Combinatorial Reconfiguration

Naomi Nishimura (University of Waterloo),
Takehiro Ito (Tohoku University),
Amer Mouawad (University of Bergen)

January 22, 2017–January 27, 2017

1 Overview of the Field

Reconfiguration is the study of relationships among solutions to an instance of a problem, and in particular the step-by-step transformation from one solution into another such that the intermediate result of each step is also a solution. As a concrete example of a reconfiguration problem, we consider the assignment of customers to power stations such that each customer obtains as much power as is required without exceeding the capacity of what each station can produce. Each such assignment can be viewed as a solution. When a station needs to be repaired, it may be necessary to change to another assignment, moving a single customer at each step in order to minimize disruptions. Commonly-asked questions in the area of reconfiguration include both structural questions concerning the solution space and algorithmic questions about the complexity of solving various related problems.

For any combinatorial problem, there are multiple ways of defining the solution space, and for each such definition, there are a variety of related problems. A *reconfiguration graph* for an instance of a problem can be defined as a set of nodes, one for each solution to the problem (where there may be constraints with respect to solution size), and a set of edges, where there is an edge between any two nodes that are adjacent (for some definition of adjacency). The definition of adjacency gives rise to the notion of a *reconfiguration step*, an operation by which a solution is transformed into an adjacent solution. The sequence of solutions traversed in a sequence of reconfiguration steps, or equivalently, a path in the reconfiguration graph through the nodes associated with those solutions, is called a *reconfiguration sequence*.

Typical questions that arise include the following:

- Given two solutions, does there exist a reconfiguration sequence transforming one into the other? This is known as the *reachability problem*.
- Given two solutions, what is the length of the shortest reconfiguration sequence transforming one into the other? This is the decision version of the *shortest transformation problem*.
- Given two solutions, produce a reconfiguration sequence (or a shortest reconfiguration sequence). This is the search version of the *shortest transformation problem*.
- Given a problem, under what additional constraints (if any) is it guaranteed that for any instance the reconfiguration graph is connected? This is the *connectivity problem*.

- Given a problem, under what additional constraints (if any) is it guaranteed that for any instance the reconfiguration graph has a bound on the diameter?

For a given problem, there may be more than one natural definition of adjacency (and, corresponding to each such definition, the notion of a reconfiguration step). For the problem of *independent set*, where the solution is a subset of the vertices such that there is no edge between any of the vertices in the set, we can view a reconfiguration step as changing the position of a *token*, where there is a token on each vertex in the independent set. This immediately implies at least three possible ways of moving tokens: *token jumping (TJ)* [14], in which a token is moved from a vertex to any other vertex in the graph, *token sliding (TS)* [8, 14], in which a token is moved from a vertex along an edge to a neighbouring vertex, and *token addition/removal (TAR)* [11, 14], where in each step a token is added or removed. For the problem of k-*colouring*, where each vertex in a graph is assigned a colour such that for each edge the endpoints have different colours, a reconfiguration step could consist of changing the colour of a single vertex or swapping two colour classes. In yet another example of adjacency, for the problem of satisfiability of Boolean formulas, a reconfiguration step might consist of the flipping of a single variable in the truth assignment (from true to false or from false to true).

For the reconfiguration steps of token jumping and token sliding, all feasible solutions have the same number of tokens. In other situations, it may be necessary to give a bound on the size of solutions under consideration. As an example problem, we consider the problem of *vertex cover*, where the goal is to determine a subset of the vertices of a graph such that each edge in the graph has at least one endpoint in the set. If we take as our feasible solutions all vertex covers, of any size, then it is easy to see that we can find a reconfiguration sequence from any vertex cover to any other: at each step we add one of the vertices in the target solution (adding a vertex to a solution gives another solution) and after all have been added, removing a vertex that is not in the target solution (which is still a solution, as it contains all the vertices in the target solution). A more interesting problem results from restricting the size of solutions to be at most k or $k+1$, where k is the size of a minimum solution.

