A Brief Introduction to Reconfiguration of Independent Sets and Related Problems

Hoang Anh Duc

April 16, 2019

hoanganhduc@ces.kyutech.ac.jp

Research Seminar at Kyutech Algorithms Group

About Me

Kyutech
Kyushu Institute of Technology

- **Basic Info:**
  - Nationality: Vietnamese
  - Full Name: Hoàng Anh Đức (in publications: Duc A. Hoang)
  - Email: `hoanganhduc@ces.kyutech.ac.jp`
  - Personal Webpage: `https://hoanganhduc.github.io/`
- **Education:**
  - **Bachelor in Mathematics**, K53 Advanced Mathematics, VNU University of Science, Vietnam (2008–2013)
  - **Master in Information Science**, Uehara Lab, School of Information Science, JAIST, Japan (2013–2015)
  - **PhD in Information Science**, Uehara Lab, School of Information Science, JAIST, Japan (2015–2018)
- **Research Interests:**
  - Graph Algorithms
  - Combinatorial Reconfiguration

Outline

Moving Tokens on Graphs

Reconfiguration of Independent Sets

My Work at JAIST: The Sliding Token problem

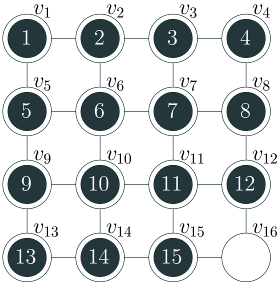
Some Open Problems

Moving Tokens on Graphs

- A token (coin) is placed at each vertex of a vertex-subset X of a graph. A rule R of moving tokens is given.
  - Checking if a token-set X is obtained from another token-set Y by applying R exactly once can be done in polynomial time.
- Each set of tokens X satisfies some property P
  - Checking if X satisfies P can be done in polynomial time.

Example: 15-PUZZLE

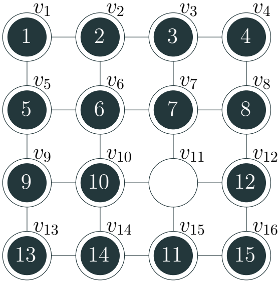
- X: fifteen labeled tokens
- R: one can slide a token to its empty adjacent neighbor (if exists)
- P: each member of X is placed at a vertex of a 4×4 grid

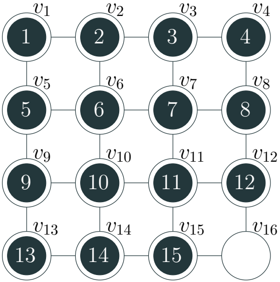


Given: two sets of tokens X, Y (both satisfy P)

Question: decide if there exists a sequence of token-sets $(X_1, X_2, \ldots, X_\ell)$, $X_1 = X$, $X_\ell = Y$ (all X_i satisfy P for $i \in \{1, 2, \ldots, \ell\}$) between X and Y such that X_i is obtained from X_{i-1} by applying R exactly once to the tokens in X_{i-1} $(i \in \{2, 3, \ldots, \ell\})$

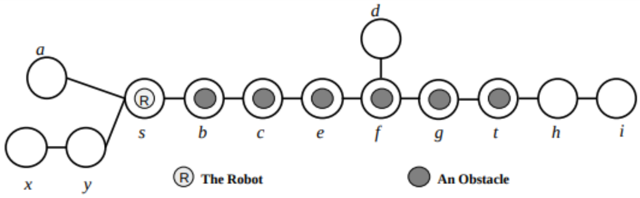
Example: 15-PUZZLE





Token Reconfiguration in planning robot motion.

- Graph Motion Planning with One Robot (GMP1R) [Papadimitriou et al. 1994]
  - Goal: move robot from s to t using a smallest number of steps possible.
  - It is NP-complete to decide if a solution of length k exists in a general graph.
- Multi-Robot Path Planning [Ryan 2007]
  - Robots may need to "detour away" from their shortest paths to let other robots pass.



