Further Evidence Towards the Multiplicative 1-2-3 Conjecture

Julien Bensmail[1], Hervé Hocquard[2], Dimitri Lajou[2], Éric Sopena[2]

[1]Université Côte d'Azur, Nice.

[2]LaBRI, Bordeaux.

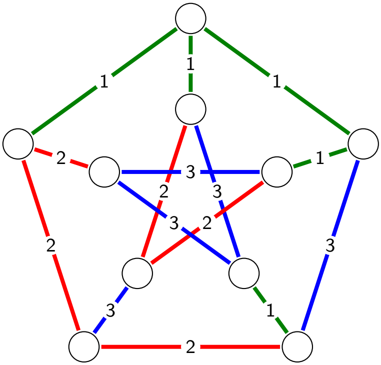
JGA 2020
16-18 novembre 2020
arXiv:2004.09090

Overview

1. The 1-2-3 Conjecture
   - Definitions
   - Variations
2. Our main results
   - Overview
   - Proof for 4-chromatic nice graphs
   - Almost p-proper labellings for all graphs
3. Concluding remarks
   - A new conjecture

Overview

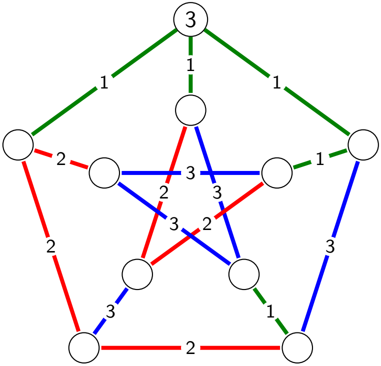
3-Edge-labelling and coloring



Definition

A k-edge-labelling is a function $\ell : E(G) \to \{1, 2, \ldots, k\}$.

3-Edge-labelling and coloring

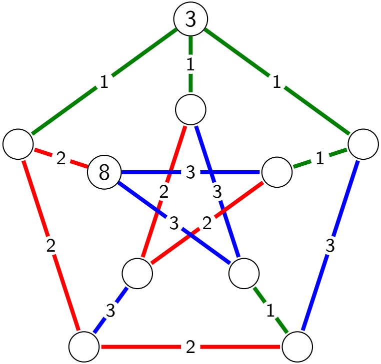


Definition

A k-edge-labelling is a function $\ell : E(G) \to \{1, 2, \ldots, k\}$.

$$\sigma_\ell(u) = \sum_{uv \in E(G)} \ell(uv)$$

3-Edge-labelling and coloring

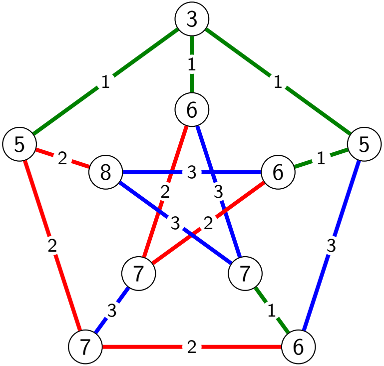


Definition

A k-edge-labelling is a function $\ell : E(G) \to \{1, 2, \ldots, k\}$.

$$\sigma_\ell(u) = \sum_{uv \in E(G)} \ell(uv)$$

3-Edge-labelling and coloring

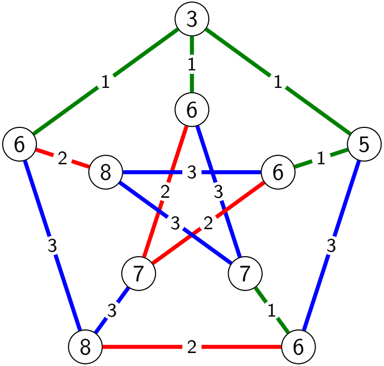


Definition

A k-edge-labelling is a function $\ell : E(G) \to \{1, 2, \ldots, k\}$.

$$\sigma_\ell(u) = \sum_{uv \in E(G)} \ell(uv)$$

3-Edge-labelling and coloring



Definition

A k-edge-labelling is a function $\ell : E(G) \to \{1, 2, \ldots, k\}$.

$$\sigma_\ell(u) = \sum_{uv \in E(G)} \ell(uv)$$

The function σ_ℓ is a proper coloring of G.

