Optimally Reconfiguring List Colorings Given Large Lists

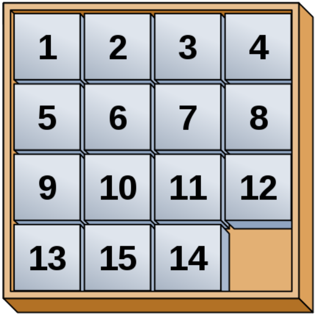
Daniel W. Cranston
Virginia Commonwealth University
dcranston@vcu.edu

Joint with Stijn Cambie and Wouter Cames van Batenburg

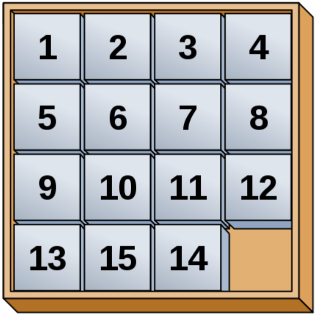
Graph Theory and Combinatorics Seminar
Georgia Tech
7 March 2023

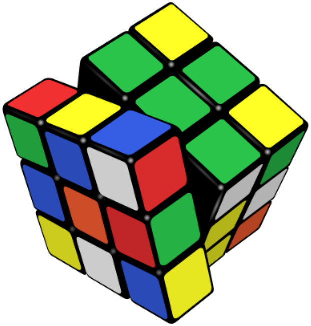
What is Reconfiguration?

What is Reconfiguration?

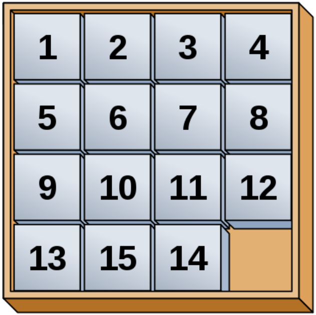


What is Reconfiguration?





What is Reconfiguration?



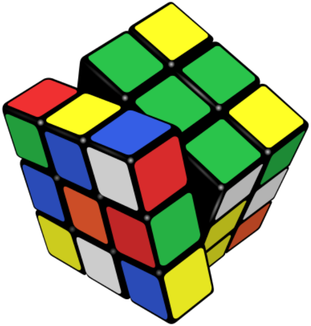
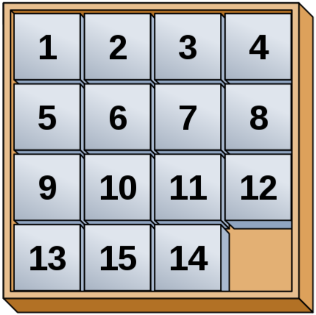


Image credit: Wikipedia

What is Reconfiguration?



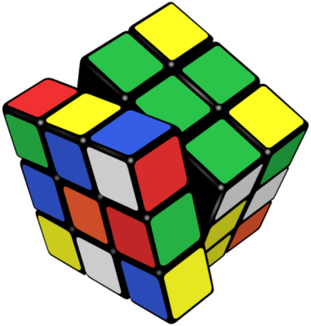
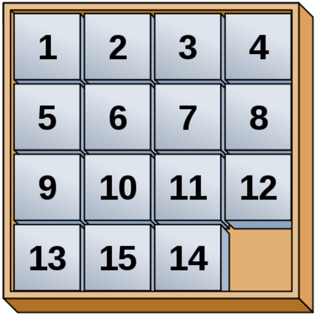


Image credit: Wikipedia

Move from one instance to another

What is Reconfiguration?



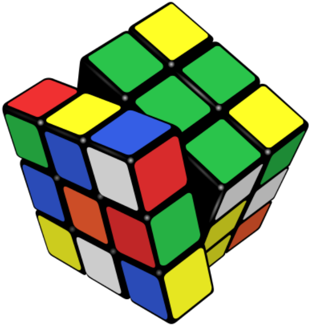
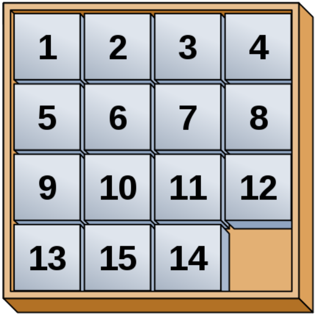


Image credit: Wikipedia

Move from one instance to another by a sequence of small steps?

What is Reconfiguration?



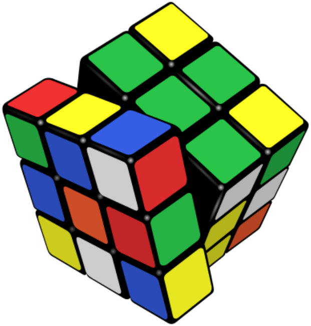
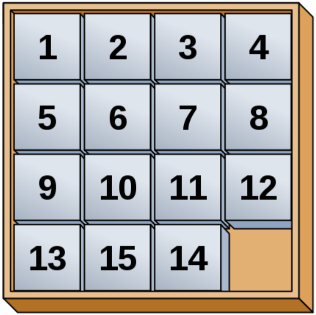


Image credit: Wikipedia

Move from one instance to another by a sequence of small steps?

- Is it always possible?

What is Reconfiguration?



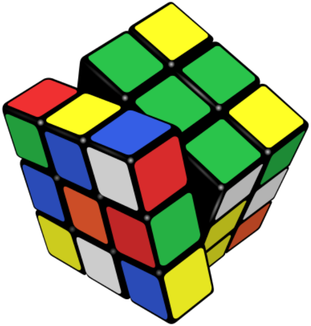
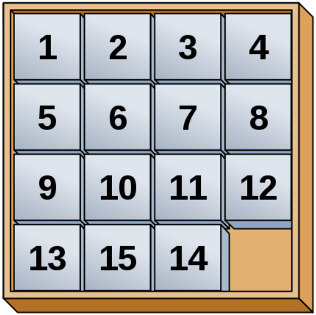


Image credit: Wikipedia

Move from one instance to another by a sequence of small steps?

- Is it always possible?
- If so, how many moves do you need?

What is Reconfiguration?



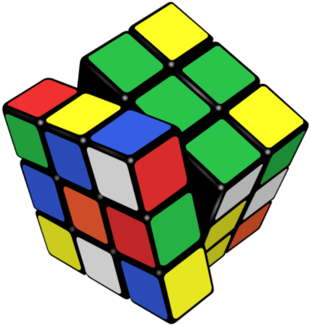
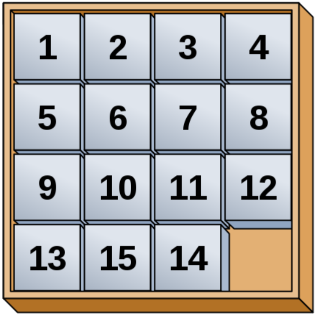


Image credit: Wikipedia

Move from one instance to another by a sequence of small steps?

- Is it always possible?
- If so, how many moves do you need?
- Can you quickly find a short sequence from one to another?

What is Reconfiguration?



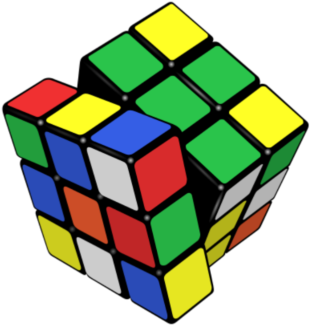


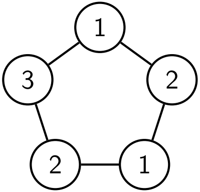
Image credit: Wikipedia

Move from one instance to another by a sequence of small steps?

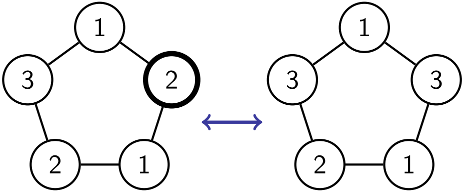
- Is it always possible?
- If so, how many moves do you need?
- Can you quickly find a short sequence from one to another?
- Can you quickly sample from all instances (nearly) uniformly?

What is Coloring Reconfiguration?

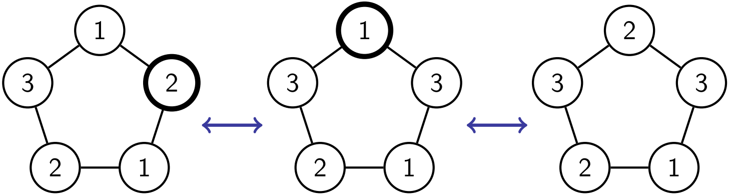
What is Coloring Reconfiguration?