Reconfiguration Problems

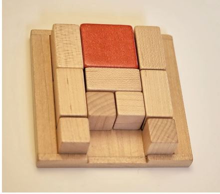
- **Given:**
  - a description of what a configuration is
  - a reconfiguration rule that describes how to modify a configuration
- **Question:** Whether there is a sequence of configurations that transforms one given configuration into another, where each member of the sequence is obtained from the previous one by applying the reconfiguration rule exactly once.

Typical Assumptions

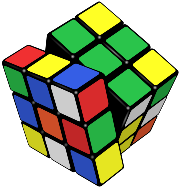
- Checking whether a given structure is a configuration can be done in polynomial time.
- Checking if one configuration is obtained from another configuration by applying the reconfiguration rule exactly once can be done in polynomial time.

Some Other Reconfiguration Problems

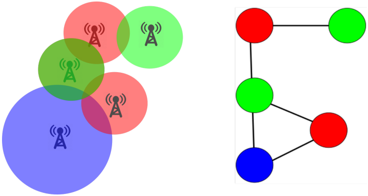
TOKEN RECONFIGURATION is a reconfiguration problem.



(a) SLIDING-BLOCK PUZZLE



(b) RUBIK'S CUBE



(c) FREQUENCY RE-ASSIGNMENT

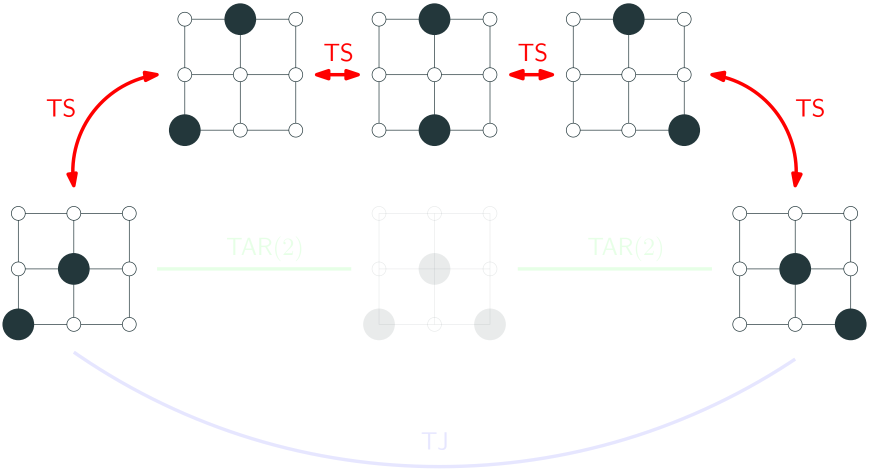
(d) RUSH HOUR

Reconfiguration of Independent Sets

