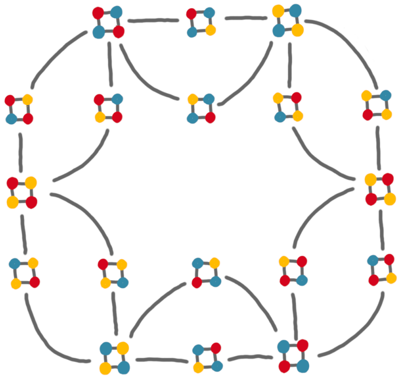
Short and local transformations between $(\Delta + 1)$-colorings

Laurent Feuilloley

CNRS and University of Lyon

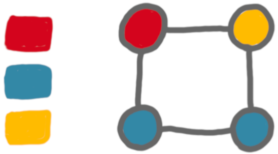
joint work with Nicolas Bousquet, Marc Heinrich, and Mikaël Rabie

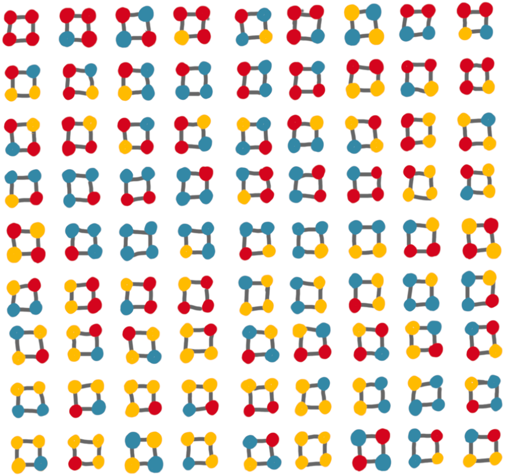
GT CoA days · Paris · November 2024



The structure of graph colorings

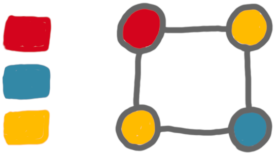
All colorings





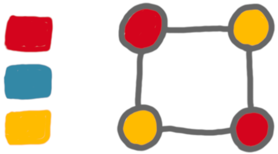
The structure of graph colorings

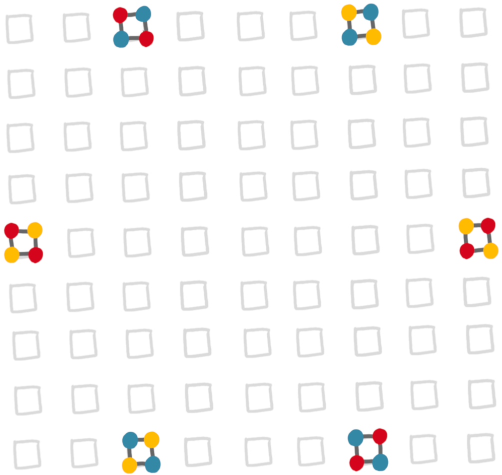
All *proper* colorings



The structure of graph colorings

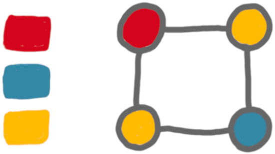
All *optimal* proper colorings

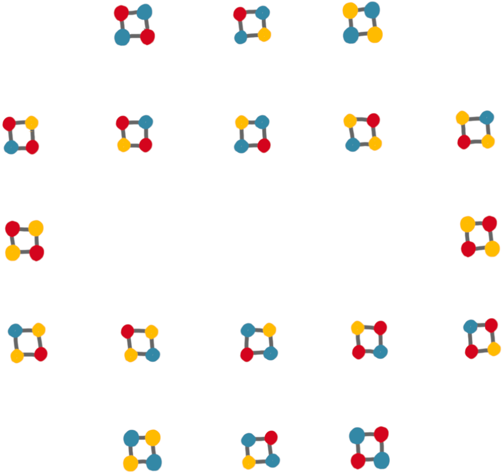




The structure of graph colorings

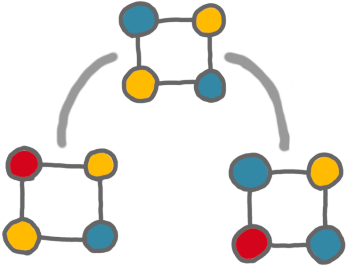
Our focus today:
All proper colorings



