j = G[X_j \cup Y_j]$. Observe that $G_\beta = G$, $Z_\beta = \emptyset$ and that Z_j contains those vertices in Y whose neighborhood starts before or at b_j and ends after b_j. For example, for the graph given in Fig. 5, $b_2 = 7, X_2 = \{x_1, x_2, \ldots, x_7\}, Y_2 = \{y_1, y_2\}$ and $Z_2 = \{y_3, y_4\}$. For sake of simplicity we assume an initial point $b_0 = 0$, so that G_0 is the empty graph.

Let K be a set of k colors. Assume that each vertex u in G has an associated list $L(u) \subseteq K$. We next define the information that we want to compute for each $1 \leq b_j \leq b_\beta$. For each $1 \leq j \leq \beta$, define $A(j) = \{a_y \mid y \in Z_j\} \cup \{b_j\}$. As before, we assume that the elements in $A(j) = \{a_{1,j}, \ldots, a_{\alpha_j,j}\}$ are increasingly ordered, $a_{1,j} < a_{2,j} < \cdots < a_{\alpha_j,j} = b_j$. To simplify notation set $a_{\alpha_j+1,j} = b_j + 1$ to make sure that a higher value always exists. For the example in Fig. 5, $A(2) = \{5, 6, 7\}$. For the fictitious initial b value $j = 0$, we take $A(0) = \{0\}$.

Fix j, $1 \leq j \leq \beta$. For each $1 \leq i \leq \alpha_j$ and $S \subsetneq K$, $T_j(i,S)$ will hold value true whenever there is a valid list coloring of G_j such that it uses no color in S for the set $X_i^j = \{x_\ell \mid a_{i,j} \leq \ell < a_{i+1,j}\}$. Observe that we are not considering K as a potential set as not using any color is impossible.

The Color Algorithm will compute those values in three steps. In going from $j-1$ to j, first it computes the values for the $x \in X_j$ that were not in X_{j-1} combining this information with the relevant information computed in the previous step. Next, it incorporates the restriction from $y \in Y_j$ that were not in Y_{j-1}. Finally, it rearranges the information to keep only the values for the index in $A(j)$.

Color Algorithm: Let j, $1 \leq j \leq \beta$. Initially set $A(0) = \{0\}$, $b_0 = 0$ and set $T_0(0,S)$ to TRUE for any S. When $j \geq 1$ assume that the values of T_{j-1} have already been computed.

Step 1   Extending to new parts.

Let $A'(j) = A(j-1) \cup \{a_y \mid b_{j-1} < a_y \leq b_j\} \cup \{b_j\}$. For $j > 1$, by construction, those values lie before b_j and some of them have no corresponding entries in T_{j-1}. Assume that $A'(j) = \{a'_1, \ldots, a'_{\gamma_j}\}$ increasingly ordered. Let $a'_{\gamma_j+1} = b_j + 1$. We set $T_{j-1}(\ell, S)$ for $\alpha_{j-1} < \ell \leq \gamma_j$ and $S \subsetneq K$ to be true whenever there is a valid list coloring of the set $X'(\ell) = \{x_i \mid a'_\ell \leq i < a'_{\ell+1}\}$. For this, the algorithm checks whether $L(x) \setminus S \neq \emptyset$ for each $x \in X'(\ell)$. If this is the case, one can select a color not in S and get a valid coloring. Accordingly we update the value of $T_{j-1}(\alpha_{j-1}, S)$ so that it remains TRUE if it was already set to TRUE and the previous condition holds for the elements in $X'(\alpha_{j-1})$

Step 2   Incorporating Y_j.

For $y \in Y_j$ and $a_i \in [a_y, b_y]$, consider any entry $T_{j-1}(i,S)$ set to TRUE. If $S \cap L(y) = \emptyset$, the corresponding entry is changed to FALSE.

Next, the values on T_{j-1} are processed in increasing order of x_i: any entry (i,S) holding value TRUE will remain TRUE whenever there is an entry $(i-1, S')$ holding value TRUE with $S \subseteq S'$. By monotonicity, the property holds whenever $T_{j-1}(i-1,S)$ is TRUE.

After processing y, if $T_{j-1}(l,S)$ holds true, for each piece $[a'_l, a'_{l+1})$ between a_y and b_y, we can pick a common color not in S but in $L(y)$ to color y that is compatible with some list coloring on the X relevant parts that do not use S.

Step 3   Compacting to get T_j.

For each $1 \leq i \leq \alpha_j$ the set X_i^j might contain several subintervals on $X'(j)$, considered either in T_{j-1} that will not be needed later on. We fusion those sets from left to right, adding one at a time, setting $T_j(i,S)$ to true whenever there are corresponding entries holding value true for sets S_1 and S_2 so that $S \subseteq S_1 \cap S_2$.

