CoRe 2019 - Open Problems

Alice Joffard[*] Paul Ouvrard[†]

Amer Mouawad

First problem We are interested in the reachabilty problem of Vertex Cover Reconfiguration under the TAR rule. More precisely, the problem is the following: we are given a graph G and two vertex covers C_s and C_t (each of size at most k) and we want to decide if there is a TAR-sequence between them such that each intermediate solution has also size at most k. What is the complexity of this decision problem if G has treewidth at most 2? Or more simply if G is an outerplanar graph: is it polynomial-time solvable, NP-complete or PSPACE-complete? Actually, this is not clear even if G is a cycle with few chords that do not intersect.

Second problem Here, we are interested in the parameterized complexity of $(s-t)$-cuts reconfiguration under the token jumping rule. More formally, we are given a graph, two vertices s and t and two $(s-t)$-cuts S_1 and S_2 of size at most k. Is this problem FPT when parameterized by k? Does it have a kernel parameterized by k or is it NP-complete?

Update: $(s-t)$-cuts is NP-complete. The reduction is from Vertex Cover Reconfiguration on bipartite graphs.

Jan van den Heuvel

A k-coloring with deficiency D of a graph G is a (not necessarily proper) vertex-coloring of G with at most k colors such that the graph induced by each color class has maximum degree D. Given a graph class, what is the smallest integer k such that there exists a k-recoloring with deficiency $O(1)$? For instance, it is known that planar graphs have a 3-coloring with deficiency 2. Here, we are interested in the reconfiguration version where at each step, we are allowed to recolor a single vertex. Does there exists a constant c, such that for any planar graph G, all 3-colorings of G with deficiency c are reachable one from another? More generally, what about graphs embedded in surfaces? (the constant c would

[*]alice.joffard@gmail.com
[†]paul.ouvrard@u-bordeaux.fr

then depend on the genus). Similarly, for general graphs, if the number of colors is a function of the degeneracy, does there exist an absolute constant?

Update: there is no constant c that depends on the genus of the surface S such that the reconfiguration graph of all 3-colorings with deficiency c for any graph embeddable in S is connected. However, given a surface S of genus g, there exists $c = O(g)$ such for any graph G embeddable in S, the collection of c-defective 4-colorings is connected. What about K_{t+1}-minor free graphs? It is known that if G is K_{t+1}-minor free then it is t-colorable with defectiveness $O(t)$. For the recoloring version, we know that we need at least $t+1$ colors.

Carl Feghali

First problem We focus on graph recoloring where we are allowed to recolor exactly one vertex at each step. Given a perfect graph G of maximum degree Δ and an integer $k \leq \Delta + 1$, we know that the reconfiguration graph $R_k(G)$ is disconnected (the counterexample is given by a complete bipartite graph $K_{k,k}$ where we remove a perfect matching). On the other hand, it is known that if k is at least $\Delta + 2$, then $R_k(G)$ is connected. For chordal graphs (which are also perfect), we also have a complete characterization since $R_{\omega+1}$ (where ω is the size of a largest clique of G) is connected and has diameter $O(n^2)$.

A graph is *weakly chordal* if it contains neither a C_h nor its complement for every $h \geq 5$ as an induced subgraph. Let G be a weakly chordal graph. It is known that $R_{\omega+1}(G)$ is not necessarily connected but what is the smallest k such that $R_k(G)$ is connected? Is $k = \omega + 2$ sufficient?

Second problem The Oberwolfach problem consists in finding a decomposition of the edges of K_n into edge-disjoint copies of disjoint cycles of given lengths (2-factors of given sizes). Consider a coloring of the edges of K_n. We ask ourselves what necessary and sufficient conditions guarantee that we can extend it to K_{n+m}, in such way that the color classes will exactly form the disjoint cycles of given lengths.

An idea would be to find one copy of them, multiply the edges of this copy, then use a sequence of well-chosen edge flips (from the edges ab and cd, we form the edges ac and bd) to remove the multiple edges and create the other copies. This method has been used by Hilton and Johnson in *An algorithm for finding factorizations of complete graphs* to find a hamiltonian cycle decomposition of K_n, and could also be applied to the Oberwolfach problem.

Jonathan Noel

(Question proposed by Jan Volec)

A fractional $(a:b)$-coloring of G is an assignment of b different colors from a set of size a to every vertex of G so that any two adjacent vertices have disjoint sets of colors.