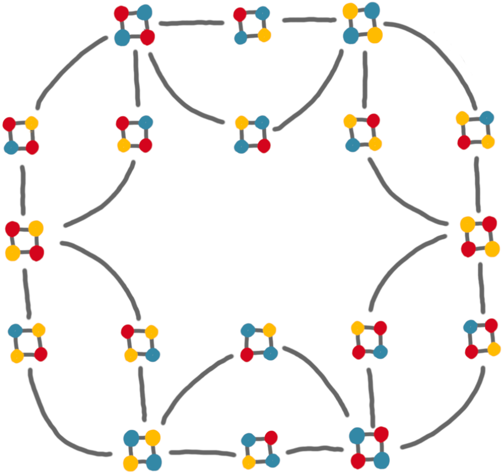


The structure of graph colorings

With an adjacency:
single-vertex recoloring



Reconfiguration graph →

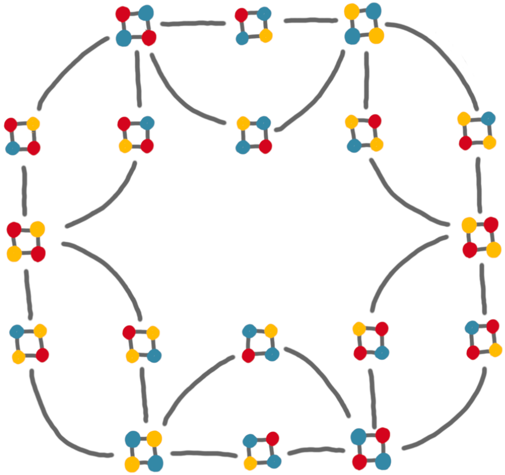


Key question # 1: Reachability/Connectivity

Setting:
For a graph G and c colors.

Algorithmic question:
Given two c-colorings of G, can I reach one from the other?

Structural question:
Is the reconfiguration graph connected?

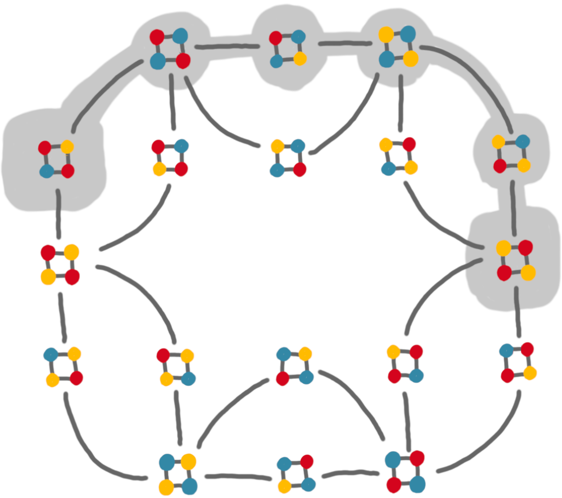


Key question # 2: Shortest path/Diameter

Setting:
For a graph G and c colors.

Algorithmic question:
Given two colorings, how fast can I go from one to the other?

Structural question:
What is the diameter of the reconfiguration graph?

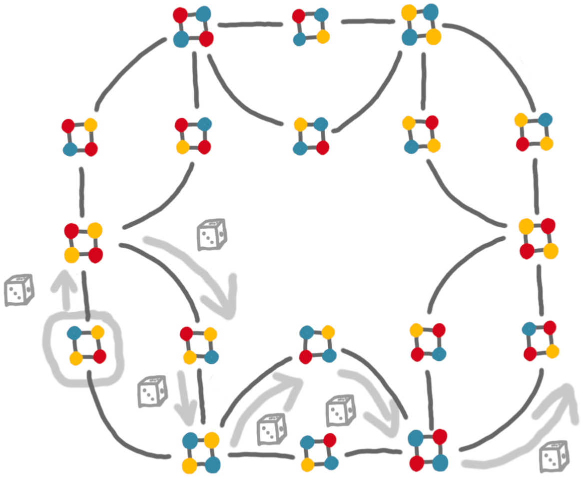


Algorithmic motivations

A generic framework:
Makes sense for any set of configurations and adjacency.

