List Coloring Ordered Graphs with Forbidden Induced Subgraphs

Marta Piecyk[1] & Paweł Rzążewski[2]

[1]CISPA Helmholtz Center for Information Security
[2]Warsaw University of Technology

Grenoble, STACS 2026

List-k-Coloring

I: Graph G and for every vertex $v \in V(G)$, we have $L(v) \subseteq \{1, 2, \ldots, k\}$.

Q: Is there a proper coloring of G that respects the lists L?

List-k-Coloring

I: Graph G and for every vertex $v \in V(G)$, we have $L(v) \subseteq \{1, 2, \ldots, k\}$.

Q: Is there a proper coloring of G that respects the lists L?

Remark 1: An instance of List-k-Coloring can be seen as an instance of List-$(k+1)$-Coloring.

List-k-Coloring

I: Graph G and for every vertex $v \in V(G)$, we have $L(v) \subseteq \{1, 2, \ldots, k\}$.

Q: Is there a proper coloring of G that respects the lists L?

Remark 1: An instance of List-k-Coloring can be seen as an instance of List-$(k+1)$-Coloring.

Remark 2: If all lists have size at most 2, then the problem can be solved in polynomial time. (Edwards '86)

H-free (un)ordered graphs

G is H-free if G does not contain H as an induced subgraph.

H-free (un)ordered graphs

G is H-free if G does not contain H as an induced subgraph.

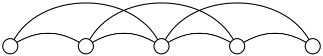
Ordered graphs: graph G and a linear ordering $\prec$ of $V(G)$.

H-free (un)ordered graphs

G is H-free if G does not contain H as an induced subgraph.

Ordered graphs: graph G and a linear ordering $\prec$ of $V(G)$.

Example:

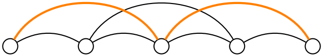


H-free (un)ordered graphs

G is H-free if G does not contain H as an induced subgraph.

Ordered graphs: graph G and a linear ordering $\prec$ of $V(G)$.

Example:



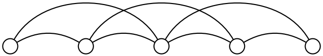
Contains induced 

H-free (un)ordered graphs

G is H-free if G does not contain H as an induced subgraph.

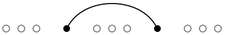
Ordered graphs: graph G and a linear ordering $\prec$ of $V(G)$.

Example:



Contains induced  Does not contain induced 

List-k-Coloring H-free ordered graphs

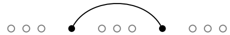
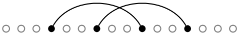
Forbidden H	$k = 3$	$k = 4$	$k \geqslant 5$
	[1]		
			
			
			
			
			
			
other			

[1] Hajebi, Li, Spirkl '24

Marta Piecyk[1] & Paweł Rzążewski[2] [1]CISPA Helmholtz Center for Information Security [2]Warsaw University of Technology

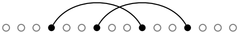
List-k-Coloring H-free ordered graphs

chordal →

Forbidden H	$k = 3$	$k = 4$	$k \geqslant 5$
	[1]		
			[1]
			
			
			
			
			
other			

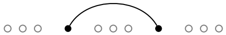
[1] Hajebi, Li, Spirkl '24

List-k-Coloring H-free ordered graphs

Forbidden H	$k = 3$	$k = 4$	$k \geqslant 5$
	[1]		
			[1]
	[1]		
	[1]		
			
			
			
other			

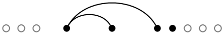
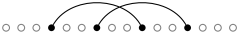
[1] Hajebi, Li, Spirkl '24

List-k-Coloring H-free ordered graphs

Forbidden H	$k = 3$	$k = 4$	$k \geqslant 5$
	[1]		
			[1]
	[1]		
	[1]		
			
			
			
other	[1]		

[1] Hajebi, Li, Spirkl '24

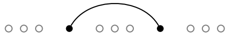
List-k-Coloring H-free ordered graphs

Forbidden H	$k = 3$	$k = 4$	$k \geqslant 5$
	[1]		★
			[1]
	[1]		
	[1]		
			
			
			
other	[1]		

