A Brief Introduction to Independent Set Reconfiguration and Related Problems

Duc A. Hoang

October 12, 2018

anhduc.hoang1990@gmail.com

Biannual Conference of the Faculty of Mathematics, Mechanics, and Informatics, VNU University of Science

Outline

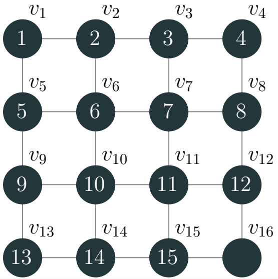
Moving Tokens on Graphs

Token Reconfiguration in a Graph

- 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, and one unlabeled token.
- R: Swap the unlabeled token with an adjacent labeled one.

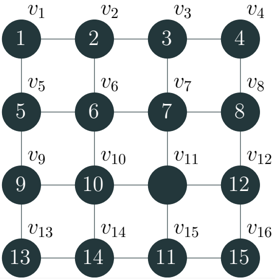


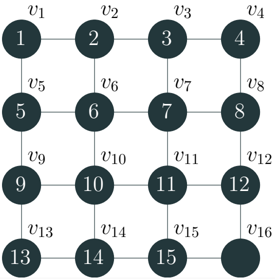
Token Reconfiguration in a Graph

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

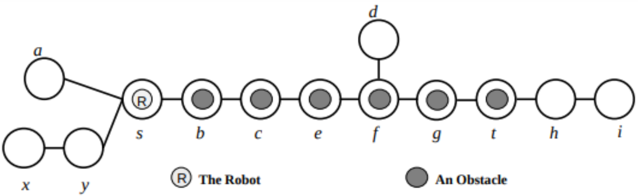




TOKEN RECONFIGURATION in a Graph

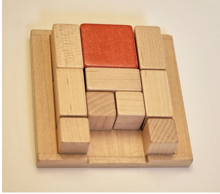
TOKEN RECONFIGURATION can be used in planning robot motion.

- GRAPH MOTION PLANNING WITH ONE ROBOT (GMP1R) [Papadimitriou et al. 1994]
    - It is NP-complete to decide if a solution of length k exists in a general graph.
- MULTI-ROBOT PATH PLANNING (for example, see [Ryan 2007])
    - The path length should be minimized.
    - Robots may need to "detour away" from their shortest paths to let other robots pass.

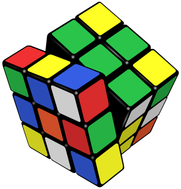


Some Other Reconfiguration Problems

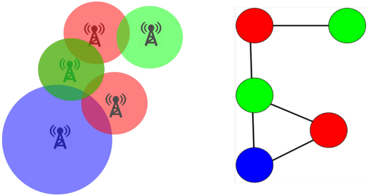
TOKEN RECONFIGURATION is a reconfiguration problem.