Motivations:

- Sampling via random walks
- Enumeration via local modifications
- Optimization algorithms visiting solutions (e.g. simplex)
- Updating a solution through safe local moves.

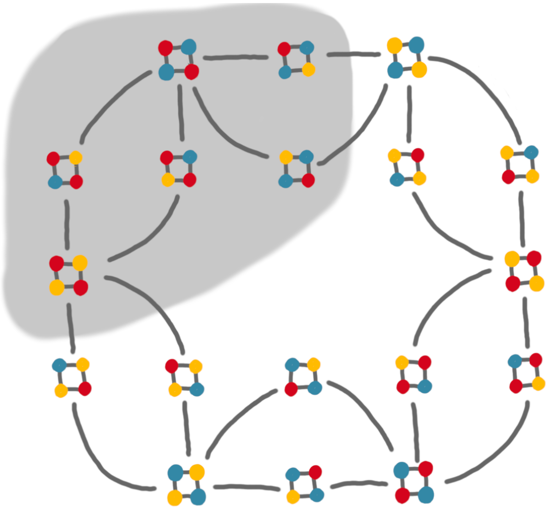


Algorithmic motivations

A generic framework:
Makes sense for any set of configurations and adjacency.

Motivations:

- Sampling via random walks
- Enumeration via local modifications
- Optimization algorithms visiting solutions (e.g. simplex)
- Updating a solution through safe local moves.

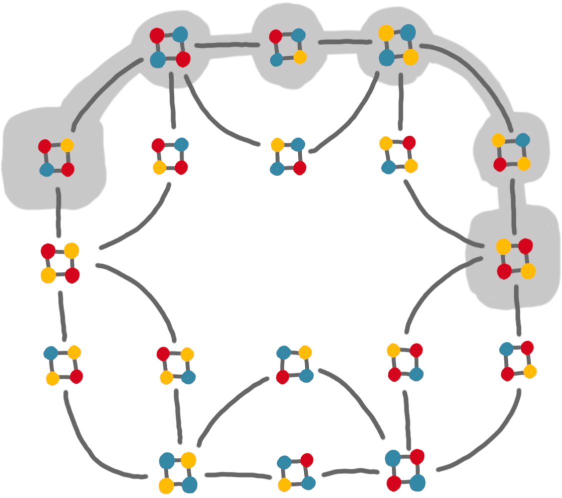


Algorithmic motivations

A generic framework:
Makes sense for any set of configurations and adjacency.

Motivations:

- Sampling via random walks
- Enumeration via local modifications
- Optimization algorithms visiting solutions (e.g. simplex)
- Updating a solution through safe local moves.



Our “theorem”

Consider a graph G

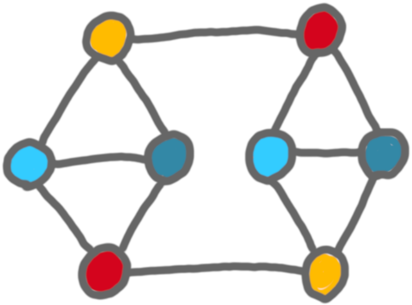
with n nodes and maximum degree Δ.

The diameter of the reconfiguration graph of $(\Delta + 1)$-colorings of G is $O_{\Delta}(n)$.

Fix #1: Frozen colorings

- A vertex is *frozen* if it cannot change color. A coloring is frozen if all nodes are frozen. (Otherwise “non-frozen”.)
- Some $\Delta + 1$ colorings are frozen.