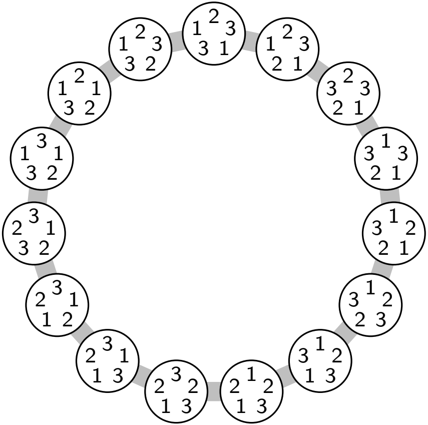
What is Coloring Reconfiguration?



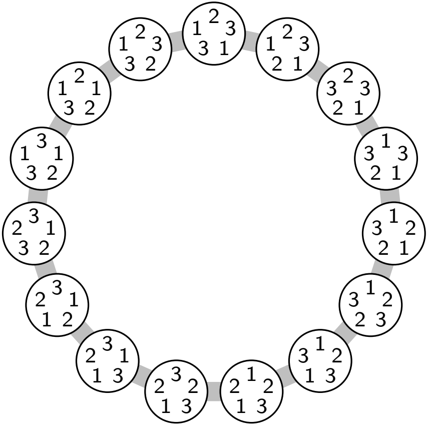
What is Coloring Reconfiguration?



What is Coloring Reconfiguration?

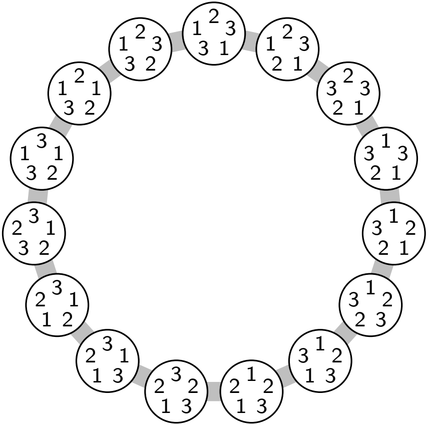


What is Coloring Reconfiguration?

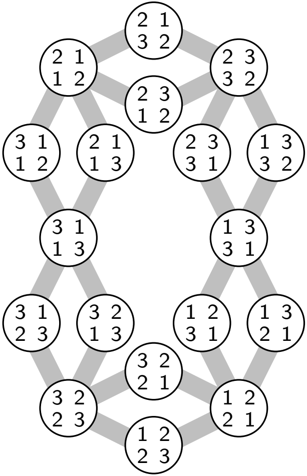


And another isomorphic component.

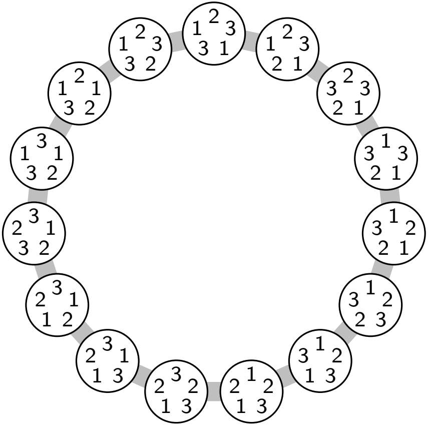
What is Coloring Reconfiguration?



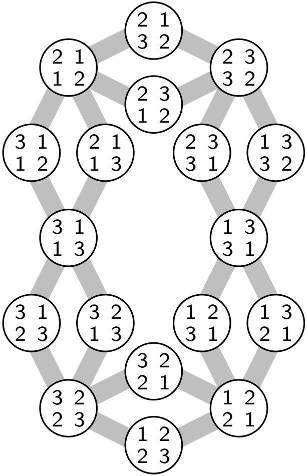
And another isomorphic component.



What is Coloring Reconfiguration?

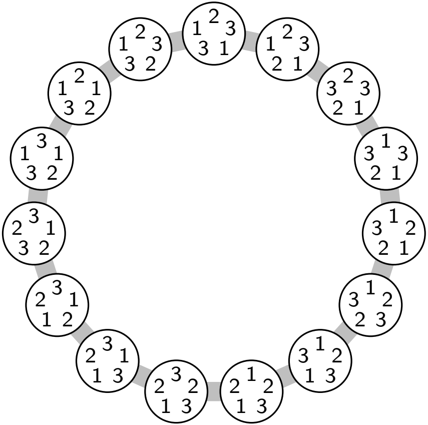


And another isomorphic component.

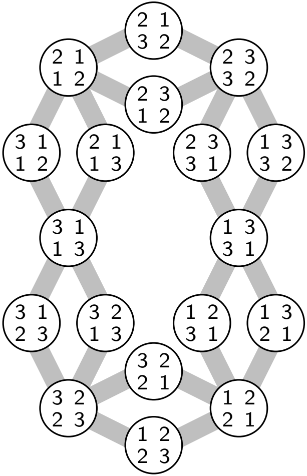


- “Reconfiguration graphs” of 3-colorings of 5-cycle and 4-cycle.

What is Coloring Reconfiguration?

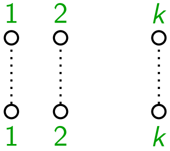


And another isomorphic component.



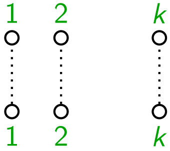
- "Reconfiguration graphs" of 3-colorings of 5-cycle and 4-cycle.
- Can ask all the same questions from the previous page.

Enlightening Examples



$G_k := K_{k,k} - kK_2$

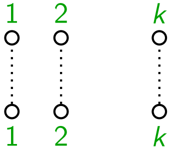
Enlightening Examples



$G_k := K_{k,k} - kK_2$

- $\mathcal{C}_k(G_k)$ is disconnected (frozen colorings)

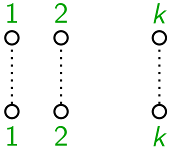
Enlightening Examples



$G_k := K_{k,k} - kK_2$

- $\mathcal{C}_k(G_k)$ is disconnected (frozen colorings)
- $\mathcal{C}_\ell(G_k)$ is connected when $3 \le \ell < k$.

Enlightening Examples

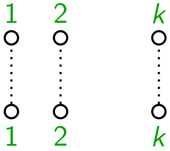


$G_k := K_{k,k} - kK_2$

- $\mathcal{C}_k(G_k)$ is disconnected (frozen colorings)
- $\mathcal{C}_\ell(G_k)$ is connected when $3 \leq \ell < k$.

  Pf: Each part has color repeated, only used on that part.

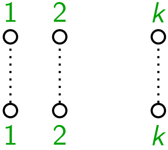
Enlightening Examples



$G_k := K_{k,k} - kK_2$

- $\mathcal{C}_k(G_k)$ is disconnected (frozen colorings)
- $\mathcal{C}_\ell(G_k)$ is connected when $3 \leq \ell < k$.
  Pf: Each part has color repeated, only used on that part. So can reach 2-coloring.

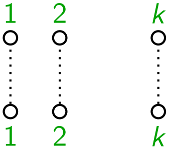
Enlightening Examples



$G_k := K_{k,k} - kK_2$

- $\mathcal{C}_k(G_k)$ is disconnected (frozen colorings)
- $\mathcal{C}_\ell(G_k)$ is connected when $3 \leq \ell < k$.
  Pf: Each part has color repeated, only used on that part. So can reach 2-coloring.
- $\mathcal{C}_\ell(G_k)$ connected when $\ell > k$ (next page)

Enlightening Examples

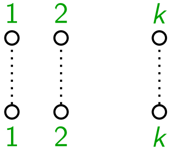


$G_k := K_{k,k} - kK_2$

- $\mathcal{C}_k(G_k)$ is disconnected (frozen colorings)
- $\mathcal{C}_\ell(G_k)$ is connected when $3 \leq \ell < k$.
  Pf: Each part has color repeated, only used on that part. So can reach 2-coloring.
- $\mathcal{C}_\ell(G_k)$ connected when $\ell > k$ (next page)

Bounds on $\mathrm{diam}(\mathcal{C}_k(G))$?

Enlightening Examples



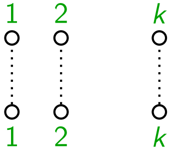
$G_k := K_{k,k} - kK_2$

- $\mathcal{C}_k(G_k)$ is disconnected (frozen colorings)
- $\mathcal{C}_\ell(G_k)$ is connected when $3 \le \ell < k$.
  Pf: Each part has color repeated, only used on that part. So can reach 2-coloring.
- $\mathcal{C}_\ell(G_k)$ connected when $\ell > k$ (next page)

Bounds on $\text{diam}(\mathcal{C}_k(G))$?

