# On List $k$-Coloring Convex Bipartite Graphs

J. Diaz[1], Ö. Y. Diner[1,2], M. Serna[1], O. Serra[1]

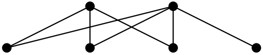
[1]Universidad Politecnica de Catalunya, Barcelona
[2]Kadir Has University, İstanbul

15 September 2020

## Outline

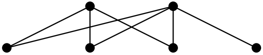
# COL: Coloring

- Given a graph $G = (V, E)$ and a non-negative integer $k$.



# COL: Coloring

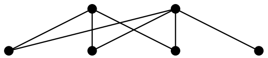
- Given a graph $G = (V, E)$ and a non-negative integer $k$.



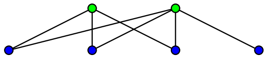
- A (proper) *coloring* of $G$ is a map $c : V \to \mathbb{N}$ such that $c(u) \neq c(v)$ if $u$ is adjacent to $v$.

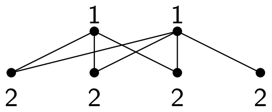
## Col: Coloring

- Given a graph $G = (V, E)$ and a non-negative integer $k$.



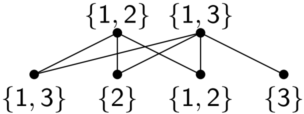
- A (proper) *coloring* of $G$ is a map $c : V \to \mathbb{N}$ such that $c(u) \neq c(v)$ if $u$ is adjacent to $v$.
- A *k-coloring* is a map $c : V \to [k]$.





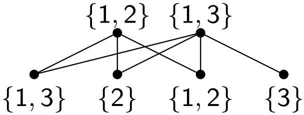
## LiCol: List Coloring

- A *list assignment* $L : V \to 2^{\mathbb{N}}$.

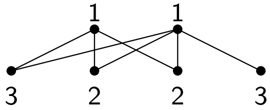


# LiCol: List Coloring

- A *list assignment* $L : V \to 2^{\mathbb{N}}$.

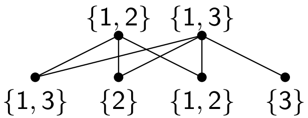



- List Coloring (LiCol) (Erdős 1979, Vizing 1976) asks for the existence of a proper coloring $c$ that obeys $L$, i.e., $\forall v \in V, c(v) \in L(v)$. If it exists, $G$ is said to be *$L$-colorable*.

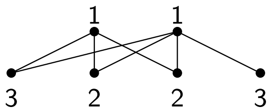


# LiCol: List Coloring

- A *list assignment* $L : V \to 2^{\mathbb{N}}$.



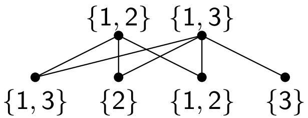

- List Coloring (LiCol) (Erdős 1979, Vizing 1976) asks for the existence of a proper coloring $c$ that obeys $L$, i.e., $\forall v \in V, c(v) \in L(v)$. If it exists, $G$ is said to be *L-colorable*.



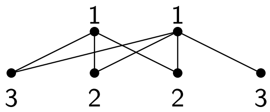

- List $k$-Coloring (Li $k$-Col), $L(v) \subseteq \{1, 2, \ldots, k\}$ for each $v \in V$.

## LiCol: List Coloring

- A *list assignment* $L : V \to 2^{\mathbb{N}}$.




- List Coloring (LiCol) (Erdős 1979, Vizing 1976) asks for the existence of a proper coloring $c$ that obeys $L$, i.e., $\forall v \in V, c(v) \in L(v)$. If it exists, $G$ is said to be *$L$-colorable*.



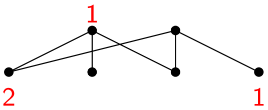

- List $k$-Coloring (Li $k$-Col), $L(v) \subseteq \{1, 2, \ldots, k\}$ for each $v \in V$.
- $k$-List Coloring ($k$-LiCol) each list $L$ has size at most $k$.

## PrExt: Pre-Coloring Extension

- A *pre-coloring* previously colors every vertex in $W$ for $W \subseteq V$.

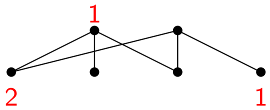
# PrExt: Pre-Coloring Extension

- A *pre-coloring* previously colors every vertex in $W$ for $W \subseteq V$.
- Precoloring Extension(PrExt) extends this pre-coloring to all of the vertices.



## PrExt: Pre-Coloring Extension

- A *pre-coloring* previously colors every vertex in $W$ for $W \subseteq V$.
- Precoloring Extension(PrExt) extends this pre-coloring to all of the vertices.




- In $k$-Precoloring Extension ($k$-PrExt) total number of colors is bounded by $k$.