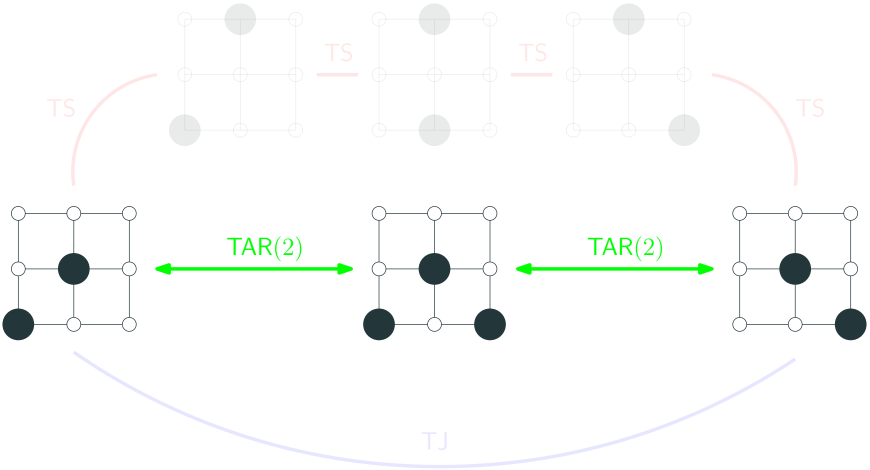
... is Token Reconfiguration where

- Each token-set X forms an independent set, i.e., no two tokens in X are connected by an edge.
- The rule R can be:
  - **Token Sliding (TS) [Hearn and Demaine 2005]:** A token can only be moved to one of its (unoccupied) neighbors.
  - **Token Addition and Removal (TAR(k)) [Ito et al. 2011]:** One can either add or remove a token such that the number of remaining tokens is at least k.
  - **Token Jumping (TJ) [Kamiński et al. 2012]:** A token can be moved to any unoccupied vertex.

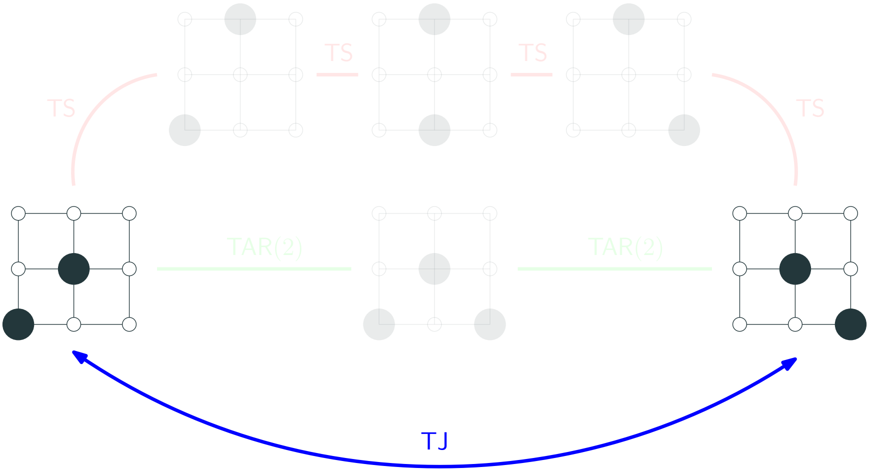
Independent Set Reconfiguration in a Graph



Independent Set Reconfiguration in a Graph

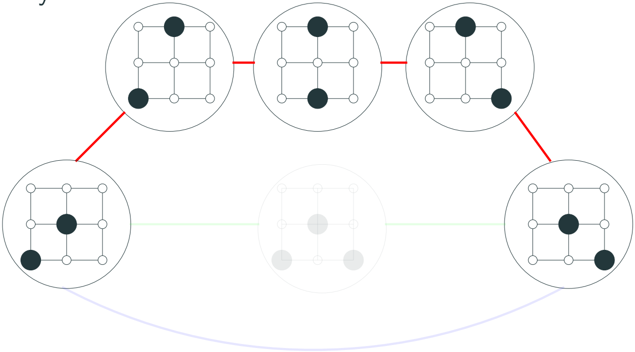


Independent Set Reconfiguration in a Graph



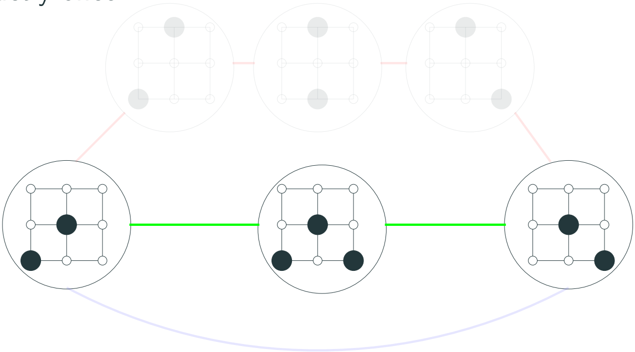
One can also form the corresponding reconfiguration graph.

- Each token-set is a node.
- Two nodes (token-sets) X, Y are adjacent if one can be obtained from the other by applying R (TS/TAR(k)/TJ) exactly once.