- Trivial upper bound: $k^{|G|}$

Enlightening Examples



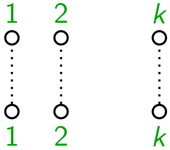
$G_k := K_{k,k} - kK_2$

- $\mathcal{C}_k(G_k)$ is disconnected (frozen colorings)
- $\mathcal{C}_\ell(G_k)$ is connected when $3 \le \ell < k$.
  Pf: Each part has color repeated, only used on that part. So can reach 2-coloring.
- $\mathcal{C}_\ell(G_k)$ connected when $\ell > k$ (next page)

Bounds on $\mathrm{diam}(\mathcal{C}_k(G))$?

- Trivial upper bound: $k^{|G|}$
- Non-trivial lower bound: $2^{\Theta(\sqrt{|G|})}$ (with 4 colors)

Enlightening Examples



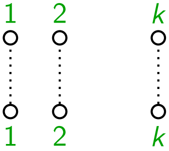
$G_k := K_{k,k} - kK_2$

- $\mathcal{C}_k(G_k)$ is disconnected (frozen colorings)
- $\mathcal{C}_\ell(G_k)$ is connected when $3 \leq \ell < k$.
  Pf: Each part has color repeated, only used on that part. So can reach 2-coloring.
- $\mathcal{C}_\ell(G_k)$ connected when $\ell > k$ (next page)

Bounds on $\text{diam}(\mathcal{C}_k(G))$?

- Trivial upper bound: $k^{|G|}$
- Non-trivial lower bound: $2^{\Theta(\sqrt{|G|})}$ (with 4 colors)
  - Encode an n-bit counter with $\Theta(n^2)$ vertices

Enlightening Examples



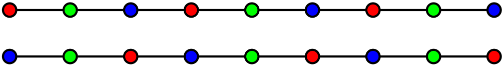
$G_k := K_{k,k} - kK_2$

- $\mathcal{C}_k(G_k)$ is disconnected (frozen colorings)
- $\mathcal{C}_\ell(G_k)$ is connected when $3 \leq \ell < k$.
  Pf: Each part has color repeated, only used on that part. So can reach 2-coloring.
- $\mathcal{C}_\ell(G_k)$ connected when $\ell > k$ (next page)

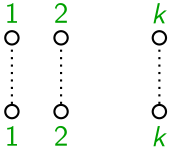
Bounds on $\mathrm{diam}(\mathcal{C}_k(G))$?

- Trivial upper bound: $k^{|G|}$
- Non-trivial lower bound: $2^{\Theta(\sqrt{|G|})}$ (with 4 colors)
  - Encode an n-bit counter with $\Theta(n^2)$ vertices

How about "nice" graphs?



Enlightening Examples

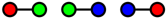
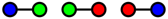


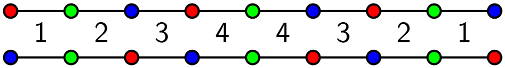
$G_k := K_{k,k} - kK_2$

- $\mathcal{C}_k(G_k)$ is disconnected (frozen colorings)
- $\mathcal{C}_\ell(G_k)$ is connected when $3 \leq \ell < k$.
  Pf: Each part has color repeated, only used on that part. So can reach 2-coloring.
- $\mathcal{C}_\ell(G_k)$ connected when $\ell > k$ (next page)

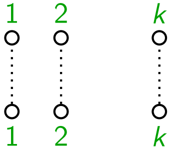
Bounds on $\text{diam}(\mathcal{C}_k(G))$?

- Trivial upper bound: $k^{|G|}$
- Non-trivial lower bound: $2^{\Theta(\sqrt{|G|})}$ (with 4 colors)
  - Encode an n-bit counter with $\Theta(n^2)$ vertices

How about "nice" graphs? '$+$':  '$-$': 



Enlightening Examples

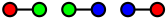
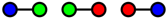


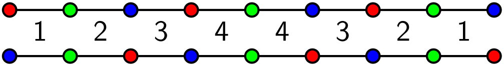
$G_k := K_{k,k} - kK_2$

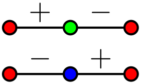
- $\mathcal{C}_k(G_k)$ is disconnected (frozen colorings)
- $\mathcal{C}_\ell(G_k)$ is connected when $3 \le \ell < k$.
  Pf: Each part has color repeated, only used on that part. So can reach 2-coloring.
- $\mathcal{C}_\ell(G_k)$ connected when $\ell > k$ (next page)

Bounds on $\mathrm{diam}(\mathcal{C}_k(G))$?

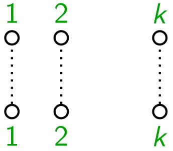
- Trivial upper bound: $k^{|G|}$
- Non-trivial lower bound: $2^{\Theta(\sqrt{|G|})}$ (with 4 colors)
  - Encode an n-bit counter with $\Theta(n^2)$ vertices

How about "nice" graphs? '$+$':  '$-$': 





Enlightening Examples

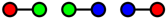
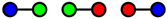


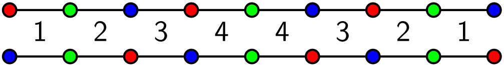
$G_k := K_{k,k} - kK_2$

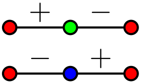
- $\mathcal{C}_k(G_k)$ is disconnected (frozen colorings)
- $\mathcal{C}_\ell(G_k)$ is connected when $3 \leq \ell < k$.
  Pf: Each part has color repeated, only used on that part. So can reach 2-coloring.
- $\mathcal{C}_\ell(G_k)$ connected when $\ell > k$ (next page)

Bounds on $\text{diam}(\mathcal{C}_k(G))$?

- Trivial upper bound: $k^{|G|}$
- Non-trivial lower bound: $2^{\Theta(\sqrt{|G|})}$ (with 4 colors)
  - Encode an n-bit counter with $\Theta(n^2)$ vertices

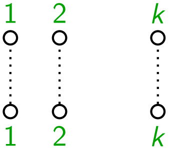
How about "nice" graphs? '+':  '−': 





- So $\text{diam}(\mathcal{C}_3(P_n)) = \Theta(n^2)$

Enlightening Examples



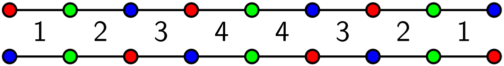
$G_k := K_{k,k} - kK_2$

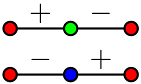
- $\mathcal{C}_k(G_k)$ is disconnected (frozen colorings)
- $\mathcal{C}_\ell(G_k)$ is connected when $3 \leq \ell < k$.
  Pf: Each part has color repeated, only used on that part. So can reach 2-coloring.
- $\mathcal{C}_\ell(G_k)$ connected when $\ell > k$ (next page)

Bounds on $\text{diam}(\mathcal{C}_k(G))$?

- Trivial upper bound: $k^{|G|}$
- Non-trivial lower bound: $2^{\Theta(\sqrt{|G|})}$ (with 4 colors)
  - Encode an n-bit counter with $\Theta(n^2)$ vertices

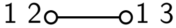
How about "nice" graphs? '$+$': 🔴–🟢 🟢–🔵 🔵–🔴 '$-$': 🔵–🟢 🟢–🔴 🔴–🔵





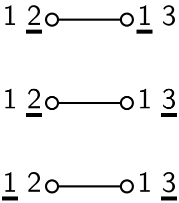
- So $\text{diam}(\mathcal{C}_3(P_n)) = \Theta(n^2)$ and $\text{diam}(\mathcal{C}_k(P_n \vee K_{k-3})) = \Theta(n^2)$

What is List Coloring Reconfiguration?



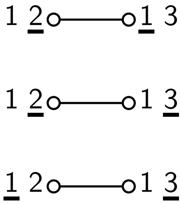
- list-assignment L: each vertex v gets allowable colors $L(v)$

What is List Coloring Reconfiguration?


$1\ \underline{2}$ $\underline{1}\ 3$
$1\ \underline{2}$ $1\ \underline{3}$
$\underline{1}\ 2$ $1\ \underline{3}$

- list-assignment L: each vertex v gets allowable colors $L(v)$
- L-coloring φ: φ is proper and $\varphi(v) \in L(v)$ for all v

What is List Coloring Reconfiguration?

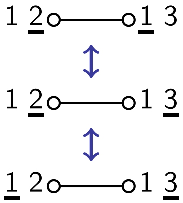


- list-assignment L: each vertex v gets allowable colors $L(v)$
- L-coloring φ: φ is proper and $\varphi(v) \in L(v)$ for all v

Main Questions:

- Given L-colorings α and β, can we change α to β by recoloring single vertices, keeping L-coloring at each step?

What is List Coloring Reconfiguration?