(a) SLIDING-BLOCK PUZZLE



(b) RUBIK'S CUBE



(c) FREQUENCY RE-ASSIGNMENT

(d) RUSH HOUR

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/

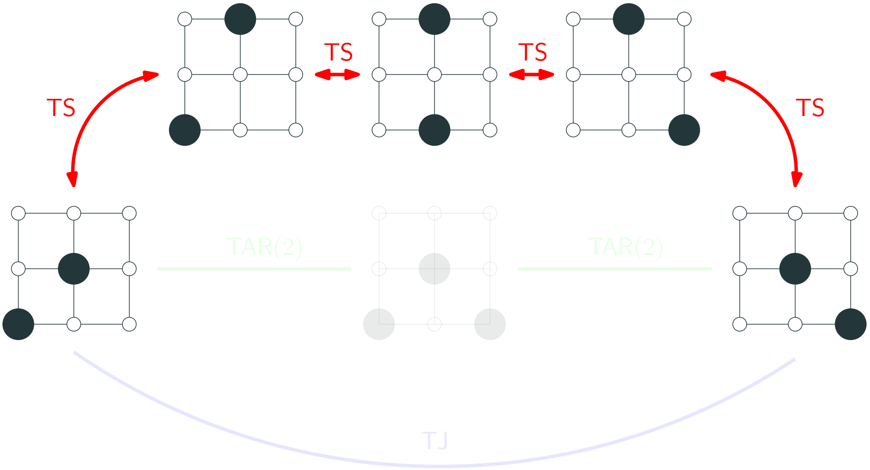
Reconfiguration of Independent Sets

Independent Set Reconfiguration in a Graph

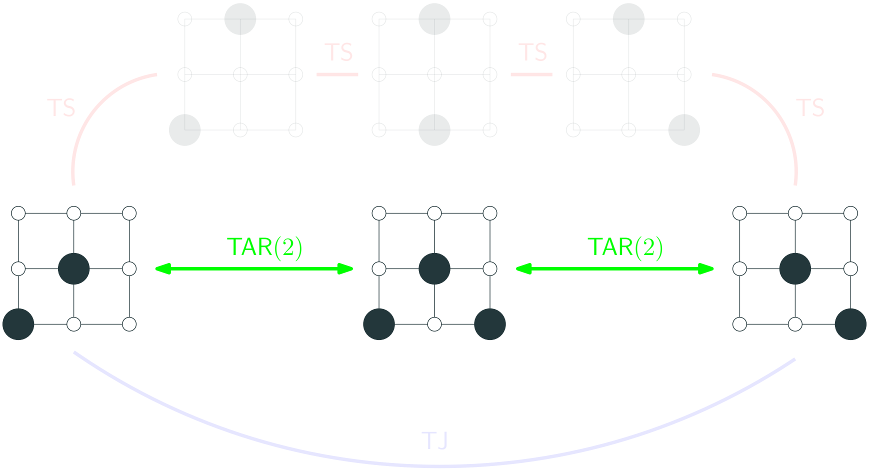
... is Token Reconfiguration where

- 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.

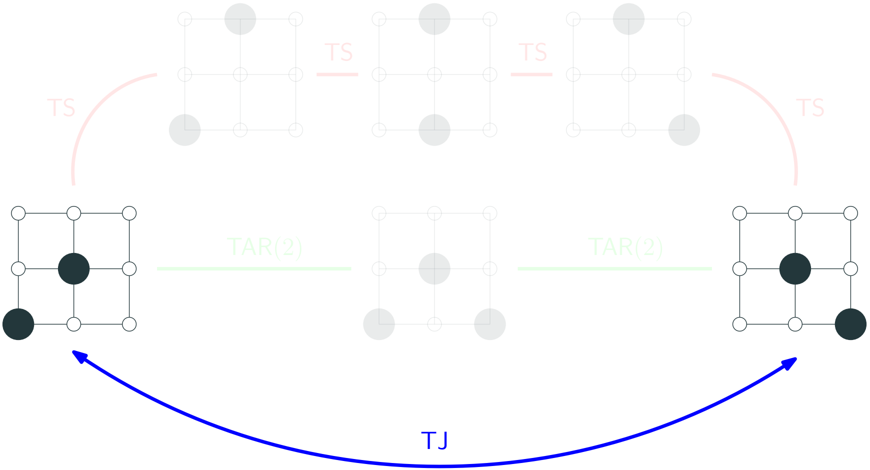
Independent Set Reconfiguration in a Graph



Independent Set Reconfiguration in a Graph



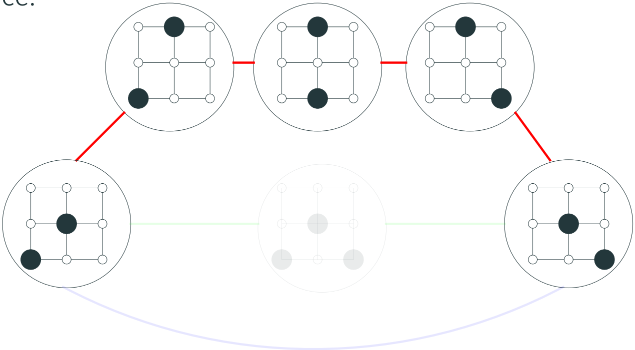
Independent Set Reconfiguration in a Graph



Independent Set Reconfiguration in a Graph

One can also form the corresponding reconfiguration graph.

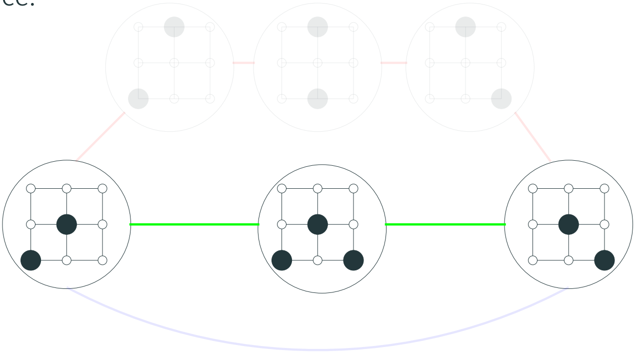
- Each token-set is a vertex.
- Two token-sets X, Y are adjacent if one can be obtained from the other by applying R (TS/TAR(k)/TJ) exactly once.



