List Homomorphism: Beyond the Known Boundaries

Sriram Bhyravarapu[1], Satyabrata Jana[1], Fahad Panolan[3], Saket Saurabh[1,2], and Shaily Verma[1]

[1] The Institute of Mathematical Sciences, HBNI, Chennai, India
[2] University of Bergen, Norway
[3] Indian Institute of Technology Hyderabad, Hyderabad, India
`{sriramb, saket, shailyverma}@imsc.res.in`
`satyamtma@gmail.com, fahad@cse.iith.ac.in`

Abstract. Given two graphs G and H, and a list $L(u) \subseteq V(H)$ associated with each $u \in V(G)$, a list homomorphism from G to H is a mapping $f : V(G) \rightarrow V(H)$ such that (i) for all $u \in V(G)$, $f(u) \in L(u)$, and (ii) for all $u, v \in V(G)$, if $uv \in E(G)$ then $f(u)f(v) \in E(H)$. The List Homomorphism problem asks whether there exists a list homomorphism from G to H. Enright, Stewart and Tardos [SIAM J. Discret. Math., 2014] showed that the List Homomorphism problem can be solved in $O(n^{k^2-3k+4})$ time on graphs where every connected induced subgraph of G admits "a multichain ordering" (see the introduction for the definition of multichain ordering of a graph), that includes permutation graphs, biconvex graphs, and interval graphs, where $n = |V(G)|$ and $k = |V(H)|$. We prove that List Homomorphism parameterized by k even when G is a bipartite permutation graph is $\mathsf{W[1]}$-hard. In fact, our reduction implies that it is not solvable in time $n^{o(k)}$, unless the Exponential Time Hypothesis (ETH) fails. We complement this result with a matching upper bound and another positive result.

1. There is a $O(n^{8k+3})$ time algorithm for List Homomorphism on bipartite graphs that admit a multichain ordering that includes the class of bipartite permutation graphs and biconvex graphs.
2. For bipartite graph G that admits a multichain ordering, List Homomorphism is fixed parameter tractable when parameterized by k and the number of layers in the multichain ordering of G.

In addition, we study a variant of List Homomorphism called List Locally Surjective Homomorphism. We prove that List Locally Surjective Homomorphism parameterized by the number of vertices in H is $\mathsf{W[1]}$-hard, even when G is a chordal graph and H is a split graph.

Keywords: List Homomorphism · FPT · W[1]-hardness· bipartite permutation graphs · chordal graphs.

1 Introduction

Given a graph G, a *proper coloring* is an assignment of colors to the vertices of G such that adjacent vertices are assigned different colors. Given a graph G and an integer k, the k-Coloring problem asks if there exists a proper coloring of G using k colors. The k-Coloring problem is known to be NP-complete even when $k = 3$ [17]. It is a very well-studied problem due to its practical applications. Many variants of coloring have been studied. In 1970's Vizing [27] and Erdös et al. [11] independently, introduced List k-Coloring which is a generalization of k-Coloring. Given a graph G and a list of admissible colors $L(v) \subseteq [k]$ for each vertex v in $V(G)$, the List k-Coloring problem asks whether there exists a proper coloring of G where each vertex is assigned a color from its list. Here, $[k] = \{1, 2, \ldots, k\}$. List k-Coloring has found practical applications in wireless networks, for example in frequency assignment problem [18,28].

Given two graphs G and H, a *graph homomorphism* from G to H is a mapping $f : V(G) \to V(H)$ such that if $uv \in E(G)$, then $f(u)f(v) \in E(H)$. Given two graphs G and H, and a list $L(v) \subseteq V(H)$ for each $v \in V(G)$, a *list homomorphism* from G to H is a graph homomorphism f from G to H such that $f(v) \in L(v)$ for each vertex v in $V(G)$. Given an instance (G, H, L), the List Homomorphism problem (LHom for short) asks whether there exists a list homomorphism from G to H. Observe that List k-Coloring is a special case of List Homomorphism where H is a simple complete graph on k vertices.

List k-Coloring is NP-complete for $k \geq 3$ as it is an extension of k-coloring problem. The problem remains NP-complete even for planar bipartite graphs [22]. On the positive side, for a fixed k, the problem is known to be polynomial time solvable on co-graphs [20], P_5-free graphs [19] and partial t-trees [20]. Considering the List Homomorphism problem, given a fixed integer $k = |V(H)|$, polynomial time algorithms are available for graphs of bounded tree-width [8], interval graphs, permutation graphs [10] and convex bipartite graphs [7]. Recently List Homomorphism on graphs with bounded tree-width has been studied in [23]. The list homomorphism has also been studied as *list H-coloring* in the literature and is a well studied problem [9,5,4,24]. Feder et al. [12,13,14] gave classifications of the complexity of LHom based on the restrictions on graph H. Recently, LHom has been studied for signed graphs [1,21,2].

Enright, Stewart and Tardos [10] showed that the List Homomorphism problem can be solved in $O(n^{k^2-3k+4})$ time on bipartite permutation graphs, interval graphs and biconvex graphs, where $n = |V(G)|$ and $k = |V(H)|$. It is natural to ask whether the running time can be improved or can we obtain a FPT algorithm when parameterized by k. Towards that we prove the following results.

Theorem 1. LHom *can be solved in time $O(n^{4k+3})$ on bipartite permutation graphs.*

Theorem 2. LHom *can be solved in $O(n^{8k+3})$ time on biconvex graphs.*

Theorem 3. List k-Coloring *parameterized by k is W[1]-hard on bipartite permutation graphs. Furthermore, there is no $f(k)n^{o(k)}$-time algorithm for* List k-Coloring*, for any computable function f unless* ETH fails.

Since List k-Coloring is a particular case of LHom, similar hardness results hold for LHom. However, we design fixed-parameter tractable (FPT) algorithms when parameterized by $|V(H)|$ and the diameter of the input graph G, where diameter of a graph is the maximum distance between any pair of vertices.

Theorem 4. LHom *is FPT on bipartite permutation graphs and biconvex graphs graphs, when parameterized by* $|V(H)|$ *and the diameter of the input graph* G.

We also a study a variant of LHom called List Locally Surjective Homomorphism. Given two graphs G and H, and a list $L(v) \subseteq V(H)$ for each $v \in V(G)$, a *list locally surjective homomorphism* from G to H is a list homomorphism $f : V(G) \to V(H)$ that is surjective in the neighborhood of each vertex in G. In other words, if $f(v) = v'$, then for every vertex $u' \in N_H(v')$, there is a vertex $u \in N_G(v)$, such that $f(u) = u'$. That is, for each connected component C of H if one vertex in C is "used" by the homomorphism, then all the vertices are used. Given as an input (G, H, L), the List Locally Surjective Homomorphism problem (LLSHom for short) asks whether there exists a list locally surjective homomorphism from G to H. We prove the following result about LLSHom.

Theorem 5 ($\star$[4])**.** *Given an instance* (G, H, L) *such that* G *is a chordal graph, and* H *is a split graph, it is* W[1]*-hard to decide whether there is a list locally surjective homomorphism from* G *to* H*, when parameterized by* $|H|$.

Other Related Works. In 1999, Feder et al. [15] studied List M-Partition problem. The input to the problem is a graph $G = (V, E)$ and a $m \times m$ matrix M with entries $M(i, j) \in \{0, 1, *\}$. The goal is to check whether there exists a partition of $V(G)$ into m parts (called M-partition) such that for distinct vertices x and y of G placed in parts i and j respectively, we have that (i) if $M(i, j) = 0$, then $xy \notin E(G)$, (ii) if $M(i, j) = 1$, then $xy \in E(G)$, and (iii) if $M(i, j) = *$, then xy may or may not be an edge of G. By considering H as a graph on m vertices and M as a matrix obtained from the adjacency matrix of H by replacing each 1 with $*$, each homomorphism corresponds to a M-partition of G. Thus List M-Partition generalizes List k-Coloring and LHom.