1-2-3-Conjecture

Definition

An *s*-**proper** labelling ℓ of a graph G is a labelling such that $\sigma_\ell = \sum_{uv \in E(G)} \ell(uv)$ is a proper coloring.

A *nice* graph is a graph with no connected component isomorphic to K_2.

Conjecture (1-2-3-Conjecture, Karonski, Luczak, and Thomason 2004)

*Every nice graph G admits an s-***proper** *3-labelling.*

Overview of some results

Theorem (Kalkowski, Karoński and Pfender 2010)

*Every nice graph G admits an s-**proper** 5-**labelling**.*

Theorem

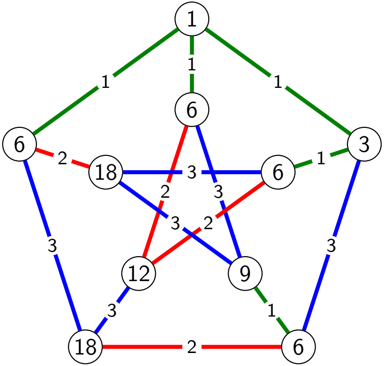
Every nice graph G such that

- *G has **chromatic number at most** 3 (Karoński, Łuczak, Thomason 2004), or*
- *G has **chromatic number at most** 4 and is 4-**edge-connected** (Wu, Zhang, Zhu 2017),*

*admits an s-**proper** 3-labelling.*

Overview

Variations



$$\pi_\ell(u) = \prod_{uv \in E(G)} \ell(uv)$$

The function π_ℓ is a proper coloring of G.

Other 1-2-3-Conjectures

Definition

An ***p*-proper** labelling ℓ of a graph G is a labelling such that $\pi_\ell = \prod_{uv \in E(G)} \ell(uv)$ is a proper coloring.

Conjecture (Multiplicative 1-2-3-Conjecture, J. Skowronek-Kaziów 2012)

*Every nice graph G admits a **p-proper** 3-labelling .*

Overview of some results

Theorem (Skowronek-Kaziów 2012)

Every nice graph G admits a **p-proper** **4-labelling**.

Theorem (Skowronek-Kaziów 2012)

Every nice graph G of **chromatic number at most 3** *admits a* **p-proper** *3-labelling.*

Other 1-2-3-Conjectures

Definition

For a labelling ℓ of a graph G, let μ_ℓ be the function which associates with a vertex u the multiset $\mu_\ell(u)$ containing the labels of the edges incident with u. If μ_ℓ is a proper coloring of G, we say that ℓ is ***m*-proper**.

Observation

If ℓ is s-proper or p-proper then it is m-proper.

Theorem (Vučković 2018)

Every nice graph G admits an ***m*-proper** *3-labelling.*

Overview

Main results

Theorem

Every nice graph G of **chromatic number at most 4** *admits a* p-**proper** *3-labelling.*

Theorem

Every graph G admits a 3-labelling such that:

- *if u and v are adjacent and $\pi_\ell(u) = \pi_\ell(v)$ then $\pi_\ell(u) = \pi_\ell(v) = 1$,*
- *the subgraph of G induced by the vertices u with $\pi_\ell(u) = 1$ contains only connected components with at most two vertices.*

We say that ℓ is **almost** p-**proper**.

Regular graphs (Pathological case)

Theorem

Every nice **regular** *graph G admits a p-***proper** *3-labelling.*

Proof. Let ℓ be an m-proper 3-labelling of G. Suppose

$$\pi_\ell(u) = \pi_\ell(v)$$
$$1^a 2^b 3^c = 1^d 2^e 3^f$$

As 2 and 3 are coprime, $b = e$ and $c = f$.
As G is k-regular, $k = a + b + c = d + e + f$.
Hence $a = d$, a contradiction.

$\square$

Notation

For a 3-labelling ℓ:

- $d_2(u)$, the 2-degree of a vertex u, is the number of 2-labelled edges incident with u,
- $d_3(u)$, the 3-degree of a vertex u, is the number of 3-labelled edges incident with u.

Notation

For a 3-labelling ℓ:

- $d_2(u)$, the 2-degree of a vertex u, is the number of 2-labelled edges incident with u,
- $d_3(u)$, the 3-degree of a vertex u, is the number of 3-labelled edges incident with u.