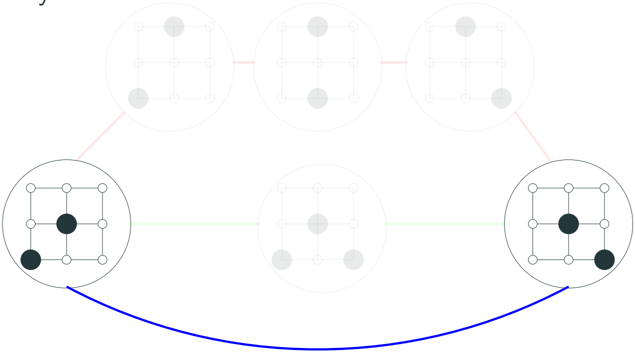
One can also form the corresponding reconfiguration graph.

- Each token-set is a node.
- Two nodes (token-sets) X, Y are adjacent if one can be obtained from the other by applying R (TS/TAR(k)/TJ) exactly once.



One can also form the corresponding reconfiguration graph.

- Each token-set is a node.
- Two nodes (token-sets) X, Y are adjacent if one can be obtained from the other by applying R (TS/TAR(k)/TJ) exactly once.



Independent Set Reconfiguration **in a Graph**

One can also form the corresponding reconfiguration graph.

- Each token-set is a node.
- Two nodes (token-sets) X, Y are adjacent if one can be obtained from the other by applying R (TS/TAR(k)/TJ) exactly once.

One may ask

- Reachability: a path between two nodes of a reconfiguration graph?
- Shortest Reconfiguration: find a shortest path (if exists) between two nodes of a reconfiguration graph?
- Connectivity: a reconfiguration graph is connected?
- Diameter: the diameter of a reconfiguration graph is bounded?

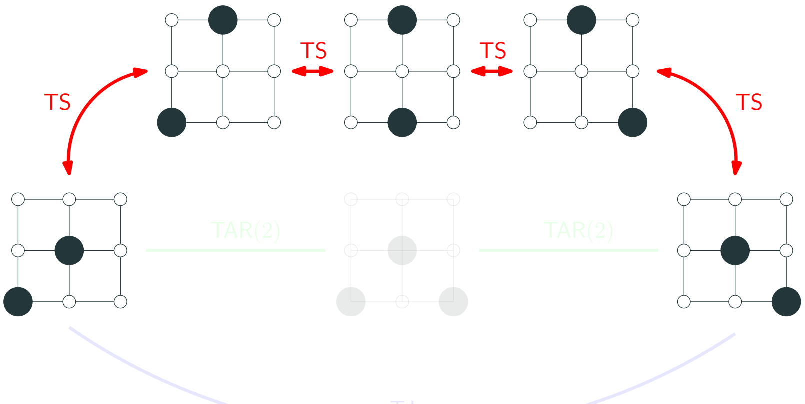
My Work at JAIST: The Sliding Token problem

The SLIDING TOKEN problem

SLIDING TOKEN [Hearn and Demaine 2005]

Given: two independent sets (token sets) I, J of a graph G, and the Token Sliding (TS) rule

Question: whether there is a TS-sequence that transforms I into J (and vice versa)



My co-author(s) and I want to know whether SLIDING TOKEN can be solved efficiently for some restricted graphs.

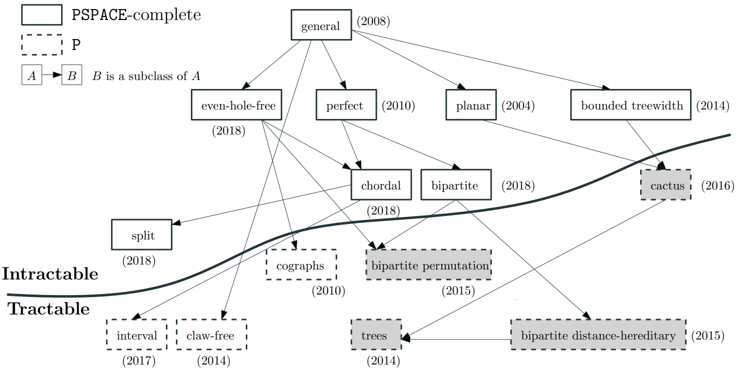
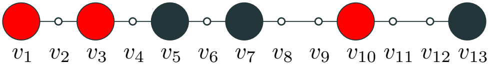


Figure 2: The complexity of SLIDING TOKEN for some well-known graph classes. Our contribution is marked with dashed gray boxes.