Valadkhan [25,26] gave polynomial time algorithms for List M-Partition for various graph classes. They gave $O(m^2 n^{4m+2})$ time algorithms for interval and permutation graphs, $O(m^2 n^{8m+2})$ time algorithms for interval bigraphs, interval containment bigraphs, and circular-arc graphs, $O(m^2 n^{4mt+2})$ time algorithm for comparability graphs with bounded clique-covering number t. The algorithm on interval graphs is an improvement over the algorithm by Enright, Stewart and Tardos [10]. Feder et al. [16] showed that List M-Partition can be solved in $O(t^{t+1} \cdot n)$ time on graphs of treewidth at most t.

Our Methods. In this paper, we study LHom on sub-classes of bipartite graphs by exploiting their structural properties. In particular, the sub-classes of bipartite graphs studied in this paper admit a "multichain" ordering (see Definition 3 in Preliminaries). Some of the graph classes that admit a multichain ordering

[4] Due to paucity of space the proofs of results marked with $\star$ are omitted here.

include interval graphs, permutation graphs, bipartite permutation graphs, biconvex graphs, etc [10]. Towards proving Theorems 1 and 2, we prove that there is a list homomorphism such that if we know the labels of $O(k)$ vertices in a layer, in polynomial time we can extend that to a list homomorphism.

In Section 3, we present a $O(n^{8k+3})$ time algorithm for LHom on bipartite graphs that admit a multichain ordering (Theorem 6). It is known that biconvex graphs and bipartite permutation graphs admit a multichain ordering. Hence Theorem 2 follows from Theorem 6. Since there are additional properties for bipartite permutation graphs, we provide an improved algorithm to bipartite permutation graphs that runs in $O(n^{4k+3})$ time (Theorem 1). These are improvements over the results from [10].

In Section 4, we show that List k-Coloring is W[1]-hard on bipartite permutation graphs (Theorem 3). We prove this result by giving a parameter preserving reduction from the Multi-colored Independent Set problem.

2 Preliminaries

Let $f : D \to R$ be a function from a set D to a set R. For a subset $A \subseteq D$, we use $f|_A : A \to R$ to denote the restriction of f to A. We will also use the words labelings and mappings for functions. A *partial labeling* on a set D is a function on a strict subset of D.

Let $G = (V, E)$ be a graph. We also use $V(G)$ and $E(G)$ to denote the vertex set and the edge set of the graph G, respectively. For a vertex $v \in V(G)$, the number of vertices adjacent with v is called the *degree* of v in G and it is denoted by $deg_G(v)$ (or simply $deg(v)$ if the graph G is clear from the context). The set of all the vertices adjacent with v is called as the neighborhood of v and it is denoted by $N_G(v)$ (or simply $N(v)$). The *distance* between two vertices $u, v \in V(G)$ is the length of a shortest path between u and v in G. Let X and Y be two disjoint subsets of $V(G)$, then $E(X, Y)$ denotes the set of edges with one endpoint in X and the other is in Y. A graph G is called a *split graph* if the vertices of G can be partitioned into two sets C and I such that $G[C]$ is a clique and $G[I]$ is an independent set. A graph is a *permutation graph* if there is some pair P_1, P_2 of permutations of the vertex set such that there is an edge between vertices x and y if and only if x precedes y in one of $\{P_1, P_2\}$, while y precedes x in the other. A graph is a *bipartite permutation graph* if it is both bipartite and a permutation graph.

Let (G, H, L) be an instance for List Homomorphism, where $V(H) = \{1, 2, \ldots, k\}$. First notice that if G is not connected, then (G, H, L) is a yes-instance if and only if for all connected components C of G, $(C, H, L|_{V(C)})$ is a yes-instance. Thus, throughout the paper, we assume that for an instance (G, H, L) of List Homomorphism, G is connected.

Definition 1 (Chain Graph [10]). *A bipartite graph $G = (A \uplus B, E)$ is a* chain graph *if and only if for any two vertices $u, v \in A$, either $N(u) \subseteq N(v)$ or $N(v) \subseteq N(u)$. It follows that, for any two vertices $u, v \in B$, either $N(u) \subseteq N(v)$ or $N(v) \subseteq N(u)$.*

Definition 2. *For a graph G and a vertex subset U, we say that an ordering σ of U is* increasing *in G, if for any $x <_\sigma y$, $N_G(x) \subseteq N_G(y)$. We say that an ordering σ' of U is* decreasing *in G, if for any $x <_{\sigma'} y$, $N_G(y) \subseteq N_G(x)$.*

For a chain graph $G = (A \uplus B, E)$, there is an ordering σ of A which is increasing in G and there is an ordering σ' of B which is decreasing in G. For a vertex $u \in A$, a vertex $v \in N(u)$ is called a *private neighbor* of u if for any vertex w such that $w <_\sigma u$, v is not a neighbor of w. In fact, for a chain graph $G = (A \uplus B, E)$, any ordering of A that is non-decreasing in its degrees increases in G. Also, any ordering of B that is non-increasing in its degrees decreases in G.

Definition 3 (Multichain ordering). *For a connected graph G, the* distance layers *of G from a vertex v_0 is a sequence $L_0, L_1, \dots, L_r$ where $L_0 = \{v_0\}$, L_i is the set of vertices that are at distance i from v_0 for each $i \in [r]$, and r is the largest integer such that $L_r \neq \emptyset$. These layers form a* multichain ordering *of G if for every two consecutive layers L_i and L_{i+1}, the edges connecting these two layers form a chain graph. That is, the graph $(L_i \cup L_{i+1}, E(L_i, L_{i+1}))$ is a chain graph. We say that G admits a multichain ordering if there is a vertex v_0 such that the distance layers of G from v_0 forms a multichain ordering.*

It is known that all connected permutation graphs and connected interval graphs have multichain orderings [10]. Let G be a graph and let $L_0, L_1, \dots, L_r$ be a multichain ordering of G. Then, for any $i \in [r]$, let G_i be the bipartite graph with vertex set $L_{i-1} \cup L_i$ and edge set $E(L_{i-1}, L_i)$. Then, we know that for each $i \in [r]$, G_i is a chain graph. Thus, for each $i \in [r-1]$, there are two orderings $\sigma_{i,1}$ and $\sigma_{i,2}$ of L_i such that $\sigma_{i,1}$ is decreasing in G_i and $\sigma_{i,2}$ is increasing in G_{i+1}. The following result implies that for connected bipartite permutation graphs there is a multichain ordering where for each layer L_i, $\sigma_{i,1}$ is same as $\sigma_{i,2}$.

Proposition 1 ([3]). *A connected graph $G = (V, E)$ is a bipartite permutation graph if and only if the vertex set $V(G)$ can be partitioned into independent sets $V_0, V_1, \dots, V_q$ such that the following holds.*

1. *Any two vertices in non-consecutive sets are non-adjacent.*
2. *Any two consecutive sets V_{i-1} and V_i, induce a chain graph denoted by G_i.*
3. *For each $j \in \{0, \dots, q\}$, there is an ordering σ_j of V_j with the following properties. For each $i \in \{1, 2, \dots, q-1\}$, σ_i is decreasing in G_i and increasing in G_{i+1}. Moreover, σ_0 is increasing in G_1 and σ_q is decreasing in G_q.*
4. *$|V_0| = 1$ and $V_0, V_1, \dots, V_q$ is the distance layers of G from the vertex in V_0.*

Observation 1 *Let G be a connected bipartite graph that admits a multichain ordering $V_0, \dots, V_q$. Then for each $i \in \{0, 1, \dots, q\}$, V_i is an independent set.*

3 XP algorithms: Proofs of Theorems 1 and 2

In this section, we give an $O(n^{8k+3})$ time algorithm for List Homomorphism on bipartite graphs that admit a multichain ordering. We first discuss an algorithm for LHom on bipartite permutation graphs that runs in $O(n^{4k+3})$ time (Theorem 1). We then extend this algorithm to bipartite graphs admitting a multichain ordering that includes biconvex graphs. Thereby, settling Theorem 2.