Note that a 3-labelling ℓ is p-proper if and only if for every pair $\{u, v\}$ of adjacent vertices $d_2(u) \neq d_2(v)$ or $d_3(u) \neq d_3(v)$.

Notation

For a 3-labelling ℓ:

- $d_2(u)$, the 2-degree of a vertex u, is the number of 2-labelled edges incident with u,
- $d_3(u)$, the 3-degree of a vertex u, is the number of 3-labelled edges incident with u.

Note that a 3-labelling ℓ is p-proper if and only if for every pair $\{u, v\}$ of adjacent vertices $d_2(u) \neq d_2(v)$ or $d_3(u) \neq d_3(v)$.
We say that

- u is 1-monochromatic if $\pi_\ell(u) = 1$,
- u is 2-monochromatic if $d_2(u) > 0$ and $d_3(u) = 0$,
- u is 3-monochromatic if $d_2(u) = 0$ and $d_3(u) > 0$, and
- u is bichromatic if $d_2(u) > 0$ and $d_3(u) > 0$.

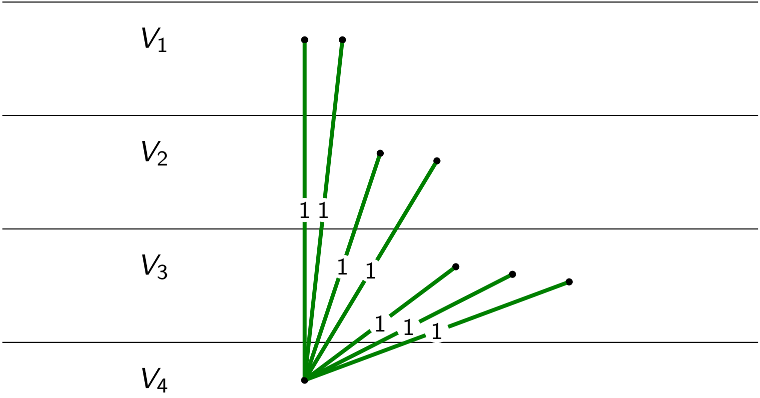
Overview

Step 1: Partition $V(G)$ into four independent sets V_1, V_2, V_3, V_4 such that each $u \in V_i$ has a neighbor in every V_j for $j < i$.

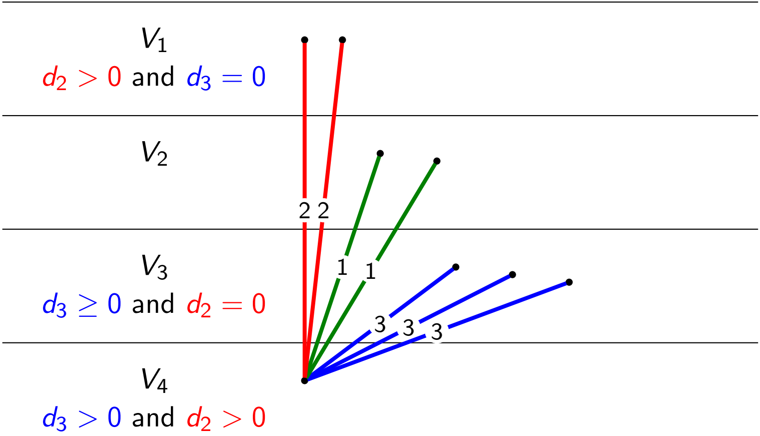
Step 1: Partition $V(G)$ into four independent sets V_1, V_2, V_3, V_4 such that each $u \in V_i$ has a neighbor in every V_j for $j < i$.
Objective: we want to have no conflict between any two V_i's.

Step 1: Partition $V(G)$ into four independent sets V_1, V_2, V_3, V_4 such that each $u \in V_i$ has a neighbor in every V_j for $j < i$.
Objective: we want to have no conflict between any two V_i's.
Step 2: Label edges incident with vertices of V_4 and V_3.

Step 1: Partition $V(G)$ into four independent sets V_1, V_2, V_3, V_4 such that each $u \in V_i$ has a neighbor in every V_j for $j < i$.
Objective: we want to have no conflict between any two V_i's.
Step 2: Label edges incident with vertices of V_4 and V_3.