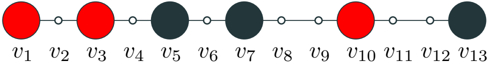
The SLIDING TOKEN problem on trees

1. What happen if the input graph is only a path?



The SLIDING TOKEN problem on trees

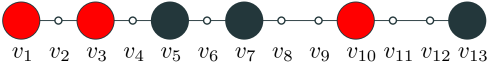
1. What happen if the input graph is only a path?



Easy. Only one way to move tokens.

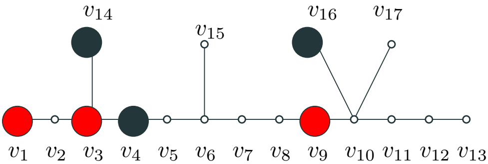
The SLIDING TOKEN problem on trees

1. What happen if the input graph is only a path?



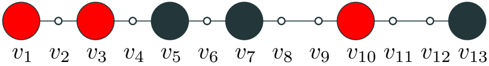
Easy. Only one way to move tokens.

2. Now, what if the input graph is a caterpillar?



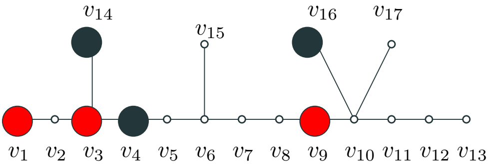
The SLIDING TOKEN problem on trees

1. What happen if the input graph is only a path?



   Easy. Only one way to move tokens.

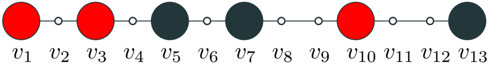
2. Now, what if the input graph is a caterpillar?



   Trouble. v_3 moves before v_1, and it moves to v_4. So how to move v_1? $\Rightarrow$ DETOUR

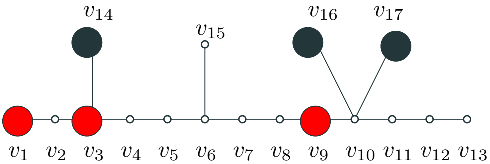
The SLIDING TOKEN problem on trees

1. What happen if the input graph is only a path?



Easy. Only one way to move tokens.

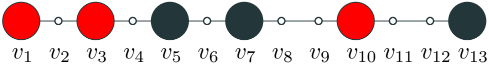
2. Now, what if the input graph is a caterpillar?



Trouble. Can we move red tokens to black ones?

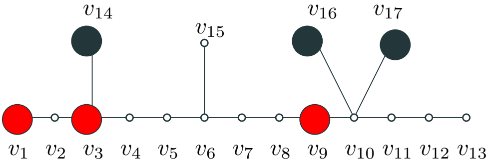
The SLIDING TOKEN problem on trees

1. What happen if the input graph is only a path?



   Easy. Only one way to move tokens.

2. Now, what if the input graph is a caterpillar?



   Trouble. Can we move red tokens to black ones?

3. Not trivial even for simple graphs like trees
   3.1 Whether there is any structure that forbids the transformation of one independent set into another?
   3.2 Whether we can handle "detour"?

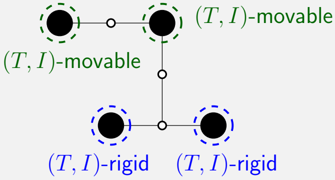
The Sliding Token problem on trees

For an instance (T, I, J) of Sliding Token, where T is a tree and I, J are independent sets of T.