  $\rightarrow$ Isolated vertices in the reconfiguration graph.
- Previous work theorem: Non-frozen colorings form a giant connected component, of diameter $O(n^2)$.

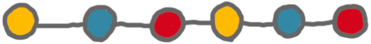


[A Reconfigurations Analogue of Brooks’ Theorem and its Consequences, Feghali, Johnson, Paulusma, 2016]

Fix #2: $\Delta = 2$ is special

- For $\Delta = 2$, our bound cannot hold: the reconfiguration graph can have diameter $\Omega(n^2)$.
- Cute lower bound.

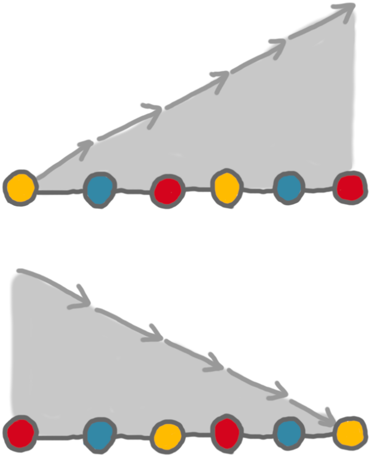
  [Reconfiguration graphs for vertex colourings of chordal and chordal bipartite graphs, Bonamy, Johnson, Lignos, Patel, Paulusma, 2014]



Fix #2: $\Delta = 2$ is special

- For $\Delta = 2$, our bound cannot hold: the reconfiguration graph can have diameter $\Omega(n^2)$.
- Cute lower bound.

  [Reconfiguration graphs for vertex colourings of chordal and chordal bipartite graphs, Bonamy, Johnson, Lignos, Patel, Paulusma, 2014]



Our theorem

Consider a graph G

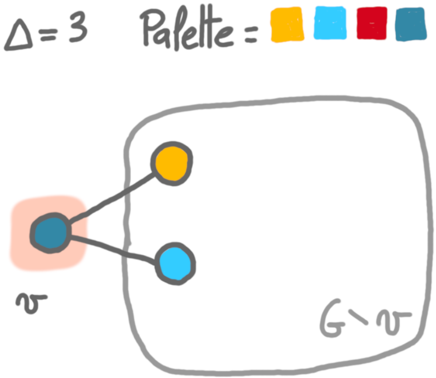
with n nodes and maximum degree $\Delta \geq 3$.

The diameter of the reconfiguration graph of non-frozen $(\Delta + 1)$-colorings of G is $O_\Delta(n)$.

Proof idea #1: Degeneracy by local warming

Classic degeneracy argument:

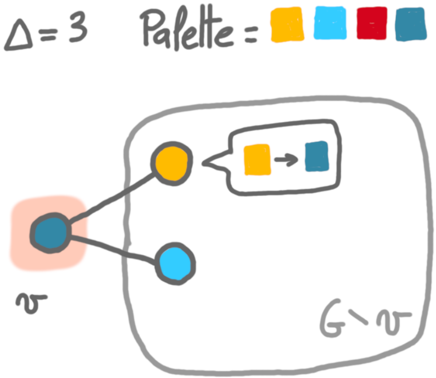
- A node of degree $< \Delta$ is always non-frozen.
- $\rightarrow$ Easy to remove/add it.
- The argument can be used recursively.



Proof idea #1: Degeneracy by local warming

Classic degeneracy argument:

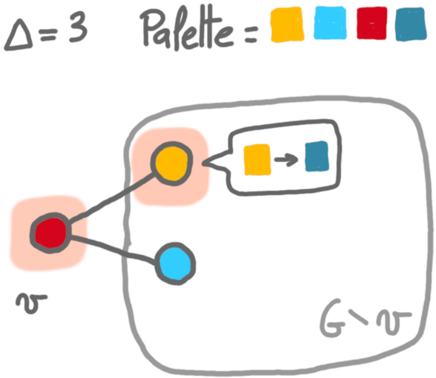
- A node of degree $< \Delta$ is always non-frozen.
- $\rightarrow$ Easy to remove/add it.
- The argument can be used recursively.