Step 1: Partition $V(G)$ into four independent sets V_1, V_2, V_3, V_4 such that each $u \in V_i$ has a neighbor in every V_j for $j < i$.
Objective: we want to have no conflict between any two V_i's.
Step 2: Label edges incident with vertices of V_4 and V_3.



Step 1: Partition $V(G)$ into four independent sets V_1, V_2, V_3, V_4 such that each $u \in V_i$ has a neighbor in every V_j for $j < i$.

Objective: we want to have no conflict between any two V_i's.

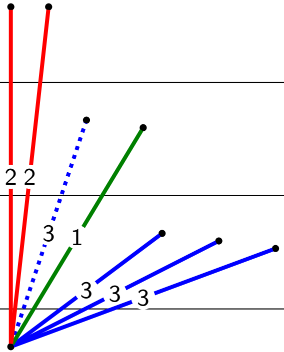
Step 2: Label edges incident with vertices of V_4 and V_3.

V_1
$d_2 > 0$ and $d_3 = 0$

V_2
$d_3 \geq 0$ and $d_2 = 0$

V_3
$d_3 \geq 0$ and $d_2 = 0$

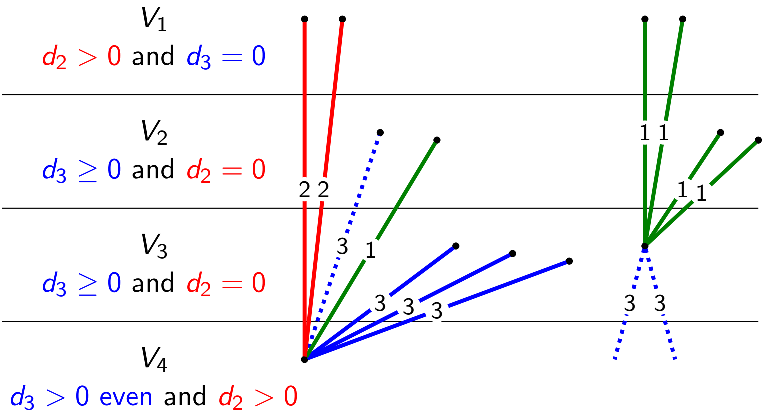
V_4
$d_3 > 0$ even and $d_2 > 0$



Step 1: Partition $V(G)$ into four independent sets V_1, V_2, V_3, V_4 such that each $u \in V_i$ has a neighbor in every V_j for $j < i$.

Objective: we want to have no conflict between any two V_i's.

Step 2: Label edges incident with vertices of V_4 and V_3.



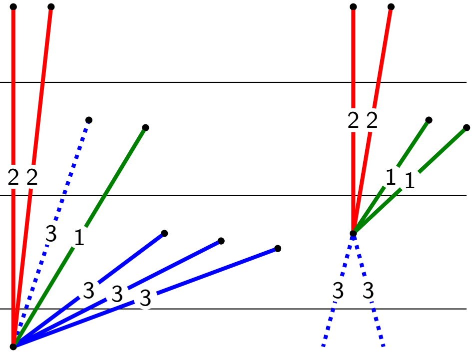
Step 1: Partition $V(G)$ into four independent sets V_1, V_2, V_3, V_4 such that each $u \in V_i$ has a neighbor in every V_j for $j < i$.
Objective: we want to have no conflict between any two V_i's.
Step 2: Label edges incident with vertices of V_4 and V_3.

V_1
$d_2 > 0$ and $d_3 = 0$

V_2
$d_3 \geq 0$ and $d_2 = 0$

V_3
$d_3 \geq 0$ and $d_2 = 0$

V_4
$d_3 > 0$ even and $d_2 > 0$



Step 1: Partition $V(G)$ into four independent sets V_1, V_2, V_3, V_4 such that each $u \in V_i$ has a neighbor in every V_j for $j < i$.

Objective: we want to have no conflict between any two V_i's.

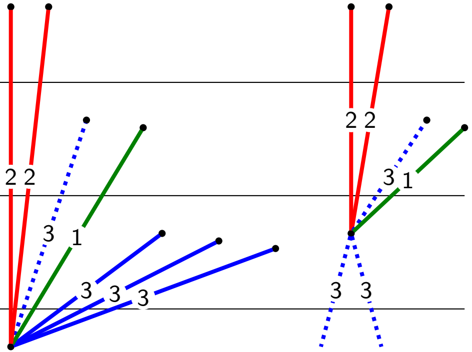
Step 2: Label edges incident with vertices of V_4 and V_3.

