A Brief Introduction to Independent Set Reconfiguration and Related Problems

Duc A. Hoang

August 31, 2018

anhduc.hoang1990@gmail.com

Seminar at Department of Informatics, VNU University of Science

Outline

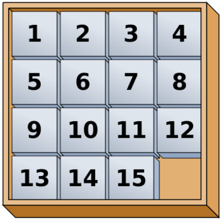
General Framework: Combinatorial Reconfiguration

Independent Set Reconfiguration

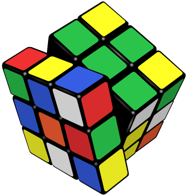
My PhD Journey: The SLIDING TOKEN problem

General Framework: Combinatorial Reconfiguration

Reconfiguration: An Overview



15-PUZZLE

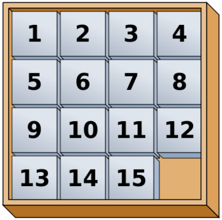


RUBIK'S CUBE

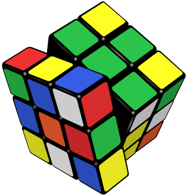
RUSH-HOUR

The figures were originally downloaded from various online sources, especially Wikipedia

Reconfiguration: An Overview



15-PUZZLE



RUBIK'S CUBE

RUSH-HOUR

They are all examples of Reconfiguration Problems:

Given two configurations, and a specific rule describing how a configuration can be transformed into a (slightly) different one

Ask whether one can transform one configuration into another by applying the given rule repeatedly

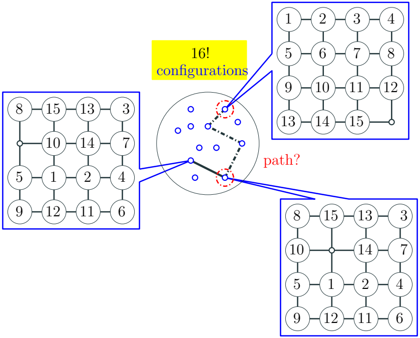
The figures were originally downloaded from various online sources, especially Wikipedia

Reconfiguration: An Overview

Reconfiguration Graph

Vertex Set: Set of all configurations

Edge Set: Two vertices (configurations) are adjacent if one can be obtained from the other by applying the given rule once



The reconfiguration graph of 15-PUZZLE

Reconfiguration: An Overview

New insights into the computational complexity theory

Given	Configurations A, B, and a transformation rule
Decision	Decide if A can be transformed into B
Find	A transformation sequence?
Shortest	A shortest transformation sequence?

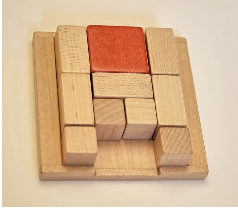
See also the “Masterclass Talk: Algorithms and Complexity for Japanese Puzzles” by R. Uehara at ICALP 2015

The figures were originally downloaded from various online sources, especially Wikipedia

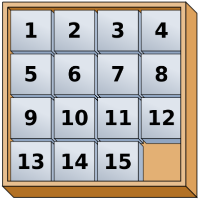
Reconfiguration: An Overview

New insights into the computational complexity theory

	Sliding-block Puzzle	15-puzzle
Decision	PSPACE-complete	Linear
Find	PSPACE-complete	Poly-time
Shortest	PSPACE-complete	NP-complete



Sliding-block Puzzle



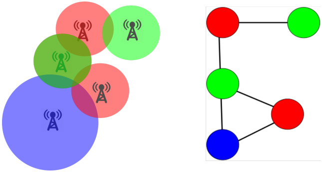
15-puzzle

See also the "Masterclass Talk: Algorithms and Complexity for Japanese Puzzles" by R. Uehara at ICALP 2015

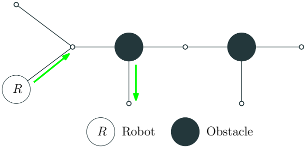
The figures were originally downloaded from various online sources, especially Wikipedia

