Finding minimal obstructions to graph colouring

Maria Chudnovsky   Jan Goedgebeur*

Oliver Schaudt   Mingxian Zhong

* Combinatorial Algorithms and Algorithmic Graph Theory
Department of Applied Mathematics and Computer Science
Ghent University, Belgium

Graph colouring

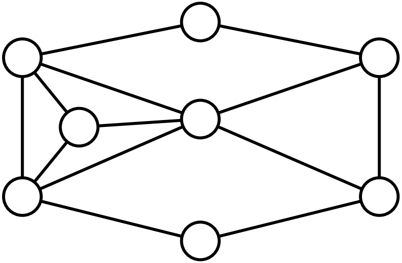
Definition (*k*-colouring)

A **k-(vertex)-colouring** of a graph G is an assignment of colours $\{1, 2, ..., k\}$ to the vertices of G such that any two adjacent vertices receive a distinct colour.

Graph colouring

Definition (k-colouring)

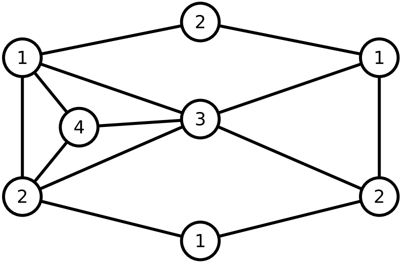
A **k-(vertex)-colouring** of a graph G is an assignment of colours $\{1, 2, ..., k\}$ to the vertices of G such that any two adjacent vertices receive a distinct colour.



Graph colouring

Definition (*k*-colouring)

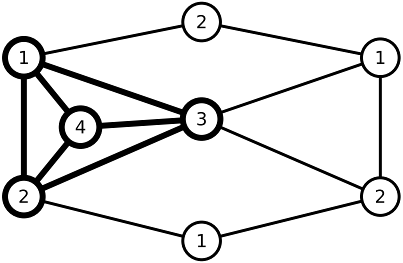
A **k-(vertex)-colouring** of a graph G is an assignment of colours $\{1, 2, ..., k\}$ to the vertices of G such that any two adjacent vertices receive a distinct colour.



Graph colouring

Definition (k-colouring)

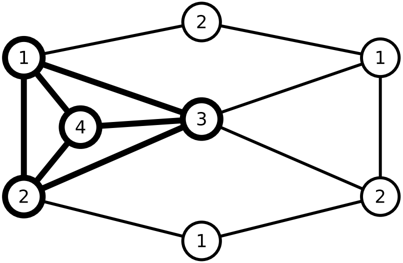
A **k-(vertex)-colouring** of a graph G is an assignment of colours $\{1, 2, ..., k\}$ to the vertices of G such that any two adjacent vertices receive a distinct colour.



Graph colouring

Definition (*k*-colouring)

A **k-(vertex)-colouring** of a graph G is an assignment of colours $\{1, 2, ..., k\}$ to the vertices of G such that any two adjacent vertices receive a distinct colour.



- We call K_4 an **obstruction** for 3-colourability.

Complexity of graph colouring

- The problem of deciding if a graph is k-colourable is called the **k-colourability problem**.
- This is a very hard problem:
  - It is NP-complete.
  - Even if k is fixed (for every $k \geq 3$).

k-colourability in H-free graphs

k-colourability in H-free graphs

- A graph G is **H-free** if it does not contain H as an induced subgraph.

k-colourability in H-free graphs

- A graph G is **H-free** if it does not contain H as an induced subgraph.
- (That is: H cannot be obtained from G by deleting vertices.)

k-colourability in H-free graphs

- A graph G is **H-free** if it does not contain H as an induced subgraph.
- (That is: H cannot be obtained from G by deleting vertices.)

Theorem (Lozin and Kamiński + Král et al., 2007)

Let H be any graph that is not the disjoint union of paths. Then k-colourability is NP-complete in the class of H-free graphs, for all $k \geq 3$.

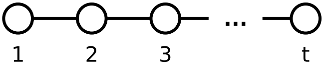
k-colourability in H-free graphs

- A graph G is **H-free** if it does not contain H as an induced subgraph.
- (That is: H cannot be obtained from G by deleting vertices.)

Theorem (Lozin and Kamiński + Král et al., 2007)

Let H be any graph that is not the disjoint union of paths. Then k-colourability is NP-complete in the class of H-free graphs, for all $k \geq 3$.