In addition to characterizing the solution spaces of classical computational problems (as well as solutions to puzzles), reconfiguration is related to problems in a variety of areas. In the area of network security, a network can be represented as a graph where each vertex corresponds to a server and each edge to a communication link; a step-by-step change in firewalls corresponds to the reconfiguration of a cut. In the Frequency Assignment Problem, the goal is to assign frequencies to users of a wireless network such that the interference between them is minimized and the range of used frequencies is as small as possible. Frequencies can be viewed as colours, giving rise to a colouring problem. Due to the frequent changes in demand and addition of new transmitters as well as the difficulty of finding optimal assignments, there is a need to change between assignments. Reconfiguration gives a method of doing so without interrupting service for all customers. As a final example, the moving of objects on a plane (modeled by moving tokens on a graph) has applications in the 3D-printing industry, where a "head" follows a path in laying down a layer of material. The use of multiple heads to reduce printing time necessitates ways to plan movement without collisions and, ideally, minimizing the distance traveled.

2 Recent Developments and Open Problems

As the field is quite young, all progress in the field qualifies as recent developments. Below are listed various types of results along with related open problems.

2.1 Defining Adjacency

Relations among different types of reconfiguration steps have been demonstrated; for independent set reconfiguration, TJ and TAR are equivalent [14], and for clique reconfiguration, TJ, TAR, and TS are all equivalent [12]. In this context, problems are equivalent if an instance is solvable using one type of reconfiguration step if and only if it is solvable using another type of reconfiguration step.

Open problems in this area include defining adjacency for problems with solutions that are, for example, given as sequences rather than sets, or for areas such as computational geometry.

2.2 Structural Results

Structural results focus on properties of the reconfiguration graph; most results in the category specify when the reconfiguration graph is connected. Most of the results of this type focus on the problems of k-colouring and *dominating set*. In the latter problem, the goal is to find a set of vertices such that each vertex outside the set has a neighbour in the set.

There are many open problems with respect to not only connectivity of the reconfiguration graph, but also other properties, such as k-connectivity, properties of isolated vertices, and characteristics of various connected components.

2.3 Hardness Results

In many cases, there is a correspondence between the complexity of a classical problem and the complexity of the related reachability reconfiguration problem: often both can be solved in polynomial time, or both are intractable (NP-complete and PSPACE-complete, respectively). Many of the PSPACE-hardness proofs follow the chain of reductions used for NP-hardness reductions, starting with the PSPACE-completeness of 3SAT reconfiguration [5].

There are, however, a few exceptions to this general pattern. Although the problem of 3-colouring is NP-complete, its reachability reconfiguration problem can be solved in polynomial time [13]. Conversely, the shortest path problem can be solved in polynomial time, but its reachability reconfiguration problem is PSPACE-complete [1]. Although exceptions to the general rules are known, currently there are no results explaining the reasons for such exceptions.

2.4 Algorithms

Greedy algorithms have been developed for matching, independent set, and minimum spanning tree, and dynamic programming algorithms for list colouring, shortest path, and k-colouring. In addition, the complexity of problems has been considered using the framework of parameterized complexity, where the goal is to find algorithms with running times that are polynomial in the size of the input but possibly exponential (or worse) in terms of one or more parameters of the problem. Various results consider as parameters the sizes of solutions, the lengths of reconfiguration sequences, and other attributes of inputs and problems [16].

One area with many open problems is that of finding shortest transformations between solutions. Among the few known results are algorithms for satisfiability [15] and independent set using token-sliding on caterpillars [19].

3 Presentation Highlights

In order to facilitate as much collaboration as possible, the workshop time was divided among technical talks, open problem presentations, and follow-up discussions. The first two types of activities are summarized here, with a selection of completed work and work in progress detailed in Section 4.

Both the high quality of the contributions and the engagement of the participants led to exciting new collaborations.

3.1 Technical Talks

3.1.1 Takehiro Ito (Tokohu University, Japan): Invitation to Combinatorial Reconfiguration

The workshop started with this introductory talk. It is possible to view the field of reconfiguration as middle ground between standard search problems, which ask for one solution to an instance of a problem, and enumeration problems, which ask for all solutions to an instance of a problem. It is important to note that since it is a decision problem, a reachability problem does not require the generation of an actual reconfiguration sequence, which may be super-polynomial in length.