Reconfiguration: An Overview

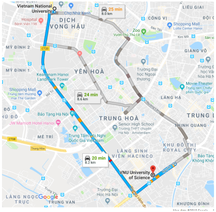
Real-world situations involving movement and change



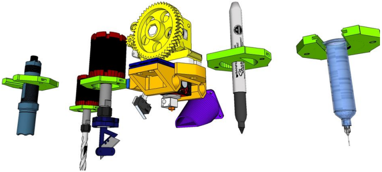
Frequency Re-Assignment



Robot Motion



Path Planning



Printer's Multi-heads Motion

The figures were originally downloaded from various online sources, especially Wikipedia

Reconfiguration: An Overview

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/

Independent Set Reconfiguration

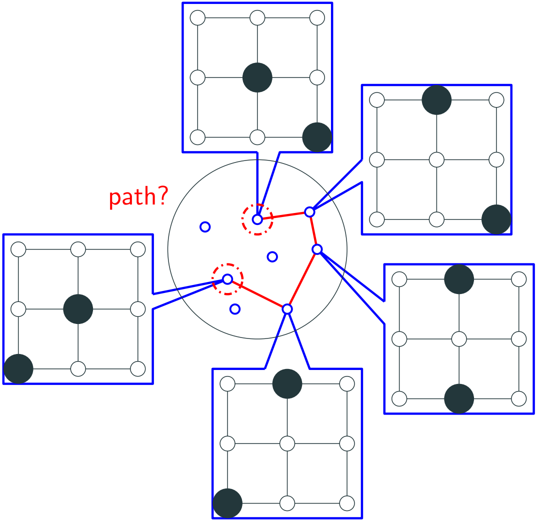
Independent Set Reconfiguration: Definition

Given two independent sets of a graph; and one of the following rules

- **Token Sliding** (TS) [Hearn and Demaine 2005]
- **Token Addition and Removal** (TAR(k)) [Ito et al. 2011]
- **Token Jumping** (TJ) [Kamiński et al. 2012]

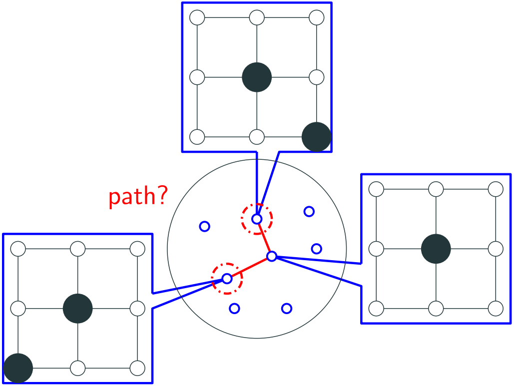
Ask whether one can transform one independent set into another by applying the given rule repeatedly

Independent Set Reconfiguration: Transformation Rule



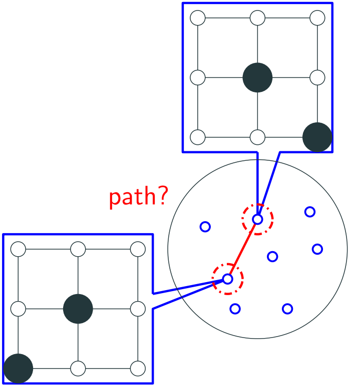
Token Sliding (TS): Swap one member for one non-member that is adjacent to the member (see [Hearn and Demaine 2005])

Independent Set Reconfiguration: Transformation Rule



Token Addition and Removal (TAR(k)): Either add or remove one vertex, while keeping the size of the independent sets at least some given threshold k (see [Ito et al. 2011])

Independent Set Reconfiguration: Transformation Rule



Token Jumping (TJ): Swap one member for one non-member
(see [Kamiński et al. 2012])