V_1
$d_2 > 0$ and $d_3 = 0$

V_2
$d_3 \geq 0$ and $d_2 = 0$

V_3
$d_3 > 0$ odd and $d_2 > 0$

V_4
$d_3 > 0$ even and $d_2 > 0$

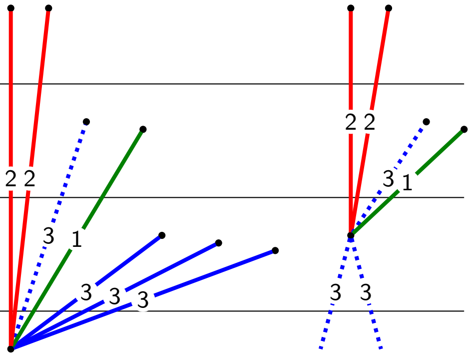


V_1
$d_2 > 0$ and $d_3 = 0$

V_2
$d_3 \geq 0$ and $d_2 = 0$

V_3
$d_3 > 0$ odd and $d_2 > 0$

V_4
$d_3 > 0$ even and $d_2 > 0$



- Vertices in V_1 are 1-monochromatic or 2-monochromatic.
- Vertices in V_2 are 1-monochromatic or 3-monochromatic.
- Vertices in V_3 are bichromatic with odd 3-degree.
- Vertices in V_4 are bichromatic with even 3-degree.

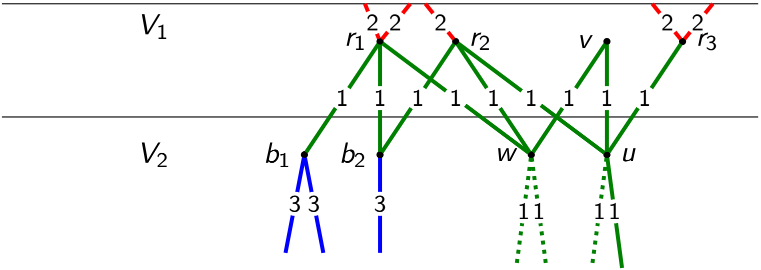
- Vertices in V_1 are 1-monochromatic or 2-monochromatic.
- Vertices in V_2 are 1-monochromatic or 3-monochromatic.
- Vertices in V_3 are bichromatic with odd 3-degree.
- Vertices in V_4 are bichromatic with even 3-degree.

- Vertices in V_1 are 1-monochromatic or 2-monochromatic.
- Vertices in V_2 are 1-monochromatic or 3-monochromatic.
- Vertices in V_3 are bichromatic with odd 3-degree.
- Vertices in V_4 are bichromatic with even 3-degree.

Step 3: Remove conflicts between vertices of V_1 and vertices of V_2.

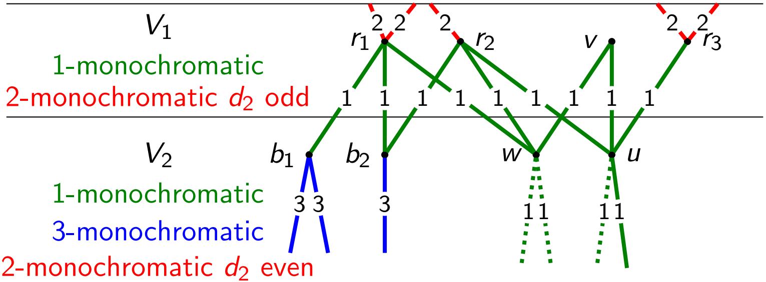
- Vertices in V_1 are 1-monochromatic or 2-monochromatic.
- Vertices in V_2 are 1-monochromatic or 3-monochromatic.
- Vertices in V_3 are bichromatic with odd 3-degree.
- Vertices in V_4 are bichromatic with even 3-degree.

Step 3: Remove conflicts between vertices of V_1 and vertices of V_2.