- This motivates the study of k-colouring of P_t-free graphs.
- P_t is the path on t vertices.



k-colourability in P_t-free graphs

Definition (k-critical H-free graph)

A graph G is **k-critical H-free** if G is H-free and has $\chi(G) = k$, but every H-free proper subgraph of G is $(k-1)$-colourable.

Definition (k-critical H-free graph)

A graph G is **k-critical H-free** if G is H-free and has $\chi(G) = k$, but every H-free proper subgraph of G is $(k-1)$-colourable.

Our goal:

- Determine finite sets S of all $(k+1)$-critical P_t-free graphs (at least if such a finite set exists).

Definition (k-critical H-free graph)

A graph G is **k-critical H-free** if G is H-free and has $\chi(G) = k$, but every H-free proper subgraph of G is $(k-1)$-colourable.

Our goal:

- Determine finite sets S of all $(k+1)$-critical P_t-free graphs (at least if such a finite set exists).
- This means: a P_t-free graph is k-colourable $\iff$ it does not contain a graph from S as subgraph.

Why is this useful?

Why is this useful?

- If S is finite this gives a polynomial algorithm to test if a P_t-free graph is k-colourable.

Why is this useful?

- If S is finite this gives a polynomial algorithm to test if a P_t-free graph is k-colourable.
- Moreover it provides a **no-certificate** for k-colourability.
  - Cf. Kuratowski graphs as no-certificate for planarity.

k-colourability in P_t-free graphs

Theorem (Bruce, Hoàng and Sawada, 2009)

There are six 4-critical P_5-free graphs.

k-colourability in P_t-free graphs

Theorem (Bruce, Hoàng and Sawada, 2009)

There are six 4-critical P_5-free graphs.

Theorem (Randerath and Schiermeyer, 2004)

There is one 4-critical (P_6, C_3)-free graph.

Theorem (Hell and Huang, 2013)

There are four 4-critical (P_6, C_4)-free graphs.

k-colourability in P_t-free graphs

Theorem (Bruce, Hoàng and Sawada, 2009)

There are six 4-critical P_5-free graphs.

Theorem (Randerath and Schiermeyer, 2004)

There is one 4-critical (P_6, C_3)-free graph.

Theorem (Hell and Huang, 2013)

There are four 4-critical (P_6, C_4)-free graphs.

- **Golovach et al., 2014:** is there a certifying algorithm for 3-colourablity of P_6-free graphs?

k-colourability in P_t-free graphs

Theorem (Bruce, Hoàng and Sawada, 2009)

There are six 4-critical P_5-free graphs.

Theorem (Randerath and Schiermeyer, 2004)

There is one 4-critical (P_6, C_3)-free graph.

Theorem (Hell and Huang, 2013)

There are four 4-critical (P_6, C_4)-free graphs.

- **Golovach et al., 2014:** is there a certifying algorithm for 3-colourablity of P_6-free graphs?
- **Seymour, 2014:** for which connected graphs H is the set of 4-critical H-free graphs finite?

How to solve the P_6-free case?

How to solve the P_6-free case?

Note:

- Probably not feasible to solve the P_6-free case by hand...
  - E.g.: proof of the P_5-free case was more than 10 pages.

How to solve the P_6-free case?

Note:

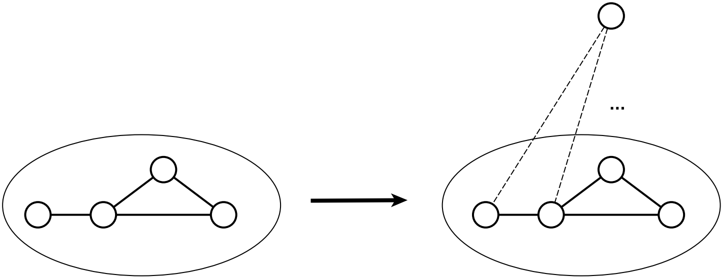
- Probably not feasible to solve the P_6-free case by hand...
  - E.g.: proof of the P_5-free case was more than 10 pages.

Our approach:

- Computer approach through graph generation:
  - Design generation algorithm for k-critical P_t-free graphs.

Basic construction operation

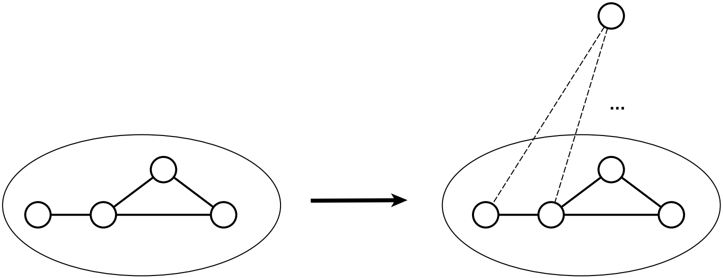
The following operation can be used to construct any simple graph:



- Create a new vertex and connect it to the vertices of the parent graph in all possible ways.