We first prove the following lemma, which is crucial to our algorithm.

Lemma 1. *Let (G, H, L) be an instance of* List Homomorphism*, where G is a connected bipartite permutation graph. Let $V_0, \ldots, V_q$ be a sequence of independent sets such that the properties mentioned in Proposition 1 hold. For each $i \in \{0, \ldots, q\}$, σ_i is the ordering of V_i and for each $j \in [q]$, G_j is the graph $G[V_{j-1} \cup V_j]$ mentioned in Proposition 1. If there exists a list homomorphism from G to H, then there exists a list homomorphism f from G to H such that for any $i \in \{0, 1, \ldots, q\}$, and any $w \in V_i$, at least one of the following is true.*

1. *w is the first vertex or the last vertex in σ_i that is assigned the label $f(w)$.*
2. *$f(w)$ is the least integer in $L(w)$ such that there exist $x, y \in V_i$ with $x <_{\sigma_i} w <_{\sigma_i} y$ and $f(x) = f(y) = f(w)$.*

Proof. Let f be a list homomorphism such that maximum number of vertices satisfy the stated properties (1) or (2). If all the vertices satisfy the stated properties, then f is our desired list homomorphism. Otherwise, let w be a vertex such that it does not satisfy (1) and (2). Let $w \in V_i$ for some $i \in \{0, 1, \ldots, q\}$. Since w does not satisfy (1), we know that there exist $v, x \in V_i$ such that $v <_{\sigma_i} w <_{\sigma_i} x$ and $f(v) = f(w) = f(x)$. Since w does not satisfy (2), there exists an integer $c \in L(w)$ and two vertices $x', y' \in V_i$ such that $c < f(w)$, $x' \leq_{\sigma_i} w \leq_{\sigma_i} y'$ and $f(x') = f(y') = c$. Without loss of generality, let c be the least integer with the above property. Now consider the following function $f' \colon V(G) \to V(H)$. For each $z \neq w$, $f'(z) = f(z)$ and $f'(w) = c$.

Now we claim that f' is a list homomorphism from G to H and the number of vertices in G that satisfies (1) or (2) with respect to f' is strictly more than the number of vertices in G that satisfies (1) or (2) with respect to f, which leads to a contradiction.

Claim. f' is a list homomorphism from G to H.

Proof. Since f is a list homomorphism and $c = f'(w) \in L(w)$, we have that for any vertex $u \in V(G)$, $f'(u) \in L(u)$. Recall that, $N(w) \cap V_i = \emptyset$ and all the neighbors of w are in $V_{i-1} \cup V_{i+1}$. Since $x' \leq_{\sigma_i} w \leq_{\sigma_i} y'$, we have $N(w) \subseteq N(x')$ in G_i and $N(w) \subseteq N(y')$ in G_{i+1}. Thus, any neighbor w' of w is adjacent to either x' or y'. This implies that $f(w')f'(w)$ is an edge in H. For any edge $zz' \in E(G)$ with $w \notin \{z, z'\}$, $f'(z)f'(z') = f(z)f(z')$ and hence $f'(z)f'(z')$ is an edge in H. Thus, we have proved that f' is a list homomorphism. □

Claim. The number of vertices in G that satisfies (1) or (2) with respect to f' is strictly more than the number of vertices in G that satisfies (1) or (2) with respect to f.

Proof. Notice that w does not satisfy (1) and (2) with respect to f, but it satisfies (2) with respect to f'.

Now we want to prove that for other vertices if they were satisfying (1) or (2) in f, then they so do in f'. Let x be the first vertex and y be the last vertex in σ_i such that $f'(x) = f'(y) = f'(w)$. Since w satisfies (2) with respect to f', we have that $x <_{\sigma_i} w <_{\sigma_i} y$. Let x_1 be the first vertex and y_1 be the last vertex in σ_i such that $f(x_1) = f(y_1) = f(w)$. Since w does not satisfy (1) with respect to f, we have that $x_1 <_{\sigma_i} w <_{\sigma_i} y_1$.

Let u be a vertex in G such that $u \neq w$ and u satisfies (1) or (2) with respect to f. We prove that u satisfies (1) or (2) with respect to f' also. If $f'(u) \notin \{f(w), f'(w)\}$ or $u \notin V_i$, then clearly u satisfies (1) or (2) with respect to f'. So we assume that $u \in V_i$ and $f'(u) \in \{f(w), f'(w)\}$

Case 1: $f'(u) = f(w)$, and u satisfies (1) with respect to f. Then u is the first vertex or the last vertex that is assigned a label $f(u)$ by f. Since w is the only vertex such that $f'(w) \neq f(w)$ and $f'(u) = f(w)$, u is the first vertex or the last vertex that is assigned a label $f'(u) = f(u)$ by f'.

Case 2: $f'(u) = f(w)$, and u satisfies (2) with respect to f. Then, $f(u)$ (which is equal to $f(w)$ and $f'(u)$) is the least integer in $L(u)$ such that there exist $x_1, y_1 \in V_i$ with $x_1 <_{\sigma_i} u <_{\sigma_i} y_1$ and $f(x_1) = f(y_1) = f(u) = f(w)$. Thus, by the definition of x_1 and y_1, we have that $x_1 <_{\sigma_i} u <_{\sigma_i} y_1$ and $f(x_1) = f(y_1) = f(u)$. This implies that u and w appears between x_1 and y_1 in the ordering σ_i. We consider the case $x_1 <_{\sigma_i} u <_{\sigma_i} w <_{\sigma_i} y_1$, and we omit the case $x_1 <_{\sigma_i} w <_{\sigma_i} u <_{\sigma_i} y_1$ as the arguments are symmetric. If $f'(w) \notin L(u)$, then u satisfies (2) with respect to f'. Now, if $f'(w) \in L(u)$, then there will not be any vertex $z <_{\sigma_i} u$ such that $f(z) = f'(w)$. Otherwise, we get $f(z) = f'(w) = f(y)$, $f'(w) \in L(u)$, and $z <_{\sigma_i} u <_{\sigma_i} y$, and it contradicts the assumption that $f(u)$ is the least integer satisfying property (2) for u with respect to f. This implies that u satisfies (2) with respect to f'.

Case 3: $f'(u) = f'(w)$ and u satisfies (1) with respect to f. Suppose u is the first vertex in σ_i that is assigned a label $f(u)$ by f. We claim that $u <_{\sigma_i} w$. For the sake of contradiction, let $w <_{\sigma_i} u$. We know that $x <_{\sigma_i} w$ and $f(x) = f'(w) = f'(u)$. This contradicts the assumption that u is the first vertex in σ_i that is assigned a label $f(u)$ by f. Since $u <_{\sigma_i} w$, u is the first vertex in σ_i that is assigned a label $f'(u)$ (which is equal to $f(u)$) by f' and hence u satisfies (1) with respect to f'. The case when u is the last vertex in σ_i that is assigned a label $f(u)$ by f, is symmetric in arguments and hence is omitted.

Case 4: $f'(u) = f'(w)$ and u satisfies (2) with respect to f. Since u satisfies (2) with respect to f, and $f'(u) = f'(w)$, we have that $x <_{\sigma_i} u <_{\sigma_i} y$ because x is the first vertex and y is the last vertex in σ_i which are assigned the label $f(u) = f'(u)$ by f. This implies that u satisfies (2) with respect to f'.