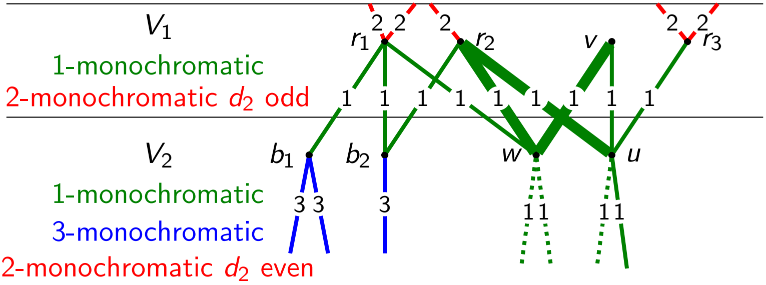
- Vertices in V_1 are 1-monochromatic or 2-monochromatic.
- Vertices in V_2 are 1-monochromatic or 3-monochromatic.
- Vertices in V_3 are bichromatic with odd 3-degree.
- Vertices in V_4 are bichromatic with even 3-degree.

Step 3: Remove conflicts between vertices of V_1 and vertices of V_2.



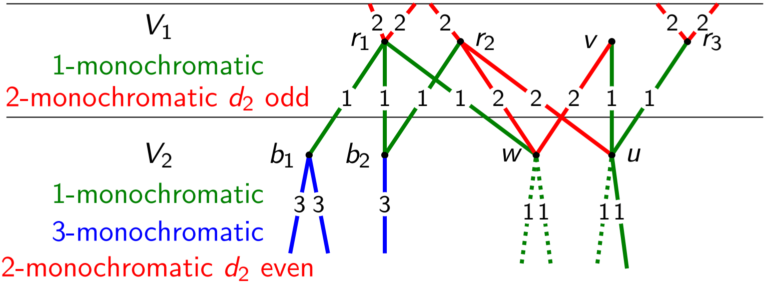
- Vertices in V_1 are 1-monochromatic or 2-monochromatic.
- Vertices in V_2 are 1-monochromatic or 3-monochromatic.
- Vertices in V_3 are bichromatic with odd 3-degree.
- Vertices in V_4 are bichromatic with even 3-degree.

Step 3: Remove conflicts between vertices of V_1 and vertices of V_2.



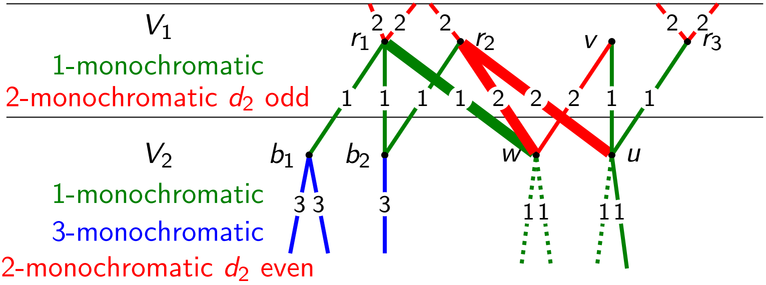
- Vertices in V_1 are 1-monochromatic or 2-monochromatic.
- Vertices in V_2 are 1-monochromatic or 3-monochromatic.
- Vertices in V_3 are bichromatic with odd 3-degree.
- Vertices in V_4 are bichromatic with even 3-degree.

Step 3: Remove conflicts between vertices of V_1 and vertices of V_2.



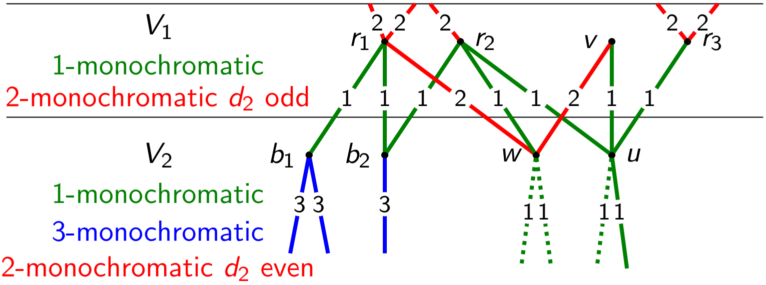
- Vertices in V_1 are 1-monochromatic or 2-monochromatic.
- Vertices in V_2 are 1-monochromatic or 3-monochromatic.
- Vertices in V_3 are bichromatic with odd 3-degree.
- Vertices in V_4 are bichromatic with even 3-degree.

Step 3: Remove conflicts between vertices of V_1 and vertices of V_2.