Independent Set Reconfiguration in a Graph

One can also form the corresponding reconfiguration graph.

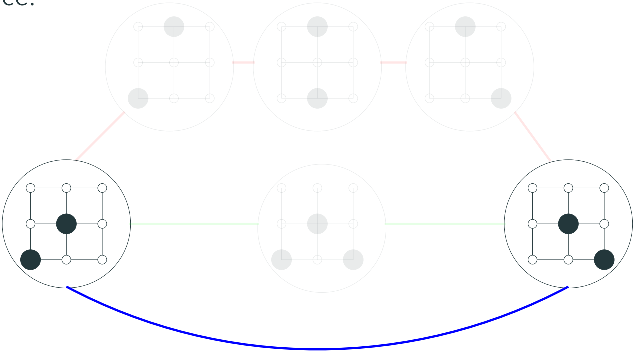
- Each token-set is a vertex.
- Two token-sets X, Y are adjacent if one can be obtained from the other by applying R (TS/TAR(k)/TJ) exactly once.



Independent Set Reconfiguration in a Graph

One can also form the corresponding reconfiguration graph.

- Each token-set is a vertex.
- Two 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 vertex.
- Two 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 vertices of a reconfiguration graph?
- Shortest Reconfiguration: find a shortest path (if exists) between two vertices of a reconfiguration graph?
- Connectivity: a reconfiguration graph is connected?
- Diameter: the diameter of a reconfiguration graph is bounded?

Why Independent Sets?

- If all tokens are identical and we have no restriction on the tokens.

Why Independent Sets?

- If all tokens are identical and we have no restriction on the tokens.

Theorem (Heuvel 2013)

One can decide in polynomial time whether it is possible to transform a token-set X into another token-set Y using at most N TS-moves.

Why Independent Sets?

- If all tokens are identical and we have no restriction on the tokens.

Theorem (Heuvel 2013)

One can decide in polynomial time whether it is possible to transform a token-set X into another token-set Y using at most N TS-moves.

- If there is exactly one special token and other tokens are identical.

Why Independent Sets?

- If all tokens are identical and we have no restriction on the tokens.

Theorem (Heuvel 2013)

One can decide in polynomial time whether it is possible to transform a token-set X into another token-set Y using at most N TS-moves.

- If there is exactly one special token and other tokens are identical.

Theorem (Heuvel 2013)

It is NP-complete to decide whether it is possible to transform a token-set X into another token-set Y using at most N TS-moves.

Why Independent Sets?

- If all tokens are identical and we have no restriction on the tokens.

Theorem (Heuvel 2013)

One can decide in polynomial time whether it is possible to transform a token-set X into another token-set Y using at most N TS-moves.

- If there is exactly one special token and other tokens are identical.

Theorem (Heuvel 2013)

It is NP-complete to decide whether it is possible to transform a token-set X into another token-set Y using at most N TS-moves.

- So, what happen when all tokens are identical and satisfy some additional property (say, independent)?

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]
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]

Complexity under TS/TJ/TAR

Theorem (Kamiński et al. 2012)

TAR *and* TJ *are equivalent, in the sense that, given two independent sets* I, J *of size* k *of a graph* G*,*

(a) *From a* TJ*-sequence between* I *and* J*, one can construct a* TAR$(k-1)$*-sequence between* I *and* J*.*

(b) *From a* TAR$(k-1)$*-sequence between* I *and* J*, one can construct a* TJ*-sequence between* I *and* J*.*

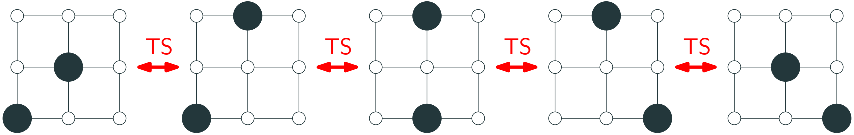
Complexity under TS/TJ/TAR

Graph	TS	TAR/TJ
planar	PSPACE-complete [Hearn and Demaine 2005]	PSPACE-complete [Hearn and Demaine 2005]
cograph (P_4-free)	P [Kamiński et al. 2012]	P [Bonsma 2014]
bipartite	PSPACE-complete [Lokshtanov and Mouawad 2018]	NP-complete [Lokshtanov and Mouawad 2018]
split	PSPACE-complete [Belmonte et al. 2018]	P [Kamiński et al. 2012]

In all problems, the REACHABILITY question is considered.