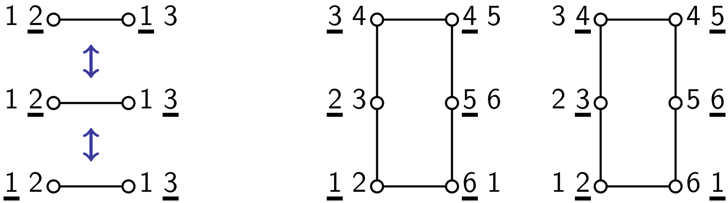


- list-assignment L: each vertex v gets allowable colors $L(v)$
- L-coloring φ: φ is proper and $\varphi(v) \in L(v)$ for all v

Main Questions:

- Given L-colorings α and β, can we change α to β by recoloring single vertices, keeping L-coloring at each step?

What is List Coloring Reconfiguration?

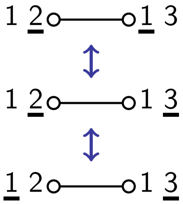


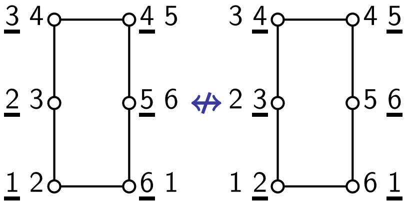
- list-assignment L: each vertex v gets allowable colors $L(v)$
- L-coloring φ: φ is proper and $\varphi(v) \in L(v)$ for all v

Main Questions:

- Given L-colorings α and β, can we change α to β by recoloring single vertices, keeping L-coloring at each step?

What is List Coloring Reconfiguration?



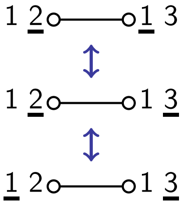


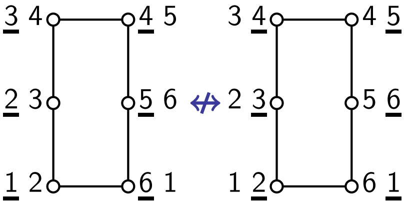
- list-assignment L: each vertex v gets allowable colors $L(v)$
- L-coloring φ: φ is proper and $\varphi(v) \in L(v)$ for all v

Main Questions:

- Given L-colorings α and β, can we change α to β by recoloring single vertices, keeping L-coloring at each step?

What is List Coloring Reconfiguration?



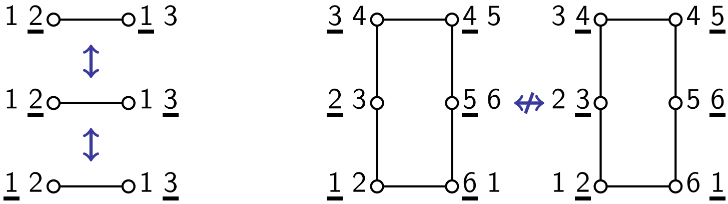


- list-assignment L: each vertex v gets allowable colors $L(v)$
- L-coloring φ: φ is proper and $\varphi(v) \in L(v)$ for all v

Main Questions:

- Given L-colorings α and β, can we change α to β by recoloring single vertices, keeping L-coloring at each step?
- If so, how many steps are needed?

What is List Coloring Reconfiguration?

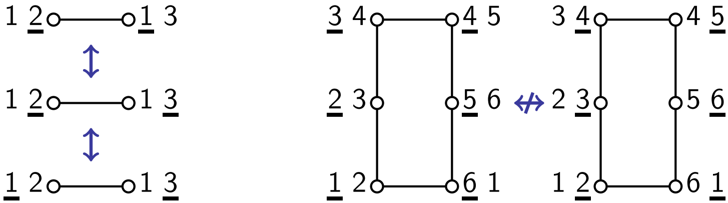


- list-assignment L: each vertex v gets allowable colors $L(v)$
- L-coloring φ: φ is proper and $\varphi(v) \in L(v)$ for all v

Main Questions:

- Given L-colorings α and β, can we change α to β by recoloring single vertices, keeping L-coloring at each step?
- If so, how many steps are needed?
- Given list-assignment L, can we transform every L-coloring α into every L-coloring β?

What is List Coloring Reconfiguration?



- list-assignment L: each vertex v gets allowable colors $L(v)$
- L-coloring φ: φ is proper and $\varphi(v) \in L(v)$ for all v

Main Questions:

- Given L-colorings α and β, can we change α to β by recoloring single vertices, keeping L-coloring at each step?
- If so, how many steps are needed?
- Given list-assignment L, can we transform every L-coloring α into every L-coloring β?
- If so, how many steps are needed in the worst case?

Degenerate Graphs: Conjectures and Tools

Cereceda's Conj: For each $d \in \mathbb{Z}^+$, $\exists$ constant C_d s.t. if G is d-degenerate and $k \geq d+2$, then $\mathrm{diam}(\mathcal{C}_k(G)) \leq C_d|G|^2$.

Degenerate Graphs: Conjectures and Tools

Cereceda's Conj: For each $d \in \mathbb{Z}^+$, $\exists$ constant C_d s.t. if G is d-degenerate and $k \geq d+2$, then $\mathrm{diam}(\mathcal{C}_k(G)) \leq C_d|G|^2$.

Strong Cereceda's Conj: $\exists$ constant C s.t., for each $d \in \mathbb{Z}^+$, if G is d-degenerate and $k \geq d+2$, then $\mathrm{diam}(\mathcal{C}_k(G)) \leq C|G|^2$.

Degenerate Graphs: Conjectures and Tools

Cereceda's Conj: For each $d \in \mathbb{Z}^+$, $\exists$ constant C_d s.t. if G is d-degenerate and $k \geq d+2$, then $\mathrm{diam}(\mathcal{C}_k(G)) \leq C_d|G|^2$.

Strong Cereceda's Conj: $\exists$ constant C s.t., for each $d \in \mathbb{Z}^+$, if G is d-degenerate and $k \geq d+2$, then $\mathrm{diam}(\mathcal{C}_k(G)) \leq C|G|^2$.

Linear Cereceda's Conj: For each $d \in \mathbb{Z}^+$, $\exists$ constant C_d s.t. if G is d-degenerate and $k \geq d+3$, then $\mathrm{diam}(\mathcal{C}_k(G)) \leq C_d|G|$.

Degenerate Graphs: Conjectures and Tools

Cereceda's Conj: For each $d \in \mathbb{Z}^+$, $\exists$ constant C_d s.t. if G is d-degenerate and $k \geq d+2$, then $\mathrm{diam}(\mathcal{C}_k(G)) \leq C_d|G|^2$.

Strong Cereceda's Conj: $\exists$ constant C s.t., for each $d \in \mathbb{Z}^+$, if G is d-degenerate and $k \geq d+2$, then $\mathrm{diam}(\mathcal{C}_k(G)) \leq C|G|^2$.

Linear Cereceda's Conj: For each $d \in \mathbb{Z}^+$, $\exists$ constant C_d s.t. if G is d-degenerate and $k \geq d+3$, then $\mathrm{diam}(\mathcal{C}_k(G)) \leq C_d|G|$.

Obs: Fix G and L. If $\exists v$ s.t. $|L(v)| \geq d(v)+2$, then $\mathcal{C}_L(G)$ is connected iff $\mathcal{C}_L(G-v)$ is connected. So $\mathcal{C}_L(G)$ is connected if G is d-degenerate and L is $(d+2)$-assignment. **Pf:** Induction on $|G|$.

Degenerate Graphs: Conjectures and Tools

Cereceda's Conj: For each $d \in \mathbb{Z}^+$, $\exists$ constant C_d s.t. if G is d-degenerate and $k \geq d+2$, then $\mathrm{diam}(\mathcal{C}_k(G)) \leq C_d|G|^2$.

Strong Cereceda's Conj: $\exists$ constant C s.t., for each $d \in \mathbb{Z}^+$, if G is d-degenerate and $k \geq d+2$, then $\mathrm{diam}(\mathcal{C}_k(G)) \leq C|G|^2$.

Linear Cereceda's Conj: For each $d \in \mathbb{Z}^+$, $\exists$ constant C_d s.t. if G is d-degenerate and $k \geq d+3$, then $\mathrm{diam}(\mathcal{C}_k(G)) \leq C_d|G|$.

Obs: Fix G and L. If $\exists v$ s.t. $|L(v)| \geq d(v)+2$, then $\mathcal{C}_L(G)$ is connected iff $\mathcal{C}_L(G-v)$ is connected. So $\mathcal{C}_L(G)$ is connected if G is d-degenerate and L is $(d+2)$-assignment. **Pf:** Induction on $|G|$.

Key Lem: Fix G, L, v, and L-colorings α and β. Let $G' := G - v$, $\alpha' := \alpha_{\upharpoonright G'}$, $\beta' := \beta_{\upharpoonright G'}$. If we can transform α' to β' only recoloring $N(v)$ at most s times, then we can tranform α to β only recoloring v at most $\lceil \frac{s}{|L(v)|-d(v)-1} \rceil + 1$ times. **Pf:** Above, more carefully.