Basic construction operation

The following operation can be used to construct any simple graph:



- Create a new vertex and connect it to the vertices of the parent graph in all possible ways.
- Note: all graphs constructed from *G* contain *G* as induced subgraph!

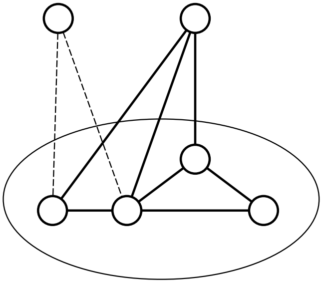
Properties of k-critical graphs

Theorem (Folklore)

Critical graphs do not contain similar vertices.

Definition (Similar vertices)

Similar vertices are vertices $v, w \in V(G) : N(v) \subseteq N(w)$.



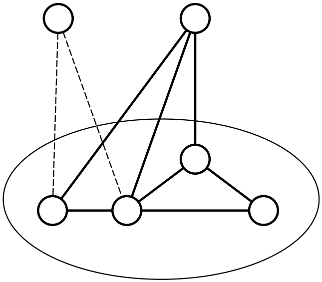
Properties of k-critical graphs

Theorem (Folklore)

Critical graphs do not contain similar vertices.

Definition (Similar vertices)

Similar vertices are vertices $v, w \in V(G) : N(v) \subseteq N(w)$.



- Note: the above theorem can also be generalised to *similar* edges, triangles, P_3's, C_4's etc.

Properties of k-critical graphs

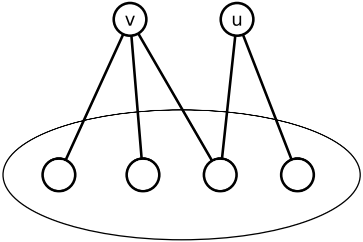
Theorem (Hidden edge)

If a graph G contains two non-adjacent vertices u, v such that for all neighbours x of u it holds that v cannot receive the same colour as x in any $(k-1)$-colouring of $G-u$, then G cannot be k-critical.

Properties of k-critical graphs

Theorem (Hidden edge)

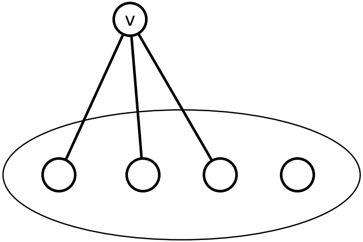
If a graph G contains two non-adjacent vertices u, v such that for all neighbours x of u it holds that v cannot receive the same colour as x in any $(k-1)$-colouring of $G - u$, then G cannot be k-critical.



Properties of k-critical graphs

Theorem (Hidden edge)

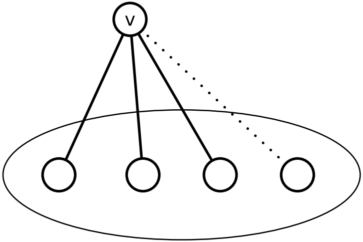
If a graph G contains two non-adjacent vertices u, v such that for all neighbours x of u it holds that v cannot receive the same colour as x in any $(k-1)$-colouring of $G - u$, then G cannot be k-critical.



Properties of k-critical graphs

Theorem (Hidden edge)

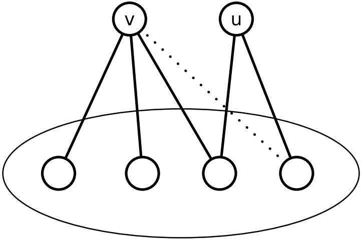
If a graph G contains two non-adjacent vertices u, v such that for all neighbours x of u it holds that v cannot receive the same colour as x in any $(k-1)$-colouring of $G - u$, then G cannot be k-critical.



Properties of k-critical graphs

Theorem (Hidden edge)

If a graph G contains two non-adjacent vertices u, v such that for all neighbours x of u it holds that v cannot receive the same colour as x in any $(k-1)$-colouring of $G-u$, then G cannot be k-critical.



