4-Coloring H-Free Graphs When H Is Small

Petr A. Golovach
Daniel Paulusma
Jian Song

Durham University

23 January 2012

Introduction

$G = (V, E)$ is finite, undirected graph, no loops, no multiple edges.

A coloring of G is a mapping $c : V \to \{1, 2, \ldots\}$ such that

$$c(u) \neq c(v) \text{ whenever } uv \in E.$$

A coloring c of G is a k-coloring if $c(u) \in \{1, \ldots, k\}$ for all $u \in V$.

COLORING
Instance: a graph G and an integer k.
Question: does G have a k-coloring?

k-COLORING
Instance: a graph G.
Question: does G have a k-coloring?

Motivation

We are interested in special graph classes because of the following well-known result.

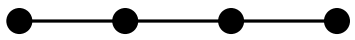
Theorem (Karp, 1972)

k-COLORING can be solved in polynomial time for $k \leq 2$, and it is NP-complete for $k \geq 3$.

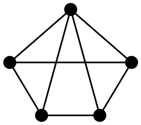
H-free graphs

Let G and H be two graphs. The graph G is H-free if G contains no *induced* subgraph isomorphic to H.

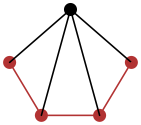
Let P_ℓ denote the path on ℓ vertices.



$H = P_4$



P_4-free



Not P_4-free

COLORING for H-free graphs

Let $F + R$ denote the disjoint union of graphs F and R.

Theorem (Král', Kratochvíl, Tuza & Woeginger, 2001)

Let H be a fixed graph.

If H is a (not necessarily proper) induced subgraph of P_4 or of $P_1 + P_3$ then COLORING *can be solved in polynomial time for H-free graphs.*

If not then COLORING *is* NP*-complete for H-free graphs.*

Cycles and Claws

We focus on the computational complexity of the k-Coloring problem for H-free graphs.

A result of Kamiński and Lozin [2007] implies the next theorem.

Theorem

For any $k \geq 3$, the k-Coloring problem is NP-complete for the class of H-free graphs whenever H contains a cycle.

Combining the results of Holyer [1981], and Leven and Galil [1983] leads to the following consequence.

Theorem

For any $k \geq 3$, the k-Coloring problem is NP-complete for the class of H-free graphs whenever H is a forest with a vertex of degree at least 3.

A linear forest is the disjoint union of a collection of paths.

Paths

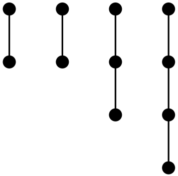
Combining the results of Hoàng et al. [2010], Randerath and Schiermeyer [2004], and Broersma et al. [2009, 2010] leads to the following table on k-COLORING for P_ℓ-free graphs.

	$k \rightarrow$						
P_ℓ-free	3	4	5	6	7	8	...
$\ell = 1$	P	P	P	P	P	P	...
$\ell = 2$	P	P	P	P	P	P	...
$\ell = 3$	P	P	P	P	P	P	...
$\ell = 4$	P	P	P	P	P	P	...
$\ell = 5$	P	P	P	P	P	P	...
$\ell = 6$	P	?	?	?	?	?	...
$\ell = 7$	?	?	?	NP-c	NP-c	NP-c	...
$\ell = 8$	?	NP-c	NP-c	NP-c	NP-c	NP-c	...
$\ell = 9$	?	NP-c	NP-c	NP-c	NP-c	NP-c	...
$\ell = 10$	?	NP-c	NP-c	NP-c	NP-c	NP-c	...
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

Disconnected Linear Forests

Let sH denote the disjoint union of s copies of H.

e.g.



$$H = 2P_2 + P_3 + P_4$$

By combining the results of Balas and Yu [1989], Tsukiyama et al. [1977], Broersma et al. [2010] and Couturier et al. [2011], we obtain the following theorem.

Theorem

- 3-Coloring *is polynomial-time solvable for H-free graphs if*
  - $H = rP_1 + P_2 + P_4$ *for any* $r \geq 0$
  - $H = rP_1 + P_6$ *for any* $r \geq 0$
  - $H = sP_3$ *for any* $s \geq 0$.

- *For any* $k \geq 4$, k-Coloring *is polynomial-time solvable for H-free graphs if*
  - $H = rP_1 + P_5$ *for any* $r \geq 0$
  - $H = sP_2$ *for any* $s \geq 0$.

Our new results

For any fixed graph H on at most 5 vertices, the computational complexity for 4-Coloring has been classified except the case where $H = P_2 + P_3$. We proved that 4-Coloring is also polynomial-time solvable for this case.