Independent Set Reconfiguration: Complexity

Graph	Adjacency			Complexity	Reference
	TS	TAR	TJ		
planar ∩ maximum degree 3	○	○	○	PSPACE-complete	[Hearn and Demaine 2005]
general	○	○	○	PSPACE-complete	[Ito et al. 2011]
line ⇐ MATCHING RECONF.		○	○	P	
perfect	○	○	○	PSPACE-complete	[Kamiński et al. 2012]
even-hole-free		○	○	P	
cograph (P_4-free)	○			P	
cograph (P_4-free)		○	○	P	[Bonsma 2014]
bounded bandwidth	○	○	○	PSPACE-complete	[Wrochna 2014]
claw-free (⊃ line)	○	○	○	P	[Bonsma et al. 2014]
tree (⊂ even-hole-free)	○	○	○	P	[Demaine et al. 2014]
bipartite permutation	○			P	[Fox-Epstein et al. 2015]
bipartite distance-hereditary	○				
planar ∩ maximum degree 3 ∩ bounded bandwidth	○	○	○	PSPACE-complete	[van der Zanden 2015]
cactus		○	○	P	[Mouawad et al. 2014]
cactus	○			P	[Hoang and Uehara 2016]
interval (⊂ even-hole-free)	○	○	○	P	[Bonamy and Bousquet 2017]
bipartite	○			PSPACE-complete	[Lokshtanov and Mouawad 2018]
		○	○	NP-complete	
split	○			PSPACE-complete	[Belmonte et al. 2018]
split (⊂ even-hole-free)		○	○	P	

Independent Set Reconfiguration: Related Problems

- Reconfiguration Graph
  - Find a (shortest) path between two vertices (configurations)
  - Determine whether the reconfiguration graph is connected
  - Determine whether its diameter is bounded
  - Determine whether the original graph and the corresponding reconfiguration graph are isomorphic/belong to the same graph class
- Optimization
  - Allow p tokens to move simultaneously (under TJ). Find smallest p such that there is a transformation sequence between any two independent sets of size k.
  - Find an independent set of maximum size that is reachable from an initial independent set I_0 of size at least k using TAR(k).

Independent Set Reconfiguration: Related Problems

- Independent Set as a Vertex-Subset
  - Reconfiguration of Cliques
  - Reconfiguration of Vertex Covers
  - Reconfiguration of (Labeled/Unlabeled) Tokens
- Independent Set as an Edgeless Induced Subgraph
  - Reconfiguration of (Induced/Spanning) Subgraphs
- Independent Set as a 1-colorable set
  - Reconfiguration of c-colorable sets ($c \geq 1$)

My PhD Journey: The SLIDING TOKEN problem

My PhD Journey: The Sliding Token problem

- Erik D. Demaine, Martin L. Demaine, Takehiro Ito, Hirotaka Ono, Yota Otachi, Ryuhei Uehara, and Takeshi Yamada: Around 2014, they designed a $O(n^2)$-time algorithm for Sliding Token for caterpillars (a subclass of trees). At that time, Sliding Token for trees remains open.

My PhD Journey: The Sliding Token problem

- Erik D. Demaine, Martin L. Demaine, Takehiro Ito, Hirotaka Ono, Yota Otachi, Ryuhei Uehara, and Takeshi Yamada: Around 2014, they designed a $O(n^2)$-time algorithm for Sliding Token for caterpillars (a subclass of trees). At that time, Sliding Token for trees remains open.
- With me joining the project, we designed a $O(n^2)$-time algorithm for Sliding Token for trees. This result was accepted to ISAAC 2014.

My PhD Journey: The Sliding Token problem

- Erik D. Demaine, Martin L. Demaine, Takehiro Ito, Hirotaka Ono, Yota Otachi, Ryuhei Uehara, and Takeshi Yamada: Around 2014, they designed a $O(n^2)$-time algorithm for Sliding Token for caterpillars (a subclass of trees). At that time, Sliding Token for trees remains open.
- With me joining the project, we designed a $O(n^2)$-time algorithm for Sliding Token for trees. This result was accepted to ISAAC 2014.
- With Eli Fox-Epstein joining the project, we improved the running time of our algorithm to $O(n)$. This result was then published in the journal "Theoretical Computer Science" (600, 132–142, 2015).