The history of combinatorial reconfiguration was presented, showing both how the field was developed and in what new directions it might go. Most of the content can be found in Sections 1 and 2.

3.1.2 Nicolas Bousquet (Institut Polytechnique de Grenoble, France): Token Sliding on Chordal Graphs

(Joint work with Bonamy)

The reconfiguration of independent sets using token sliding was first considered by Hearn and Demaine [8], who showed that the reachability problem is PSPACE-complete on planar graphs. In later results, polynomial-time algorithms have been found for trees, cographs, claw-free graphs, and bipartite permutation graphs.

In new work, the authors show that both the reachability and connectivity problems can be solved in polynomial-time on interval graphs. The algorithm makes use of the geometric representation of an interval graph as an intersection graph of intervals on the line, which can be obtained in polynomial time. Using this representation, one can define the leftmost independent set. Although a naive approach may not yield a polynomial-time algorithm, dynamic programming can be employed to obtain algorithms for both problems.

In addition, the authors show that on *split graphs* (where the vertex set can be partitioned in two sets, one inducing a clique and the other inducing a set of isolated vertices), the connectivity problem is co-NP-hard and co-W[2]-hard. In particular, they use the fact that k-Dominating Set is NP-hard and W[2]-hard and then show that the result still holds when there is no *blocking set* of size at most $k+1$, where in a blocking set no vertex has a private neighbour. Using a similar construction, the same hardness results can be derived for the connectivity problem on bipartite graphs.

3.1.3 Akira Suzuki (Tohoku University, Japan): Reduction Tools on NCL

Nondeterministic Constraint Logic was first introduced by Hearn and Demaine [8] to show the hardness of sliding block puzzles. In a constraint graph, the directed edges have weights of 1 or 2 and are either legal or not. In the legal direction, each vertex is the head of at least two weight 2 edges. The NCL problem is to determine whether a target edge can be reversed by a sequence of *legal moves*, where in a legal move an edge is reversed such that the resulting direction is legal.

The problem NCL is difficult even if each vertex is either an AND vertex (the output edge is outgoing if both input edges are incoming) or an OR vertex (the output edge is outgoing if at least one input edge is incoming). Using this fact, PSPACE reductions are possible using NCL with just AND and OR gadgets. Demonstrations were given for sliding blocks and for edge colouring.

In addition, NCL can be generalized by introducing the notion of a neutral, undirected edge. This new version can accomplish all that can be accomplished using NCL, but often results in fewer gadgets. One of the new types of gadgets (in addition to AND and OR gadgets) is a *link gadget*.

In new work [18], it can be shown that reconfiguration of list edge-colouring and edge-colouring are PSPACE-complete using AND, OR, and link gadgets. This result completes the classification of the complexity of the reconfiguration of list edge-colouring by showing that the problem is in P for $k \leq 3$ and PSPACE-complete otherwise. For edge-colouring, the complexity is still open for $k = 4$, though this result proves PSPACE-completeness for $k \geq 5$.

The talk ended with some open questions about the complexity of more restricted constraint graphs.

3.1.4 Karen Seyffarth (University of Calgary, Canada): Reconfiguring Vertex Colourings of 2-Trees

(Joint work with Cavers)

Recent work was presented in the area of the connectivity problem for k-colouring for various classes of graphs. In 2008, Cereceda, van den Heuvel, and Johnson [3] proved that the reconfiguration graph for k-colouring is connected for all $k \geq col(H)+1$, where $col(H) = \max\{\delta(G) \mid G \subseteq H\}+1$ is the *colouring number* of H.

More recently, Choo and MacGillivray [4] proved a connection between colouring numbers and Hamiltonicity, namely, that $k_0(H) \leq col(H)+2$, for $k_0(H)$ the *Gray code number* of H, the least integer such that the reconfiguration graph for k-colouring has a Hamiltonian cycle for all $k \geq k_0(H)$. In the same paper, values of k_0 were shown for complete graphs, trees, and cycles. Later, Gray code numbers were found for complete bipartite graphs [2].