Thus, all the vertices which satisfy (1) or (2) with respect to f also satisfy (1) or (2) with respect to f'. The vertex w does not satisfy (1) or (2) with respect to f, but satisfies (2) with respect to f'. This completes the proof of the claim. □

This completes the proof of the lemma. □

Proof (Proof of Theorem 1). Let (G, H, L) be an instance of LHom where $G = (V, E)$ is a bipartite permutation graph and $V(H) = \{1, 2, \ldots, k\}$. By Proposition 1, there exists a partition of V into $V_0, V_1, \ldots, V_q$ satisfying properties (1)-(4). Because of property (4), such a partition can be constructed in polynomial time.

We now discuss the overall idea of the algorithm. In each set V_i, $i \in \{0, 1, \ldots, q\}$, for each label $j \in [k]$, we guess whether the label j is assigned to 0, 1 or at least 2 vertices in V_i. For the latter case, when at least two vertices are assigned the label j, we guess two vertices with label j and extend the labeling to other vertices. Depending on the guess for the label j, we guess the first vertex and the last vertex (the first and the last vertices are the same when there is exactly one vertex assigned the label j) in σ_i that are assigned the label j, in a list homomorphism from G to H. Using the partial labeling obtained from each guess, we obtain a full labeling of V_i maintaining the property of list homomorphism using Lemma 1. That is, for each vertex that is not assigned a label, we choose a label satisfying property (2) of Lemma 1. Then we construct a directed graph G' using the labelings obtained at each V_i and solve the directed s-t path problem on G' to decide if a list homomorphism exists from G to H.

Now we explain the algorithm in detail. We process the vertices of G in the order $V_0, V_1, \ldots, V_q$. From (3) of Definition 1, there exists an ordering σ_i of V_i that is decreasing in G_i and increasing in G_{i+1}. At each V_i, $0 \le i \le q$, for each label $j \in [k]$, we guess the first and the last vertices in σ_i that are assigned the label j. That is, we guess a partial labeling $\widehat{c}$ of V_i such that at most $2k$ vertices are assigned labels. Then we extend $\widehat{c}$ to a full labeling $c : V_i \to \{1, 2, \ldots, k\}$. For each vertex u labeled under $\widehat{c}$, we set $c(u) = \widehat{c}(u)$. For each of the remaining vertices, we use Lemma 1 to assign a label. We say a labeling c of V_i is *feasible* if there exists a partial labeling $\widehat{c}$ of V_i that can be extended to c using Lemma 1. Let C_i denote the set of all feasible labelings of V_i. Hence $|C_i| \le n^{2k}$.

We now construct an auxiliary directed graph G' with $V(G') = \{s, t\} \cup C_0 \cup C_1 \cup \cdots \cup C_q$, where C_i contains a vertex corresponding to every feasible labeling of V_i, $0 \le i \le q$. We add edges between vertices of two consecutive sets C_i and C_{i+1}, for each $0 \le i \le q-1$, in the following manner. We add a directed edge from $c \in C_i$ to $c' \in C_{i+1}$ if the labeling $c \cup c'$ is a list homomorphism from $G[V_i \cup V_{i+1}]$ to H, where c and c' are feasible labelings of V_i and V_{i+1}, respectively. We add directed edges from s to all vertices in C_0. Similarly, we add directed edges from all vertices in C_q to t. If we find a $s - t$ path in G', then such a path indicates the existence of list homomorphism from G to H.

Next, we show that there exists a list homomorphism from G to H if and only if there is a directed path from s to t in G'. If there exists a directed path from s to t in G', say P, then the number of vertices in P is $q + 3$. Moreover, $|P \cap C_i| = 1$, for each $0 \le i \le q$. This is due to the fact that there are edges only between consecutive sets C_i and C_{i+1} and the directed edges are from vertices in C_i to C_{i+1}. In addition, the edge from $c \in C_i$ to $c' \in C_{i+1}$ indicates the existence of list homomorphism from $G[V_i \cup V_{i+1}]$ to H, where c and c' are feasible labelings of V_i and V_{i+1} respectively. Let c_i be the vertex at distance $i+1$

from s in P. The vertex c_i represents a feasible labeling of V_i. Thus the feasible labelings $c_1, \ldots, c_q$ assigned to $V_0, V_1, \ldots, V_q$, respectively, together obtain a list homomorphism from G to H.

For the forward direction, let f be a list homomorphism from G to H such that f satisfies the properties mentioned in Lemma 1. Then, there exists a vertex $c_i \in V(G')$ that captures the labeling of V_i with respect to the labeling f, for each $0 \leq i \leq q$. Since f is a list homomorphism, there exists an edge from c_i to c_{i+1}, for all $0 \leq i \leq q-1$. This leads to a directed path from s to t.

Next, we do the runtime analysis. Because of the property (4) in Proposition 1, the partition $V_0, \ldots, V_q$ can be computed in $O(n^3)$ time. In our process, for each V_i, $i \in \{0, \ldots, q\}$, we guess whether a label is assigned to none of the vertices, one vertex, or more than one vertex in V_i. Since the number of labels is k, the above guessing takes $O(3^k)$ time. Then, we guess at most $2k$ vertices from each V_i that are "critical" (the first and the last vertices assigned a label in V_i) for the labeling resulting in $O(3^k n^{2k})$ partial labelings. We extend a partial labeling to a full labeling by assigning a label to an unlabelled vertex using (2) of Lemma 1, which takes $O(n^2)$ time. The number of edges between a pair of layers in G' is $O(3^{2k} n^{4k})$. Since there are q pairs of layers in G', the total number of edges is $O(q 3^{2k} n^{4k})$. Since $q \leq n$, and checking if an edge corresponds to a valid list homomorphism takes $O(n^2)$ time, we need $O(3^{2k} n^{4k+3})$ time to complete the construction of G'. The final step of the algorithm is to find a directed $s-t$ path in G' which can be done in $O(9^k n^{4k+3})$ time. Thus, the overall running time is $O(9^k n^{4k+3})$. □

The above algorithm can be extended when the input graph is a bipartite graph that admits a multichain ordering property. Theorem 2 is a corollary of Theorem 6.

Theorem 6 ($\star$). List Homomorphism *can be solved in* $O(n^{8k+3})$ *time on bipartite graphs that admit a multichain ordering property.*

4 Hardness: Proof of Theorem 3

In this section, we prove Theorem 3. To prove that, we use a specific type of chain graph. Let $G = (A, B, E)$ be a bipartite graph with $|A| = r$ and $|B| = s$ such that $(x_1, x_2, \ldots, x_r)$ and $(y_1, y_2, \ldots, y_s)$ be the chain orderings of A and B, respectively. That is $(x_1, x_2, \ldots, x_r)$ is increasing in G and $(y_1, y_2, \ldots, y_s)$ is decreasing in G. We call G, an *incremental* chain graph if $E(G) = \{x_i y_j \colon 1 \leq i \leq r,\ 1 \leq j \leq i,\ j \leq s\}$.

Towards proving the hardness, we give a polynomial-time parameter preserving reduction from the Multicolored Independent Set (McIS for short) problem to List k-Coloring. In McIS, the input is a graph G, a positive integer k, and a partition $(X_1, \ldots, X_k)$ of $V(G)$. The goal is to check if there exists a k-sized independent set $S \subseteq V(G)$ such that for all $i \in [k]$, $|S \cap X_i| = 1$. The problem is known to be W[1]-hard [6]. In fact it is known that McIS can not be solved in time $n^{o(k)}$ unless the Exponential Time Hypothesis fails. Let

$(G, k, (X_1, \ldots, X_k))$ be an instance of McIS such that m be the number of edges in G and X_i be an independent set with cardinality n, for each $i \in [k]$ (without loss of generality we can assume this).