Theorem

For any fixed graph H on at most 5 vertices, 4-Coloring is polynomial-time solvable on H-free graphs whenever H is a linear forest and NP-complete otherwise.

Terminology

- A list assignment of a graph $G = (V, E)$ is a function L that assigns a list $L(u)$ of so-called admissible colors to each $u \in V$. If $L(u) \subseteq \{1, \ldots, k\}$ for $u \in V$, then L is also called a k-list assignment.

- We say that a coloring $c: V \rightarrow \{1, 2, \ldots\}$ respects L if $c(u) \in L(u)$ for all $u \in V$.

Outline of algorithm

Phase 1.

Determine in polynomial time a polynomial-bounded set $\mathcal{L}$ of list assignments for a $(P_2 + P_3)$-free graph G that have the following two properties.

- The graph G has a 4-coloring if and only if G has a coloring that respects at least one list assignment in $\mathcal{L}$.
- Every list assignment in $\mathcal{L}$ is a good list assignment, i.e., we either have that all its lists have size at most two or else that the union of its lists that contain at least 2 colors has size 3.

Phase 2.

Process every list assignment $L \in \mathcal{L}$.

Remark. Due to a result of Broersma, Fomin, Golovach and Paulusma [2009], this can be solved in polynomial time for P_6-free graph, and therefore also for $(P_2 + P_3)$-free graphs.

Terminology

Let $G = (V, E)$ be a graph.

- For a subset $U \subseteq V$ we define $N_G(U) = \{v \in V \setminus U \mid uv \in E \text{ for some } u \in U\}$.
- A set $D \subseteq V$ dominates a set $S \subseteq V$ if $S \subseteq D \cup N_G(D)$; if $S = V$ then we say that D is a dominating set of G.
- We write $G[U]$ to denote the subgraph of G induced by the vertices in U.
- If we say that we "color the vertices of a set U according to their lists", then we mean that we assign every vertex $u \in U$ a color that is in the list of u, and moreover, such that two adjacent vertices in U do not get the same color. Afterwards, for every $u \in U$, we remove the color of u from the list of every neighbor of u in $N_G(U)$. This is what we call updating the list assignment.

Algorithm

Input: A $(P_2 + P_3)$-free graph G.

We may assume that G has minimum degree at least 4. Otherwise we remove a vertex of degree at most 3 until G has no such vertices anymore. The new graph has a 4-coloring if and only if G has.

Step 1.

Check if G has a dominating set of size at most 39. If such a set does not exist, then return `No`. Otherwise, let D be such a dominating set.

Remark. If G has a 4-coloring, then G contains a dominating set D of size at most 39.

Step 2.

Check if $G[D]$ is 4-colorable.

If not, then return `No`.

Otherwise set $\mathcal{L} = \emptyset$.

For every 4-coloring of $G[D]$, perform the following steps.

Step 3.

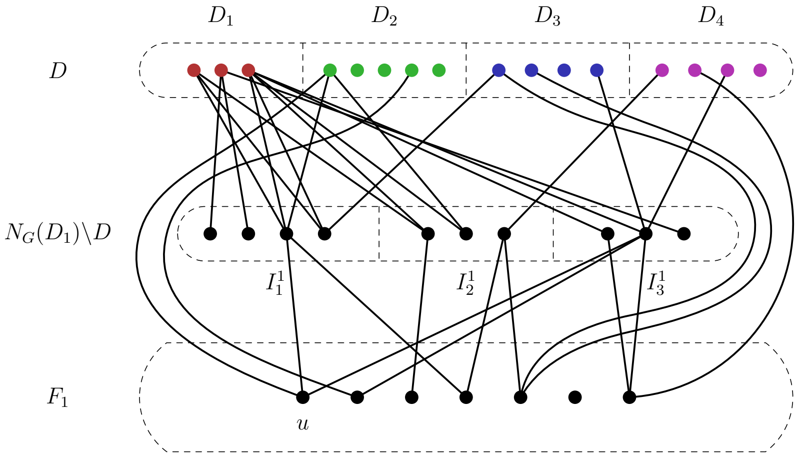
Update the list assignment.

Since D is a dominating set, now each list has size at most 3.

The union of the lists may have size 4. Hence the existing results may not apply here.

