FAMNIT Mathematical Research Seminar, 29 September 2014

Finding (Shortest) Paths between Graph Colourings

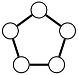
Matthew Johnson
Durham University

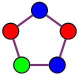
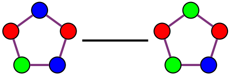
Reconfiguration graphs

Given an instance of any combinatorial search problem (colouring, independent set, clique . . .), define the reconfiguration graph:

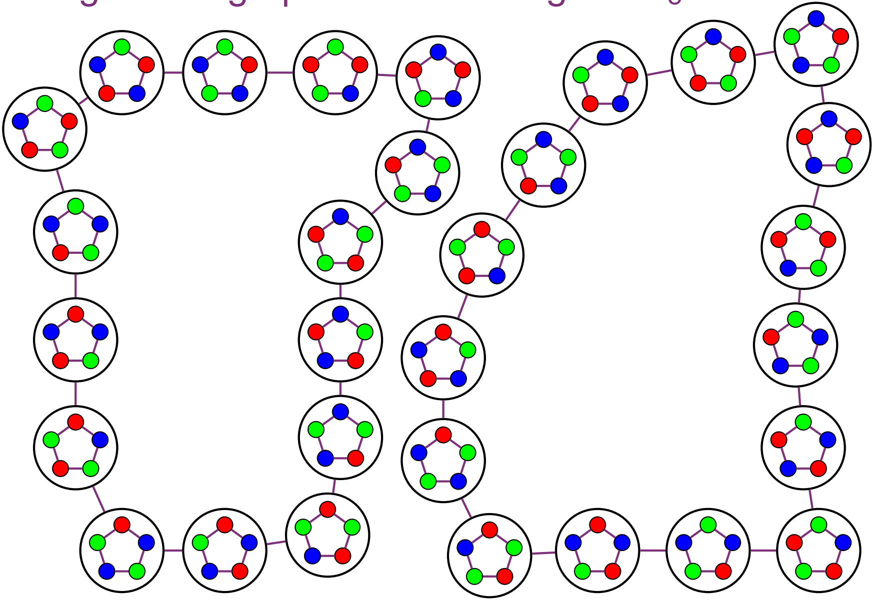
- the vertex set is the set of all feasible solutions;
- the edge relation typically relates solutions that are "close" (eg. their symmetric difference is size one).

Example: Reconfiguration graphs of vertex colourings

Given an instance  of 3-colouring,

- each feasible solution  is a vertex of the reconfiguration graph,
- and pairs of solutions  are joined by an edge if they differ in colour on exactly one vertex.

Reconfiguration graph of 3-colourings of C_5

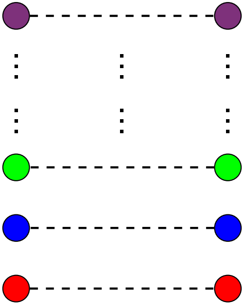


Terminology

- If two colourings differ only on a vertex v (so there is an edge between them in the reconfiguration graph), then we say v can be recoloured.
- A sequence of recolourings corresponds to a path in the reconfiguration graph.
- The available colours at a vertex are those that do not appear on any of its neighbours.

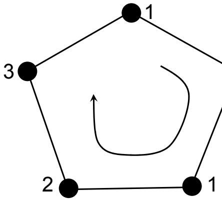
When is the reconfiguration graph connected?

There is no bound $b(\chi)$ such that, for all graphs G, for all integers $k \geq b(\chi(G))$, the reconfiguration graph of k-colourings of G is connected.

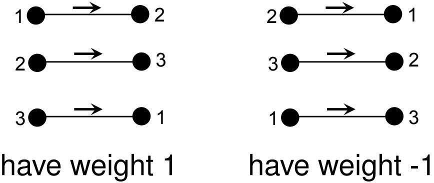


This colouring of $K_{n,n} - I$ with n colours is frozen (it is a isolated vertex in the reconfiguration graph)

3-Colourings of Cycles



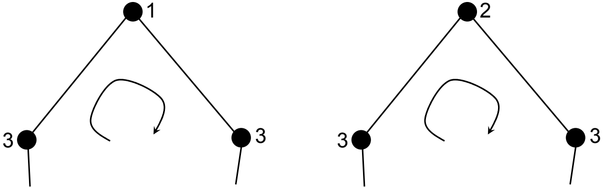
To describe a 3-colouring of a cycle we orient the cycle, and put weights on the edges.