Forbidden Structure: Rigid Tokens

Intuitively, a token t placed on vertex $u \in I$ is (T, I)-rigid if it cannot be moved at all. If t is not (T, I)-rigid, we say that it is (T, I)-movable.

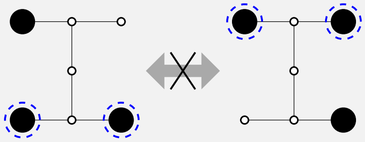
One can find the set $\mathsf{R}(T, I)$ of all vertices where (T, I)-rigid tokens are placed in $O(n^2)$ time, where n is the number of vertices of T.



The Sliding Token problem on trees

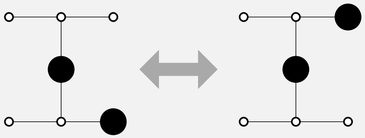
Observation 1

If $\mathsf{R}(T, I) \neq \mathsf{R}(T, J)$ then there is no reconfiguration sequence of token-slides between I and J.

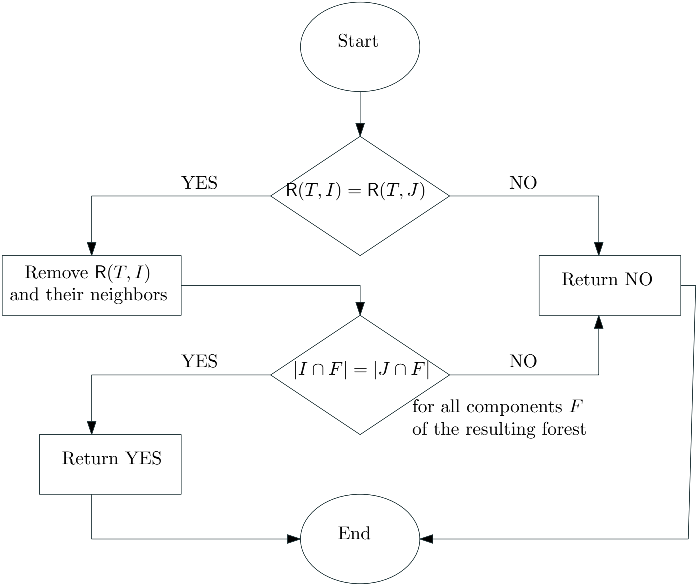


Observation 2

If $\mathsf{R}(T, I) = \mathsf{R}(T, J) = \emptyset$ then there is a reconfiguration sequence of token-slides between I and J if and only if $|I| = |J|$.



The Sliding Token problem on trees



Some Open Problems

Independent Set and its reconfiguration variants

- The Independent Set problem asks if there exists an independent set of size at least k in a given graph.

Graph	Independent Set	Independent Set Reconf.[1] (TS, TJ, TAR)
general	NP-complete [Garey and Johnson 1979]	PSPACE-complete [Ito et al. 2011]
perfect	P [Grötschel et al. 1981]	PSPACE-complete [Kamiński et al. 2012]
interval	P [Frank 1975]	P [Kamiński et al. 2012; Bonamy and Bousquet 2017]
Unknown[2]	NP-hard	P

[1] In all problems, the Reachability question is considered.
[2] This open question was first proposed in [Kamiński et al. 2012]

Hardness with small graph parameters

Theorem (Wrochna 2014)

Independent Set Reconfiguration *remains* PSPACE-*complete even for graphs of bandwidth at most* c*, for some constant* c.

- The bandwidth $\mathrm{bw}(G)$ of a graph G is defined as follows

$$\mathrm{bw}(G) = \min_{f} \max_{uv \in E(G)} |f(u) - f(v)|,$$

where $f : V(G) \to \{1, 2, \ldots, |V(G)|\}$ represents a way of labeling vertices of G with integers from 1 to $|V(G)|$.