For our reduction, we require that m is a multiple of 2 and 3. Suppose m is not a multiple of 6. In this case, we can modify our instance (G, k) to $(G', k+1)$ such that the number of edges in G' is a multiple of 6. Let $m = b \mod 6$ where $b \in \{1, 2, \ldots 5\}$. We add one new set of vertices X_{k+1} of size n, and add b number of edges between some vertex of X_{k+1} to b vertices in X_k. Additionally, we update the parameter k to $k+1$. Observe that G' has a multicolored independent set of size $k+1$ if and only if G has a multicolored independent set of size k. Thus without loss of generality, we can assume that m is a multiple of 6 for the given instance $(G, k, (X_1, X_2, \ldots, X_k))$.

First of all, we fix an arbitrary ordering of the vertices in X_i, for each $i \in [k]$. Let $\sigma(V(G))$ be a vertex ordering of G such that the vertices of X_1 appear in the above mentioned fixed order in $\sigma(V(G))$ and then the vertices of X_2 and so on. Let $E(G) = \{e_1, e_2, \ldots, e_m\}$ be the set of edges in G. From this we construct an instance (G', k') of List k-Coloring.

Construction of a block. First we explain a construction of a *block* and later we explain how to construct G' from the blocks.

1. Let $X'_i = X_i \cup \{x_i\}$ where x_i is a new element. We take one copy of each part X'_i of G and we call the union of these copies, a *layer*. We mention that later we add two or three more vertices to each layer. We take $2m$ copies of a layer, say $D_j = (X_{j1} \cup X_{j2} \cup \cdots \cup X_{jk})$, for $j \in [2m]$. We call this union of $2m$ layers, a *block*. For each D_j, we define an order σ_j on the vertices in D_j as follows. In the order $\sigma(V(G))$ insert x_i just before the first vertex of X_i for all $i \in [k]$.
2. For each edge e_ℓ, we add three new vertices in $D_{2\ell-1}$ and two new vertices in $D_{2\ell}$. Towards explaining this, let us fix an edge $e_\ell = uv \in E(G)$ such that $u \in X_i$ and $v \in X_j$ for some $i, j \in [k]$, where $i < j$. Let $u^{2\ell-1} \in X_{(2\ell-1)i}$ and $v^{2\ell-1} \in X_{(2\ell-1)j}$ be the copies of the vertices u and v in layer $D_{2\ell-1}$, respectively. We add two new vertices $\alpha(e_\ell)$ and $\beta(e_\ell)$ just before the vertices $u^{2\ell-1}$ and $v^{2\ell-1}$ in $\sigma_{2\ell-1}$, respectively. Similarly, we add two new vertices $\alpha'(e_\ell)$ and $\beta'(e_\ell)$ just before the vertices $u^{2\ell}$ and $v^{2\ell}$ in $\sigma_{2\ell}$, respectively. Also, we add one new vertex $\gamma(e_\ell)$ at the end of the ordering $\sigma_{2\ell-1}$ in layer $D_{2\ell-1}$ and this vertex we call an *edge vertex*. Let $Q = \{\alpha(e_\ell), \alpha'(e_\ell), \beta(e_\ell), \beta'(e_\ell), \gamma(e_\ell) \ : \ \ell \in [m]\}$. So, notice that now a layer D_j is a union of $X_{j1} \cup X_{j2} \cup \cdots \cup X_{jk}$ and two or three vertices from Q. This completes the description of the vertex set of a block B.
   Moreover, we use $\sigma_{2\ell-1}$ and $\sigma_{2\ell}$ to represent the order of vertices in $D_{2\ell-1}$ and $D_{2\ell}$, respectively, (including the new vertices) that is naturally derived from the old $\sigma_{2\ell-1}$ and $\sigma_{2\ell}$ as per the explanation of the new vertices added.
3. Next we explain the edges of G'. For each $i \in \{1, 3, \ldots, 2m-1\}$, we add edges between D_i and D_{i+1} such that the graph induced on $D_i \cup D_{i+1}$ is an incremental chain graph. Observe that except the edge vertex, every vertex in

D_i has a private neighbor in D_{i+1} now and the edge vertex in D_i is adjacent with every vertex in D_{i+1}. In the next step, we will add more edges.

4. For each $\ell \in [m]$, we add the following edges between $D_{2\ell-1}$ and $D_{2\ell}$. Let $e_\ell = uv$. We make $\alpha(e_\ell)$ adjacent to $u^{2\ell}$ and $\beta(e_\ell)$ adjacent to $v^{2\ell}$. Additionally, we add an edge between the vertex $\alpha'(e_\ell)$ and the vertex that appears just before the vertex $\alpha(e_\ell)$ in the ordering $\sigma_{2\ell-1}$ of layer $D_{2\ell-1}$. Similarly, we add an edge between the vertex $\beta'(e_\ell)$ and the vertex that appears just before the vertex $\beta(e_\ell)$ in the ordering $\sigma_{2\ell-1}$ of layer $D_{2\ell-1}$. See Figure 1 for an illustration.
5. Next we explain the edges between D_i and D_{i+1}, where $i \in \{2, 4, \ldots, 2m\}$. Recall that $Q = \{\alpha(e_\ell), \alpha'(e_\ell), \beta(e_\ell), \beta'(e_\ell), \gamma(e_\ell) \ : \ \ell \in [m]\}$. Let us fix an $i \in \{2, 4, \ldots, 2m\}$. We add edges between D_i and D_{i+1} such that the graph induced on $(D_i \cup D_{i+1}) \setminus Q$ with bipartition $D_i \setminus Q$ and $D_{i+1} \setminus Q$ forms an incremental chain graph with respect to the orderings σ_i and σ_{i+1} restricted on $D_i \setminus Q$ and $D_{i+1} \setminus Q$, respectively. Let ℓ and ℓ' be the integers such that $2\ell = i$ and $2\ell' - 1 = i + 1$. Notice that $\ell' = \ell + 1$. Also, notice that $\alpha'(e_\ell), \beta'(e_\ell) \in D_i$ and $\alpha(e_{\ell'}), \beta(e_{\ell'}) \in D_{i+1}$. Add the minimum number of edges on $\alpha'(e_\ell), \beta'(e_\ell), \alpha(e_{\ell'})$ and $\beta(e_{\ell'})$, between D_i and D_{i+1} such that it forms a chain graph with respect to orders σ_i and σ_{i+1}. For example, let w be the endpoint of the edge e_ℓ in graph G with minimum index in the ordering σ of graph G'. Let w^i and w^{i+1} be the copies of w in D_i and D_{i+1}, respectively. Note that w^i be the vertex that appears just before $\alpha'(e_\ell)$. Then, add edges between $\alpha'(e_\ell)$ and all the vertices that appear before w^{i+1} in σ_{i+1}. We also add an edge between $\alpha'(e_\ell)$ and w^{i+1}. Similarly, let z be the vertex that appears just after the minimum index endpoint of the edge $e_{\ell'}$ in the ordering σ of graph G. Let z^{i+1} be the copy of vertex z in layer D_{i+1} and appears just after $\alpha(e_{\ell'})$. Then add edges between $\alpha(e_{\ell'})$ and vertices that appear after z^i in σ_i. We also add an edge between $\alpha(e_{\ell'})$ and z^i. The cases of $\beta'(e_\ell)$ and $\beta(e_{\ell'})$ are symmetric. This completes the description of the edge set of the block B.
6. We denote the first vertex and the last vertex of any set X_{ij} in layer D_i as $first(X_{ij})$ and $last(X_{ij})$, respectively. Notice that $first(X_{ij})$ is the copy of x_j in $X_{i,j}$ and it will not corresponds to a vertex in $V(G)$. Next, we define a list function $L \colon V(B) \to [3k] \cup \{c_1, c_2, c_3, c_4, \widehat{c}_1, \widehat{c}_2, \widehat{c}_3, \widehat{c}_4\}$. For each $i \in [2m]$, $j \in [k]$, and $v \in X_{ij}$