The presentation gave a proof of the Gray code numbers for 2-trees. Except for certain cases in which $k_0(H) = 5$, the Gray code number for 2-trees is 4.

3.1.5 Matthew Johnson (Durham University, UK): Kempe Equivalence in Regular Graphs

(Joint work with Bonamy, Bousquet, Feghali, and Paulusma)

For a k-colouring, an (a, b)-*component* is a maximal connected subgraph whose vertices are coloured a or b; such components are called *Kempe chains*. A *Kempe change* is the exchanging of the colours a and b of the vertices in an (a, b)-component, resulting in another proper colouring. A *Kempe class* consists of k-colourings that are *Kempe equivalent* (one can be obtained from another by a sequence of Kempe changes).

In 1981, Las Vergnas and Meyniel showed that the set of k-colourings of a d-degenerate graph form a Kempe class, $k > d$, where a graph is d-*degenerate* if every induced subgraph has a vertex of degree at most d. Mohar conjectured in 2007 that for $k \geq 3$, the k-colourings of a k-regular non-complete graph form a Kempe class, but in 2013 van den Heuvel found a counterexample for $k = 3$ [10].

Bonamy, Bousquet, Feghali, Johnson, and Paulusma have proved that for $k \geq 3$, if G is a connected k-regular graph that is neither complete nor the triangular prism, then the k-colourings of G form a Kempe class. In their proof, they make use of the clique cutset lemma of Las Vergnas and Meyniel, and consider separately k-regular graphs that are not 3-connected and k-regular graphs that are 3-connected. In the latter case, they make use of a matching lemma, and consider k-regular 3-connected graphs of diameter 2.

The talk ended with several open problems, including the question of whether the 5-colourings of a *toroidal triangular lattice* form a Kempe class (which would have an application in physics, proving the validity of the WSK algorithm for simulating the antiferromagnetic Potts model) and what can be determined about the number of Kempe changes needed to transform one colouring into another.

3.1.6 Jan van den Heuvel (London School of Economics and Political Science, UK): Token-Sliding Problems

(Joint work with Brightwell and Trakultraipruk)

The classic 15-puzzle can be interpreted as a problem of moving tokens on a graph, which can then be generalized to consider different graphs. A result by Wilson in 1974 shows that the reconfiguration graph for a sliding block puzzle is connected except if the puzzle graph is a cycle on $n \geq 4$ vertices, is bipartite and not a cycle, or the exceptional graph Θ_0 on 7 vertices.

To further generalize the problem, one can ask what would happen if there were more than a single uncovered vertex and/or if not all tokens were considered to be identical. The authors show that the reconfiguration graph is connected except if the puzzle graph is not connected, a path with at least two token labels, a cycle with at least two token labels, a cycle with two token labels, with at least two of the same label, a 2-connected bipartite graph with $n - 1$ distinct tokens, the graph Θ_0 with one of four possible labelings of tokens, or has connectivity 1 with at least two token labels and a separating path preventing tokens from moving between blocks.

A related problem is that of finding the shortest reconfiguration sequence, which Goldreich showed to be NP-complete for cases with $n - 1$ different tokens and van den Heuvel and Trakultriapruk (2014) showed to be in P when all tokens are the same, but NP-complete when at least one token is special and all others are identical. The proof is based on a result for robot motion by Papadimitriou, Raghavan, Sudan, and Tamaki (1994).

3.1.7 Moritz Mühlenthaler (Technische Universität Dortmund, Germany): Reconfiguration in Matroids Revisited

Reconfiguration properties of matroids have been studied, in different guises, at least since the 1970s. The talk touched very briefly on several interesting results from the past 40 years in this area and put them in the current unified setting of reconfiguration problems [11, 10]. Furthermore, some new aspects of reconfiguration in matroids were investigated, namely the complexity of reconfiguring common independent sets of two or more partition matroids. In particular, it was shown that reachability of two common independent sets of two partition matroids can be decided in polynomial time, while for three or more matroids, the task becomes PSPACE-complete. Since many combinatorial optimization problems can be phrased in terms of finding maximum common independent sets of partition matroids, there are many interesting applications of these results.