## LI $H$-COL: List $H$–Coloring

- A *graph homomorphism* from $G$ to $H$ is a function $f : V(G) \to V(H)$ s. t. $f(u)f(v) \in E(H)$ whenever $uv \in E(G)$.

## LI $H$-COL: List $H$–Coloring

- A *graph homomorphism* from $G$ to $H$ is a function $f : V(G) \to V(H)$ s. t. $f(u)f(v) \in E(H)$ whenever $uv \in E(G)$.
- For a fixed graph $H$ and for an input $G$, $H$-COLORING ($H$-COL) asks whether there is a $G$ to $H$ homomorphism.

## LI $H$-COL: List $H$–Coloring

- A *graph homomorphism* from $G$ to $H$ is a function $f : V(G) \to V(H)$ s. t. $f(u)f(v) \in E(H)$ whenever $uv \in E(G)$.
- For a fixed graph $H$ and for an input $G$, $H$-COLORING ($H$-COL) asks whether there is a $G$ to $H$ homomorphism.
- In LIST $H$-COLORING (LI $H$-COL), each vertex of the input graph $G$ is associated with a list of vertices of $H$. The question is whether a $G$ to $H$ homomorphism exists that maps each vertex to a member of its list.

# LI $H$-COL: List $H$–Coloring

- A *graph homomorphism* from $G$ to $H$ is a function $f : V(G) \to V(H)$ s. t. $f(u)f(v) \in E(H)$ whenever $uv \in E(G)$.
- For a fixed graph $H$ and for an input $G$, $H$-COLORING ($H$-COL) asks whether there is a $G$ to $H$ homomorphism.
- In LIST $H$-COLORING (LI $H$-COL), each vertex of the input graph $G$ is associated with a list of vertices of $H$. The question is whether a $G$ to $H$ homomorphism exists that maps each vertex to a member of its list.
- LI $H$-COL is a generalization of LI $k$-COL.

## Relationships Between Coloring and its Variants

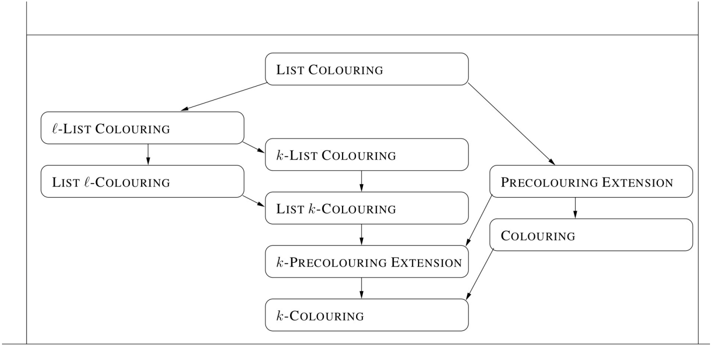
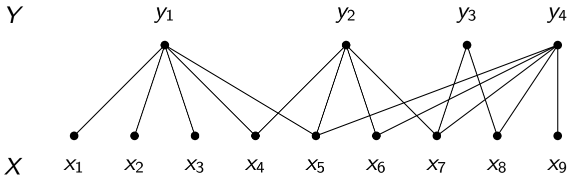



Figure from Golovach, Johnson, Paulusma and Song 2014.

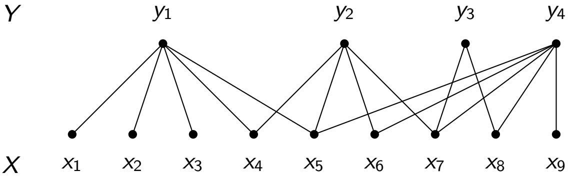
## Bipartite Graphs: Adjacency and Enclosure Properties

- Given a bipartite graph $G = (X \cup Y, E)$, $X$ and $Y$ form a bipartition of the vertex set into stable sets.



## Bipartite Graphs: Adjacency and Enclosure Properties

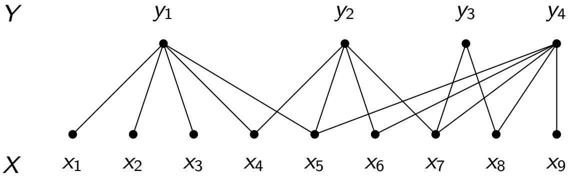
- Given a bipartite graph $G = (X \cup Y, E)$, $X$ and $Y$ form a bipartition of the vertex set into stable sets.




- An ordering for $X$ has the *adjacency property* or the ordering is *convex w. r. t. $X$* if, for each vertex $v \in Y$, $N(v)$ consists of vertices which are consecutive in the ordering of $X$.