Degenerate Graphs: Applications of Key Lemma

Key Lem: Fix G, L, v, and L-colorings α and β. Let $G' := G - v$, $\alpha' := \alpha_{\restriction G'}$, $\beta' := \beta_{\restriction G'}$. If we can transform α' to β' only recoloring $N(v)$ at most s times, then we can tranform α to β only recoloring v at most $\lceil \frac{s}{|L(v)|-d(v)-1} \rceil + 1$ times. **Pf:** Above, more carefully.

Degenerate Graphs: Applications of Key Lemma

Key Lem: Fix G, L, v, and L-colorings α and β. Let $G' := G - v$, $\alpha' := \alpha_{\restriction G'}$, $\beta' := \beta_{\restriction G'}$. If we can transform α' to β' only recoloring $N(v)$ at most s times, then we can tranform α to β only recoloring v at most $\lceil \frac{s}{|L(v)|-d(v)-1} \rceil + 1$ times. **Pf:** Above, more carefully.

Thm:[Cereceda] If G is d-degenerate and $|L(v)| \geq 2d + 1$ for all v, then $\mathrm{diam}(\mathcal{C}_L(G)) \leq \binom{n+1}{2}$.

Degenerate Graphs: Applications of Key Lemma

Key Lem: Fix G, L, v, and L-colorings α and β. Let $G' := G - v$, $\alpha' := \alpha_{\restriction G'}$, $\beta' := \beta_{\restriction G'}$. If we can transform α' to β' only recoloring $N(v)$ at most s times, then we can tranform α to β only recoloring v at most $\lceil \frac{s}{|L(v)|-d(v)-1} \rceil + 1$ times. **Pf:** Above, more carefully.

Thm:[Cereceda] If G is d-degenerate and $|L(v)| \geq 2d+1$ for all v, then $\mathrm{diam}(\mathcal{C}_L(G)) \leq \binom{n+1}{2}$.
Pf: By induction, v_i recolored $\leq i$ times. By IH, true for $G - v_n$. Extend, recoloring v_n at most $\lceil \frac{d(n-1)}{2d+1-d-1} \rceil + 1 = n$ times.

Degenerate Graphs: Applications of Key Lemma

Key Lem: Fix G, L, v, and L-colorings α and β. Let $G' := G - v$, $\alpha' := \alpha_{\restriction G'}$, $\beta' := \beta_{\restriction G'}$. If we can transform α' to β' only recoloring $N(v)$ at most s times, then we can tranform α to β only recoloring v at most $\lceil \frac{s}{|L(v)|-d(v)-1} \rceil + 1$ times. **Pf:** Above, more carefully.

Thm:[Cereceda] If G is d-degenerate and $|L(v)| \geq 2d+1$ for all v, then $\mathrm{diam}(\mathcal{C}_L(G)) \leq \binom{n+1}{2}$.
Pf: By induction, v_i recolored $\leq i$ times. By IH, true for $G - v_n$. Extend, recoloring v_n at most $\lceil \frac{d(n-1)}{2d+1-d-1} \rceil + 1 = n$ times.

Thm:[Bousquet-Perarnau] If G is d-degenerate and $|L(v)| \geq 2d+2$ for all v, then $\mathrm{diam}(\mathcal{C}_L(G)) \leq (d+1)n$.

Degenerate Graphs: Applications of Key Lemma

Key Lem: Fix G, L, v, and L-colorings α and β. Let $G' := G - v$, $\alpha' := \alpha_{\upharpoonright G'}$, $\beta' := \beta_{\upharpoonright G'}$. If we can transform α' to β' only recoloring $N(v)$ at most s times, then we can tranform α to β only recoloring v at most $\lceil \frac{s}{|L(v)|-d(v)-1} \rceil + 1$ times. **Pf:** Above, more carefully.

Thm:[Cereceda] If G is d-degenerate and $|L(v)| \geq 2d+1$ for all v, then $\text{diam}(\mathcal{C}_L(G)) \leq \binom{n+1}{2}$.
Pf: By induction, v_i recolored $\leq i$ times. By IH, true for $G - v_n$. Extend, recoloring v_n at most $\lceil \frac{d(n-1)}{2d+1-d-1} \rceil + 1 = n$ times.

Thm:[Bousquet-Perarnau] If G is d-degenerate and $|L(v)| \geq 2d+2$ for all v, then $\text{diam}(\mathcal{C}_L(G)) \leq (d+1)n$.
Pf: By induction, v_i recolored $\leq d+1$ times. By IH, for $G - v_n$. Extend, recoloring v_n at most $\lceil \frac{d(d+1)}{2d+2-d-1} \rceil + 1 = d+1$ times.

Degenerate Graphs: Applications of Key Lemma

Key Lem: Fix G, L, v, and L-colorings α and β. Let $G' := G - v$, $\alpha' := \alpha_{\upharpoonright G'}$, $\beta' := \beta_{\upharpoonright G'}$. If we can transform α' to β' only recoloring $N(v)$ at most s times, then we can tranform α to β only recoloring v at most $\lceil \frac{s}{|L(v)|-d(v)-1} \rceil + 1$ times. **Pf:** Above, more carefully.

Thm:[Cereceda] If G is d-degenerate and $|L(v)| \geq 2d+1$ for all v, then $\text{diam}(\mathcal{C}_L(G)) \leq \binom{n+1}{2}$.
Pf: By induction, v_i recolored $\leq i$ times. By IH, true for $G - v_n$. Extend, recoloring v_n at most $\lceil \frac{d(n-1)}{2d+1-d-1} \rceil + 1 = n$ times.

Thm:[Bousquet-Perarnau] If G is d-degenerate and $|L(v)| \geq 2d+2$ for all v, then $\text{diam}(\mathcal{C}_L(G)) \leq (d+1)n$.
Pf: By induction, v_i recolored $\leq d+1$ times. By IH, for $G - v_n$. Extend, recoloring v_n at most $\lceil \frac{d(d+1)}{2d+2-d-1} \rceil + 1 = d+1$ times.

Thm:[B-P] Fix $k > d$. For all $\epsilon > 0$, $\exists$ $C_{d,\epsilon}$ s.t. if $|L(v)| \geq k$ for all v and $\text{mad}(G) \leq d - \epsilon$, then $\text{diam}(\mathcal{C}_L(G)) = O(n^{C_{d,\epsilon}})$.

Degenerate Graphs: Applications of Key Lemma

Key Lem: Fix G, L, v, and L-colorings α and β. Let $G' := G - v$, $\alpha' := \alpha_{\upharpoonright G'}$, $\beta' := \beta_{\upharpoonright G'}$. If we can transform α' to β' only recoloring $N(v)$ at most s times, then we can tranform α to β only recoloring v at most $\lceil \frac{s}{|L(v)|-d(v)-1} \rceil + 1$ times. **Pf:** Above, more carefully.

Thm:[Cereceda] If G is d-degenerate and $|L(v)| \geq 2d+1$ for all v, then $\mathrm{diam}(\mathcal{C}_L(G)) \leq \binom{n+1}{2}$.
Pf: By induction, v_i recolored $\leq i$ times. By IH, true for $G - v_n$. Extend, recoloring v_n at most $\lceil \frac{d(n-1)}{2d+1-d-1} \rceil + 1 = n$ times.

Thm:[Bousquet-Perarnau] If G is d-degenerate and $|L(v)| \geq 2d+2$ for all v, then $\mathrm{diam}(\mathcal{C}_L(G)) \leq (d+1)n$.
Pf: By induction, v_i recolored $\leq d+1$ times. By IH, for $G - v_n$. Extend, recoloring v_n at most $\lceil \frac{d(d+1)}{2d+2-d-1} \rceil + 1 = d+1$ times.

Thm:[B-P] Fix $k > d$. For all $\epsilon > 0$, $\exists\ C_{d,\epsilon}$ s.t. if $|L(v)| \geq k$ for all v and $\mathrm{mad}(G) \leq d - \epsilon$, then $\mathrm{diam}(\mathcal{C}_L(G)) = O(n^{C_{d,\epsilon}})$.
Pf: Like above, but delete many vertices at once.

Coloring Reconfig: Best Bounds for Degenerate Graphs

Thm:[Bousquet–Heinrich] Fix $\epsilon \in (0,1)$. There exist C_1, C_2, C_ϵ s.t. for all $d, k \in \mathbb{Z}^+$, if G is d-degenerate, then:

Coloring Reconfig: Best Bounds for Degenerate Graphs

Thm:[Bousquet–Heinrich] Fix $\epsilon \in (0,1)$. There exist C_1, C_2, C_ϵ s.t. for all $d, k \in \mathbb{Z}^+$, if G is d-degenerate, then:

1. If $k \geq \frac{3}{2}(d+1)$, then $\mathrm{diam}(\mathcal{C}_k(G)) \leq C_1 n^2$.

Coloring Reconfig: Best Bounds for Degenerate Graphs

Thm:[Bousquet–Heinrich] Fix $\epsilon \in (0,1)$. There exist C_1, C_2, C_ϵ s.t. for all $d, k \in \mathbb{Z}^+$, if G is d-degenerate, then:

1. If $k \geq \frac{3}{2}(d+1)$, then $\mathrm{diam}(\mathcal{C}_k(G)) \leq C_1 n^2$.
2. If $k \geq (1+\epsilon)(d+2)$, then $\mathrm{diam}(\mathcal{C}_k(G)) \leq C_\epsilon n^{\lceil 1/\epsilon \rceil}$.