3.1.8 Ruth Haas (University of Hawaii, USA): Reconfiguration of Dominating Sets

(Joint work with Seyffarth)

A subset S of the vertices of a graph is a *dominating set* if and only if every other vertex is adjacent to a vertex of S. The *domination number*, $\gamma(G)$, is the minimum cardinality of a dominating set of G, and the *upper domination number*, $\Gamma(G)$, is the maximum cardinality of a minimal dominating set of G. Various models of domination reconfiguration have been studied, e.g. by Subramaniam, Sridharan, and Fricke and by Hedetniemi, Hedetniemi, and Hutson; in addition, the γ-graph uses only γ sets and token jumping.

The goal is to find $d_0(G)$, the smallest value of k such that the reconfiguration graph is connected for all $k \geq d_0(G)$. The authors showed in 2014 that $d_o(G) = \Gamma(G) + 1$ for bipartite graphs and chordal graphs [6], with further results obtained by Suzuki, Mouawad, and Nishimura. Alikhani, Fatehi, and Klavzar considered which graphs might be reconfiguration graphs.

In the talk, the following new results were presented:

- All independent dominating sets are in the same connected component of the reconfiguration graph for $k = \Gamma(G) + 1$, where an *independent dominating set* is a maximal independent set.
- If G is both perfect and irredundant perfect, then $d_0(G) = \Gamma(G)+1$, where a set is *irredundant* if every vertex in the set has a private neighbour, and precise definitions of the terms relate sizes of independent sets and clique cover numbers.
- For certain classes of well-covered graphs, $d_0(G) = \Gamma(G) + 1$, where a graph is *well-covered* if every maximal independent set has the same cardinality.

3.2 Open Problem Presentations

3.2.1 Henning Fernau (Universität Trier, Germany)

Various extensions and generalizations of reconfiguration were proposed, as detailed below.

Building on ideas on generalizing the notion of adjacency, Fernau proposed approaching problems by determining what definition of adjacency would result in a connected reconfiguration graph. It was noted that for many problems there is a trivial solution, such as in the case of vertex cover, when all intermediate solutions are vertex covers of any size (a vertex cover V_1 can be reconfigured to a vertex cover V_2 by adding all vertices in V_2 and then deleting all vertices in V_1).

Another proposal was to consider the relationship between reconfiguration and *reoptimization*, where the goal is to determine a solution for an instance to a problem given a "similar" instance and its solution. Combining and generalizing the concepts could lead to the notion of a sequence of steps that changes both the instance and the solution.

It was further observed that although many reconfiguration problems turn out to be computationally difficult, it may be beneficial to focus on results in areas where hardness is a positive attribute. One example could be taken from computational social choice, considering how easy it is for an election to be rigged.

Other possible topics include the reconfiguration of problems in computational geometry as well as that of string editing.

3.2.2 Tatsuhiko Hatanaka (Tokohu University, Japan): Optimizing a Colouring via a Reconfiguration Sequence

In this presentation, a new variant of reconfiguration was introduced. In the *optimization variant*, the goal is to find the optimal solution over all solutions reachable from a given solution. In particular, for the problem of colouring reconfiguration, an optimal solution is one that minimizes the number of colours used.

Preliminary results include a proof that the problem is NP-hard when the number of colours is at least five, and a polynomial-time algorithm for numbers of colours no greater than three.

3.2.3 Jan van den Heuvel (London School of Economics and Political Science, UK)

The focus of this presentation was the k-colouring reconfiguration problem, considering both the reachability problem and the connectivity problem. The reachability problem can be solved in polynomial time when the number of colours, k, is at most three, and is PSPACE-complete otherwise (as a consequence of [8]).