Proof idea #1: Degeneracy by local warming

Classic degeneracy argument:

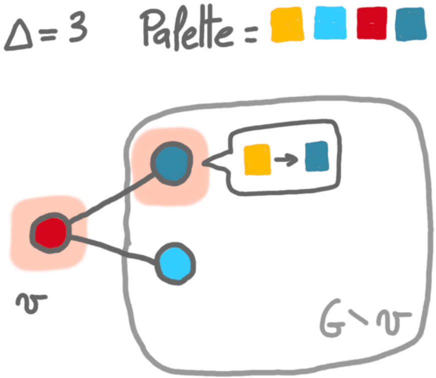
- A node of degree $< \Delta$ is always non-frozen.
- $\rightarrow$ Easy to remove/add it.
- The argument can be used recursively.



Proof idea #1: Degeneracy by local warming

Classic degeneracy argument:

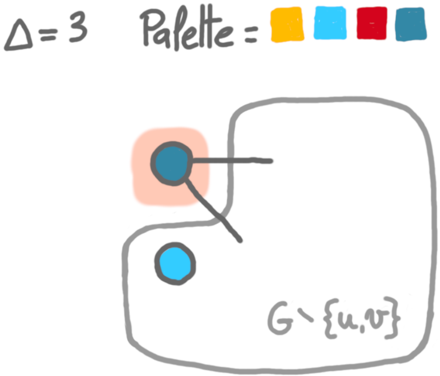
- A node of degree $< \Delta$ is always non-frozen.
- $\rightarrow$ Easy to remove/add it.
- The argument can be used recursively.



Proof idea #1: Degeneracy by local warming

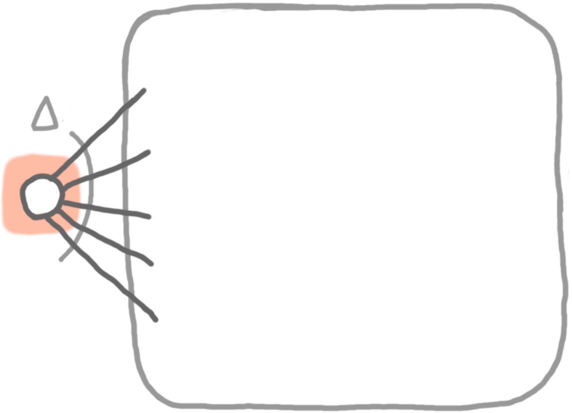
Classic degeneracy argument:

- A node of degree $< \Delta$ is always non-frozen.
- $\rightarrow$ Easy to remove/add it.
- The argument can be used recursively.



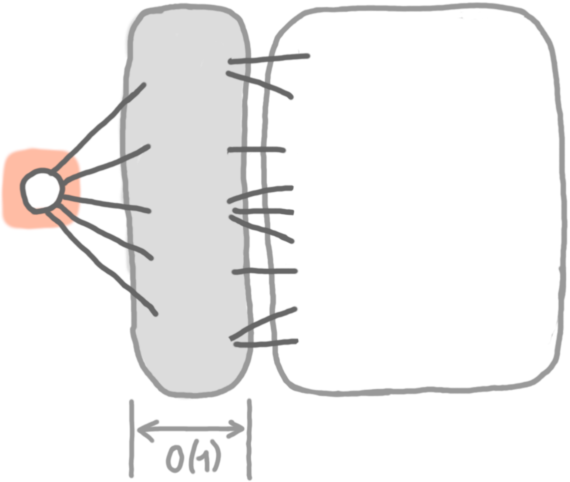
Proof idea #1: Degeneracy by local warming

- Given a non-frozen node, need a buffer of constant diameter.
- Can simulate the degeneracy argument.
- Duplicate non-frozeness on the boundary of the buffer.