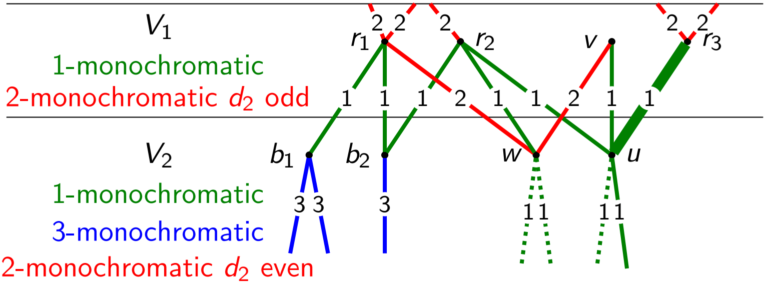
- Vertices in V_1 are 1-monochromatic or 2-monochromatic.
- Vertices in V_2 are 1-monochromatic or 3-monochromatic.
- Vertices in V_3 are bichromatic with odd 3-degree.
- Vertices in V_4 are bichromatic with even 3-degree.

Step 3: Remove conflicts between vertices of V_1 and vertices of V_2.



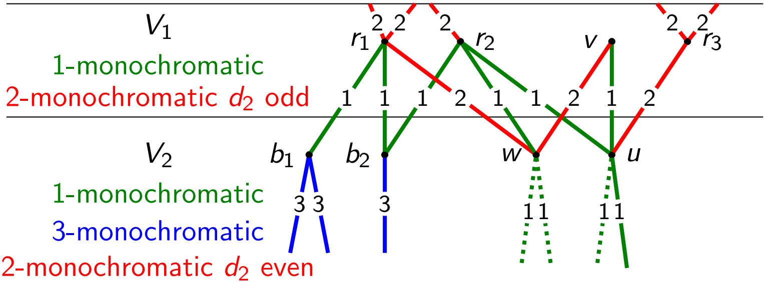
- Vertices in V_1 are 1-monochromatic or 2-monochromatic.
- Vertices in V_2 are 1-monochromatic or 3-monochromatic.
- Vertices in V_3 are bichromatic with odd 3-degree.
- Vertices in V_4 are bichromatic with even 3-degree.

Step 3: Remove conflicts between vertices of V_1 and vertices of V_2.



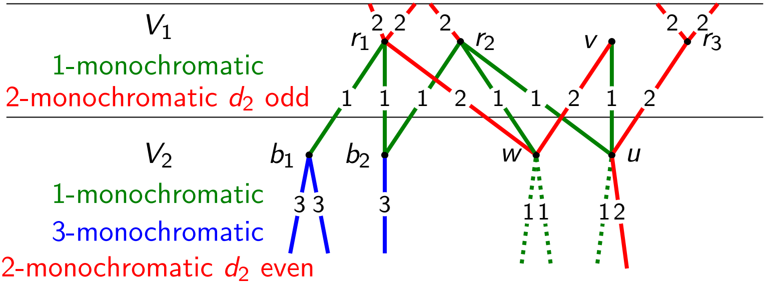
- Vertices in V_1 are 1-monochromatic or 2-monochromatic.
- Vertices in V_2 are 1-monochromatic or 3-monochromatic.
- Vertices in V_3 are bichromatic with odd 3-degree.
- Vertices in V_4 are bichromatic with even 3-degree.

Step 3: Remove conflicts between vertices of V_1 and vertices of V_2.



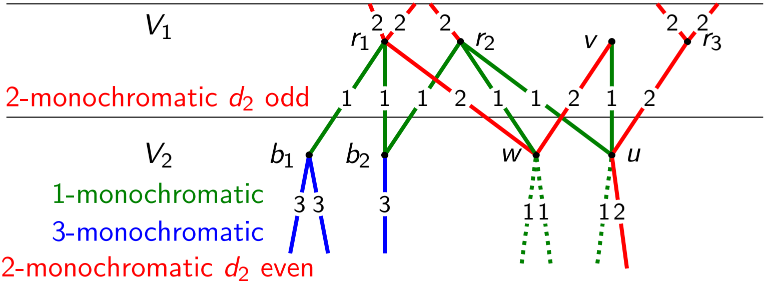
- Vertices in V_1 are 1-monochromatic or 2-monochromatic.
- Vertices in V_2 are 1-monochromatic or 3-monochromatic.
- Vertices in V_3 are bichromatic with odd 3-degree.
- Vertices in V_4 are bichromatic with even 3-degree.