What is the complexity of reconfiguring fractional colorings? (Changing at each step the set of colors assigned to a vertex).

Benjamin Hellouin

First problem Consider an undirected graph H (which might contain loops). We construct a reconfiguration graph $\mathrm{Walk}_n(H)$ where the vertices are the walks of length n on the vertices of H, and two walks $(x_1, x_2, ..., x_n)$ and $(y_1, y_2, ..., y_n)$ share an edge iff x_i and y_i are adjacent in H for every i.

The question is the following: How does the diameter of $\mathrm{Walk}_n(H)$ grows with n ? In particular, the conjecture is that it is either in $\mathcal{O}(1)$ or in $n + \mathcal{O}(1)$ depending on the graphs (the characterization remains to be found).

Update: The conjecture was already proved for C_4-free graphs. Jan van den Heuvel and Benjamin Hellouin proved it for graphs whose cycles are not sums of C_4.

Second problem We say that a graph H has the pivot property if for any two homomorphisms h_1 and h_2 from $\mathbb{Z}^2$ to H that differ on finitely many points, we can go from h_1 to h_2 by changing one point at a time while keeping a homomorphism. This property can be true for some graphs like C_4-free graphs, and false for other ones like K_4 and K_5.

We focus here on the generalized pivot property, where we can change several points at a time at each step, but there should exist a radius $r > 0$ such that we only change at each step points within a ball of radius r. Again, the property holds if the reconfiguration graph is connected.

A conjecture is that the property of the generalized pivot holds for any H.

Update: this conjecture has already be proven to be false.

Alice Joffard

For an ordered sequence $S = (d_1, ..., d_n)$ of natural numbers, we say that a graph G on the ordered vertices $v_1, ..., v_n$ realizes S if for all i, $d(v_i) = d_i$.

We construct a reconfiguration graph where the vertices are the graphs that realize S, and where two graphs G and H share an edge iff we can go from G to H by performing a flip, which is a swap from the two disjoint edges (a, b) and (c, d) to the two disjoint edges (a, c) and (b, d). Hakimi showed that the reconfiguration graph was connected. Then came the question of finding a Polytime algorithm to go from any G to any H using the minimum number of flips. The best result so far is a 1.5-approximation algorithm given by Bereg and Ito in 2017.

We are interested here in another version of this problem, where G and H are both connected, and the graph has to stay connected after any flip. Again, it is known that we can go from any G to any H using such flips, and we want to find a Polytime algorithm to approximate the optimal number of flips needed.

The problem being easier in the case where G and H have cycles, we can restrict ourselves to the case where both are trees. The best result we have so far is a 2.5-approximation, and we would like to improve it.

Duc A. Hoang

We are interested here in the Token Swapping problem. Let us consider a set of tokens $\{1, 2, ..., n\}$ placed on the vertices $\{v_1, v_2, ...v_n\}$ of a graph G in any order. We want to move the tokens so that each token i will be placed on v_i, by applying a sequence of swaps between two tokens whose position are adjacent in G.

We know that the reconfiguration graph is connected, so that we can always do it, but we now want to do it using the minimum number of swaps possible. The complexity of this problem when G is a tree is unknown.

Valentin Bartier

Let G be a bipartite graph and I_1, I_2 be two independent sets of G of size k. We already know that the reconfiguration problem under the TJ rule is NP-complete. However, is it FPT when parameterized by k?

Aline Parreau

The Eternal Domination problem on a graph G can be seen as an infinite game between a defender, and an attacker. The defender starts by choosing a set of vertices. At turn i, the attacker attacks a vertex r_i and the defender must defend by moving to r_i a guard from an adjacent vertex. All the other guards can also move to an adjacent vertex. The next turn starts from this new configuration.

The defender wins the game if it can defend against any infinite sequence of attacks. The eternal domination number, denoted by $\gamma_m^\infty(G)$, is the minimum number of guards necessary for the defender to win.

We know that determining the value of $\gamma_m^\infty(G)$ is a NP-hard problem, but we do not know if it is in NP, or in PSPACE.