Specialised construction algorithm

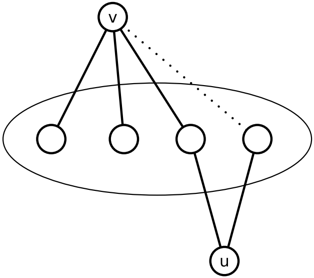
if Graph is P_t-free AND not generated before [nauty] **then**
  if Graph is not $(k-1)$-colourable **then**
    if Graph is k-critical **then**
      Output graph
    end if
  else
    // Apply expansions: see next slide...
  end if
end if

Specialised construction algorithm

```
if Graph is P_t-free AND not generated before [nauty] then
    if Graph is not (k − 1)-colourable then
        if Graph is k-critical then
            Output graph
        end if
    else
        // Apply expansions:
        if Graph contains similar vertices then
            Destroy the *best* similar vertex in all possible ways
        else if Graph contains similar vertices including hidden edges then
            Destroy the *best* similar vertex in all possible ways
        else if Graph contains similar edges then
            Destroy the *best* similar edge in all possible ways
        else if Graph contains ... (i.e. try to apply other lemmas) then
            Destroy ... in all possible ways
        else
            Apply all possible expansions
        end if
    end if
end if
```

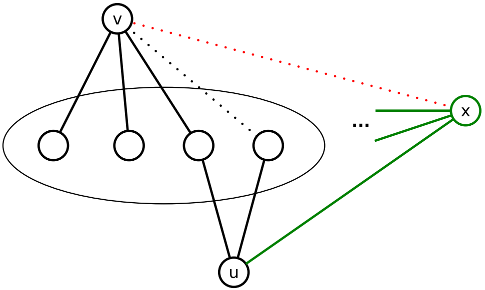
Destroying similar vertices

How to destroy similar vertices u, v of G (with $N(u) \subseteq N(v)$)?



Destroying similar vertices

How to destroy similar vertices u, v of G (with $N(u) \subseteq N(v)$)?



- Connect new vertex x to u but not to v (in all possible ways).
- Make sure there is no hidden edge between x and v in $G - u + x$ (helps a lot in most cases!).

Results – 4-critical P_6-free

Results – 4-critical P_6-free

- We generated all 4-critical P_6-free graphs up to 28 vertices (this took approximately 9 CPU years)...
- ... but unfortunately the number of (colourable) graphs generated keeps growing. ☹

Results – 4-critical P_6-free

- We generated all 4-critical P_6-free graphs up to 28 vertices (this took approximately 9 CPU years)...
- ... but unfortunately the number of (colourable) graphs generated keeps growing. ☹

Theorem

There are 24 4-critical P_6-free graphs with at most 28 vertices.

```
 1 graphs : n=4
 1 graphs : n=6
 2 graphs : n=7
 3 graphs : n=8
 4 graphs : n=9
 6 graphs : n=10
 2 graphs : n=11
 1 graphs : n=12
 3 graphs : n=13
 1 graphs : n=16
24 graphs altogether;
```

Results – 4-critical P_6-free

- We generated all 4-critical P_6-free graphs up to 28 vertices (this took approximately 9 CPU years)...
- ... but unfortunately the number of (colourable) graphs generated keeps growing. ☹

Theorem

There are 24 4-critical P_6-free graphs with at most 28 vertices.

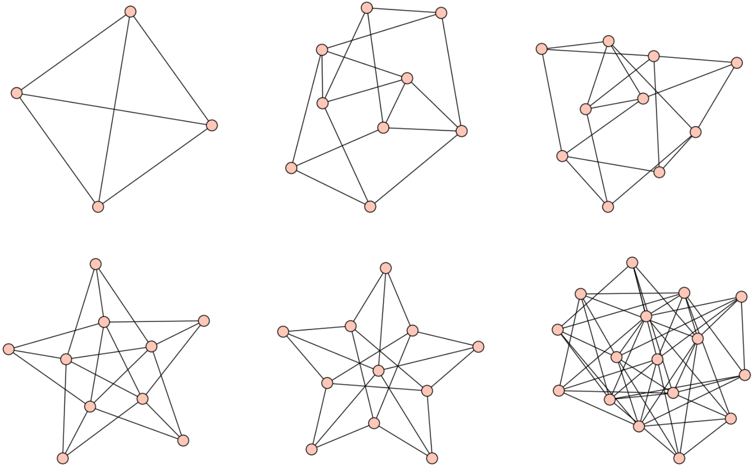
```
 1 graphs : n=4
 1 graphs : n=6
 2 graphs : n=7
 3 graphs : n=8
 4 graphs : n=9
 6 graphs : n=10
 2 graphs : n=11
 1 graphs : n=12
 3 graphs : n=13
 1 graphs : n=16
24 graphs altogether;
```

So it is VERY likely that these are the only 4-critical P_6-free graphs...