Lemma 2. *Let $G = (X \cup Y, E)$ be a connected convex bipartite graph, L be a color assignment for G. There is an L–coloring of G if and only if there is $S \subseteq K$ such that at the end of the execution of the Color Algorithm $T_\beta(\alpha_\beta, S) = true$.*

Proof.   Assume that G admits a list coloring. Let c be an L-coloring of G. For $U \subseteq X$ let $S_U = K \setminus c(X)$. Observe that L does not use any color in S_U on U and furthermore, for any $y \in Y$ so that $N(y) \cap U \neq \emptyset$, $L(y) \cap S_U \neq \emptyset$. Using this fact, it follows that the

entries in the tables for the corresponding sets get the value true and at the end of the algorithm $T(\beta, \{c(x_n)\})$ will be true.

Conversely, we can prove that the Color Algorithm correctly computes the values of T_j for $1 \leq j \leq \beta$. The proof is by induction. Observe that for $j = 1$ the table R provides the right indices and the initialization step provides the correct values for the table on an empty graph. By induction hypothesis, we assume that the values of T_j are correctly computed. Step 1 guarantees that the desired coloring exists when adding only the X part on G_j to G_{j-1}. Step 2 has two parts. The first one guarantees that only those entries with sets that are compatible with the list of the vertices in Y_j are still alive. The second one ensures that when combining two consecutive pieces having a common neighborhood on Y_j a common set of colors (a subset) is available to color these vertices. Finally, Step 3 merges tables for pieces that have the same Y neighborhood outside G_j. Again, we need to maintain a common set of colors free for potential use on these neighbors.

Finally, observe that all the running time of the color algorithm is polynomial in $|G|$ and in 2^k. Furthermore, the k-PREXT can be polynomially reduced to LI k-COL. Therefore, we get our main result.

Theorem 3. *For $k \geq 3$, LI k-COL and k-PREXT on convex bipartite graphs can be solved in polynomial time.*

The color algorithm can be modified to solve the LI H-COL on convex bipartite graphs. For this, the algorithm keeps track instead of the unused color on the X part of the used ones. For doing that, we have to consider some longer subdivision of the intervals in the X part. Step 2 will check that at least one of the colors in the list of y is connected to all the used colors in the X part. Step 3 is also modified as the global set of used colors will be the union.

Theorem 4. *For any H, LI H-COL on convex bipartite graphs can be solved in polynomial time.*

5 Conclusions

In this paper, the problem posed by Huang et al. [21] on the computational complexity of the LI 3-COL and 3-PREXT on chordal bipartite graphs is addressed. A partial answer to a general version of this question is given by increasing the subclasses of chordal bipartite graphs for which polynomial time algorithms for the LI k-COL are known to biconvex bipartite graphs and convex bipartite graphs. Note that the later class includes convex bipartite graphs with bounded degree, complete bipartite graphs which have unbounded treewidth, as well as graphs with unbounded cliquewidth. Interestingly enough, the second result can also be extended, with a slight modification, to solve LI H-COL for the same graph class. The paper includes another result of independent interest: any connected biconvex bipartite graph admit a multichain ordering.

On the other hand, chordal bipartite graphs form a much larger graph class. Using the terminology of [35], it is a superfactorial graph class whereas convex bipartite graphs is a factorial graph class. Although L₁ k-COL is hard for $k \geq 4$ when restricted to chordal bipartite graphs, finding the computational complexity of L₁ 3-COL for chordal bipartite graphs is the next natural open question.

Acknowledgments

References

1. Abbas, N., Stewart, L. K.: Biconvex graphs: Ordering and algorithms. Discrete Applied Mathematics. 103(1–3), 1–19 (2000)
2. Biro, M., Hujter, M., Tuza, Zs.: Precoloring extension. I. Interval graphs. Discrete Mathematics. 100(1-3), 267–279 (1992)
3. Bonomo, F., Chudnovsky, M., Maceli, P., Schaudt, O., Stein, M., and Zhong, M.: Three-Coloring and List Three-Coloring of Graphs Without Induced Paths on Seven Vertices. Combinatorica. 38, 779–801 (2018)
4. Brandstädt, A., Le, V. B., Spinrad, J. P.: Graph classes: a survey, Society for Industrial and Applied Mathematics. (1999)
5. Brandstädt, A. and Lozin, V. V.: On the linear structure and clique-width of bipartite permutation graphs. Ars Combinatoria. 67, 273–281 (2003)
6. Broersma, H. J., Fomin, F. V., Golovach, P.A., and Paulusma, D.: Three complexity results on coloring P_k-freegraphs. European Journal of Combinatorics. 34, 609-619 (2013)
7. Chlebik, M., Chlebikova, J.: Hard coloring problems in low degree planar bipartite graphs. Discrete Applied Mathematics. 154, 1960–1965 (2006)
8. Courcelle, B., Makowsky, J. A., Rotics, U.: Linear time solvable optimization problems on graphs of bounded clique-width. Theory of Computing Systems. 33 (2), 125–150 (2000)
9. Díaz, J., Diner, Ö. Y., Serna, M., and Serra, O.: On List k-Coloring Convex Bipartite Graphs. arXiv:2002.02729 [cs.CC] (2020)
10. Díaz, J., Serna, M., and Thilikos, D. M.: Counting H-colorings of partial k-trees. Theoret. Comput. Sci. 281, 291-309 (2002)
11. Diestel, R. Graph Theory. Graduate Texts in Mathematics 173(5) Heidelberg, Springer-Verlag (2017)
12. Enright, J. Stewart, T., Tardos, G.: On List Coloring and List Homomorphism of Permutation and Interval Graphs. SIAM Journal on Discrete Mathematics. 28(4), 1675-1685 (2014)
13. Erdös, P., Rubin, A. L., Taylor, H.: Choosability in graphs. Proceedings of the West Coast Conference on Combinatorics, Graph Theory and Computing, Humboldt State University, Utilitas Mathematica. 125-157 (1979)
14. Garey, M. R., Johnson, D. S. Computers and Intractability: A Guide to the Theory of NP-Completeness. New York, W. H. Freeman (1979)