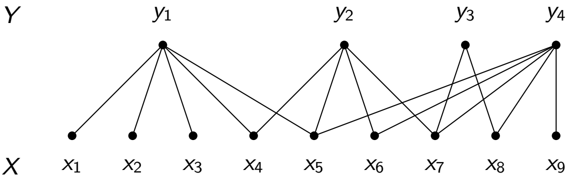
## Bipartite Graphs: Adjacency and Enclosure Properties

- Given a bipartite graph $G = (X \cup Y, E)$, $X$ and $Y$ form a bipartition of the vertex set into stable sets.

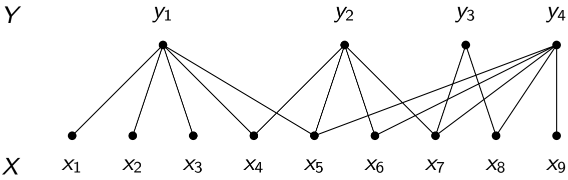



- An ordering for $X$ has the *adjacency property* or the ordering is *convex w. r. t.* $X$ if, for each vertex $v \in Y$, $N(v)$ consists of vertices which are consecutive in the ordering of $X$.
- An ordering for $X$ has the *enclosure property* if for every pair of vertices $u, v \in Y$ s. t. $N(u) \subseteq N(v)$, the vertices in $N(v) \backslash N(u)$ occur consecutively in the ordering of $X$.

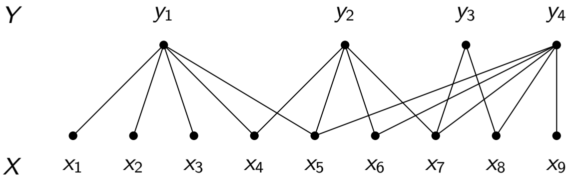
- *Chordal bipartite graphs* are bipartite graphs s. t. every cycle $C_n$ of length at least 6 has a chord.



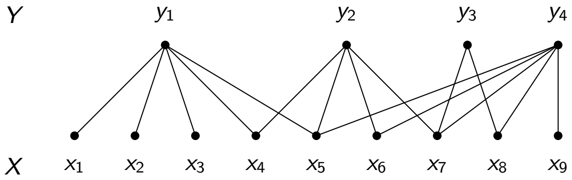
- *Chordal bipartite graphs* are bipartite graphs s. t. every cycle $C_n$ of length at least 6 has a chord.
- *Convex bipartite graphs* have the adjacency property on one of the partite sets.



- *Chordal bipartite graphs* are bipartite graphs s. t. every cycle $C_n$ of length at least 6 has a chord.
- *Convex bipartite graphs* have the adjacency property on one of the partite sets.
- *Biconvex bipartite graphs* have the adjacency property on both partite sets $X$ and $Y$.



- *Chordal bipartite graphs* are bipartite graphs s. t. every cycle $C_n$ of length at least 6 has a chord.
- *Convex bipartite graphs* have the adjacency property on one of the partite sets.
- *Biconvex bipartite graphs* have the adjacency property on both partite sets $X$ and $Y$.
- *Bipartite permutation graphs* are biconvex bipartite graphs in which one of the partite sets obeys both the adjacency and the enclosure properties.



## Computational Complexity

- COL, LICOL and PREXT are NP-C (Karp 1972, Garey and Johnson 1979)

## Computational Complexity

- COL, LICOL and PREXT are NP-C (Karp 1972, Garey and Johnson 1979)
- When $k$ is fixed: $k$-COL, $k$-LICOL, LI $k$-COL and $k$-PREXT are NP-C when $k \geq 3$ (Lovasz 1973); and they are PTIME when $k \leq 2$ (Erdos 1979, Vizing 1976).

## Computational Complexity

- COL, LICOL and PREXT are NP-C (Karp 1972, Garey and Johnson 1979)
- When $k$ is fixed: $k$-COL, $k$-LICOL, LI $k$-COL and $k$-PREXT are NP-C when $k \geq 3$ (Lovasz 1973); and they are PTIME when $k \leq 2$ (Erdos 1979, Vizing 1976).
- Perfect Graphs COL PTIME (Grostchel, Lovasz and Schrijver 1984).

## Computational Complexity

- COL, LICOL and PREXT are NP-C (Karp 1972, Garey and Johnson 1979)
- When $k$ is fixed: $k$-COL, $k$-LICOL, LI $k$-COL and $k$-PREXT are NP-C when $k \geq 3$ (Lovasz 1973); and they are PTIME when $k \leq 2$ (Erdos 1979, Vizing 1976).
- Perfect Graphs COL PTIME (Grostchel, Lovasz and Schrijver 1984).
- Subclasses of Perfect Graphs: LICOL NP-C for split graphs, bipartite graphs (Kubale 1992) and interval graphs (Biro 1992).