[1] Hajebi, Li, Spirkl '24

★ Our results

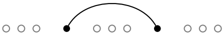
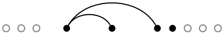
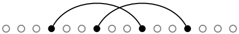
List-k-Coloring H-free ordered graphs

Forbidden H	$k = 3$	$k = 4$	$k \geqslant 5$
	[1]		★
			[1]
	[1]	★	
	[1]		
			
			
			
other	[1]		

[1] Hajebi, Li, Spirkl '24

★ Our results

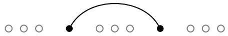
List-k-Coloring H-free ordered graphs

Forbidden H	$k = 3$	$k = 4$	$k \geqslant 5$
	[1]		★
			[1]
	[1]	★	
	[1]	★	
			
			
			
other	[1]		

[1] Hajebi, Li, Spirkl '24

★ Our results

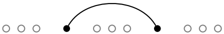
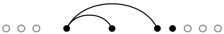
List-k-Coloring H-free ordered graphs

Forbidden H	$k = 3$	$k = 4$	$k \geqslant 5$
	[1]		★
			[1]
	[1]	★	
	[1]	★	
		★	
			
			
other	[1]		

[1] Hajebi, Li, Spirkl '24

★ Our results

List-k-Coloring H-free ordered graphs

Forbidden H	$k = 3$	$k = 4$	$k \geqslant 5$
	[1]		★
			[1]
	[1]	★	?
	[1]	★	
	?	★	
	?		
	?	?	?
other	[1]		

[1] Hajebi, Li, Spirkl '24

★ Our results

List-k-Coloring H-free ordered graphs

Forbidden H	$k = 3$	$k = 4$	$k \geqslant 5$
	[1]		★
			[1]
	[1]	★	?
	[1]	★	
	?	★	
	?		
	?	?	?
other	[1]		

[1] Hajebi, Li, Spirkl '24

★ Our results

List-4-Coloring  -free

Theorem. LIST-4-COLORING is polynomial-time solvable on  -free ordered graphs.

List-4-Coloring  -free

Theorem. List-4-Coloring is polynomial-time solvable on  -free ordered graphs.

Proof: Let (G, L) be an instance of List-4-Coloring s.t. G is  -free.

List-4-Coloring  -free

Theorem. List-4-Coloring is polynomial-time solvable on  -free ordered graphs.

Proof: Let (G, L) be an instance of List-4-Coloring s.t. G is  -free.

Basic reduction rules.

(R1) If G contains K_5, then return NO.

List-4-Coloring  -free

Theorem. List-4-Coloring is polynomial-time solvable on  -free ordered graphs.

Proof: Let (G, L) be an instance of List-4-Coloring s.t. G is  -free.

Basic reduction rules.

(R1) If G contains K_5, then return NO.

(R2) If there is $v \in V(G)$ with $L(v) = \emptyset$, then return NO.

List-4-Coloring  -free

Theorem. List-4-Coloring is polynomial-time solvable on  -free ordered graphs.

Proof: Let (G, L) be an instance of List-4-Coloring s.t. G is  -free.

Basic reduction rules.

(R1) If G contains K_5, then return NO.

(R2) If there is $v \in V(G)$ with $L(v) = \emptyset$, then return NO.

(R3) If there is $v \in V(G)$ with $L(v) = \{a\}$ for some $a \in [4]$, then remove a from list of every neighbor of v, and remove v from G.

List-4-Coloring  -free cnd.

Guess first vertex x_i in each color $i \in [4]$.

List-4-Coloring  -free cnd.

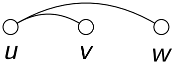
Guess first vertex x_i in each color $i \in [4]$. – $\mathcal{O}(n^4)$ branches

Marta Piecyk[1] & Paweł Rzążewski[2] [1]CISPA Helmholtz Center for Information Security [2]Warsaw University of Technology

List-4-Coloring  -free cnd.

Guess first vertex x_i in each color $i \in [4]$. – $\mathcal{O}(n^4)$ branches

Observation 1: There is no induced

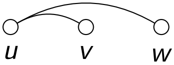


with $L(u) \cap L(v) \cap L(w) \neq \emptyset$.