For the connectivity problem, the answer is always no when $k = 2$ and is co-NP-complete when $k = 3$. The complexity remains open for $k \geq 4$. There followed a discussion of the co-NP-completeness result, and in particular the *folding* operation, where a vertex has two nonadjacent neighbours that can be folded together. It can be shown that a bipartite instance G is a no-instance of the connectivity problem for $k = 3$ if and only if G can be folded to a cycle on six vertices. One might wish to try to generalize this to $k = 4$ by showing that G can be folded to a cube; however, this is not the case. Further work is required on this problem, perhaps using another technique, and also perhaps by considering techniques for nonbipartite graphs.

3.2.4 Nicolas Bousquet (Institut Polytechnique de Grenoble, France): Graph Recolouring on Sparse Classes of Graphs

Based on a conjecture by Cereceda in 2007, one possible approach to characterizing the reconfiguration graph for k-colouring is by considering the degeneracy of the graph and using that to determine the diameter of the reconfiguration graph. In particular, the conjecture states that the reconfiguration graph is of quadratic diameter when the number of colours is two greater than the degeneracy of the input. The conjecture has been shown to be true for k-regular graphs, trees, chordal graphs, bounded treewidth graphs, and distance-hereditary graphs.

So far, all the results have used nothing more than a tree decomposition of the input graph, raising the question as to whether other techniques might be applied.

Another question concerns the relationship between mixing time and the diameter of the reconfiguration graph, and in particular a characterization of classes of graphs for which the diameter of the reconfiguration graph is linear. Results are known for trees and k-degenerate graphs.

3.2.5 Kunihiro Wasa (National Institute of Informatics, Japan)

This talk focused on the reconfiguration of optimal amidakujis. An *amidakuji*, also known as a *ladder lottery*, is a way of generating a permutation by creating a series of vertical and horizontal bars. The number of vertical bars is the number of items to be permuted. The placement of horizontal bars joining adjacent vertical bars dictates the permutation, where from the top of each vertical bar, a path is traced following all horizontal bars encountered. An amidakuji is *optimal* if it uses the minimum number of bars to achieve the particular permutation.

The talk presented an algorithm that can be used to determine the shortest reconfiguration sequence between optimal amidakujis, and presented open problems on various variants.

3.2.6 Ryuhei Uehara (Japan Advanced Institute of Science and Technology (JAIST), Japan)

Starting with a general overview of the complexity of various games and puzzles, it was observed that most one-player games are NP-complete whereas most two-player games are PSPACE-complete or EXP-complete [7]. As an exception to this general rule, many sliding block puzzles, in which it is possible to return to the same state, are PSPACE-complete.

Uehara is the director of the JAIST Gallery, which houses NOB's Puzzle Collection, a collection of about 10,000 puzzles from around the world. Although he did not bring the entire collection to share with us, he did share his latest purchase, the Qubigon, which is a generalization of the well-studied slide-block 15-puzzle. The observation that the puzzle makes use of 18 of the 20 possible locations at once led to questions about the relationship between the number of tokens and the number of locations more generally. (This was later addressed in a presentation by van den Heuvel.)

An additional open problem that was discussed related to token sliding for independent sets.

4 Scientific Progress Made

4.1 Matching Shortest Transformation

Many techniques have been developed in the last few years in order to design polynomial time algorithms to determine the existence of a transformation between two configurations (reachability). On the other hand, very few of them have been proposed to compute shortest transformations between two configurations. One of the main reasons is that the shortest transformation problem seems to be much harder than the reachability problem even if only little evidence has been found so far in that direction (i.e. we need a very simple reconfiguration problem in order to obtain a polynomial-time algorithm for the shortest transformation problem). One of our motivations was to find problems on which the shortest transformation problem is hard while the reachability problem is very simple. Another motivation consisted in developing new tools to prove that the shortest transformation problem is hard.