## Computational Complexity

- COL, LICOL and PREXT are NP-C (Karp 1972, Garey and Johnson 1979)
- When $k$ is fixed: $k$-COL, $k$-LICOL, LI $k$-COL and $k$-PREXT are NP-C when $k \geq 3$ (Lovasz 1973); and they are PTIME when $k \leq 2$ (Erdos 1979, Vizing 1976).
- Perfect Graphs COL PTIME (Grostchel, Lovasz and Schrijver 1984).
- Subclasses of Perfect Graphs: LICOL NP-C for split graphs, bipartite graphs (Kubale 1992) and interval graphs (Biro 1992).
- Subclasses of Chordal Bipartite Graphs: LI 4-COL NP-C for those that are $P_8$–free and 4–PREXT is NP-C for $P_{10}$–free (Huang, Johnson and Paulusma 2015).

## Computational Complexity

- Col, LiCol and PrExt are NP-C (Karp 1972, Garey and Johnson 1979)
- When $k$ is fixed: $k$-Col, $k$-LiCol, Li $k$-Col and $k$-PrExt are NP-C when $k \geq 3$ (Lovasz 1973); and they are PTIME when $k \leq 2$ (Erdos 1979, Vizing 1976).
- Perfect Graphs Col PTIME (Grostchel, Lovasz and Schrijver 1984).
- Subclasses of Perfect Graphs: LiCol NP-C for split graphs, bipartite graphs (Kubale 1992) and interval graphs (Biro 1992).
- Subclasses of Chordal Bipartite Graphs: Li 4-Col NP-C for those that are $P_8$–free and 4–PrExt is NP-C for $P_{10}$–free (Huang, Johnson and Paulusma 2015).
- LiCol PTIME for trees, complete graphs and graphs of bounded treewidth (Jansen and Scheffler 1997).

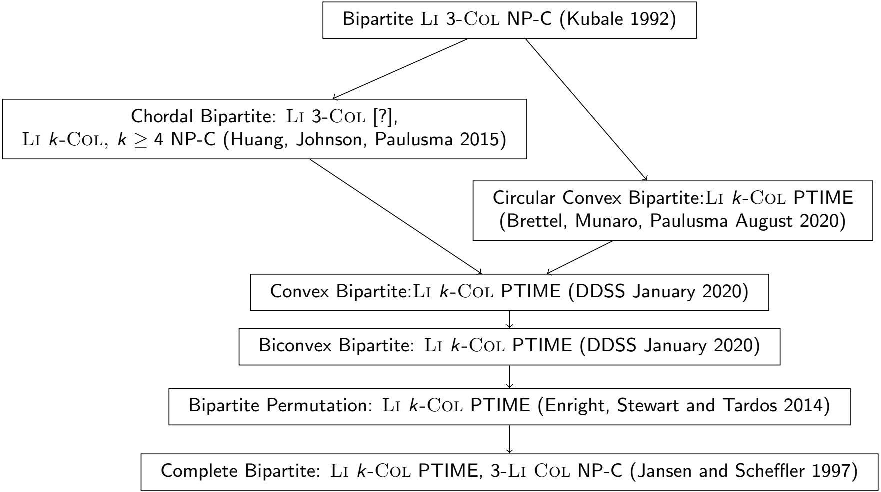
## Computational Complexity

- Col, LiCol and PrExt are NP-C (Karp 1972, Garey and Johnson 1979)
- When $k$ is fixed: $k$-Col, $k$-LiCol, Li $k$-Col and $k$-PrExt are NP-C when $k \geq 3$ (Lovasz 1973); and they are PTIME when $k \leq 2$ (Erdos 1979, Vizing 1976).
- Perfect Graphs Col PTIME (Grostchel, Lovasz and Schrijver 1984).
- Subclasses of Perfect Graphs: LiCol NP-C for split graphs, bipartite graphs (Kubale 1992) and interval graphs (Biro 1992).
- Subclasses of Chordal Bipartite Graphs: Li 4-Col NP-C for those that are $P_8$–free and 4–PrExt is NP-C for $P_{10}$–free (Huang, Johnson and Paulusma 2015).
- LiCol PTIME for trees, complete graphs and graphs of bounded treewidth (Jansen and Scheffler 1997).
- The complexities of $H$–Col (Hell and Nešetřil, 1994) and Li $H$–Col (Feder, Hell, Huang 1999) for arbitrary input graphs are completely characterized in terms of the structure of $H$.