Step 3: Remove conflicts between vertices of V_1 and vertices of V_2.



- Vertices in V_1 are 1-monochromatic or 2-monochromatic.
- Vertices in V_2 are 1-monochromatic or 3-monochromatic.
- Vertices in V_3 are bichromatic with odd 3-degree.
- Vertices in V_4 are bichromatic with even 3-degree.

Step 3: Remove conflicts between vertices of V_1 and vertices of V_2.



Overview

Proof idea for “almost p-proper labellings”

Step 1: Partition $V(G)$ into k independent sets $V_1, \ldots, V_k$ such that each $u \in V_i$ has a neighbor in every V_j for $j < i$.

Proof idea for "almost p-proper labellings"

Step 1: Partition $V(G)$ into k independent sets $V_1, \dots, V_k$ such that each $u \in V_i$ has a neighbor in every V_j for $j < i$.

Step 2: Label edges incident with vertices of $V_k, V_{k-1}, \dots, V_3$.

- $v \in V_1$: 1-monochromatic or 2-monochromatic;
- $v \in V_2$: 1-monochromatic or 3-monochromatic;
- $v \in V_3$: bichromatic, $d_2(v) = 1$, and $d_2(v) + d_3(v)$ **even**;
- $v \in V_4$: bichromatic, $d_3(v) = 2$, and $d_2(v) + d_3(v)$ **odd**;
- $v \in V_5$: bichromatic, $d_2(v) = 2$, and $d_2(v) + d_3(v)$ **even**;
- ...
- $v \in V_{2n}$: bichromatic, $d_3(v) = n$, and $d_2(v) + d_3(v)$ **odd**;
- $v \in V_{2n+1}$: bichromatic, $d_2(v) = n$, and $d_2(v) + d_3(v)$ **even**;
- ...

Proof idea for "almost p-proper labellings"

Step 1: Partition $V(G)$ into k independent sets $V_1, \dots, V_k$ such that each $u \in V_i$ has a neighbor in every V_j for $j < i$.
Step 2: Label edges incident with vertices of $V_k, V_{k-1}, \dots, V_3$.

- $v \in V_1$: 1-monochromatic or 2-monochromatic;
- $v \in V_2$: 1-monochromatic or 3-monochromatic;
- $v \in V_3$: bichromatic, $d_2(v) = 1$, and $d_2(v) + d_3(v)$ **even**;
- $v \in V_4$: bichromatic, $d_3(v) = 2$, and $d_2(v) + d_3(v)$ **odd**;
- $v \in V_5$: bichromatic, $d_2(v) = 2$, and $d_2(v) + d_3(v)$ **even**;
- ...
- $v \in V_{2n}$: bichromatic, $d_3(v) = n$, and $d_2(v) + d_3(v)$ **odd**;
- $v \in V_{2n+1}$: bichromatic, $d_2(v) = n$, and $d_2(v) + d_3(v)$ **even**;
- ...

Step 3: Try to fix conflicts between vertices of V_1 and V_2.

Overview

Corollary

Every graph G admits a **3-labelling** *such that, for every $x \in \mathbb{N}$, the subgraph induced by x-colored vertices is a* **forest**.

Corollary

Every graph G admits a **3-labelling** *such that, for every $x \in \mathbb{N}$, the subgraph induced by x-colored vertices is a* **forest***.*

Conjecture

Every graph G admits a **2-labelling** *such that, for every $x \in \mathbb{N}$, the subgraph induced by x-colored vertices is a* **forest***.*

Corollary

Every graph G admits a **3-labelling** *such that, for every $x \in \mathbb{N}$, the subgraph induced by x-colored vertices is a* **forest**.

Conjecture

Every graph G admits a **2-labelling** *such that, for every $x \in \mathbb{N}$, the subgraph induced by x-colored vertices is a* **forest**.

Theorem

A graph G admits a **2-labelling** *such that, for every $x \in \mathbb{N}$, the subgraph induced by x-colored vertices is a* **forest**, *if*

- *G is a* **complete graph**, *or*
- *G is* **bipartite**, *or*
- *G is* **subcubic**.

Thank you for your attention.