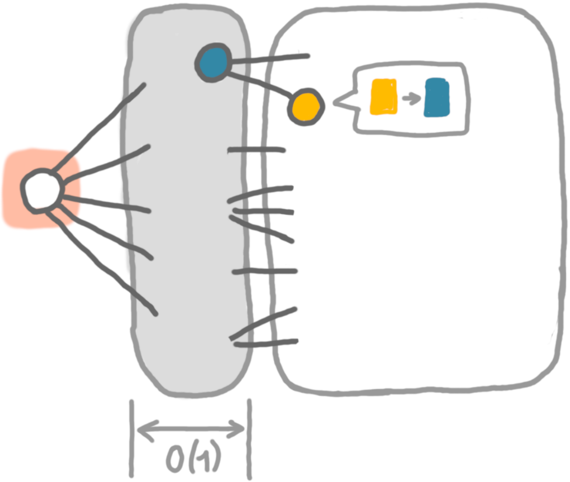
Proof idea #1: Degeneracy by local warming

- Given a non-frozen node, need a buffer of constant diameter.
- Can simulate the degeneracy argument.
- Duplicate non-frozeness on the boundary of the buffer.



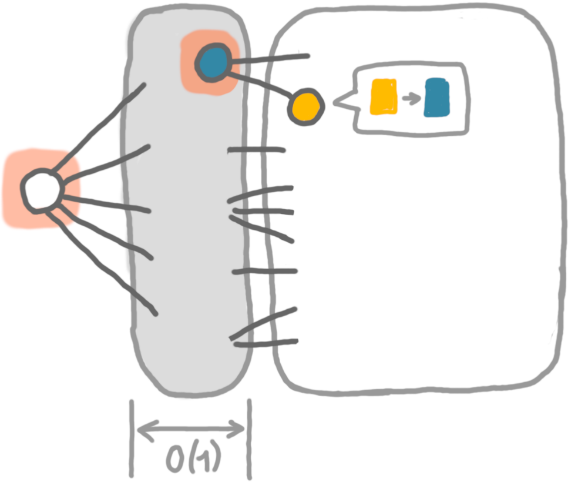
Proof idea #1: Degeneracy by local warming

- Given a non-frozen node, need a buffer of constant diameter.
- Can simulate the degeneracy argument.
- Duplicate non-frozeness on the boundary of the buffer.



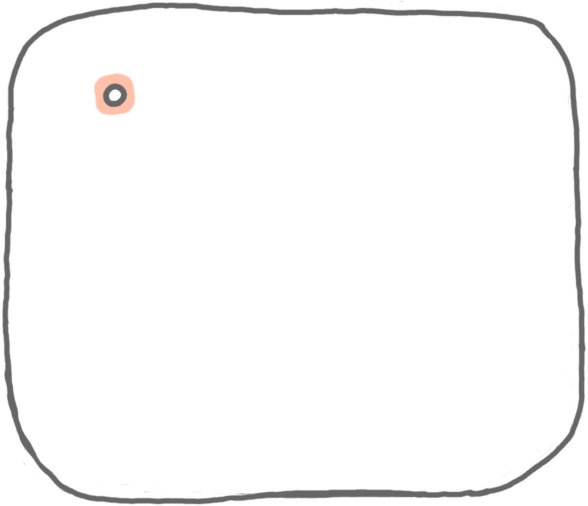
Proof idea #1: Degeneracy by local warming

- Given a non-frozen node, need a buffer of constant diameter.
- Can simulate the degeneracy argument.
- Duplicate non-frozeness on the boundary of the buffer.