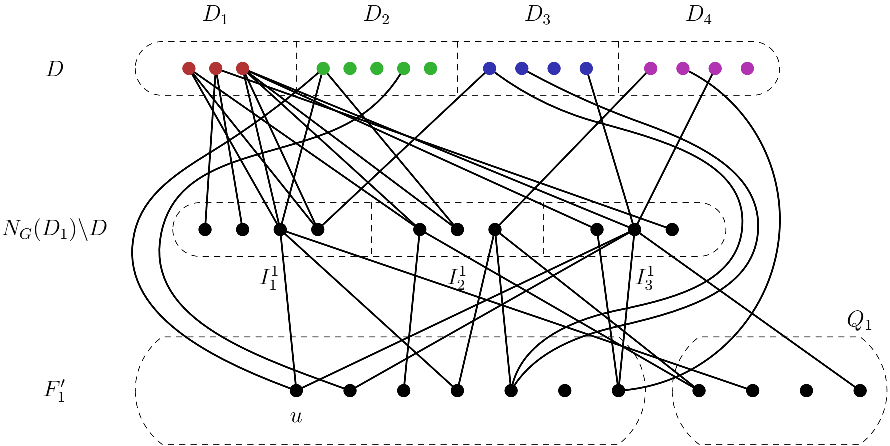
- For $i = 1, \ldots, 4$, let $D_i \subseteq D$ be the subset of vertices with color i, and let $F_i = G[V \setminus (D \cup N_G(D_i)]$.

- Check whether $N_G(D_i) \setminus D$ can be partitioned into three independent sets I_1^i, I_2^i, I_3^i for each i. If not, stop considering this 4-coloring of D.



Step 4.

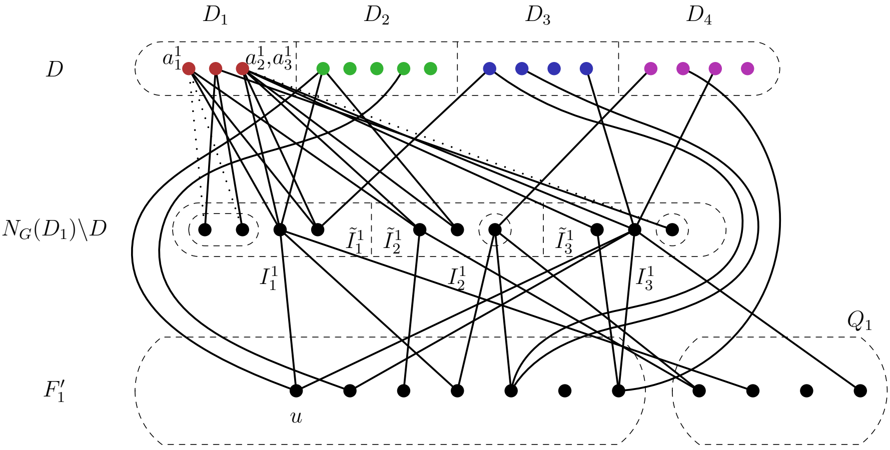
For $i = 1, \ldots, 4$, determine the set Q_i of isolated vertices of F_i, i.e., that have no neighbors in F_i. Let F_i' be the graph obtained from F_i by removing all vertices of Q_i.



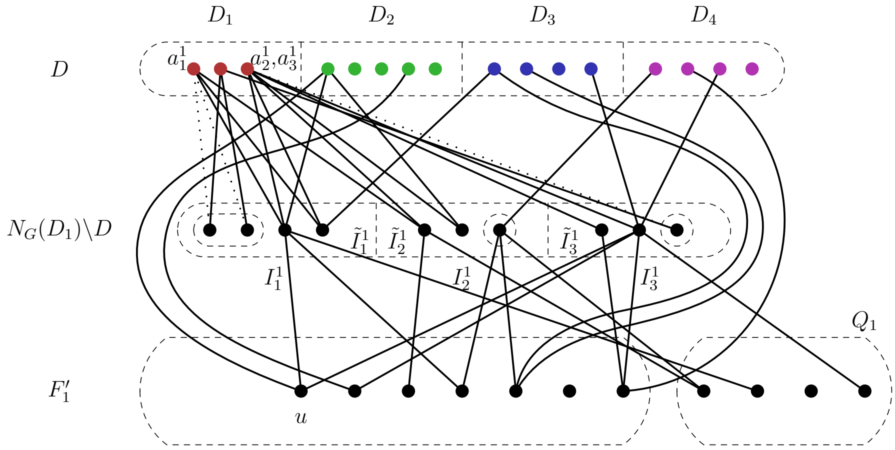
Step 5.

For $i = 1, \ldots, 4$ and $j = 1, \ldots, 3$ do as follows. Find a vertex $a_j^i \in D_i$ that has the maximum number of neighbors in I_j^i over all vertices in D_i.