## Computational Complexity

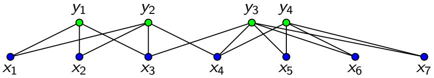
- Col, LiCol and PrExt are NP-C (Karp 1972, Garey and Johnson 1979)
- When $k$ is fixed: $k$-Col, $k$-LiCol, Li $k$-Col and $k$-PrExt are NP-C when $k \geq 3$ (Lovasz 1973); and they are PTIME when $k \leq 2$ (Erdos 1979, Vizing 1976).
- Perfect Graphs Col PTIME (Grostchel, Lovasz and Schrijver 1984).
- Subclasses of Perfect Graphs: LiCol NP-C for split graphs, bipartite graphs (Kubale 1992) and interval graphs (Biro 1992).
- Subclasses of Chordal Bipartite Graphs: Li 4-Col NP-C for those that are $P_8$–free and 4–PrExt is NP-C for $P_{10}$–free (Huang, Johnson and Paulusma 2015).
- LiCol PTIME for trees, complete graphs and graphs of bounded treewidth (Jansen and Scheffler 1997).
- The complexities of $H$–Col (Hell and Nešetřil, 1994) and Li $H$–Col (Feder, Hell, Huang 1999) for arbitrary input graphs are completely characterized in terms of the structure of $H$.
- Surveys: Tuza 1997, Paulusma 2016.

# Chart


Bipartite Li 3-Col NP-C (Kubale 1992)
Chordal Bipartite: Li 3-Col [?],
Li $k$-Col, $k \geq 4$ NP-C (Huang, Johnson, Paulusma 2015)
Circular Convex Bipartite:Li $k$-Col PTIME
(Brettel, Munaro, Paulusma August 2020)
Convex Bipartite:Li $k$-Col PTIME (DDSS January 2020)
Biconvex Bipartite: Li $k$-Col PTIME (DDSS January 2020)
Bipartite Permutation: Li $k$-Col PTIME (Enright, Stewart and Tardos 2014)
Complete Bipartite: Li $k$-Col PTIME, 3-Li Col NP-C (Jansen and Scheffler 1997)

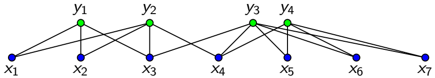
## Multi Chain Ordering

- A *chain graph* is a bipartite graph that contains no induced $2K_2$ (Yannakakis 1982).

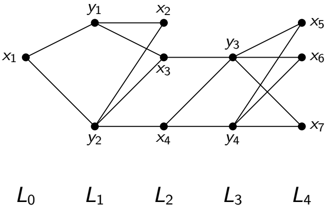


## Multi Chain Ordering

- A *chain graph* is a bipartite graph that contains no induced $2K_2$ (Yannakakis 1982).

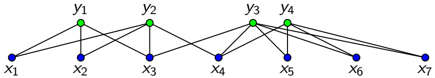



- The *distance layers* of a connected graph $G = (V, E)$ from a vertex $v_0$ are $L_0, L_1, ..., L_z$, where $L_0 = \{v_0\}$ and, for $i > 0$, $L_i$ consists of the vertices at distance $i$ from $v_0$ and $z$ is the largest integer for which this set is non-empty.

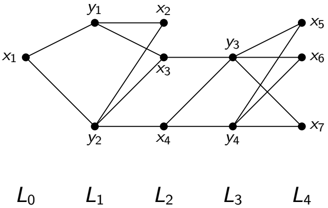


## Multi Chain Ordering

- A *chain graph* is a bipartite graph that contains no induced $2K_2$ (Yannakakis 1982).



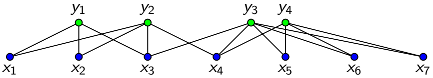

- The *distance layers* of a connected graph $G = (V, E)$ from a vertex $v_0$ are $L_0, L_1, \ldots, L_z$, where $L_0 = \{v_0\}$ and, for $i > 0$, $L_i$ consists of the vertices at distance $i$ from $v_0$ and $z$ is the largest integer for which this set is non-empty.



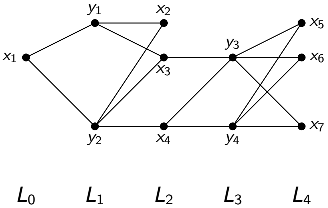

- These layers form a *multi-chain ordering* of $G$ if, for every two consecutive layers $L_i$ and $L_{i+1}$, the edges connecting them form a chain graph (not necessarily the layers themselves).

## Multi Chain Ordering

- A *chain graph* is a bipartite graph that contains no induced $2K_2$ (Yannakakis 1982).




- The *distance layers* of a connected graph $G = (V, E)$ from a vertex $v_0$ are $L_0, L_1, \ldots, L_z$, 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.




- These layers form a *multi-chain ordering* of $G$ if, for every two consecutive layers $L_i$ and $L_{i+1}$, the edges connecting them form a chain graph (not necessarily the layers themselves).
- Chain graphs are a proper subclass of convex bipartite graphs.