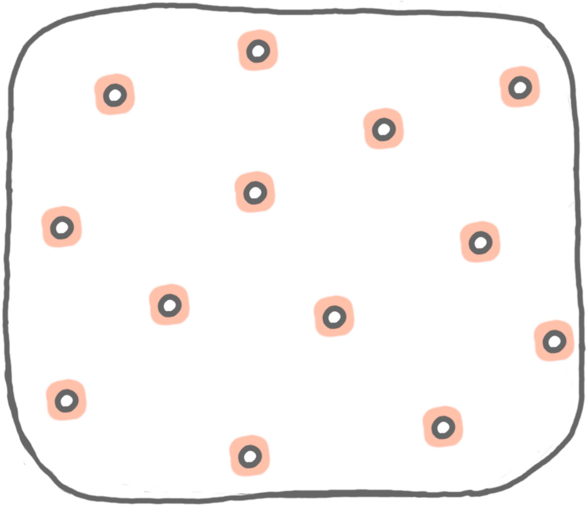
Proof idea #2: Parallelize

- Given a non-frozen vertex, we can unfreeze any well-spread set of vertices.
- Partition the graph into zones centered around the non-frozen vertices and their buffers.
- From there, the recoloring can be computed in parallel, efficiently in a distributed way.



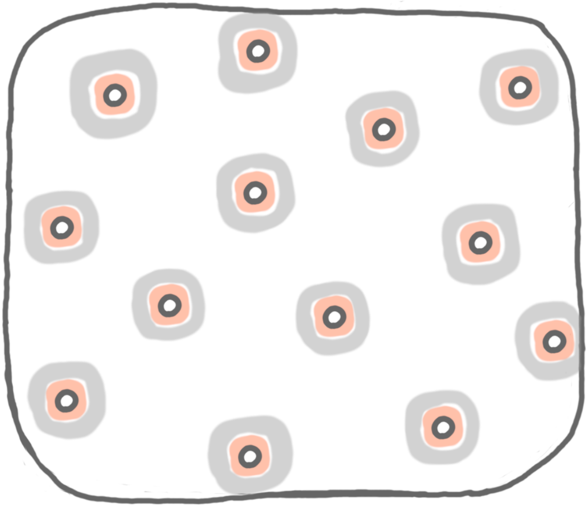
Proof idea #2: Parallelize

- Given a non-frozen vertex, we can unfreeze any well-spread set of vertices.
- Partition the graph into zones centered around the non-frozen vertices and their buffers.
- From there, the recoloring can be computed in parallel, efficiently in a distributed way.



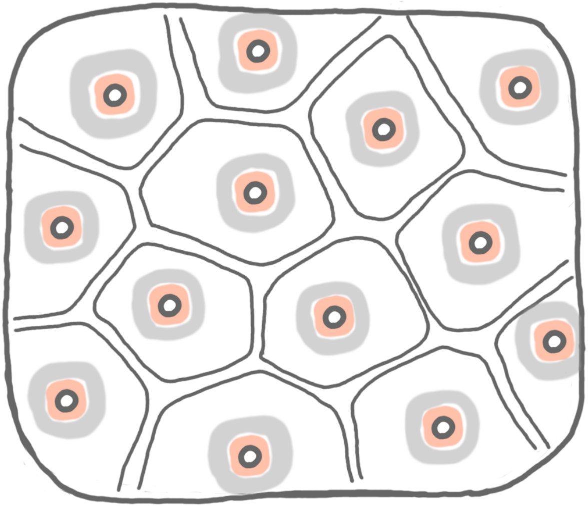
Proof idea #2: Parallelize

- Given a non-frozen vertex, we can unfreeze any well-spread set of vertices.
- Partition the graph into zones centered around the non-frozen vertices and their buffers.
- From there, the recoloring can be computed in parallel, efficiently in a distributed way.



Proof idea #2: Parallelize

- Given a non-frozen vertex, we can unfreeze any well-spread set of vertices.
- Partition the graph into zones centered around the non-frozen vertices and their buffers.
- From there, the recoloring can be computed in parallel, efficiently in a distributed way.



Back to the result and open questions

Theorem: Consider a graph G
with n nodes and maximum degree $\Delta \geq 3$.

The diameter of the reconfiguration graph
of non-frozen $(\Delta + 1)$-colorings of G is $O_\Delta(n)$.

Related open questions: Complexity in Δ, lower bounds, mixing time, palette size depending on the degeneracy, applying the distributed lens to other graph problems.