Coloring Reconfig: Best Bounds for Degenerate Graphs

Thm:[Bousquet–Heinrich] Fix $\epsilon \in (0,1)$. There exist C_1, C_2, C_ϵ s.t. for all $d, k \in \mathbb{Z}^+$, if G is d-degenerate, then:

1. If $k \geq \frac{3}{2}(d+1)$, then $\mathrm{diam}(\mathcal{C}_k(G)) \leq C_1 n^2$.
2. If $k \geq (1+\epsilon)(d+2)$, then $\mathrm{diam}(\mathcal{C}_k(G)) \leq C_\epsilon n^{\lceil 1/\epsilon \rceil}$.
3. If $k \geq d+2$, then $\mathrm{diam}(\mathcal{C}_k(G)) \leq (C_2 n)^{d+1}$

Coloring Reconfig: Best Bounds for Degenerate Graphs

Thm:[Bousquet–Heinrich] Fix $\epsilon \in (0,1)$. There exist C_1, C_2, C_ϵ s.t. for all $d, k \in \mathbb{Z}^+$, if G is d-degenerate, then:

1. If $k \geq \frac{3}{2}(d+1)$, then $\text{diam}(\mathcal{C}_k(G)) \leq C_1 n^2$.
2. If $k \geq (1+\epsilon)(d+2)$, then $\text{diam}(\mathcal{C}_k(G)) \leq C_\epsilon n^{\lceil 1/\epsilon \rceil}$.
3. If $k \geq d+2$, then $\text{diam}(\mathcal{C}_k(G)) \leq (C_2 n)^{d+1}$

Thm:[Feghali] Fix $k > d$. Fix $\epsilon \in (0,1)$. There exists $C_{d,\epsilon}$ s.t. if $\text{mad}(G) \leq d - \epsilon$, then $\text{diam}(\mathcal{C}_k(G)) \leq C_{d,\epsilon} n (\log n)^{d-1}$.

Coloring Reconfig: Best Bounds for Degenerate Graphs

Thm:[Bousquet–Heinrich] Fix $\epsilon \in (0,1)$. There exist C_1, C_2, C_ϵ s.t. for all $d, k \in \mathbb{Z}^+$, if G is d-degenerate, then:

1. If $k \geq \frac{3}{2}(d+1)$, then $\text{diam}(\mathcal{C}_k(G)) \leq C_1 n^2$.
2. If $k \geq (1+\epsilon)(d+2)$, then $\text{diam}(\mathcal{C}_k(G)) \leq C_\epsilon n^{\lceil 1/\epsilon \rceil}$.
3. If $k \geq d+2$, then $\text{diam}(\mathcal{C}_k(G)) \leq (C_2 n)^{d+1}$

Thm:[Feghali] Fix $k > d$. Fix $\epsilon \in (0,1)$. There exists $C_{d,\epsilon}$ s.t. if $\text{mad}(G) \leq d - \epsilon$, then $\text{diam}(\mathcal{C}_k(G)) \leq C_{d,\epsilon} n (\log n)^{d-1}$.

Thm:[Bartier–Bousquet–Feghali–Heinrich–Moore–Pierron]
If G is planar with girth 5, then $\mathcal{C}_4(G)$ is connected.

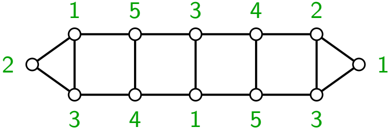
Avoiding Frozen Colorings

Prop: Every graph G has list assignment L with $|L(v)| = d(v) + 1$ for all v and L-colorings α and β s.t. we cannot reach β from α.

Avoiding Frozen Colorings

Prop: Every graph G has list assignment L with $|L(v)| = d(v) + 1$ for all v and L-colorings α and β s.t. we cannot reach β from α.

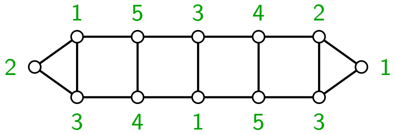
Pf: Color G^2 arbitrarily; call it α.

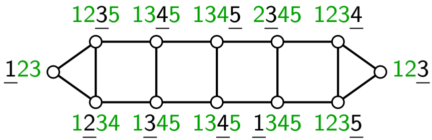


Avoiding Frozen Colorings

Prop: Every graph G has list assignment L with $|L(v)| = d(v) + 1$ for all v and L-colorings α and β s.t. we cannot reach β from α.

Pf: Color G^2 arbitrarily; call it α. Let $L(v) := \{\alpha(w) : \ w \in N[v]\}$. Now let β be another L-coloring (color greedily in any order).

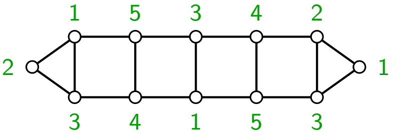


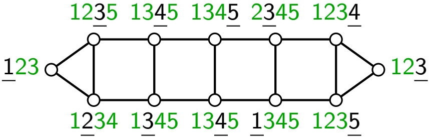


Avoiding Frozen Colorings

Prop: Every graph G has list assignment L with $|L(v)| = d(v) + 1$ for all v and L-colorings α and β s.t. we cannot reach β from α.

Pf: Color G^2 arbitrarily; call it α. Let $L(v) := \{\alpha(w) : \ w \in N[v]\}$. Now let β be another L-coloring (color greedily in any order). Note that α is frozen (no recoloring is possible), so cannot reach β.

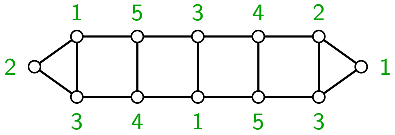


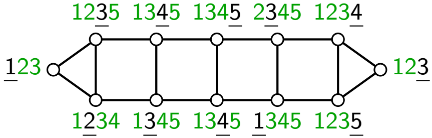


Avoiding Frozen Colorings

Prop: Every graph G has list assignment L with $|L(v)| = d(v) + 1$ for all v and L-colorings α and β s.t. we cannot reach β from α.

Pf: Color G^2 arbitrarily; call it α. Let $L(v) := \{\alpha(w) : \ w \in N[v]\}$. Now let β be another L-coloring (color greedily in any order). Note that α is frozen (no recoloring is possible), so cannot reach β.



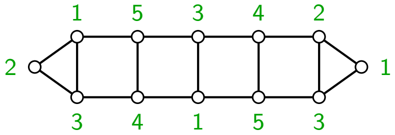


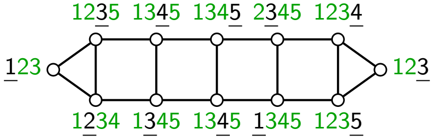
Prop: Fix an acyclic orientation D of a graph G and a list assignment L for G. If $|L(v)| \geq d_D(v) + 2$ for all $v \in V(G)$, then every two L-colorings α and β can reach each other by single vertex recolorings.

Avoiding Frozen Colorings

Prop: Every graph G has list assignment L with $|L(v)| = d(v) + 1$ for all v and L-colorings α and β s.t. we cannot reach β from α.

Pf: Color G^2 arbitrarily; call it α. Let $L(v) := \{\alpha(w) \colon\ w \in N[v]\}$. Now let β be another L-coloring (color greedily in any order). Note that α is frozen (no recoloring is possible), so cannot reach β.





Prop: Fix an acyclic orientation D of a graph G and a list assignment L for G. If $|L(v)| \geq d_D(v) + 2$ for all $v \in V(G)$, then every two L-colorings α and β can reach each other by single vertex recolorings.

Pf sketch: Induction on $|V(G)|$.

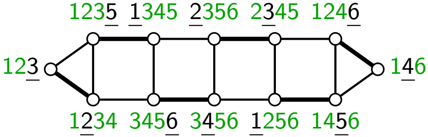
How Many Moves are Needed?

Prop: For every G and every f, with $f(v) \geq 2$ for all v, there is list assignment L with $|L(v)| = f(v)$ for all v and L-colorings α and β where changing α to β needs $n(G) + \mu(G)$ moves.

How Many Moves are Needed?