The weight of a 3-coloured oriented cycle is the sum of the weights of its edges.

3-Colourings of Cycles

Compare the weights of cycles under 3-colourings that are adjacent in the reconfiguration graph; that is, colourings that differ on only one vertex.



- Both neighbours must have the same colour.
- So the incident edges have opposite sign, and their combined weight is zero in both colourings.
- 3-colourings in the same component of the reconfiguration graph of a cycle have the same weight.

Reconfiguration graphs of 3-colourings

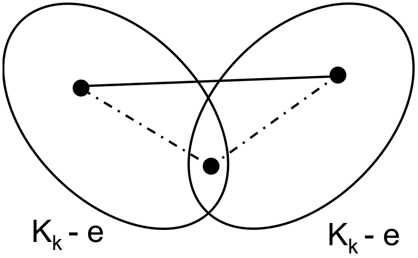
Proposition

The reconfiguration graph of 3-colourings of any 3-chromatic graph is not connected.

- A 3-chromatic graph G contains a cycle C with an odd number of vertices.
- For some 3-colouring of G, fix an orientation of C. Let w be the weight of C. Notice that $w \neq 0$.
- Obtain a new colouring by exchanging colours 1 and 2. The weight of C is $-w$.
- Thus these two colourings belong to different components of the reconfiguration graph.

k-Chromatic Graphs

For $k \geq 4$, there are k-chromatic graphs that have reconfiguration graphs that are not connected (complete graphs, for example), but also k-chromatic graphs that have connected reconfiguration graphs; for example:



Proposition

For $k \geq 2$, $\ell \geq k$ there are k-chromatic graphs whose ℓ-colourings reconfiguration graphs are not connected, but also k-chromatic graphs that have connected ℓ-colourings reconfiguration graphs unless $k = \ell = 2$ or $k = \ell = 3$.

Theorem (Cereceda, van den Heuvel, MJ 2006)

Deciding whether the reconfiguration graph of 3-colourings of a bipartite graph is connected is coNP-complete. (There is a polynomial-time algorithm for planar graphs.)

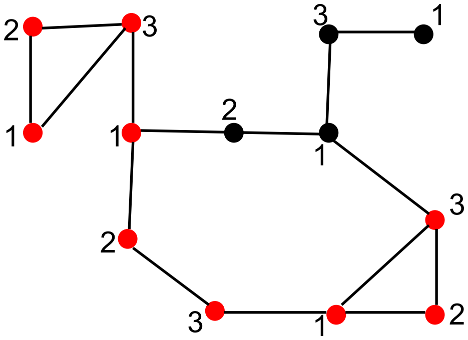
The complexity of deciding whether reconfiguration graphs of k-colourings are connected is open for $k \geq 4$.

Reconfiguration problem

- The reconfiguration problem is to decide whether two solutions belong to the same component of the reconfiguration graph?
- Let's start with 3-colourings: we know that the colourings induce weights on cycles and that recolouring vertices cannot change the weight. So two colourings belong to the same component only if every cycle has the same weight in both colourings.
  - Is this condition sufficient?
  - Can it be checked (in polynomial time)?

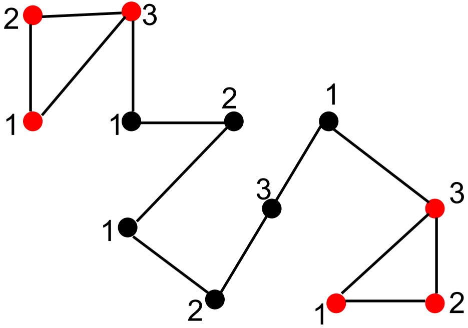
Locked Vertices

A vertex *v* in a graph coloured with α is locked if it has the same colour in every colouring in the same component as α of the reconfiguration graph of 3-colouring.