List-4-Coloring  -free cnd.

Guess first vertex x_i in each color $i \in [4]$. – $\mathcal{O}(n^4)$ branches

Observation 1: There is no induced



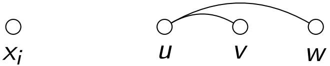
$i \in L(u) \cap L(v) \cap L(w)$

with $L(u) \cap L(v) \cap L(w) \neq \emptyset$.

List-4-Coloring  -free cnd.

Guess first vertex x_i in each color $i \in [4]$. – $\mathcal{O}(n^4)$ branches

Observation 1: There is no induced



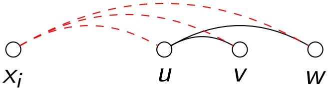
$i \in L(u) \cap L(v) \cap L(w)$

with $L(u) \cap L(v) \cap L(w) \neq \emptyset$.

List-4-Coloring  -free cnd.

Guess first vertex x_i in each color $i \in [4]$. – $\mathcal{O}(n^4)$ branches

Observation 1: There is no induced



with $L(u) \cap L(v) \cap L(w) \neq \emptyset$.

$i \in L(u) \cap L(v) \cap L(w)$

List-4-Coloring  -free cnd.

Let v be a forward neighbor of u. We say that v for u is:

- safe if $L(u) \cap L(v) \neq \emptyset$,
- dangerous, otherwise.

List-4-Coloring  -free cnd.

Let v be a forward neighbor of u. We say that v for u is:

- safe if $L(u) \cap L(v) \neq \emptyset$,
- dangerous, otherwise.

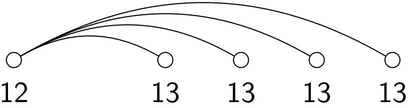
Observation 2: Every vertex has at most 12 safe forward neighbors.

List-4-Coloring  -free cnd.

Let v be a forward neighbor of u. We say that v for u is:

- safe if $L(u) \cap L(v) \neq \emptyset$,
- dangerous, otherwise.

Observation 2: Every vertex has at most 12 safe forward neighbors.

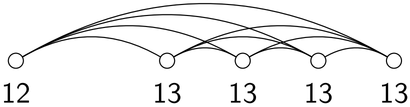


List-4-Coloring  -free cnd.

Let v be a forward neighbor of u. We say that v for u is:

- safe if $L(u) \cap L(v) \neq \emptyset$,
- dangerous, otherwise.

Observation 2: Every vertex has at most 12 safe forward neighbors.

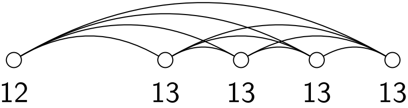


List-4-Coloring  -free cnd.

Let v be a forward neighbor of u. We say that v for u is:

- safe if $L(u) \cap L(v) \neq \emptyset$,
- dangerous, otherwise.

Observation 2: Every vertex has at most 12 safe forward neighbors.



K_5!

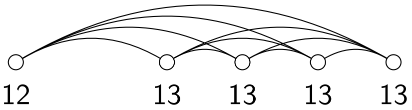
Marta Piecyk[1] & Paweł Rzążewski[2] [1]CISPA Helmholtz Center for Information Security [2]Warsaw University of Technology

List-4-Coloring  -free cnd.

Let v be a forward neighbor of u. We say that v for u is:

- safe if $L(u) \cap L(v) \neq \emptyset$,
- dangerous, otherwise.

Observation 2: Every vertex has at most 12 safe forward neighbors.



K_5!

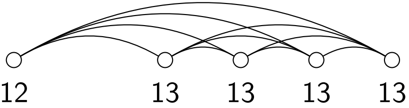
We say that v is good if it has at most 18 forward neighbors, otherwise v is bad.

List-4-Coloring  -free cnd.

Let v be a forward neighbor of u. We say that v for u is:

- safe if $L(u) \cap L(v) \neq \emptyset$,
- dangerous, otherwise.