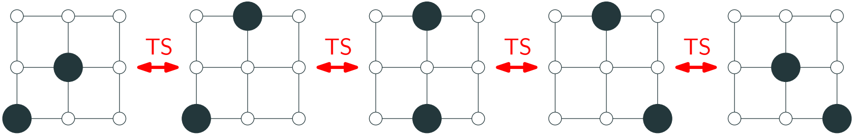
Detour of token(s)

- Under TS, sometimes a token needs to make detour.



Detour of token(s)

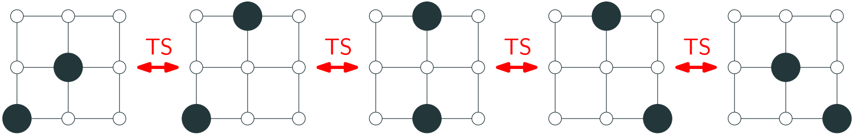
- Under TS, sometimes a token needs to make detour.



- This makes the problem of finding a shortest TS-sequence between two independent sets (if exists) quite challenging, even on a tree.

Detour of token(s)

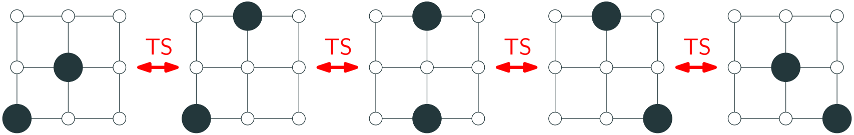
- Under TS, sometimes a token needs to make detour.



- This makes the problem of finding a shortest TS-sequence between two independent sets (if exists) quite challenging, even on a tree.
  - Very recently, K. Sugimori (University of Tokyo) announced at AAAC 2018 (the 11th Annual Meeting of the Asian Association for Algorithms and Computation) that the problem can be solved in polynomial time for trees.

Detour of token(s)

- Under TS, sometimes a token needs to make detour.



- This makes the problem of finding a shortest TS-sequence between two independent sets (if exists) quite challenging, even on a tree.
  - Very recently, K. Sugimori (University of Tokyo) announced at AAAC 2018 (the 11th Annual Meeting of the Asian Association for Algorithms and Computation) that the problem can be solved in polynomial time for trees.
  - To the best of our knowledge, it is unknown whether the problem can be solved efficiently when the given graph contains cycle(s).

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 $\mathsf{bw}(G)$ of a graph G is defined as follows

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

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

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.

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 $\mathsf{bw}(G)$ of a graph G is defined as follows

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

  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 whether the problem can be solved efficiently even for graphs of bandwidth 2.

Open Problems

Open Problems

Problem 1

Is there any graph class $\mathcal{G}$ such that INDEPENDENT SET for $\mathcal{G}$ is NP-hard, while some variant of INDEPENDENT SET RECONFIGURATION for $\mathcal{G}$ is in P?

Conjecture

$\mathcal{G}$ is even-hole-free, i.e., for a graph $G \in \mathcal{G}$, G contains no induced n-cycles for $n \geq 4$.

Problem 2

What is the complexity of deciding if there is a TS-sequence containing at most N moves between two independent sets when the given graph contains cycle(s)?

Conjecture

The problem of deciding if there is a TS-sequence containing at most N TS-moves between two independent sets is NP-hard for cactus graphs.

Open Problems

Problem 3

What is the complexity of Independent Set Reconfiguration for graphs of bandwidth 2?

Conjecture

Independent Set Reconfiguration for graphs of bandwidth 2 can be solved in polynomial time.

Thank you very much for your attention!

Belmonte, Rémy, Eun Jung Kim, Michael Lampis, Valia Mitsou, Yota Otachi, and Florian Sikora (2018). "Token Sliding on Split Graphs." In: *arXiv preprint*. arXiv: 1807.05322.

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.

Bonsma, Paul (2014). "Independent set reconfiguration in cographs." In: *Proceedings of the 40th International Workshop on Graph-Theoretic Concepts in Computer Science, WG 2014*. Vol. 8747. Lecture Notes in Computer Science. Springer, pp. 105–116. DOI: 10.1007/978-3-319-12340-0_9.

Bibliography ii

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.

Lokshtanov, Daniel and Amer E. Mouawad (2018). “The complexity of independent set reconfiguration on bipartite graphs.” In: *Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018*. Ed. by Artur Czumaj. SIAM, pp. 185–195. DOI: 10.1137/1.9781611975031.13.

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.