During the workshop, Nicolas Bousquet, Tatsuhiko Hatanaka, Takehiro Ito and Moritz Muehlenthaler found preliminary results on the hardness of shortest transformation for the matching problem on graphs of maximum degree 4. In particular it implies that the shortest transformation is hard for the reconfiguration of independent sets on the token jumping and token sliding variants on line graphs. It answers a question of Ryuhei Uehara on the complexity of the TJ problem on claw-free graphs raised during the workshop. The proof technique might be generalized in order to show that there is no constant factor approximation algorithm for the Matching Shortest Transformation problem.

4.2 Equitable Colouring Reconfiguration

A *colouring* of a graph is an assignment of colours to its vertices so that no two adjacent vertices have the same colour, whereas an *equitable colouring* of a graph is a colouring where the difference between the number of vertices coloured by any two colours is at most one. Equitable colourings have many real-world applications in scheduling, load-balancing, and timetabling, and has been the subject of much research interest. The question of whether one colouring can be transformed to another colouring by changing the colour of only one vertex at each step has also been studied in the field of reconfiguration problems and has been proved to be PSPACE-complete.

Tatsuhiko Hatanaka, Haruka Mizuta, and Krishna Vaidyanathan considered the reconfiguration of equitable colourings, showing that the problem is PSPACE-complete, even for planar graphs using only four colours.

4.3 String Editing Reconfiguration

Discussions among Henning Fernau, Ruth Haas, Matthew Johnson, Naomi Nishimura, and Karen Seyffarth focused on the issues surrounding solution spaces in which each solution consists of a sequence (in this case, a sequence of edit operations). Preliminary observations included the fact that a reconfiguration sequence is itself a solution in the form of a sequence, allowing for the discussion of the metaproblem of reconfiguration of reconfiguration.

Additional observations found connections to work on permutations, including Kunihiro Wasa's open problem presentation on amidakujis.

4.4 Homomorphism Reconfiguration

Vijay Subramanya and Ben Moore considered the homomorphism reconfiguration problem for graphs on at most four vertices. They showed that H-reconfiguration is in P when H is C_4 or the diamond graph.

Furthermore, they conjectured that if H_1- and H_2-reconfiguration are in P, then H-reconfiguration is also in P, where H is obtained by joining H_1 and H_2 at a vertex. They proved the conjecture when H_1 and H_2 are both equal to C_3. This result extends to the case where H is a series of C_3's.

4.5 NCL and Matroids

Two open problems presented by Akira Suzuki about restricted variants of nondeterministic constraint logic can be rephrased in terms of connectivity of common independent sets of two partition matroids. Since NCL is a tool which has been developed for proving hardness of puzzles, this is quite an unexpected connection.

5 Outcome of the Meeting

The objectives of the workshop were all met, as elaborated below; further outcomes are listed in Section 5.5.

5.1 Providing an Opportunity for Joint Discussion by Researchers in Reconfiguration from All over the World

The workshop was very successful in bringing together researchers from different countries and different research groups. Unfortunately, funding and visa issues resulted in last-minute cancellations by researchers from India and Lebanon.

In the interests of maintaining our world-wide inclusiveness, in our business meeting we decided to try to alternate the continents on which future workshops will be held, with the 2019 workshop to be held in France.

5.2 Identifying Future Research Directions

The many open problem sessions and follow-up discussions generated many future research directions. As detailed in Section 4, progress has already been made in several of these directions.

5.3 Deepening the Area by Establishing a Set of Common Methods and Algorithmic Techniques

At the business meeting we discussed possible approaches to collecting and displaying known results. The prototype of a website was shown and discussed.

5.4 Broadening the Area by Making Connections to Related Areas and Problems

Several of the presentations introduced the audience to new applications of reconfiguration, such as the application of colouring to the Potts model (Section 3.1.5). In other talks, unexpected connections were discovered, such as between ladder lotteries, string editing, and permutations (Sections 3.2.5 and 4.4) and between NCL and matroids (Section 4.5).

5.5 Additional Outcomes

In addition to the planned outcomes, to further the research community we are planning to hold a minisymposium on reconfiguration at the 6^{th} biennial Canadian Discrete and Algorithmic Mathematics Conference in 2017, and to have a special issue on reconfiguration in the journal *Algorithms*.