**Theorem (Enright, Stewart and Tardos 2014)**

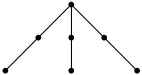
- *For a fixed graph $H$, Li $H$-Col is solvable in polynomial time for $G$ s. t. every connected induced subgraph of $G$ admits a multichain ordering.*
- *All connected bipartite permutation graphs and interval graphs admit multichain orderings.*

### Theorem (Enright, Stewart and Tardos 2014)

- *For a fixed graph $H$, LI $H$-COL is solvable in polynomial time for $G$ s. t. every connected induced subgraph of $G$ admits a multichain ordering.*
- *All connected bipartite permutation graphs and interval graphs admit multichain orderings.*

### Lemma (DDSS, 2020)

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

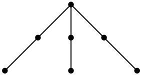


### Theorem (Enright, Stewart and Tardos 2014)

- *For a fixed graph $H$,* LI $H$-COL *is solvable in polynomial time for $G$ s. t. every connected induced subgraph of $G$ admits a multichain ordering.*
- *All connected bipartite permutation graphs and interval graphs admit multichain orderings.*

### Lemma (DDSS, 2020)

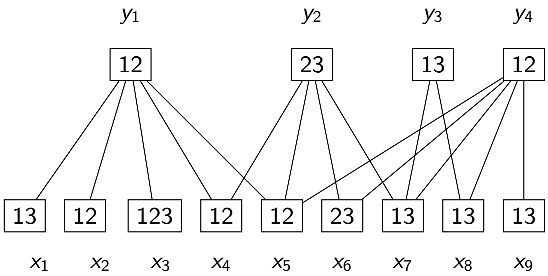
*If $G$ is a biconvex graph, then $G$ does not contain subd$(K_{1,3})$ as an induced subgraph.*



### Theorem (DDSS, 2020)

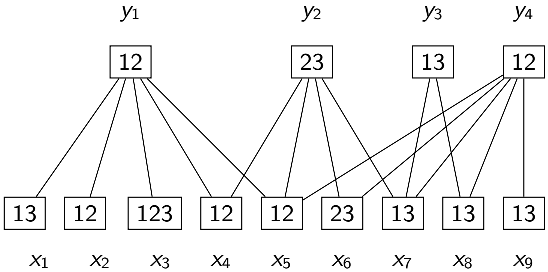
- *Every connected biconvex graph admits a multichain ordering.*
- *For any $H$,* LI $H$-COL *is solvable in polynomial time when restricted to biconvex graphs.*
- LI $k$-COL *and* $k$-PREXT *are solvable in polynomial time when restricted to biconvex graphs.*

## Color Algorithm



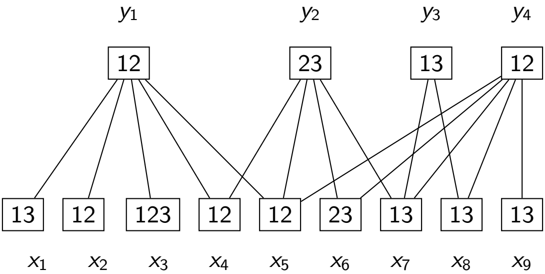

• Let $G = (X \cup Y, E)$ be a connected bipartite graph that is convex w.r.t. $X = \{x_1, \ldots, x_n\}$, thus for each $y \in Y$ there are two positive integers $a_y \leq b_y$ s. t. $N(y) = \{x_i \mid a_y \leq i \leq b_y\}$.

## Color Algorithm




- Let $G = (X \cup Y, E)$ be a connected bipartite graph that is convex w.r.t. $X = \{x_1, \ldots, x_n\}$, thus for each $y \in Y$ there are two positive integers $a_y \leq b_y$ s. t. $N(y) = \{x_i \mid a_y \leq i \leq b_y\}$.
- Define $A := \{a_y \mid y \in Y\}$ and $B := \{b_y \mid y \in Y\}$.

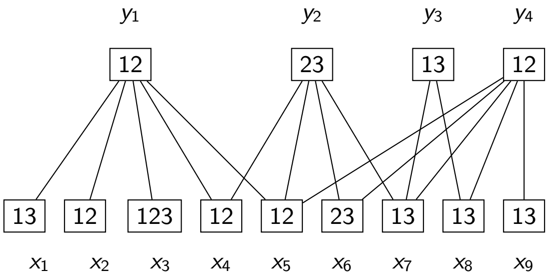
## Color Algorithm




- Define $A := \{a_y \mid y \in Y\}$ and $B := \{b_y \mid y \in Y\}$.

$A = \{1, 4, 5, 7\}$ and $B = \{5, 7, 8, 9\}$.

