Reconstructing Graphs from their Recolouring Graphs

Postgraduate Combinatorial Conference 2025

Gaétan Berthe[1], Caroline Brosse[2], Brian Hearn[3], Jan van den Heuvel[3], Pierre Hoppenot[4], Théo Pierron[5]

[1]LIRMM, Université de Montpellier, CNRS, Montpellier, France

[2]Université d'Orléans, INSA CVL, LIFO, UR 4022, Orléans, France

[3]Department of Mathematics, London School of Economics & Political Science, Houghton Street, London WC2A 2AE, United Kingdom

[4]Univ. Grenoble Alpes, CNRS, Grenoble INP, G-SCOP, 38000 Grenoble, France

[5]Univ. Lyon, UCBL, CNRS, INSA Lyon, LIRIS, UMR5205, F-69622 Villeurbanne, France

1 May 2025

What is a k-colouring?

Definition

A k-colouring of a graph G is an assignment of **at most** k colours to the vertices of G, where no two adjacent vertices are assigned the same colour.

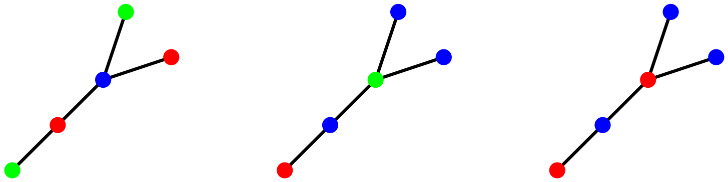


Figure: Three 3-colourings of a graph.

What is a recolouring graph?

Definition

Let G be a graph. Let $k \in \mathbb{N}$.

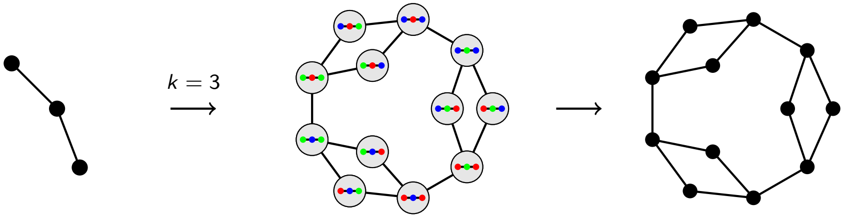
The k-recolouring graph $\mathcal{C}_k(G)$ of G is the graph whose vertex set is the set of k-colourings of G, with an edge between two k-colourings if and only if they differ at exactly one vertex.

What is a recolouring graph?

Definition

Let G be a graph. Let $k \in \mathbb{N}$.

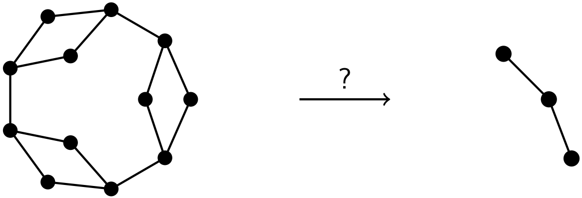
The k-recolouring graph $\mathcal{C}_k(G)$ of G is the graph whose vertex set is the set of k-colourings of G, with an edge between two k-colourings if and only if they differ at exactly one vertex.



Can we reconstruct G from its recolouring graphs?

Question

Given a non-empty recolouring graph $\mathcal{C}_k(G)$ (where k and G are unknown), can we reconstruct G?



Recent history

Conjecture (Asgarli, Krehbiel, Levinson, Russell (2024))

Every graph G is uniquely determined by the collection $\{\mathcal{C}_k(G)\}_{k\in\mathbb{N}}$.

Recent history

Conjecture (Asgarli, Krehbiel, Levinson, Russell (2024))

Every graph G is uniquely determined by the collection $\{\mathcal{C}_k(G)\}_{k\in\mathbb{N}}$.

Conjecture (Asgarli, Krehbiel, Levinson, Russell (2024))

Every graph G is uniquely determined by some finite subcollection of $\{\mathcal{C}_k(G)\}_{k\in\mathbb{N}}$.

Recent history

Conjecture (Asgarli, Krehbiel, Levinson, Russell (2024))

Every graph G is uniquely determined by the collection $\{\mathcal{C}_k(G)\}_{k\in\mathbb{N}}$.

Conjecture (Asgarli, Krehbiel, Levinson, Russell (2024))

Every graph G is uniquely determined by some finite subcollection of $\{\mathcal{C}_k(G)\}_{k\in\mathbb{N}}$.

Theorem (Hogan, Scott, Tamitegama, Tan (2024))

Let G be a graph on n vertices. Given $k > 5n^2$, we can determine G from $\mathcal{C}_k(G)$.

Our result

Theorem

Let G be a graph. We can determine G from $\mathcal{C}_k(G)$ for any $k > \chi(G)$.

Recall: the chromatic number $\chi(G)$ of G is the minimum number of colours needed for a proper colouring of G.