15. Golovach, P. A., Paulusma, D.: List coloring in the absence of two subgraphs. Discrete Applied Mathematics. 166, 123–130 (2014)
16. Golovach, P. A., Johnson, M., Paulusma, D. and Song, J.: A Survey on the Computational Complexity of Colouring Graphs with Forbidden Subgraphs. Journal of Graph Theory. 84(4), 331–363 (2016)
17. Golumbic, M. C., Goss, C. F.: Perfect elimination and chordal bipartite graphs. Journal of Graph Theory. 2, 155–163 (1978)
18. Gröstchel, M., Lovasz, L., Schrijver, A.: Polynomial algorithms for perfect graphs. Ann. Discret. Math. 21, 325–356 (1984)
19. Hell, P. and Nešetřil, J.: Graphs and Homomorphisms. Oxford University Press, Oxford (2004)
20. Hoàng, C., Kamiński, M., Lozin, V., Sawada, J., and Shu, X.: Deciding k-colorability of P_5-free graphs in polynomial time. Algorithmica. 57, 74-81 (2010)
21. Huang, S., Johnson H., Paulusma, D.: Narrowing the complexity gap for coloring (C_s, P_t)–Free Graphs. The Computer Journal. 58(11), 3074-3088 (2015)
22. Hujter, M., Tuza, Zs.: Precoloring extension. III. Classes of perfect graphs. Combinatorics, Probability and Computing. 5, 35–56 (1996)
23. Jansen, K., Scheffler, P.: Generalized coloring for tree-like graphs. Discrete Appl Math. 75, 135–155 (1997)
24. Karp, R. M.: Reducibility among combinatorial problems. In: Compexity of Computer Computations, Miller, J. D. B. R. E., and Thatcher, J. W. Editors, Plenum Press, New York, 85-103 (1972)
25. Kloks, T. and Kratsch, D.: Treewidth of chordal bipartite graphs. Journal of Algorithms. 19, 266–281 (1995)
26. Kratochvil, J.: Precoloring extension with fixed color bound. Acta Mathematica Universitatis Comenianae. 62, 139–153 (1993)
27. Kratochvil, J., Tsuza, Z.: Algorithmic complexity of list colorings, Discrete Applied Mathematics. 50, 297–302 (1994)
28. Kubale, M.: Some results concerning the complexity of restricted colorings of graphs. Discrete Applied Mathematics. 36, 35–46 (1992)
29. Lovasz, L.: Coverings and colorings of hypergraphs. Proceedings of the 4th Southeastern Conference on Combinatorics, Graph Theory and Computing, Utilitas Mathematica. 3–12 (1973)
30. Lozin, V., Rautenbach, D.: Chordal bipartite graphs of bounded tree- and clique-width. Discrete Mathematics. 283, 151–158 (2004)
31. Nussbaum, D., Pu, S., Sack, J.-R., Uno, T., Zarrabi-Zadeh, H.: Finding Maximum Edge Bicliques in Convex Bipartite Graphs. Algorithmica. 64(2), 140–149 (2010)
32. Paulusma, D. 2016. Open Problems on Graph Coloring for Special Graph Classes. Lecture Notes in Computer Science 9224, 16–30 (2016)
33. Paige, R. and Tarjan, R. E.: Three partition refinement algorithms. SIAM J. Comput. 16, 973-989 (1987)
34. Schaefer, T. J.: The complexity of satisfiability problems. Proceedings of the 10th Annual ACM Symposium on Theory of Computing. 216–226 (1978)
35. Scheinerman, E. R. and Zito, J.: On the size of hereditary classes of graphs. J. Combin. Theory Ser. B. 61, 16–39 (1994)
36. Spinrad, J. P., Brandstädt, A. and Stewart, L.: Bipartite Permutation Graphs. Discrete Applied Mathematics. 18, 279–292 (1987)
37. Spinrad, J.: Efficient Graph Representations. Fields Institute Monographs 19, American Mathematical Society (2003)
38. Tuza, Zs.: Graph Coloring with Local constrains - A survey. Discussioners Mathematicae, Graph Theory. 17, 161–228 (1997)
39. Vizing, V. G.: Coloring the vertices of a graph in prescribed colors. (in Russian) Metody Diskret. Anal. v Teorii Kodov i Schem 29, 3–10 (1976)
40. Yannakakis, M.: The complexity of the partial order dimension problem. SIAM Journal on Algebraic and Discrete Methods. 3, 351–358 (1982)