My PhD Journey: The SLIDING TOKEN problem

- The ideas of Eli for improving our algorithm for trees are quite useful. We (Eli Fox-Epstein, Yota Otachi, Ryuhei Uehara, and I) thought (at that time) that our algorithm can be extended to the case for bipartite graphs using similar ideas.

My PhD Journey: The Sliding Token problem

- The ideas of Eli for improving our algorithm for trees are quite useful. We (Eli Fox-Epstein, Yota Otachi, Ryuhei Uehara, and I) thought (at that time) that our algorithm can be extended to the case for bipartite graphs using similar ideas.
- Though we has not been successful with bipartite graphs, we designed tools and techniques for showing that Sliding Token for bipartite permutation graphs and bipartite distance-hereditary graphs can be solved in $O(n^3)$ time. This result was accepted to ISAAC 2015 (presented by Eli).

My PhD Journey: The SLIDING TOKEN problem

- The ideas of Eli for improving our algorithm for trees are quite useful. We (Eli Fox-Epstein, Yota Otachi, Ryuhei Uehara, and I) thought (at that time) that our algorithm can be extended to the case for bipartite graphs using similar ideas.
- Though we has not been successful with bipartite graphs, we designed tools and techniques for showing that SLIDING TOKEN for bipartite permutation graphs and bipartite distance-hereditary graphs can be solved in $O(n^3)$ time. This result was accepted to ISAAC 2015 (presented by Eli).
- Very recently, Daniel Lokshtanov and Amer E. Mouawad showed that SLIDING TOKEN for bipartite graphs is PSPACE-complete. Their result (presented at SODA 2018) implied that our conjecture is indeed wrong.

My PhD Journey: The Sliding Token problem

- From April to July 2016, I visited Zhou&Ito Lab, Tohoku University. At that time, I had some rough ideas for designing a poly-time algorithm for Sliding Token on cactus graphs (generalizing the techniques for trees).

My PhD Journey: The SLIDING TOKEN problem

- From April to July 2016, I visited Zhou&Ito Lab, Tohoku University. At that time, I had some rough ideas for designing a poly-time algorithm for SLIDING TOKEN on cactus graphs (generalizing the techniques for trees).
- After several useful discussions with Takehiro Ito and the members of his laboratory, including Tatsuhiko Hatanaka, Haruka Mizuta, and Yuhei Moriya, I have finally been able to complete the algorithm. The result (co-authored with my supervisor Ryuhei Uehara) was then accepted to ISAAC 2016.

My PhD Journey: The Sliding Token problem

- From the algorithm for cacti, we (Eli Fox-Epstein, Ryuhei Uehara, and I) designed a $O(n^3)$-time algorithm for Sliding Token for block graphs. This result was presented at WALCOM 2017.

My PhD Journey: The Sliding Token problem

- From the algorithm for cacti, we (Eli Fox-Epstein, Ryuhei Uehara, and I) designed a $O(n^3)$-time algorithm for Sliding Token for block graphs. This result was presented at WALCOM 2017.
- Very recently (around March 2018), Mariana Teatini Ribeiro and Vinícius Fernandes dos Santos showed us a counter-example of a proposition in our algorithm (for block graphs). As a result, the complexity of Sliding Token for block graphs has not yet been settled.

My PhD Journey: The SLIDING TOKEN problem

- Even though SLIDING TOKEN can be solved in linear time for trees, finding a shortest TS-sequence between two independent sets of a tree remains open. We (my supervisor and I) conjectured that the problem can be solved in poly-time for trees, and becomes NP-hard for graphs containing cycles.

My PhD Journey: The SLIDING TOKEN problem

- Even though SLIDING TOKEN can be solved in linear time for trees, finding a shortest TS-sequence between two independent sets of a tree remains open. We (my supervisor and I) conjectured that the problem can be solved in poly-time for trees, and becomes NP-hard for graphs containing cycles.
- Recently, Ken Sugimori announced at AAAC 2018 that finding a shortest TS-sequences in trees can be done in $O(poly(n))$ time using a dynamic programming approach. We are still waiting for his detailed proof, even though we believe that his approach is correct.