Marc Heinrich

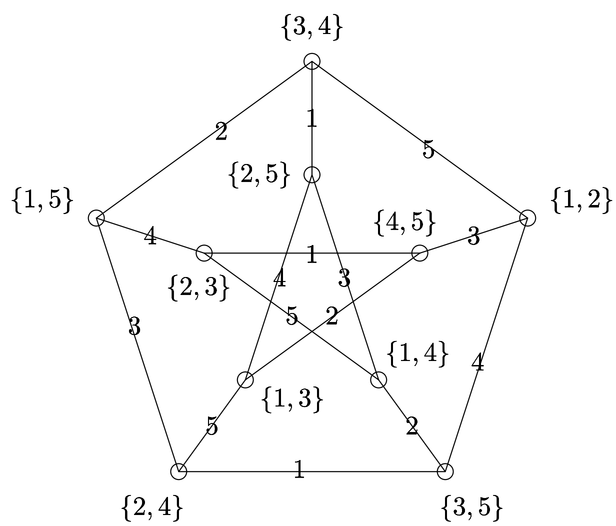
When it comes to the VERTEX COLORING RECONFIGURATION problem, it is known that for $k \geq \Delta+2$ with Δ the maximum degree of G, the reconfiguration graph $R_k(G)$ is connected, meaning that we can go from any coloring with at most k colors to any other one, with a sequence of modifications of a single vertex's color, maintaining the coloring proper.

The same question for the EDGE COLORING RECONFIGURATION problem can then be asked: what is the minimum value k_{min} of k, as a function of Δ, from which we can guarantee that the reconfiguration graph of proper k-edge colorings is connected ?

We already know that $\frac{3}{2}\Delta \leq k_{min} \leq 2\Delta$.

Update: This question has just been solved by Aline Parreau, Jonathan Noel, Marc Heinrich and Alice Joffard: the following coloring of Petersen graph with 5 colors is frozen, and we can generalize this example to the Kneser graph $KG_{2\Delta-1,\Delta-1}$ of degree Δ, colored with $2\Delta - 1$ colors in such a way that every edge is assigned the color corresponding to the missing number in the union of the sets of its two vertices. This proves that $k_{min} = 2\Delta$.

A question that remains is the one of the example that was used to prove $\frac{3}{2}\Delta \leq k_{min}$: we know that for edge colorings of $K_{n,n,n}$ with $3n$ colors, the reconfiguration graph is disconnected, but is it connected with $3n + 1$ colors ? In particular, can we go from any coloring of $K_{n,n}$ with n colors, where every color class forms a perfect matching, to any other one, using $n + 1$ colors ?



Haruka Mizuta

Given an undirected graph $G = (V, E)$ and a subset of vertices $S \subseteq V$ called terminals, a *Steiner tree* is a tree that contains S. In the Steiner Tree Reconfiguration problem, we are given two Steiner trees T_1 and T_2 with the same number of edges and we want to find a transformation between T_1 and T_2 by exchanging exactly one edge at each step. This problem has been studied under the parameterized complexity framework under seven graph parameters: ω, the treewidth, the maximum degree Δ, the bandwidth, the minimum size of a vertex cover, the number of terminals, the solution size k (ie. the number of edges). For each of them, the complexity (W[1]-hardness, PSPACE-hardness of tractability) is already known. This also known for some combinations of these parameters. However, it is still open when parameterized by k+tw.

Nicolas Bousquet

Given a hypergraph $H = (V, E)$, a *packing* is a subset of pairwise disjoint hyperedges (therefore, this is a generalization of matching for instance). Bonamy et al. recently proved that the Matching Reconfiguration problem is polynomial-time solvable. Unfortunately, this result can not be extended to Packing Reconfiguration since it is already knwown that this problem is PSPACE-complete under the token jumping rule (ie. we can remove and add one hyperedge at each step). However, the maximum packing problem is in P if the constraint matrix is Totally Unimodular ie. the determinant of each square submatrix is 0, -1 or +1. Therefore, can we extend this result to the Packing Reconfiguration problem for the connectivity question? One idea could be to use a kind of Seymour's Decomposition Theorem.

Akira Suzuki

We focus on Feedback Vertex Set Reconfiguration under the token jumping rule. This problem is PSPACE-complete in general. However, we are interested in the case where the input graph G has bounded maximum degree Δ. If $\Delta = 2$, this is trivial because G simply is a collection of paths and/or cycles. If $k = 4$, Suzuki and Yota showed that the problem is PSPACE-complete. However, it is still open for $\Delta = 3$. We know that there exist no-instances: for instance if $G = K_{3,3}$ and the two solutions are two non-adjacent vertices. Note that the original problem is in P when $\Delta = 3$.