Fixed Paths

If a path joins two locked vertices then it is a fixed path: no sequence of recolourings will change the weight of the path (the weight of a path is the sum of the weights of its edges under some orientation of the path).



Theorem (Cereceda, van den Heuvel, MJ 2008)

Two colourings α and β of a graph G on n vertices and m edges belong to the same component of the reconfiguration graph of the 3-colourings of G if and only if

- *every oriented cycle and fixed path of G has the same weight under α and β, and*
- *the locked vertices of G are the same — and have the same colours — under α and β.*

There is an $O(n^2)$ algorithm that will either find a path from α to β or exhibit one of these obstacles.

Finding Locked Vertices

To find the locked vertices of a 3-coloured graph:

- Let B_1, B_2 and B_3 be three sets of vertices initially equal to the colour classes.
- Remove a vertex from B_i if it is not adjacent to both a vertex in B_j and a vertex in B_k (i, j, k distinct)
- Continue removing vertices as long as possible.
- The vertices that remain in B_1, B_2 and B_3 are locked.

This process can be done using a modified breadth-first search in time $O(m)$.

Algorithm for connecting 3-colourings

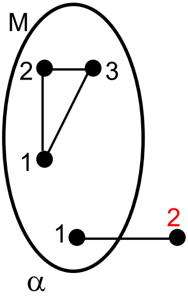
- If two colourings α and β of G have the same set of locked vertices (and they are coloured alike), then identify like-coloured locked vertices (so there are three locked vertices which induce K_3).
- This transforms fixed-weight paths into cycles — so now the only obstacle is cycles of different weights.

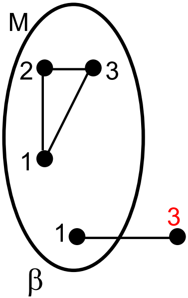
Algorithm for connecting 3-colourings

- Aim: to recolour from α to β
- We can assume G has 0 or 3 locked vertices and that it is connected.
- Let M be a connected set of vertices on which α and β agree (assume M contains the locked vertices).
- Let u be a vertex adjacent to M with $\alpha(u) \neq \beta(u)$.

Our approach is to recolour so that u is coloured with $\beta(u)$ and no vertex in M is recoloured.

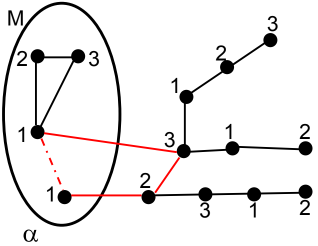
Algorithm for connecting 3-colourings





- Assume $\alpha(u) = 2$, $\beta(u) = 3$; so all the vertices in M adjacent to u are coloured 1.
- Want to recolour from α so that u is coloured 3 (and the colours in M don't change).

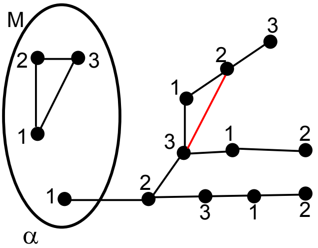
Algorithm for connecting 3-colourings



From u, do a depth-first search: from each vertex v look for vertices w such that $\alpha(w) = \alpha(v) + 1 \mod 3$

If at some point in the search a vertex in M is found, then we can find a cycle whose weight is different under α and β.

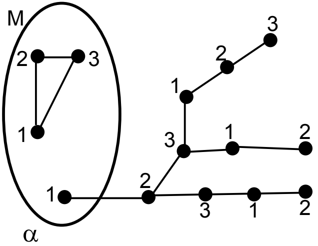
Algorithm for connecting 3-colourings



From u, do a depth-first search: from each vertex v look for vertices w such that $\alpha(w) = \alpha(v) + 1 \mod 3$

During the search a vertex cannot find its descendant as this would imply a cycle of locked vertices outside M.