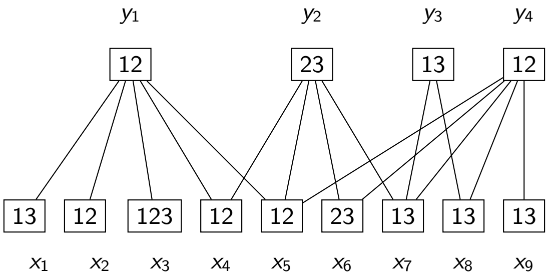
## Color Algorithm




$A = \{1, 4, 5, 7\}$ and $B = \{5, 7, 8, 9\}$.

• Assume that $B = \{b_1, \ldots, b_\beta\}$ are sorted so that $b_1 < b_2 < \cdots < b_\beta$. By the connectivity of $G$, $b_\beta = n$.

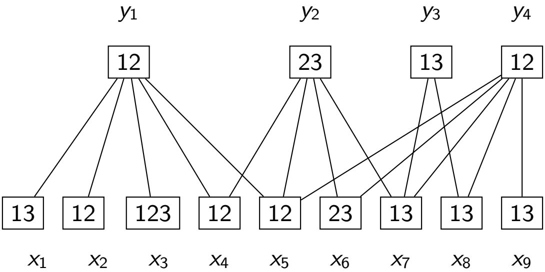
## Color Algorithm




$A = \{1, 4, 5, 7\}$ and $B = \{5, 7, 8, 9\}$.

- 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 $Z_j = \{y \in Y \mid a_y \leq b_j < b_y\}$.

## Color Algorithm

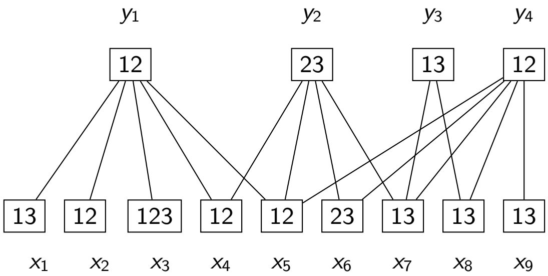



$A = \{1, 4, 5, 7\}$ and $B = \{5, 7, 8, 9\}$.

- 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 $Z_j = \{y \in Y \mid a_y \leq b_j < b_y\}$.

$b_2 = 7, X_2 = \{x_1, x_2, ..., x_7\}$, $Y_2 = \{y_1, y_2\}$ and $Z_2 = \{y_3, y_4\}$.

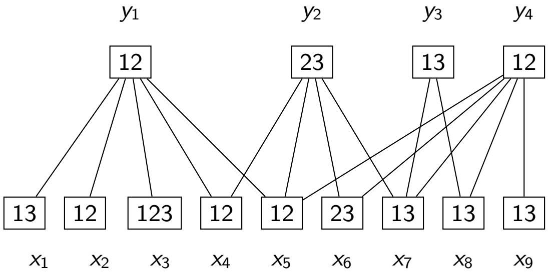
## Color Algorithm




$A = \{1, 4, 5, 7\}$ and $B = \{5, 7, 8, 9\}$.
$b_2 = 7, X_2 = \{x_1, x_2, ..., x_7\}, Y_2 = \{y_1, y_2\}$ and $Z_2 = \{y_3, y_4\}$.

- $Z_j$ contains vertices in $Y$ whose neighborhood starts before or at $b_j$ and ends after $b_j$.

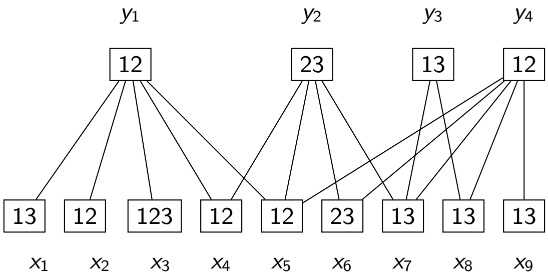
## Color Algorithm




$A = \{1, 4, 5, 7\}$ and $B = \{5, 7, 8, 9\}$.
$b_2 = 7, X_2 = \{x_1, x_2, ..., x_7\}, Y_2 = \{y_1, y_2\}$ and $Z_2 = \{y_3, y_4\}$.

- $Z_j$ contains vertices in $Y$ whose neighborhood starts before or at $b_j$ and ends after $b_j$.
- Let $G_j := G[X_j \cup Y_j], b_0 := 0$ and $G_0 := \emptyset$. Observe that $G_\beta = G$, $Z_\beta = \emptyset$

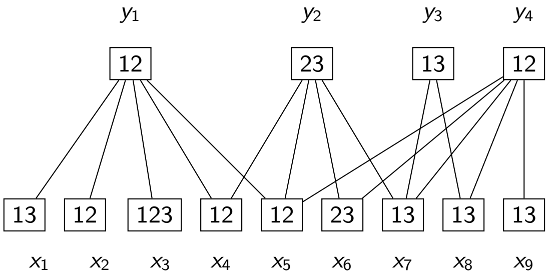
## Color Algorithm




$A = \{1, 4, 5, 7\}$ and $B = \{5, 7, 8, 9\}$.
$b_2 = 7, X_2 = \{x_1, x_2, ..., x_7\}, Y_2 = \{y_1, y_2\}$ and $Z_2 = \{y_3, y_4\}$.