References

[1] P. Bonsma, The complexity of rerouting shortest paths, In *Proceedings of the* 37^{th} *International Symposium on Mathematical Foundations of Computer Science*, 2012, 222–233.

[2] M. Celaya, K. Choo, G. MacGillivray, and K. Seyffarth, Reconfiguring k-colourings of complete bipartite graphs, *Kyungpook Math. J.* **56** (2016), 647–655.

[3] L. Cereceda, J. van den Heuvel, and M. Johnson, Connectedness of the graph of vertex colourings, *Discrete Math.* **308** (2008), 913–919.

[4] K. Choo and G. MacGillivary, Gray code numbers for graphs, *Ars Math. Contemp.* **4** (2011), 125–139.

[5] P. Gopalan, P. G. Kolaitis, E. N. Maneva, and C. H. Papadimitriou, The connectivity of Boolean satisfiability: computational and structural dichotomies, *SIAM Journal on Computing* **38**(6) (2009), 2330–2355.

[6] R. Haas and K. Seyffarth, The k-dominating graph, *Graphs and Combinatorics*, **30**(3) (2014), 609–617.

[7] R. A. Hearn and E. D. Demaine, *Games, Puzzles, and Computation*, A K Peters, Wellesley Massachusetts, 2009.

[8] R. A. Hearn and E. D. Demaine, PSPACE-completeness of sliding-block puzzles and other problems through the nondeterministic constraint logic model of computation, *Theoretical Computer Science*, **343**(1–2) (2005), 72–96

[9] R. Haas and K. Seyffarth, The k-dominating graph, *Graphs and Combinatorics*, **30**(3) (2014), 609–617.

[10] J. van den Heuvel, The complexity of change, In *Surveys in Combinatorics*, Cambridge University Press (2013), 127–160.

[11] T. Ito, E. D. Demaine, N. J. A. Harvey, C. H. Papadimitriou, M. Sideri, R. Uehara, and Y. Uno On the complexity of reconfiguration problems, *Theoretical Computer Science*, **412**(12-14) (2011), 1054–1065.

[12] T. Ito, H. Ono, and Y. Otachi, Reconfigurations of cliques in a graph, In *Proceedings of the* 12^{th} *Annual Conference on Theory and Applications of Models of Computation*, 2015, 212–223.

[13] M. Johnson, D. Kratsch, S. Kratsch, V. Patel, and D. Paulusma, Finding shortest paths between graph colourings, In *Proceedings of the* 9^{th} *International Symposium on Parameterized and Exact Computation*, 2014, 221-233.

[14] M. Kamiński, P. Medvedev, and M. Milanič, Complexity of independent set reconfigurability problems, *Theoretical Computer Science*, **439** (2012), 9–15.

[15] A. E. Mouawad, N. Nishimura, V. Pathak, and V. Raman, Shortest reconfiguration paths in the solution space of Boolean formulas, In *Proceedings of* 42^{nd} *International Colloquium on Automata, Languages, and Programming*, 2015, 985–996.

[16] A. E. Mouawad, N. Nishimura, V. Raman, N. Simjour, and A. Suzuki, On the parameterized complexity of reconfiguration problems, In *Proceedings of the* 8^{th} *International Symposium on Parameterized and Exact Computation*, 2013, 281–294.

[17] A. E. Mouawad, N. Nishimura, V. Raman, and M. Wrochna, Reconfiguration over tree decompositions, In *Proceedings of the* 9^{th} *International Symposium on Parameterized and Exact Computation*, 2014, 246–257.

[18] H. Osawa, A. Suzuki, T. Ito, and X. Zhou, The complexity of (list) edge-coloring reconfiguration problem, In *Proceedings of the* 11^{th} *International Conference and Workshops on Algorithms and Computation*, to appear, 2017.

[19] T. Yamada and R. Uehara, Shortest reconfiguration of sliding tokens on a caterpillar, In *Proceedings of the* 10^{th} *International Workshop on Algorithms and Computation*, 2016, 236–248.