Algorithm for connecting 3-colourings



From u, do a depth-first search: from each vertex v look for vertices w such that $\alpha(w) = \alpha(v) + 1 \mod 3$

Use the depth-first tree obtained to recolour *u* but not *M*.

Visit the vertices in the order that they were finished by the search and increase the colour by $1 \mod 3$.

More than 3 colours

Theorem (Bonsma and Cereceda 2008)

Deciding whether two colourings are in the same component of the reconfiguration graph of k colourings is **PSPACE***-complete for* $k \geq 4$.

- Restriction to bipartite graphs is also **PSPACE**-complete (and to planar graphs for $k \in \{4, 5, 6\}$).
- There are reconfiguration graphs of k-colourings with superpolynomial diameter.

Reconfigurations of colourings of chordal graphs

A graph is chordal if it has no induced cycle of length more than 3.

Theorem (Bonamy, MJ, Lignos, Paulusma, Patel 2010)

For any chordal graph G on n vertices with chromatic number k, for all $\ell \geq k+1$, the reconfiguration graph for ℓ-colourings of G is connected with diameter at most $2n^2$, and paths between pairs of ℓ-colourings can be found in polynomial time.

More generally . . .

Definition

A graph G is *k-colour-dense* if either

(i) it is the disjoint union of cliques each with at most k vertices, or

(ii) it has a separator S such that $G - S$ has components D_1 and D_2 with vertices $u \in D_1$ and $v \in D_2$ and

  (a) $|V_{D_1}| \leq \max\{1, k - |S|\}$,

  (b) $S \subseteq N(v)$, and

  (c) identifying u and v results in a k-colour-dense graph.

Theorem (Bonamy, MJ, Lignos, Paulusma, Patel 2010)

For any k-colour-dense graph G on n vertices, for all $\ell \geq k + 1$, the diameter of the reconfiguration graph for ℓ-colourings of G is at most $2n^2$.

Shortest paths between colourings

SHORTEST PATH RECONFIGURATION
Instance: graph G, k-colourings α and β, positive integer ℓ
Question: Is there a path between α and β in the reconfiguration graph of k-colourings of length at most ℓ?

Theorem (MJ, Kratsch, Kratsch, Patel, Paulusma 2014)

There is an algorithm for Shortest Path Reconfiguration with running time $O((k\ell)^{\ell^2+\ell} \cdot \ell n^2$ (fixed-parameter tractable)

Algorithmic approach

- The aim is to find a path in the reconfiguration graph from α to β of length at most ℓ — that is, a sequence of k-colourings $\alpha = c_0, c_1, \ldots, c_\ell = \beta$ such that each colouring differs from the last on at most one vertex.
- First observation: If α and β differ on more than ℓ vertices there is no path.
- Second observation: If a vertex v has more than ℓ neighbours of colour q, then v will not be coloured q in any c_i.

Lemma

There is a set A^* of size at most $\ell \cdot (k\ell)^{\ell}$ such that the colours of all vertices not in A^* are fixed on a shortest path from α to β.

- The idea is that to recolour a vertex with $\alpha(v) \neq \beta(v)$ we first have to recolour its neighbours.
- This argument cascades for ℓ steps, but the number of neighbours considered is bounded.

Then use brute force to search for a sequence of ℓ recolourings of vertices of A^*.

Theorem (MJ, Kratsch, Kratsch, Patel, Paulusma 2014)

For $k = 3$, there is an polynomial-time algorithm for Shortest Path Reconfiguration

- We already saw an algorithm to find a path between 3-colourings. This path finds the shortest path unless there are no locked vertices.
- If there are locked vertices, the problem is where to start the algorithm: we showed that the optimal path could be a found from few guesses.

Open problems

- What is the complexity of deciding whether the reconfiguration graph of k-colourings is connected, $k \geq 4$.
- How many extra colours do you need to connect a pair of k-colourings.
- What is the complexity of finding a path between a pair of solutions of, say, the Travelling Salesman Problem.