Our result

Theorem

Let G be a graph. We can determine G from $\mathcal{C}_k(G)$ for any $k > \chi(G)$.

Recall: the chromatic number $\chi(G)$ of G is the minimum number of colours needed for a proper colouring of G.

Observation

$\mathcal{C}_k(G)$ is empty (has no vertices) if $k < \chi(G)$.

Our result

Theorem

Let G be a graph. We can determine G from $\mathcal{C}_k(G)$ for any $k > \chi(G)$.

Recall: the chromatic number $\chi(G)$ of G is the minimum number of colours needed for a proper colouring of G.

Observation

$\mathcal{C}_k(G)$ is empty (has no vertices) if $k < \chi(G)$.

Remark

The proof is based on ideas from the Hogan, Scott, Tamitegama, and Tan proof.

Detecting vertices

Observations

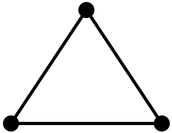
Each **edge** *of* $\mathcal{C}_k(G)$ *corresponds to a* **vertex** *of* G.

Detecting vertices

Observations

Each **edge** *of* $\mathcal{C}_k(G)$ *corresponds to a* **vertex** *of* G.

Each **clique** *in* $\mathcal{C}_k(G)$ *corresponds to a* **vertex** *of* G.



Detecting vertices

Fix some colouring $c \in \mathcal{C}_k(G)$.

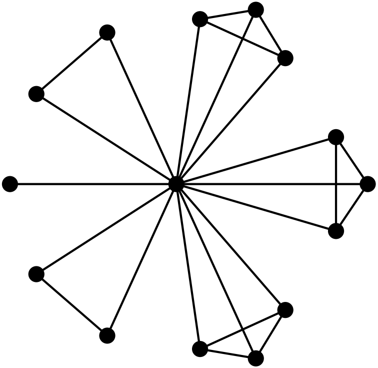


Figure: A colouring $c \in \mathcal{C}_k(G)$ and the maximal cliques containing c.

Detecting vertices

Fix some colouring $c \in \mathcal{C}_k(G)$.

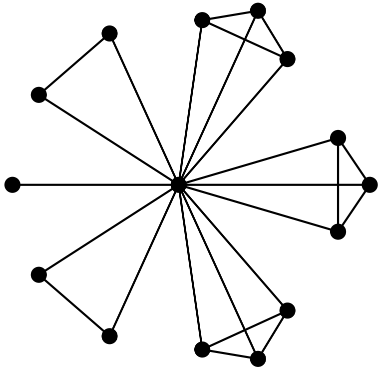


Figure: A colouring $c \in \mathcal{C}_k(G)$ and the maximal cliques containing c.

Observations

Each clique containing c corresponds to some vertex of G.

Detecting vertices

Fix some colouring $c \in \mathcal{C}_k(G)$.

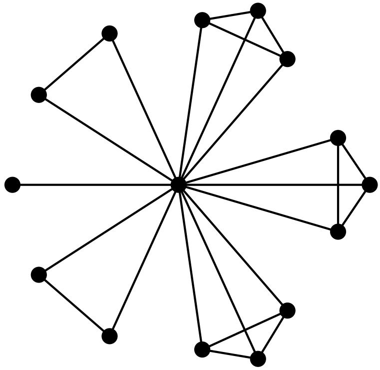


Figure: A colouring $c \in \mathcal{C}_k(G)$ and the maximal cliques containing c.

Observations

Each clique containing c corresponds to some vertex of G.

Different *cliques correspond to* **different** *vertices of G.*

Detecting vertices

Fix some colouring $c \in \mathcal{C}_k(G)$.

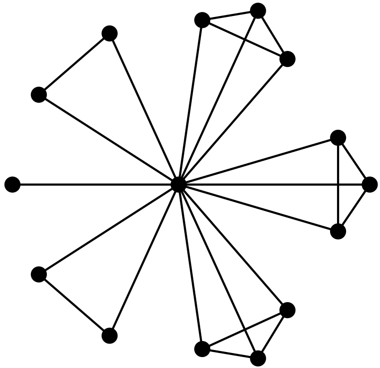


Figure: A colouring $c \in \mathcal{C}_k(G)$ and the maximal cliques containing c.

Observations

Each clique containing c corresponds to some vertex of G.

Different *cliques correspond to* **different** *vertices of G.*

So every colouring $c \in \mathcal{C}_k(G)$ is in **at most** *n cliques.*

(Where n denotes the number of vertices of G.)

Detecting vertices

Observations

Each clique containing c corresponds to a vertex which can be 'recoloured'.

Detecting vertices

Observations

Each clique containing c corresponds to a vertex which can be 'recoloured'.