My PhD Journey: The SLIDING TOKEN problem

- Even though SLIDING TOKEN can be solved in linear time for trees, finding a shortest TS-sequence between two independent sets of a tree remains open. We (my supervisor and I) conjectured that the problem can be solved in poly-time for trees, and becomes NP-hard for graphs containing cycles.
- Recently, Ken Sugimori announced at AAAC 2018 that finding a shortest TS-sequences in trees can be done in $O(poly(n))$ time using a dynamic programming approach. We are still waiting for his detailed proof, even though we believe that his approach is correct.
- Sugimori's $O(poly(n))$-time algorithm seems to have $poly(n)$ with high degree. With Amanj Khorramian joining the project, we designed a $O(n^2)$-time algorithm for finding shortest TS-sequences in spider graphs (a subclass of trees). We hope that our result provides a framework for improving Sugimori's algorithm. This result was recently presented at WAAC 2018 (August 27, 2018) by my supervisor Ryuhei Uehara.

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

Bonsma, Paul, Marcin Kamiński, and Marcin Wrochna (2014). “Reconfiguring independent sets in claw-free graphs.” In: *Proceedings of the 14th Scandinavian Symposium and Workshops on Algorithm Theory, SWAT 2014*. Ed. by R. Ravi and Inge Li Gørtz. Vol. 8503. Lecture Notes in Computer Science. Springer, pp. 86–97. DOI: 10.1007/978-3-319-08404-6_8.

Demaine, Erik D., Martin L. Demaine, Eli Fox-Epstein, Duc A. Hoang, Takehiro Ito, Hirotaka Ono, Yota Otachi, Ryuhei Uehara, and Takeshi Yamada (2014). “Polynomial-time algorithm for sliding tokens on trees.” In: *Proceedings of the 25th International Symposium on Algorithms and Computation, ISAAC 2014*. Ed. by Hee-Kap Ahn and Chan-Su Shin. Vol. 8889. Lecture Notes in Computer Science. Springer, pp. 389–400. DOI: 10.1007/978-3-319-13075-0_31.

Fox-Epstein, Eli, Duc A. Hoang, Yota Otachi, and Ryuhei Uehara (2015). "Sliding token on bipartite permutation graphs." In: *Proceedings of the 26th International Symposium on Algorithms and Computation, ISAAC 2015*. Ed. by K. Elbassioni and K. Makino. Vol. 9472. Lecture Notes in Computer Science. Springer, pp. 237–247. DOI: 10.1007/978-3-662-48971-0_21.

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.

Hoang, Duc A. and Ryuhei Uehara (2016). “Sliding tokens on a cactus.” In: *Proceedings of the 27th International Symposium on Algorithms and Computation, ISAAC 2016*. Ed. by Seok-Hee Hong. Vol. 64. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 37:1–37:26. DOI: 10.4230/LIPIcs.ISAAC.2016.37.

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`.

Mouawad, Amer E., Naomi Nishimura, and Venkatesh Raman (2014). "Vertex cover reconfiguration and beyond." In: *Proceedings of the 25th International Symposium on Algorithms and Computation, ISAAC 2014*. Ed. by Hee-Kap Ahn and Chan-Su Shin. Vol. 8889. Lecture Notes in Computer Science. Springer, pp. 452–463. DOI: `10.1007/978-3-319-13075-0_36`.

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

van der Zanden, Tom C. (2015). "Parameterized Complexity of Graph Constraint Logic." In: *Proceedings of the 10th International Symposium on Parameterized and Exact Computation, IPEC 2015*. Ed. by Thore Husfeldt and Iyad Kanj. Vol. 43. Leibniz International Proceedings in Informatics. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, pp. 282–293. DOI: 10.4230/LIPIcs.IPEC.2015.282.

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