Prop: For every G and every f, with $f(v) \geq 2$ for all v, there is list assignment L with $|L(v)| = f(v)$ for all v and L-colorings α and β where changing α to β needs $n(G) + \mu(G)$ moves.

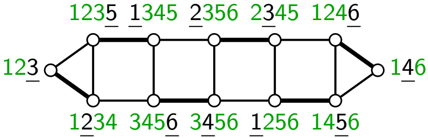
Pf: Every vert needs recolored; every edge of M needs extra step.



How Many Moves are Needed?

Prop: For every G and every f, with $f(v) \geq 2$ for all v, there is list assignment L with $|L(v)| = f(v)$ for all v and L-colorings α and β where changing α to β needs $n(G) + \mu(G)$ moves.

Pf: Every vert needs recolored; every edge of M needs extra step.

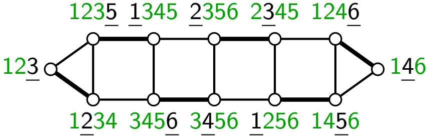


Conj:[Cambie–Cames van Batenburg–C.] For list assignment L with $|L(v)| \geq d(v) + 2$ for all v and L-colorings α and β, can always change α to β in at most $n(G) + \mu(G)$ steps.

How Many Moves are Needed?

Prop: For every G and every f, with $f(v) \geq 2$ for all v, there is list assignment L with $|L(v)| = f(v)$ for all v and L-colorings α and β where changing α to β needs $n(G) + \mu(G)$ moves.

Pf: Every vert needs recolored; every edge of M needs extra step.



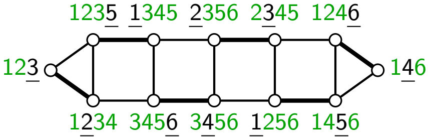
Conj:[Cambie–Cames van Batenburg–C.] For list assignment L with $|L(v)| \geq d(v) + 2$ for all v and L-colorings α and β, can always change α to β in at most $n(G) + \mu(G)$ steps.

Thm:[Cambie–Cames van Batenburg–C.] `arXiv:2204.07928`

How Many Moves are Needed?

Prop: For every G and every f, with $f(v) \geq 2$ for all v, there is list assignment L with $|L(v)| = f(v)$ for all v and L-colorings α and β where changing α to β needs $n(G) + \mu(G)$ moves.

Pf: Every vert needs recolored; every edge of M needs extra step.



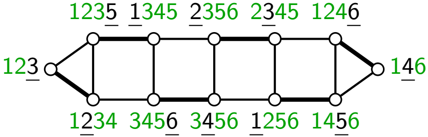
Conj:[Cambie–Cames van Batenburg–C.] For list assignment L with $|L(v)| \geq d(v) + 2$ for all v and L-colorings α and β, can always change α to β in at most $n(G) + \mu(G)$ steps.

Thm:[Cambie–Cames van Batenburg–C.] `arXiv:2204.07928`
(a) If $|L(v)| \geq 2d(v) + 1$, then $n(G) + \mu(G)$ steps suffice.

How Many Moves are Needed?

Prop: For every G and every f, with $f(v) \geq 2$ for all v, there is list assignment L with $|L(v)| = f(v)$ for all v and L-colorings α and β where changing α to β needs $n(G) + \mu(G)$ moves.

Pf: Every vert needs recolored; every edge of M needs extra step.



Conj:[Cambie–Cames van Batenburg–C.] For list assignment L with $|L(v)| \geq d(v) + 2$ for all v and L-colorings α and β, can always change α to β in at most $n(G) + \mu(G)$ steps.

Thm:[Cambie–Cames van Batenburg–C.] arXiv:2204.07928

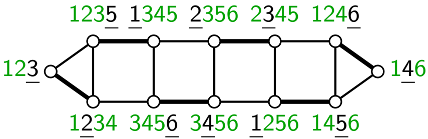
(a) If $|L(v)| \geq 2d(v) + 1$, then $n(G) + \mu(G)$ steps suffice.

(b) If $|L(v)| \geq d(v) + 2$, then $n(G) + 2\mu(G)$ steps suffice.

How Many Moves are Needed?

Prop: For every G and every f, with $f(v) \geq 2$ for all v, there is list assignment L with $|L(v)| = f(v)$ for all v and L-colorings α and β where changing α to β needs $n(G) + \mu(G)$ moves.

Pf: Every vert needs recolored; every edge of M needs extra step.



Conj:[Cambie–Cames van Batenburg–C.] For list assignment L with $|L(v)| \geq d(v) + 2$ for all v and L-colorings α and β, can always change α to β in at most $n(G) + \mu(G)$ steps.

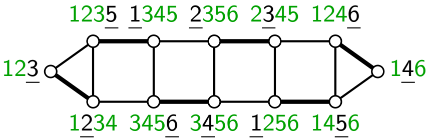
Thm:[Cambie–Cames van Batenburg–C.] `arXiv:2204.07928`
(a) If $|L(v)| \geq 2d(v) + 1$, then $n(G) + \mu(G)$ steps suffice.
(b) If $|L(v)| \geq d(v) + 2$, then $n(G) + 2\mu(G)$ steps suffice.

Correspondence Coloring: $\mu(G) \to \tau(G)$.

How Many Moves are Needed?

Prop: For every G and every f, with $f(v) \geq 2$ for all v, there is list assignment L with $|L(v)| = f(v)$ for all v and L-colorings α and β where changing α to β needs $n(G) + \mu(G)$ moves.

Pf: Every vert needs recolored; every edge of M needs extra step.



Conj:[Cambie–Cames van Batenburg–C.] For list assignment L with $|L(v)| \geq d(v) + 2$ for all v and L-colorings α and β, can always change α to β in at most $n(G) + \mu(G)$ steps.

Thm:[Cambie–Cames van Batenburg–C.] `arXiv:2204.07928`
(a) If $|L(v)| \geq 2d(v) + 1$, then $n(G) + \mu(G)$ steps suffice.
(b) If $|L(v)| \geq d(v) + 2$, then $n(G) + 2\mu(G)$ steps suffice.

Correspondence Coloring: $\mu(G) \to \tau(G)$. Conj. and Theorems

Two Proof Sketches

Two Proof Sketches

Lem: Fix list-assignment L with $|L(v)| \geq d(v) + 2$ for all v. Given L-colorings α and β, can recolor α to β in at most $2n(G) - 1$ steps.

Two Proof Sketches

Lem: Fix list-assignment L with $|L(v)| \geq d(v) + 2$ for all v. Given L-colorings α and β, can recolor α to β in at most $2n(G) - 1$ steps.
Pf Sketch: Induction on number of colors k used by β.

Two Proof Sketches

Lem: Fix list-assignment L with $|L(v)| \geq d(v) + 2$ for all v. Given L-colorings α and β, can recolor α to β in at most $2n(G) - 1$ steps.
Pf Sketch: Induction on number of colors k used by β. Base case, $k = 1$, is easy.

Two Proof Sketches

Lem: Fix list-assignment L with $|L(v)| \geq d(v) + 2$ for all v. Given L-colorings α and β, can recolor α to β in at most $2n(G) - 1$ steps.
Pf Sketch: Induction on number of colors k used by β. Base case, $k = 1$, is easy. By Pigeonhole, some color c used at least as often by β as by α.

Two Proof Sketches

Lem: Fix list-assignment L with $|L(v)| \geq d(v) + 2$ for all v. Given L-colorings α and β, can recolor α to β in at most $2n(G) - 1$ steps.
Pf Sketch: Induction on number of colors k used by β. Base case, $k = 1$, is easy. By Pigeonhole, some color c used at least as often by β as by α. From α, recolor every vertex using c.

Two Proof Sketches

Lem: Fix list-assignment L with $|L(v)| \geq d(v) + 2$ for all v. Given L-colorings α and β, can recolor α to β in at most $2n(G) - 1$ steps.
Pf Sketch: Induction on number of colors k used by β. Base case, $k = 1$, is easy. By Pigeonhole, some color c used at least as often by β as by α. From α, recolor every vertex using c. Now recolor every vertex using c in β.

Two Proof Sketches

Lem: Fix list-assignment L with $|L(v)| \geq d(v)+2$ for all v. Given L-colorings α and β, can recolor α to β in at most $2n(G)-1$ steps.
Pf Sketch: Induction on number of colors k used by β. Base case, $k=1$, is easy. By Pigeonhole, some color c used at least as often by β as by α. From α, recolor every vertex using c. Now recolor every vertex using c in β. Finish by induction.

Two Proof Sketches

Lem: Fix list-assignment L with $|L(v)| \geq d(v) + 2$ for all v. Given L-colorings α and β, can recolor α to β in at most $2n(G) - 1$ steps.
Pf Sketch: Induction on number of colors k used by β. Base case, $k = 1$, is easy. By Pigeonhole, some color c used at least as often by β as by α. From α, recolor every vertex using c. Now recolor every vertex using c in β. Finish by induction.