So any colouring which doesn't use every colour is in **exactly** *n cliques.*

Detecting vertices

Observations

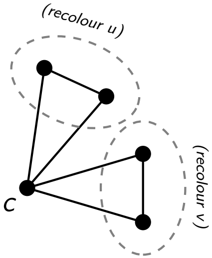
Each clique containing c corresponds to a vertex which can be 'recoloured'.

So any colouring which doesn't use every colour is in **exactly** *n cliques.*

So if $k > \chi(G)$, we can determine n from $\mathcal{C}_k(G)$.

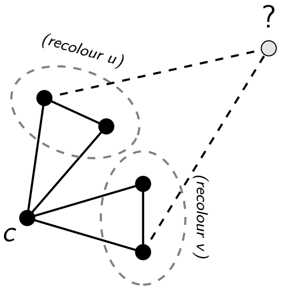
What about edges?

Fix $c \in \mathcal{C}_k(G)$ and focus on two cliques containing c, corresponding to two vertices u and v.



What about edges?

Fix $c \in \mathcal{C}_k(G)$ and focus on two cliques containing c, corresponding to two vertices u and v.



Then look at **common neighbours** of these cliques.

What about edges?

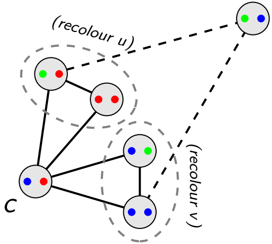


Figure: If uv is **not** an edge, then every pair closes to a 4-cycle.

What about edges?

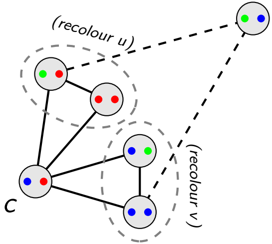


Figure: If uv is **not** an edge, then every pair closes to a 4-cycle.

Main idea of the proof

For each $c \in \mathcal{C}_k(G)$, build a **candidate graph** *G_c as follows:*

- *Look at the cliques $C_{v_a}, \ldots, C_{v_p}$ containing c.*
- *Add an edge uv whenever we find a pair $c_u \in C_u$ and $c_v \in C_v$ which does* **not** *close to a 4-cycle.*

What about edges?

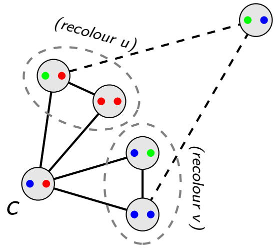


Figure: If uv is **not** an edge, then every pair closes to a 4-cycle.

Main idea of the proof

For each $c \in \mathcal{C}_k(G)$, build a **candidate graph** *G_c as follows:*

- *Look at the cliques $C_{v_a}, \ldots, C_{v_p}$ containing c.*
- *Add an edge uv whenever we find a pair $c_u \in C_u$ and $c_v \in C_v$ which does* **not** *close to a 4-cycle.*

Each candidate graph G_c is a **subgraph** *of G.*

What about edges?

If there is a colour **not** used by c, then the converse also holds!

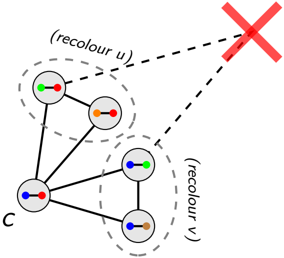


Figure: In this case: if uv is an edge, then some pairs do **not** close to a 4-cycle.

What about edges?

If there is a colour **not** used by c, then the converse also holds!

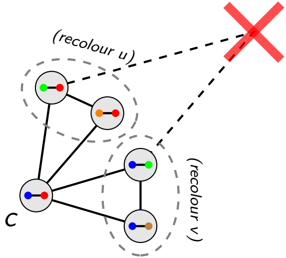


Figure: In this case: if uv is an edge, then some pairs do **not** close to a 4-cycle.

Observations

In this case, G_c is **isomorphic** *to G.*

What about edges?

If there is a colour **not** used by c, then the converse also holds!

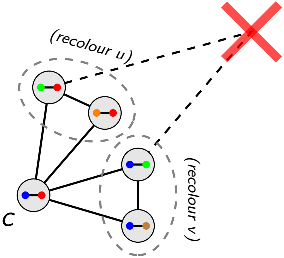


Figure: In this case: if uv is an edge, then some pairs do **not** close to a 4-cycle.

Observations

In this case, G_c *is* **isomorphic** *to* G.

Hence if $k > \chi(G)$*, any candidate with a maximal number of vertices and edges is isomorphic to* G*. (This completes the proof.)*

What if $k = \chi(G)$?

Observation

If $k = \chi(G)$, it is **not possible** *in general to determine G from $\mathcal{C}_k(G)$.*

What if $k = \chi(G)$?

Observation

If $k = \chi(G)$, it is **not possible** *in general to determine G from $\mathcal{C}_k(G)$.*

Examples

Any connected bipartite graph has the same 2-recolouring graph.

Any k-tree has the same $(k+1)$-recolouring graph.

...and so on.