- Let $G_j := G[X_j \cup Y_j], b_0 := 0$ and $G_0 := \emptyset$. Observe that $G_\beta = G$, $Z_\beta = \emptyset$
- Fix $j$, $1 \leq j \leq \beta$, define $A(j) = \{a_y \mid y \in Z_j\} \cup \{b_j\}$ and order it increasingly $\{a_{1,j}, \ldots, a_{\alpha_j,j} = b_j\}$.

## Color Algorithm




$A = \{1, 4, 5, 7\}$ and $B = \{5, 7, 8, 9\}$.
$b_2 = 7, X_2 = \{x_1, x_2, ..., x_7\}, Y_2 = \{y_1, y_2\}$ and $Z_2 = \{y_3, y_4\}$.

- Fix $j$, $1 \leq j \leq \beta$, define $A(j) = \{a_y \mid y \in Z_j\} \cup \{b_j\}$ and order it increasingly $\{a_{1,j}, \ldots, a_{\alpha_j,j} = b_j\}$.
- For each $1 \leq i \leq \alpha_j$ and $S \subsetneq [k]$, $T_j(i, S)$ will hold value $\mathrm{TRUE}$ whenever there is a valid list coloring of $G_j$ s. t. it uses no color in $S$ for the set $X_i^j = \{x_\ell \mid a_{i,j} \leq \ell < a_{i+1,j}\}$.

## What does the color algorithm do?

### The Color Algorithm

- In going from $j-1$ to $j$, compute 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, incorporate the restriction from $y \in Y_j$ that were not in $Y_{j-1}$.
- Finally, rearrange the information to keep only the values for the index in $A(j)$.

## What does the color algorithm do?

### The Color Algorithm

- In going from $j-1$ to $j$, compute 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, incorporate the restriction from $y \in Y_j$ that were not in $Y_{j-1}$.
- Finally, rearrange the information to keep only the values for the index in $A(j)$.

## What does the color algorithm do?

### The Color Algorithm

- In going from $j-1$ to $j$, compute 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, incorporate the restriction from $y \in Y_j$ that were not in $Y_{j-1}$.
- Finally, rearrange the information to keep only the values for the index in $A(j)$.

**Lemma (DDSS, 2020)**

*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 $T_\beta(\alpha_\beta, S) = \mathrm{TRUE}$.*

**Lemma (DDSS, 2020)**

*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 $T_\beta(\alpha_\beta, S) = \mathrm{TRUE}$.*

**Theorem (DDSS, 2020)**

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

## Open Problems

- What is the compuational complexity of Li 3-Col when restricted to chordal bipartite graphs?

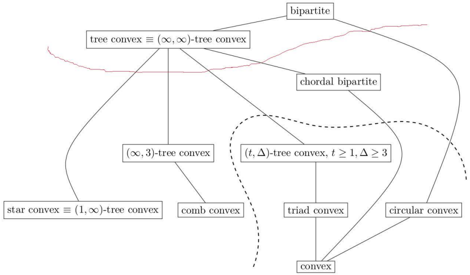



Figure -modified - from Brettel, Munaro, Paulusma 2020.

## Open Problems

- What is the compuational complexity of Li 3-Col when restricted to chordal bipartite graphs?
- .. Li $k$-Col for $k \geq 3$ when restricted to star convex graphs or to comb convex graphs?

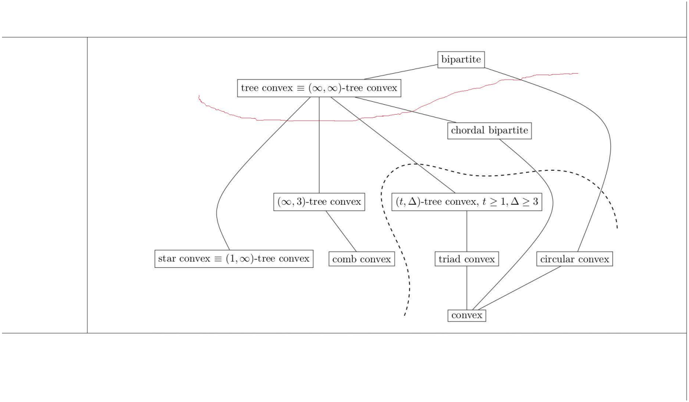



Figure -modified - from Brettel, Munaro, Paulusma 2020.

Thank you for your attention.