Define $\tilde{I}_j^i = I_j^i \cap N_G(a_j^i)$ and $I^i = I_1^i \cup I_2^i \cup I_3^i \setminus (\tilde{I}_1^i \cup \tilde{I}_2^i \cup \tilde{I}_3^i)$.



Remark. For $i = 1, \ldots, 4$, the number of vertices of I^i is at most 114.



Step 6.

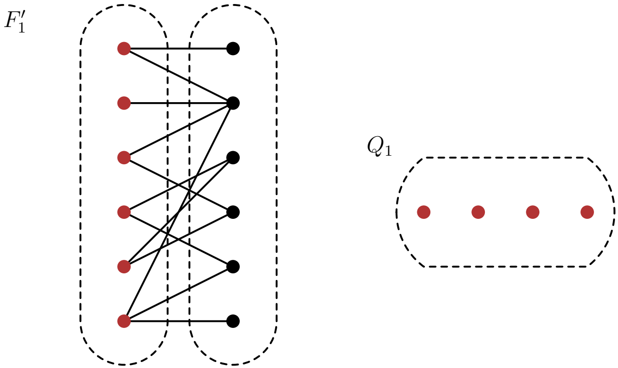
For $i = 1, \dots, 4$ do as follows.

6a. If F_i' is connected and bipartite:

Give all the vertices of one partition class color i. Consider both possibilities.

Color the vertices of Q_i with color i. Update the list assignment.

The resulting list assignment is good, and we put it in $\mathcal{L}$.



6b. If F_i' is disconnected:

Due to the $(P_2 + P_3)$-freeness, there are edges only in F_i'.

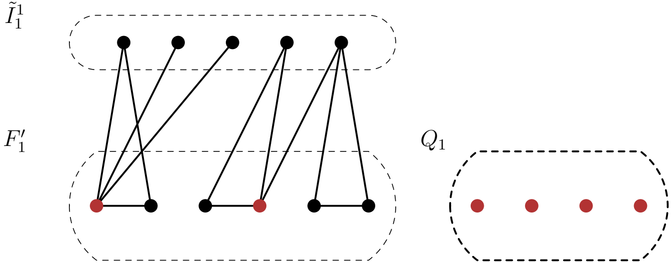
For every j, consider every edge in F_i'.

If at least one of the end-vertices is adjacent to all but at most three vertices of $\tilde{I}_j^i$, then color such a vertex with color i.

Otherwise, we color both end-vertices according to their lists. We consider all possible colorings of these end-vertices.

Color the vertices of Q_i with color i. Update the list assignment.

The resulting list assignment is good, and we put it in $\mathcal{L}$.



Step 7.

For $i = 1, \ldots, 4$, do as follows.

If F_i' is connected, then choose an edge $e^i = u^i v^i$ of F_i'.

If F_i' is disconnected, then choose for all $1 \leq j \leq 3$ a vertex u_j^i that is adjacent to all but at most three vertices in $\tilde{I}_j^i$.

Let M be a set consisting of:

u^i, v^i for every connected F'_i,

u^i_1, u^i_2, u^i_3 for every disconnected F'_i.

A suitable coloring of $G[M]$:

$c(u^i), c(v^i) \neq i$ for every connected F'_i,

$c(u^i_1), c(u^i_2), c(u^i_3) \neq i$ for every disconnected F'_i.

Color M with a suitable coloring.

If F_i' is connected, then color all vertices in $(\tilde{I}_1^i \cup \tilde{I}_2^i \cup \tilde{I}_3^i) \setminus M$ that are neither adjacent to u^i nor to v^i according to their lists.

If F_i' is disconnected, then color all vertices in $\tilde{I}_j^i \setminus M$ that are not adjacent to u_j^i according to their lists.

Color all remaining uncolored vertices in $I^1 \cup I^2 \cup I^3 \cup I^4$ according to their lists.

The resulting list assignment is good, and we put it in $\mathcal{L}$.

Remark. We branch over all possibilities.

Future work

Classifying the computational complexity of 4-Coloring for H-free graphs is not complete. Some borderline cases are:

- Is 4-Coloring polynomially solvable for $(P_1 + P_2 + P_3)$-free graphs?
- Is 4-Coloring polynomially solvable for $2P_3$-free graphs?

The computational complexity of 5-Coloring for H-free graphs is still widely open. A borderline case is:

- Is 5-Coloring polynomially solvable for $(P_2 + P_3)$-free graphs?

Thank you!