Results – 4-critical (P_6, diamond)-free

Theorem

There are 6 4-critical (P_6, diamond)-free graphs.



Results – 4-critical P_6-free

By combining:

- Knowledge of all 4-critical (P_6, diamond)-free graphs (computer-aided).
- Knowledge of all 4-critical P_6-free graphs up to 28 vertices (computer-aided).
- Extensive structural analysis by hand (see arXiv paper for details).

Proven that:

Theorem

There are 24 4-critical P_6-free graphs.

Results – 4-critical P_6-free

By combining:

- Knowledge of all 4-critical (P_6, diamond)-free graphs (computer-aided).
- Knowledge of all 4-critical P_6-free graphs up to 28 vertices (computer-aided).
- Extensive structural analysis by hand (see arXiv paper for details).

Proven that:

Theorem

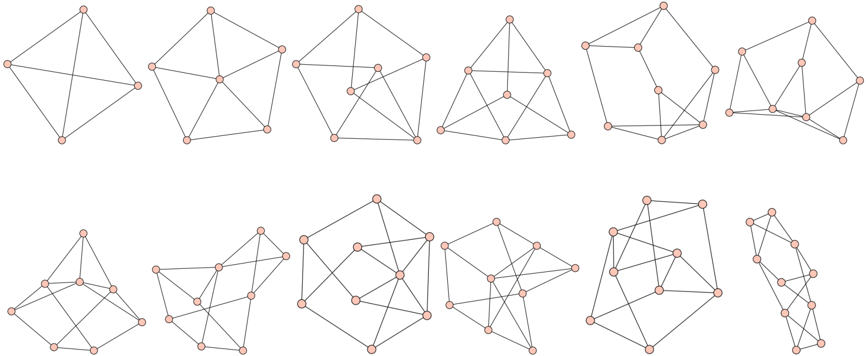
There are 24 4-critical P_6-free graphs.

Moreover, given a connected graph H, there are finitely many 4-critical H-free graphs if and only if H is a subgraph of P_6.

Results – 4-critical P_6-free: graphs 1 to 12

Theorem

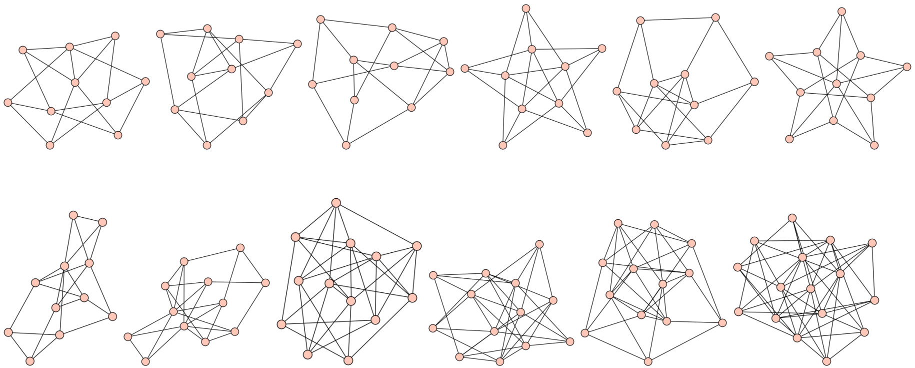
There are 24 4-critical P_6-free graphs.



[Can be downloaded from the House of Graphs (`http://hog.grinvin.org`)]

Theorem

There are 24 4-critical P_6-free graphs.



[Can be downloaded from the House of Graphs (`http://hog.grinvin.org`)]

Next steps

Next steps

Theorem (Dichotomy)

Let H be a connected graph. There are finitely many 4-critical H-free graphs if and only if H is a subgraph of P_6.

Next steps

Theorem (Dichotomy)

Let H be a connected graph. There are finitely many 4-critical H-free graphs if and only if H is a subgraph of P_6.

- What happens for disconnected H?

Next steps

Theorem (Dichotomy)

Let H be a connected graph. There are finitely many 4-critical H-free graphs if and only if H is a subgraph of P_6.

- What happens for disconnected H?
- Initial experiments indicate that $H = P_3 + P_3$ will be the most difficult case.

Next steps

Theorem (Dichotomy)

Let H be a connected graph. There are finitely many 4-critical H-free graphs if and only if H is a subgraph of P_6.

- What happens for disconnected H?
- Initial experiments indicate that $H = P_3 + P_3$ will be the most difficult case.

Other ongoing work:

- Obstructions for list k-colourability.
- Each vertex v has a list $L(v) \subseteq \{1, ..., k\}$ with available colours.

Thanks for your attention!