Observation 2: Every vertex has at most 12 safe forward neighbors.



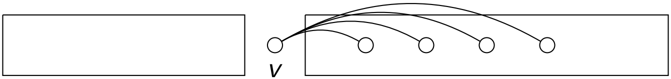
K_5!

We say that v is good if it has at most 18 forward neighbors, otherwise v is bad.

Remark: All bad vertices have lists of size 2.

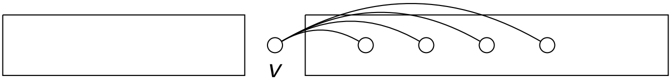
List-4-Coloring ⋯-free cnd.

Let v be the first good vertex. We guess coloring on v and its forward neighbors.



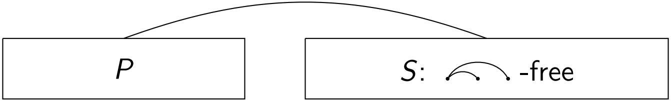
List-4-Coloring -free cnd.

Let v be the first good vertex. We guess coloring on v and its forward neighbors. – at most 4^{19} branches



List-4-Coloring -free cnd.

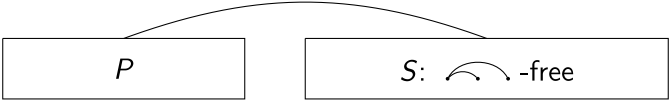
Let v be the first good vertex. We guess coloring on v and its forward neighbors. – at most 4^{19} branches



- Suffix S is a chordal graph.

List-4-Coloring -free cnd.

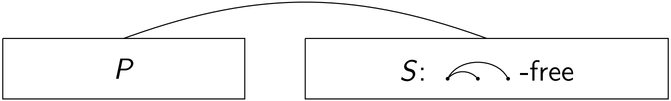
Let v be the first good vertex. We guess coloring on v and its forward neighbors. – at most 4^{19} branches



- Suffix S is a chordal graph.
- Prefix P contains only the vertices that were bad.

List-4-Coloring -free cnd.

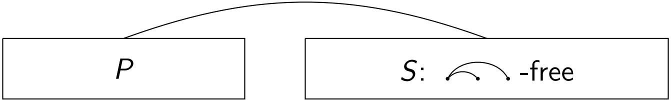
Let v be the first good vertex. We guess coloring on v and its forward neighbors. – at most 4^{19} branches



- Suffix S is a chordal graph.
- Prefix P contains only the vertices that were bad.
- By additional guesses, we can assume that there are at most two distinct lists on P (and disjoint), say $\{1, 2\}$ and $\{3, 4\}$.

List-4-Coloring -free cnd.

Let v be the first good vertex. We guess coloring on v and its forward neighbors. – at most 4^{19} branches



- Suffix S is a chordal graph.
- Prefix P contains only the vertices that were bad.
- By additional guesses, we can assume that there are at most two distinct lists on P (and disjoint), say $\{1,2\}$ and $\{3,4\}$.
- It turns out that we can remove some edges with lists $\{1,2\}$-$\{3,4\}$, so that the remaining graph is chordal.

Open questions

- Determine the complexity of List-k-Coloring in the remaining cases.

Open questions

- Determine the complexity of List-k-Coloring in the remaining cases.
- In particular: List-3-Coloring on  -free graphs.

Marta Piecyk[1] & Paweł Rzążewski[2] [1]CISPA Helmholtz Center for Information Security [2]Warsaw University of Technology

Open questions

- Determine the complexity of List-k-Coloring in the remaining cases.
- In particular: List-3-Coloring on  -free graphs.
- Subexponential-time algorithms?

Open questions

- Determine the complexity of List-k-Coloring in the remaining cases.
- In particular: List-3-Coloring on  -free graphs.
- Subexponential-time algorithms?
- What if we consider non-list coloring?

Open questions

- Determine the complexity of List-k-Coloring in the remaining cases.
- In particular: List-3-Coloring on  -free graphs.
- Subexponential-time algorithms?
- What if we consider non-list coloring?
- Other problems on ordered graphs with forbidden induced subgraphs.