$$\text{If } i = 1 \mod 3, \text{ then } L(v) = \begin{cases} \{3j-2\} & \text{if } v = first(X_{ij}) \\ \{3j-1\} & \text{if } v = last(X_{ij}) \\ \{3j-2, 3j-1\} & \text{if } first(X_{ij}) < v < last(X_{ij}) \end{cases}$$

$$\text{If } i = 2 \mod 3, \text{ then } L(v) = \begin{cases} \{3j\} & \text{if } v = first(X_{ij}) \\ \{3j-2\} & \text{if } v = last(X_{ij}) \\ \{3j-2, 3j\} & \text{if } first(X_{ij}) < v < last(X_{ij}) \end{cases}$$

$$\text{If } i = 0 \mod 3, \text{ then } L(v) = \begin{cases} \{3j-1\} & \text{if } v = first(X_{ij}) \\ \{3j\} & \text{if } v = last(X_{ij}) \\ \{3j-1, 3j\} & \text{if } first(X_{ij}) < v < last(X_{ij}) \end{cases}$$

7. For each $\ell \in [m]$, we explain the lists of $\alpha(e_\ell), \alpha'(e_\ell), \beta(e_\ell), \beta'(e_\ell)$, and $\gamma(e_\ell)$ as follows. Towards that let us fix $\ell \in [m]$. Let $e_\ell = uv$, where $u \in X_i$ and $v \in X_j$ for some $1 \leq i < j \leq k$. If ℓ is an odd number, then

$$\begin{aligned} L(\alpha(e_\ell)) &= L(first(X_{(2\ell-1)i}) \cup \{c_1\} \\ L(\alpha'(e_\ell)) &= L(first(X_{(2\ell-1)i})) \cup \{c_1, c_3\} \\ L(\beta(e_\ell)) &= L(first(X_{(2\ell-1)j}) \cup \{c_2\} \\ L(\beta'(e_\ell)) &= L(first(X_{(2\ell-1)j}) \cup \{c_2, c_4\} \\ L(\gamma(e_\ell)) &= \{c_3, c_4\} \end{aligned}$$

If ℓ is a even number, then replace each c_r with $\widehat{c}_r$ in the above equations, where $r \in \{1, 2, 3, 4\}$.

Construction of G'. We take $(nk+1)$ copies of a block with the same list function, say $B_1, B_2, \ldots, B_{nk+1}$. For any two consecutive blocks B_i and B_{i+1}, where $i \in [nk]$, we add edges between the last layer of B_i and the first layer of B_{i+1} according to item (5) in the construction of a block. Observe that the color lists of the vertices of the last layer of B_i and the first layer of B_{i+1} are *compatible* as the number of layers in each block is a multiple of m, which is a multiple of 3 and by the definition of list function. This completes the construction of graph G' with a list function $L \colon V(G') \to \mathcal{C}$, where $\mathcal{C} = [3k] \cup \{c_1, c_2, c_3, c_4, \widehat{c}_1, \widehat{c}_2, \widehat{c}_3, \widehat{c}_4\}$.

It is easy to verify that the obtained graph G' is a bipartite permutation graph as the layers of each block partition the vertex set of G' and the edges connecting any two consecutive layers induce a chain graph. Moreover, every vertex v in G' has a list $L(v) \subseteq \mathcal{C}$.

Note that for every layer D_i, the first vertex and the last vertex in each part X_{ij} has exactly one color in its list, say c and c', respectively such that $c \neq c'$, where $i \in [2m]$, $j \in [k]$ and $c, c' \in \mathcal{C}$. All the other vertices in the same part X_{ij} contain c and c' in their lists. It follows that in any list coloring, the first vertex in X_{ij} gets the color c and the last vertex in X_{ij} gets the color c' and all the other vertices get the color either c or c'. Note that there exists a vertex $w \in X_{ij}$ such that w is the first vertex in the ordering of X_{ij} which gets the color c', we call such a vertex *switch*. Moreover, a switch corresponds to a vertex in $V(G)$. Observe that each X_{ij} $(i \in [2m],\ j \in [k])$ contains at least one switch because of the list assignment of the vertices of X_{ij}. Additionally, each X_{ij} contains at most one switch by the definition of a switch. Therefore, in each layer D_i there are k switches, exactly one in each part X_{ij}, where $i \in [2m]$ and $j \in [k]$. We call a block B, a *consistent block* if for any pair of layers D_i and $D_{i'}$ and for any $j \in [k]$, the switches in X_{ij} and $X_{i'j}$ corresponds to the same vertex in G (i.e., these switches are copies of "a vertex" in $V(G)$).

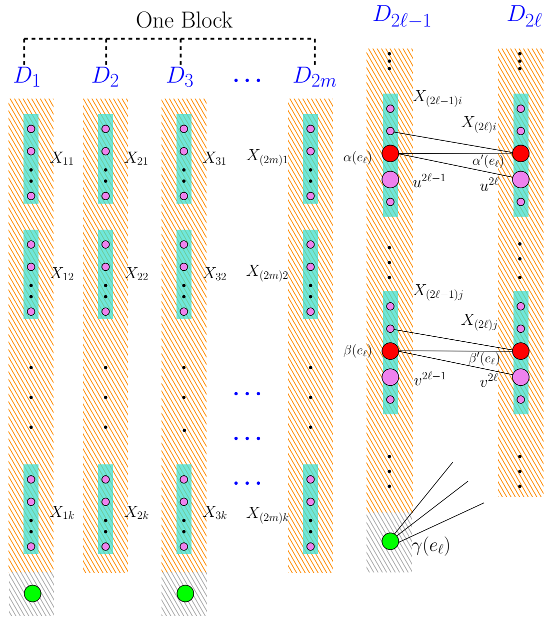

Fig. 1. A block on $2m$ layers in G' is illustrated on the left side of the figure, where each D_i represents a layer. The vertices and edges (apart from the edges of induced incremental chain graph) between two consecutive layers $D_{2\ell-1}$ and $D_{2\ell}$ are illustrated on the right side of the figure (the last two layers), where the green and red colored vertices are the newly added vertices corresponding to the edge $e_\ell = uv$ in G.

Correctness proof. Next, we show that G has a multicolored independent set of size k if and only if G' has a list k'-coloring, where $k' = 3k + 8$.

Lemma 2. *If G has a multicolored independent set of size k, then G' has a list k'-coloring, where $k' = 3k + 8$.*

Proof. Let $I = \{y_1, y_2, \ldots, y_k\}$ be an independent set in G such that $y_i \in X_i$. Our goal is to construct a list coloring $\phi : V(G') \to \mathcal{C}$. First, in each layer, we color the first and the last vertex of each set X_{ij} with the (only) color present in their lists. Next, we color all other vertices (except the vertices from Q) in every layer D_i ($i \in [2m]$) of each block such that the block gets consistent; that is, in every layer D_i of each block, switches corresponds to same vertices. Here, they correspond to the copies of $y_1, \ldots, y_k$. That is, for any vertex $y_j \in I$ and layer D_i, we make the vertex corresponding to y_j a switch. That means we color any vertex (except the vertices in Q) that appears before the vertex corresponding to y_j in the set X_{ij} with the color given to the first vertex of X_{ij} (which is a copy of x_j) and color all the other vertices (except the vertices in Q) in X_{ij} with the color given to the last vertex of the set X_{ij}, for $i \in [2m]$ and $j \in [k]$.

Observe that all the colored vertices maintain the proper coloring property by the chain ordering of each layer at this step. Also, every vertex gets color from its associated list. Thus, all the colored vertices maintain the list coloring property. The only uncolored vertices are the vertices in Q and the last vertices (called edge vertices, $\gamma(e_\ell)$ for all $\ell \in [m]$) in each layer. Next, we explain how to color those vertices.