Thm: Fix L with $|L(v)| \geq d(v) + 2$ for all v and L-colorings α and β. Can recolor α to β in at most $n(G) + 2\mu(G)$ steps.

Two Proof Sketches

Lem: Fix list-assignment L with $|L(v)| \geq d(v) + 2$ for all v. Given L-colorings α and β, can recolor α to β in at most $2n(G) - 1$ steps.
Pf Sketch: Induction on number of colors k used by β. Base case, $k = 1$, is easy. By Pigeonhole, some color c used at least as often by β as by α. From α, recolor every vertex using c. Now recolor every vertex using c in β. Finish by induction.

Thm: Fix L with $|L(v)| \geq d(v) + 2$ for all v and L-colorings α and β. Can recolor α to β in at most $n(G) + 2\mu(G)$ steps.
Pf Sketch: Induction on $n(G)$. Find v with $\mu(G - v) = \mu(G) - 1$.

Two Proof Sketches

Lem: Fix list-assignment L with $|L(v)| \geq d(v) + 2$ for all v. Given L-colorings α and β, can recolor α to β in at most $2n(G) - 1$ steps.
Pf Sketch: Induction on number of colors k used by β. Base case, $k = 1$, is easy. By Pigeonhole, some color c used at least as often by β as by α. From α, recolor every vertex using c. Now recolor every vertex using c in β. Finish by induction.

Thm: Fix L with $|L(v)| \geq d(v) + 2$ for all v and L-colorings α and β. Can recolor α to β in at most $n(G) + 2\mu(G)$ steps.
Pf Sketch: Induction on $n(G)$. Find v with $\mu(G - v) = \mu(G) - 1$. If no such v exists, finish by lemma.

Two Proof Sketches

Lem: Fix list-assignment L with $|L(v)| \geq d(v) + 2$ for all v. Given L-colorings α and β, can recolor α to β in at most $2n(G) - 1$ steps.
Pf Sketch: Induction on number of colors k used by β. Base case, $k = 1$, is easy. By Pigeonhole, some color c used at least as often by β as by α. From α, recolor every vertex using c. Now recolor every vertex using c in β. Finish by induction.

Thm: Fix L with $|L(v)| \geq d(v) + 2$ for all v and L-colorings α and β. Can recolor α to β in at most $n(G) + 2\mu(G)$ steps.
Pf Sketch: Induction on $n(G)$. Find v with $\mu(G - v) = \mu(G) - 1$. If no such v exists, finish by lemma. By Pigeonhole, find c in $L(v)$ used at most once on $N(v)$ by α and β (in total).

Two Proof Sketches

Lem: Fix list-assignment L with $|L(v)| \geq d(v) + 2$ for all v. Given L-colorings α and β, can recolor α to β in at most $2n(G) - 1$ steps.
Pf Sketch: Induction on number of colors k used by β. Base case, $k = 1$, is easy. By Pigeonhole, some color c used at least as often by β as by α. From α, recolor every vertex using c. Now recolor every vertex using c in β. Finish by induction.

Thm: Fix L with $|L(v)| \geq d(v) + 2$ for all v and L-colorings α and β. Can recolor α to β in at most $n(G) + 2\mu(G)$ steps.
Pf Sketch: Induction on $n(G)$. Find v with $\mu(G - v) = \mu(G) - 1$. If no such v exists, finish by lemma. By Pigeonhole, find c in $L(v)$ used at most once on $N(v)$ by α and β (in total). Assume α uses c once on $w \in N(v)$.

Two Proof Sketches

Lem: Fix list-assignment L with $|L(v)| \geq d(v) + 2$ for all v. Given L-colorings α and β, can recolor α to β in at most $2n(G) - 1$ steps.
Pf Sketch: Induction on number of colors k used by β. Base case, $k = 1$, is easy. By Pigeonhole, some color c used at least as often by β as by α. From α, recolor every vertex using c. Now recolor every vertex using c in β. Finish by induction.

Thm: Fix L with $|L(v)| \geq d(v) + 2$ for all v and L-colorings α and β. Can recolor α to β in at most $n(G) + 2\mu(G)$ steps.
Pf Sketch: Induction on $n(G)$. Find v with $\mu(G - v) = \mu(G) - 1$. If no such v exists, finish by lemma. By Pigeonhole, find c in $L(v)$ used at most once on $N(v)$ by α and β (in total). Assume α uses c once on $w \in N(v)$. From α: recolor w, recolor v with c.

Two Proof Sketches

Lem: Fix list-assignment L with $|L(v)| \geq d(v) + 2$ for all v. Given L-colorings α and β, can recolor α to β in at most $2n(G) - 1$ steps.
Pf Sketch: Induction on number of colors k used by β. Base case, $k = 1$, is easy. By Pigeonhole, some color c used at least as often by β as by α. From α, recolor every vertex using c. Now recolor every vertex using c in β. Finish by induction.

Thm: Fix L with $|L(v)| \geq d(v) + 2$ for all v and L-colorings α and β. Can recolor α to β in at most $n(G) + 2\mu(G)$ steps.
Pf Sketch: Induction on $n(G)$. Find v with $\mu(G - v) = \mu(G) - 1$. If no such v exists, finish by lemma. By Pigeonhole, find c in $L(v)$ used at most once on $N(v)$ by α and β (in total). Assume α uses c once on $w \in N(v)$. From α: recolor w, recolor v with c. Finish on $G - v$ by induction.

Two Proof Sketches

Lem: Fix list-assignment L with $|L(v)| \geq d(v) + 2$ for all v. Given L-colorings α and β, can recolor α to β in at most $2n(G) - 1$ steps.
Pf Sketch: Induction on number of colors k used by β. Base case, $k = 1$, is easy. By Pigeonhole, some color c used at least as often by β as by α. From α, recolor every vertex using c. Now recolor every vertex using c in β. Finish by induction.

Thm: Fix L with $|L(v)| \geq d(v) + 2$ for all v and L-colorings α and β. Can recolor α to β in at most $n(G) + 2\mu(G)$ steps.
Pf Sketch: Induction on $n(G)$. Find v with $\mu(G - v) = \mu(G) - 1$. If no such v exists, finish by lemma. By Pigeonhole, find c in $L(v)$ used at most once on $N(v)$ by α and β (in total). Assume α uses c once on $w \in N(v)$. From α: recolor w, recolor v with c. Finish on $G - v$ by induction. Recolor v to $\beta(v)$.

Two Proof Sketches

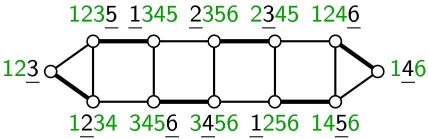
Lem: Fix list-assignment L with $|L(v)| \geq d(v) + 2$ for all v. Given L-colorings α and β, can recolor α to β in at most $2n(G) - 1$ steps.
Pf Sketch: Induction on number of colors k used by β. Base case, $k = 1$, is easy. By Pigeonhole, some color c used at least as often by β as by α. From α, recolor every vertex using c. Now recolor every vertex using c in β. Finish by induction.

Thm: Fix L with $|L(v)| \geq d(v) + 2$ for all v and L-colorings α and β. Can recolor α to β in at most $n(G) + 2\mu(G)$ steps.
Pf Sketch: Induction on $n(G)$. Find v with $\mu(G - v) = \mu(G) - 1$. If no such v exists, finish by lemma. By Pigeonhole, find c in $L(v)$ used at most once on $N(v)$ by α and β (in total). Assume α uses c once on $w \in N(v)$. From α: recolor w, recolor v with c. Finish on $G - v$ by induction. Recolor v to $\beta(v)$. Extra steps:
$2 + 1 = n(G) - n(G - v) + 2(\mu(G) - \mu(G - v))$.

How Many Moves are Needed?

Prop: For every G and every f, with $f(v) \geq 2$ for all v, there is list assignment L with $|L(v)| = f(v)$ for all v and L-colorings α and β where changing α to β needs $n(G) + \mu(G)$ moves.

Pf: Every vert needs recolored; every edge of M needs extra step.



Conj:[Cambie–Cames van Batenburg–C.] For list assignment L with $|L(v)| \geq d(v) + 2$ for all v and L-colorings α and β, can always change α to β in at most $n(G) + \mu(G)$ steps.

Thm:[Cambie–Cames van Batenburg–C.] arXiv:2204.07928
(a) If $|L(v)| \geq 2d(v) + 1$, then $n(G) + \mu(G)$ steps suffice.
(b) If $|L(v)| \geq d(v) + 2$, then $n(G) + 2\mu(G)$ steps suffice.

Correspondence Coloring: $\mu(G) \to \tau(G)$. Conj. and Theorems