František Kardoš

A *stable-tree decomposition* of G is a partition of the vertex set on two parts B and W such that $G[B]$ is a tree and $G[W]$ is a stable set. Payan and Sakarovitch proved that if G is a cyclically 4-edge-connected cubic graph and $n = 4k + 2$

then a stable-tree decomposition exists. The question is the following: for which (planar) cubic graphs can we guarantee to have (exponentially) many stable-tree decompositions? What can we say about the structure of the reconfiguration graph where a vertex is a stable-tree decomposition and two vertices (B_1, W_1) and (B_2, W_2) are adjacent iff W_2 is obtained from W_1 by removing a vertex v and adding the only vertex that lies in the intersection of the three paths joining the three black neighbors of v in $G[B_1]$?

Update: if G is cubic and planar, then it has at least n^3 stable-tree decompositions. Using the planarity of G, it might be possible to increase this exponent.

Kunihiro Wasa

We are interested in the problem Maximal Induced Tree Reconfiguration. The input of the problem is two maximal induced trees S and T of G, and we want to know if there exists a reconfiguration sequence between S and T, by Token Sliding or Token Jumping. The question is to determine the complexity of the problem.

Update: The problem has been solved by Haruka Mizuta. Using Steiner trees, we can prove that it is W[2]-hard.

Anna Lubiw

First problem Consider the reconfiguration graph of non-crossing spanning trees of a set of n points in the plane. A flip removes one edge and adds a new edge.

1. The worst case diameter is known to be between $1.5n$ and $2n$. Narrow this gap.
2. What is the complexity of computing distance in the reconfiguration graph?

Second problem Consider labelled flips of non-crossing spanning trees of a set of n points in the plane. Does the analogue of the orbit theorem hold? Prove or disprove: if we have two labelled non-crossing spanning trees S and T, and for each label λ, the edge e of S with label λ and the edge f of T with label λ lie in the same orbit (i.e. there is a flip sequence that moves λ from e to f) then S can be reconfigured to T.

Third problem A *branching* rooted at vertex r in a directed graph G is a directed tree rooted at r (i.e. it is a tree and every vertex except r has one edge directed in to it). This is a simple case of intersection of two matroids: a graphic matroid, where a set of edges is independent if it contains no (undirected) cycle;

and a partition matroid, where a set of edges is independent if every vertex has at most one incoming edge. Let S and T be two branchings rooted at r.

Question: Is it possible to reconfigure S to T? A flip removes one edge and adds a new one.

Answer: Yes. Here is the proof outline. Let $e = (u, v)$ be an edge of $T - S$. We try to add e to S and recover a branching by removing one element of S. Let e_S be the edge of S that is directed into v. Adding e to S gives two edges, e and e_S directed into v and also creates a cycle C. We want to remove one edge of S to remedy both these things. Consider 3 cases:

1. if u is an ancestor of v in S, then the cycle C contains e_S so removing e_S yields a branching.
2. if u and v are incomparable in S, the same holds.
3. if u is a descendant of v in S, then follow the path backwards in T from v to r. There must eventually be an edge e' of $T - S$ that falls into case 1 or 2.

Follow-up question: Does the above extend to the intersection of a graphic matroid and a partition matroid that arises by colouring the edges of the underlying graph G and defining a set to be independent if it has at most one edge of each colour.

Spanning trees with one edge of each colour are called *rainbow* spanning trees. Thus, the question is whether one rainbow spanning tree can be reconfigured to another rainbow spanning tree by exchanging one edge at a time.

Answer: No. Here is an example found by Ruth Hass and Anna Lubiw. In this example, the edges of each colour are all incident to one vertex, so it is very close to the branching situation — the only difference is that one vertex, t, has two colour classes associated with it.

The graph G has 5 vertices, u_1, u_2, v_1, v_2, t. The first spanning tree S contains the paths u_1, u_2, t and v_1, v_2, t. The second spanning tree T contains the paths v_1, u_2, t and u_1, v_2, t. There are no further edges in G. The colour classes are $\{u_1u_2, v_1u_2\}$, $\{v_1v_2, u_1, v_2\}$, $\{u_2t\}$, and $\{v_2t\}$. We claim that S is frozen. By symmetry, it suffices to show that we cannot add the edge v_1u_2 to S. If this edge is added, we must remove the same-colour edge u_1u_2. But then vertex u_1 becomes isolated.

Moritz Mühlenthaler had a similar example.