Let $e_\ell = uv$ be an edge in G such that $u \in X_{i'}$ and $v \in X_{j'}$, where $i', j' \in [k]$ and $i' < j'$. Recall that corresponding to edge e_ℓ, we have three vertices $\alpha(e_\ell), \in X_{(2\ell-1)i'}$, $\beta(e_\ell) \in X_{(2\ell-1)j'}$, and $\gamma(e_\ell)$ in layer $D_{2\ell-1}$; and two vertices $\alpha'(e_\ell) \in X_{(2\ell)i'}$ and $\beta'(e_\ell) \in X_{(2\ell)j'}$ of layer $D_{2\ell}$. These are the only uncolored vertices so far in layer $D_{2\ell-1}$ and $D_{2\ell}$ of each block, for all $\ell \in [m]$. Note that according to our obtained (partial) list coloring, for any $j \in [k]$ and $i \in [2m]$, the switch in X_{ij} is y_j^i (i.e., the copy of y_j in the layer D_i). Now, there are three cases based on the position of the vertex $u^{2\ell-1}$ (or $v^{2\ell-1}$) with respect to the switches $y_{i'}^{2\ell-1}, y_{j'}^{2\ell-1}$, $y_{i'}^{2\ell}, y_{j'}^{2\ell}$ in layers $D_{2\ell-1}$ and $D_{2\ell}$. First we explain the colors of $\alpha(e_\ell)$ and $\alpha'(e_\ell)$. For this, we have three cases based the position of $u^{2\ell-1}$ compared with $y_{i'}^{2\ell-1}$.

Case 1: $u^{2\ell-1} <_{\sigma_{2\ell-1}} y_{i'}^{2\ell-1}$. In this case, we have $\alpha(e_\ell) <_{\sigma_{2\ell-1}} u^{2\ell-1} <_{\sigma_{2\ell-1}} y_{i'}^{2\ell-1}$ and $\alpha'(e_\ell) <_{\sigma_{2\ell}} u^{2\ell} <_{\sigma_{2\ell}} y_{i'}^{2\ell}$. Recall that the list of the vertex $\alpha(e_\ell)$ contains the color given to the first vertex $x_{i'}^{2\ell-1}$ of $X_{(2\ell-1)i'}$. Moreover, in our partial coloring, we colored $u^{2\ell-1}$ with the color of $x_{i'}^{2\ell-1}$. We color $\alpha(e_\ell)$ with the color of $x_{i'}$. Notice that the neighbours of $\alpha(e_\ell)$ in $D_{2\ell}$ is a subset of the neighbours of $u^{2\ell-1}$ in $D_{2\ell}$. Similarly, the neighbours of $\alpha(e_\ell)$ in $D_{2\ell-2}$ is a subset of the neighbours of $x_{i'}^{2\ell-1}$ in $D_{2\ell-2}$. So, as long as the colors on $x_{i'}^{2\ell-1}$ and $u^{2\ell-1}$ do not violate the proper coloring property, it holds on the vertex $\alpha(e_\ell)$. We color $\alpha'(e_\ell)$ with the unique color c_1' in $L(\alpha'(e_\ell)) \cap \{c_1, \widehat{c_1}\}$. Notice that this color is available only in the list of $\alpha(e_\ell)$ in $D_{2\ell-1}$ and we colored that vertex with a different color. Moreover, c_1' is not present in the list of any vertex in the layer $D_{2\ell+1}$.

Case 2: $y_{i'}^{2\ell-1} <_{\sigma_{2\ell-1}} u^{2\ell-1}$. In this case, we have $y_{i'}^{2\ell-1} <_{\sigma_{2\ell-1}} \alpha(e_\ell) <_{\sigma_{2\ell-1}} u^{2\ell-1}$ and $y_{i'}^{2\ell} <_{\sigma_{2\ell}} \alpha'(e_\ell) <_{\sigma_{2\ell}} u^{2\ell}$. Notice that $y_{i'}^{2\ell}$ and $u^{2\ell}$ are colored with the color q of $last(X_{(2\ell)i'})$ (which is same as the color of $x_{i'}^{2\ell-1}$). We color $\alpha'(e_\ell)$ with color q. Using arguments similar to that in the Case 1, one can argue that as long as the colors on $y_{i'}^{2\ell}$ and $u^{2\ell}$ do not violate the proper coloring property, it holds on the vertex $\alpha'(e_\ell)$. Now we color $\alpha(e_\ell)$ with the unique color c_1' in $L(\alpha(e_\ell)) \cap \{c_1, \widehat{c_1}\}$. Notice that this color is available only in the list of $\alpha'(e_\ell)$ in $D_{2\ell}$ and we colored that vertex with a different color. Moreover, c_1' is not present in the list of any vertex in the layer $D_{2\ell-2}$.

Case 3: $u^{2\ell-1} = y_{i'}^{2\ell-1}$. In this case, we have $u^{2\ell} = y_{i'}^{2\ell}$ and e_ℓ incident on $y_{i'}$ in G. Note that in this case, the vertices $\alpha(e_\ell)$ and $\alpha'(e_\ell)$ appear just before $y_{i'}^{2\ell-1}$ and $y_{i'}^{2\ell}$, respectively. Recall that the lists of both the vertices $\alpha(e_\ell)$ and $\alpha'(e_\ell)$ contain the color given to the first vertex $x_{i'}$ of $X_{(2\ell-1)i'}$. Observe that the vertex $u^{2\ell}$ is the switch in $X_{(2\ell)i'}$ and $\alpha(e_\ell)$ is adjacent to $u^{2\ell}$. Since the vertex $u^{2\ell}$ is the switch, $u^{2\ell}$ is colored with the color $\phi(last(X_{(2\ell)i'}))$, which is same as $\phi(x_{i'}^{2\ell-1})$. Therefore, we cannot color the vertex $\alpha(e_\ell)$ with the same color $\phi(x_{i'}^{2\ell-1})$. In this case, we color the vertex $\alpha(e_\ell)$ with the unique color present in

$L(\alpha(e_\ell)) \cap \{c_1, \widehat{c}_1\}$. Also, we color the vertex $\alpha'(e_\ell)$ with the unique color present in $L(\alpha(e_\ell)) \cap \{c_3, \widehat{c}_3\}$. It is easy to argue that the obtained partial coloring does not violate any constraint so far.

Similar to the Cases 1-3, we color the vertices $\beta(e_\ell)$ and $\beta'(e_\ell)$ based on one of the cases. Lastly, we color the edge vertex $\gamma(e_\ell)$ from its list. Recall that if ℓ is odd, $L(\gamma(e_\ell)) = \{c_3, c_4\}$ and if ℓ is even $L(\gamma(e_\ell)) = \{\widehat{c}_3, \widehat{c}_4\}$. We consider the case when ℓ is odd. The other case is symmetric in arguments and hence omitted. Notice that $L(\gamma(e_\ell)) = \{c_3, c_4\}$. Observe that we use a color from the set $\{c_3, c_4\}$ to color a vertex $\alpha'(e_\ell)$ or $\beta'(e_\ell)$, only in the Case 3. In order to (properly) color the vertex $\gamma(e_\ell)$ from its list, we need to prove that at most one of the vertices $\alpha'(e_\ell)$ and $\beta'(e_\ell)$ belong to Case 3 and use a color from the set $\{c_3, c_4\}$. Therefore, we prove the following claim.

Claim. For $e_\ell = uv \in E(G)$, exactly one of the vertices $\alpha'(e_\ell)$ or $\beta'(e_\ell)$ use the color from the set $\{c_3, c_4\}$.