- It is well-known that c is very large, but to the best of our knowledge, it is unknown how large c is.
- To the best of our knowledge, it is unknown if Sliding Token can be solved in polynomial time even for graphs of bandwidth 2.

Learn More About Reconfiguration

Surveys on Reconfiguration

Jan van den Heuvel (2013). “The Complexity of Change.” In: *Surveys in Combinatorics*. Vol. 409. London Mathematical Society Lecture Note Series. Cambridge University Press, pp. 127–160. DOI: 10.1017/CBO9781139506748.005

Naomi Nishimura (2018). “Introduction to Reconfiguration.” In: *Algorithms* 11.4. (article 52). DOI: 10.3390/a11040052

Online Web Portal (maintained by Takehiro Ito)

http://www.ecei.tohoku.ac.jp/alg/core/

Thank you very much for your attention!

Kyutech
Kyushu Institute of Technology

Bonamy, Marthe and Nicolas Bousquet (2017). "Token Sliding on Chordal Graphs." In: *Proceedings of the 43rd International Workshop on Graph-Theoretic Concepts in Computer Science, WG 2017*. Ed. by H. Bodlaender and G. Woeginger. Vol. 10520. Lecture Notes in Computer Science. Springer, pp. 136–149. DOI: 10.1007/978-3-319-68705-6_10.

Frank, András (1975). "Some polynomial algorithms for certain graphs and hypergraphs." In: *Proceedings of the 5th British Combinatorial Conference, 1975*. Utilitas Mathematica.

Garey, Michael R. and David S. Johnson (1979). *Computers and Intractability: A Guide to the Theory of NP-completeness*. W.H. Freeman & Company. ISBN: 978-0-716-71044-8.

Grötschel, M., L. Lovász, and A. Schrijver (1981). "The ellipsoid method and its consequences in combinatorial optimization." In: *Combinatorica* 1.2, pp. 169–197. DOI: 10.1007/BF02579273.

Hearn, Robert A. and Erik D. Demaine (2005). "PSPACE-Completeness of Sliding-Block Puzzles and Other Problems through the Nondeterministic Constraint Logic Model of Computation." In: *Theoretical Computer Science* 343.1-2, pp. 72–96. DOI: 10.1016/j.tcs.2005.05.008.

Heuvel, Jan van den (2013). "The Complexity of Change." In: *Surveys in Combinatorics*. Vol. 409. London Mathematical Society Lecture Note Series. Cambridge University Press, pp. 127–160. DOI: 10.1017/CBO9781139506748.005.

Ito, Takehiro, Erik D. Demaine, Nicholas J. A. Harvey, Christos H. Papadimitriou, Martha Sideri, Ryuhei Uehara, and Yushi Uno (2011). "On the Complexity of Reconfiguration Problems." In: *Theoretical Computer Science* 412.12-14, pp. 1054–1065. DOI: 10.1016/j.tcs.2010.12.005.

Kamiński, Marcin, Paul Medvedev, and Martin Milanič (2012). “Complexity of independent set reconfigurability problems.” In: *Theoretical Computer Science* 439, pp. 9–15. DOI: 10.1016/j.tcs.2012.03.004.

Nishimura, Naomi (2018). “Introduction to Reconfiguration.” In: *Algorithms* 11.4. (article 52). DOI: 10.3390/a11040052.

Papadimitriou, C. H., P. Raghavan, M. Sudan, and H. Tamaki (1994). “Motion planning on a graph.” In: *Proceedings of SFCS 1994*, pp. 511–520. DOI: 10.1109/SFCS.1994.365740.

Ryan, Malcolm R.K. (2007). “Graph Decomposition for Efficient Multi-Robot Path Planning..” In: *Proceedings of IJCAI 2007*, pp. 2003–2008. URL: http://www.aaai.org/Papers/IJCAI/2007/IJCAI07-323.pdf.

Wrochna, Marcin (2014). “Reconfiguration in bounded bandwidth and treedepth.” In: *arXiv preprint*. arXiv: 1405.0847.