Proof. Suppose that both the vertices $\alpha'(e_\ell)$ and $\beta'(e_\ell)$ use the color from the set $\{c_3, c_4\}$. It follows that both the vertices $\alpha'(e_\ell)$ and $\beta'(e_\ell)$ are colored by the Case 3. It implies, the vertices $u^{2\ell-1} = y_{i'}^{2\ell-1}$ and $v^{2\ell-1} = y_{j'}^{2\ell-1}$. Moreover, the endpoints of the edge e_ℓ are $y_{i'}$ and $y_{j'}$. This is a contradiction to the fact that $y_{i'}$ and $y_{j'}$ belong to the independent set I. $\square$

Thus, when ℓ is odd, we can color the edge vertex $\gamma(e_\ell)$ with an available color from its list $\{c_3, c_4\}$ that is not used to color the vertices in Q that are corresponding to the edge e_ℓ. Also, notice that $\gamma(e_\ell)$ does not have any neighbours in the layer $D_{2\ell-2}$. To argue that the given coloring does not violate any edge constraints between two consecutive blocks, one can use the subset of the arguments used in Cases 1-3 and the fact that the last layer of a block is an even layer and m is a multiple of 3. Hence, we obtained a list k'-coloring of G'. $\square$

Lemma 3 ($\star$). *If G' has a list k'-coloring, then G has a multicolored independent set of size k, where $k' = 3k + 8$.*

5 Conclusion

In this paper, we study LHom on bipartite graphs that admit a multichain ordering and give efficient algorithms. However, we could not extend the algorithm to interval graphs because the graph induced by the vertices in a layer (in a multichain ordering) need not be an independent set. It is interesting to get faster algorithms for LHom on interval graphs.

Acknowledgements: We would like to thank anonymous referees for their helpful comments. The first author acknowledges SERB-DST for supporting this research via grant PDF/2021/003452.

References

1. Jan Bok, Richard Brewster, Tomás Feder, Pavol Hell, and Nikola Jedličková. List homomorphism problems for signed graphs. *arXiv preprint arXiv:2005.05547*, 2020.
2. Jan Bok, Richard Brewster, Tomás Feder, Pavol Hell, and Nikola Jedličková. List Homomorphisms to Separable Signed Graphs. In *Conference on Algorithms and Discrete Applied Mathematics*, pages 22–35. Springer, 2022.
3. Andreas Brandstädt and Vadim V. Lozin. On the linear structure and clique-width of bipartite permutation graphs. *Ars Comb.*, 67, 2003.
4. Hubie Chen, Bart M. P. Jansen, Karolina Okrasa, Astrid Pieterse, and Pawel Rzazewski. Sparsification Lower Bounds for List H-Coloring. In Yixin Cao, Siu-Wing Cheng, and Minming Li, editors, *31st International Symposium on Algorithms and Computation, ISAAC 2020, December 14-18, 2020*, volume 181 of *LIPIcs*, pages 58:1–58:17, 2020.
5. Rajesh Chitnis, László Egri, and Dániel Marx. List h-coloring a graph by removing few vertices. *Algorithmica*, 78(1):110–146, 2017.
6. Marek Cygan, Fedor V Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized algorithms*, volume 5. Springer, 2015.
7. Josep Díaz, Öznur Yaşar Diner, Maria Serna, and Oriol Serra. On list k-coloring convex bipartite graphs. In *Graphs and Combinatorial Optimization: from Theory to Applications*, pages 15–26. Springer, 2021.
8. Josep Díaz, Maria J. Serna, and Dimitrios M. Thilikos. Counting h-colorings of partial k-trees. *Theor. Comput. Sci.*, 281(1-2):291–309, 2002.
9. László Egri, Andrei Krokhin, Benoit Larose, and Pascal Tesson. The complexity of the list homomorphism problem for graphs. *Theory of Computing Systems*, 51(2):143–178, 2012.
10. Jessica A. Enright, Lorna Stewart, and Gábor Tardos. On List Coloring and List Homomorphism of Permutation and Interval Graphs. *SIAM J. Discret. Math.*, 28(4):1675–1685, 2014.
11. Paul Erdös, Arthur L Rubin, and Herbert Taylor. Choosability in graphs. In *Proc. West Coast Conf. on Combinatorics, Graph Theory and Computing, Congressus Numerantium*, volume 26, pages 125–157, 1979.
12. Tomás Feder and Pavol Hell. List homomorphisms to reflexive graphs. *Journal of Combinatorial Theory, Series B*, 72(2):236–250, 1998.
13. Tomás Feder, Pavol Hell, and Jing Huang. List homomorphisms and circular arc graphs. *Combinatorica*, 19(4):487–505, 1999.
14. Tomás Feder, Pavol Hell, and Jing Huang. Bi-arc graphs and the complexity of list homomorphisms. *Journal of Graph Theory*, 42(1):61–80, 2003.
15. Tomás Feder, Pavol Hell, Sulamita Klein, and Rajeev Motwani. List Partitions. *SIAM J. Discret. Math.*, 16(3):449–478, 2003.
16. Tomás Feder, Pavol Hell, Sulamita Klein, Loana Tito Nogueira, and Fábio Protti. List matrix partitions of chordal graphs. *Theor. Comput. Sci.*, 349(1):52–66, 2005.
17. Michael R Garey, David S Johnson, and Larry Stockmeyer. Some simplified NP-complete problems. In *Proceedings of the sixth annual ACM symposium on Theory of computing*, pages 47–63, 1974.
18. N. Garg, M. Papatriantafilou, and P. Tsigas. Distributed list coloring: how to dynamically allocate frequencies to mobile base stations. In *Proceedings of SPDP '96: 8th IEEE Symposium on Parallel and Distributed Processing*, pages 18–25, 1996.

19. Chính T Hoàng, Marcin Kamiński, Vadim Lozin, Joe Sawada, and Xiao Shu. Deciding k-colorability of P_5-free graphs in polynomial time. *Algorithmica*, 57(1):74–81, 2010.
20. Klaus Jansen and Petra Scheffler. Generalized coloring for tree-like graphs. *Discrete Applied Mathematics*, 75(2):135–155, 1997.
21. Hyobin Kim and Mark Siggers. Towards a dichotomy for the switch list homomorphism problem for signed graphs. *arXiv preprint arXiv:2104.07764*, 2021.
22. Jan Kratochvil and Zsolt Tuza. Algorithmic complexity of list colorings. *Discrete Applied Mathematics*, 50(3):297–302, 1994.
23. Karolina Okrasa, Marta Piecyk, and Pawel Rzazewski. Full complexity classification of the list homomorphism problem for bounded-treewidth graphs. In *28th Annual European Symposium on Algorithms, ESA 2020, September 7-9, 2020, Pisa, Italy (Virtual Conference)*, pages 74:1–74:24, 2020.
24. Karolina Okrasa and Pawel Rzazewski. Complexity of the List Homomorphism Problem in Hereditary Graph Classes. In Markus Bläser and Benjamin Monmege, editors, *38th International Symposium on Theoretical Aspects of Computer Science, STACS 2021*, volume 187 of *LIPIcs*, pages 54:1–54:17, 2021.
25. Payam Valadkhan. *List matrix partitions of special graphs*. PhD thesis, Applied Sciences: School of Computing Science, 2013.
26. Payam Valadkhan. List matrix partitions of graphs representing geometric configurations. *Discret. Appl. Math.*, 260:237–243, 2019.
27. Vadim G Vizing. Vertex colorings with given colors. *Diskret. Analiz*, 29:3–10, 1976.
28. Wei Wang and Xin Liu. List-coloring based channel allocation for open-spectrum wireless networks. In *VTC-2005-Fall. 2005 IEEE 62nd Vehicular Technology Conference, 2005.*, volume 1, pages 690–694. Citeseer, 2005.