Hoàng Anh Đức
hoanganhduc@hus.edu.vn

VNU University of Science
Hanoi, Vietnam

October 30, 2024

An Introduction to Combinatorial Reconfiguration

Seminar at VNU-HUS

Contents

A Quick Note

- We talk about *decision problems* (output YES or NO)
- Complexity Classes (time (# steps) and space (# memory cells) are w.r.t length of the input)
  - P: Problems can be "solved efficiently" in polynomial time
  - NP: Problems can be "verified efficiently" in polynomial time
  - PSPACE: Problems can be "solved efficiently" in polynomial space

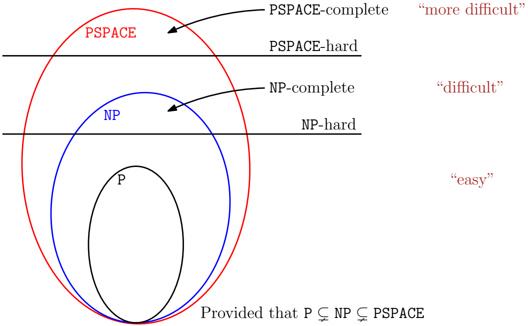
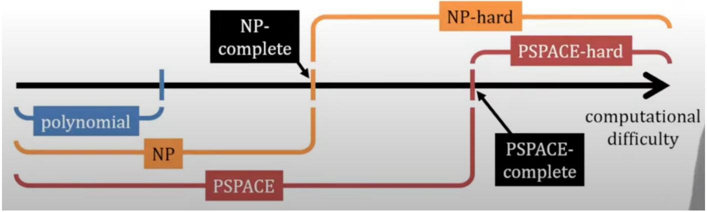
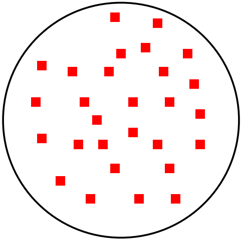


Figure: Complexity Classes P, NP, and PSPACE

A Quick Note

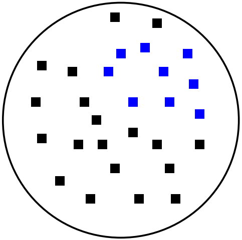


“easy” problem



any instance can be solved in **polynomial time**

“hard” problem



some instance **cannot** be solved in **polynomial time** (unless P = NP)

A Brief Overview of Reconfiguration

Reconfiguration Setting

- A description of what *states* (≡ *configurations*) are
- One or more *allowed moves* between states (≡ *reconfiguration rule(s)*)

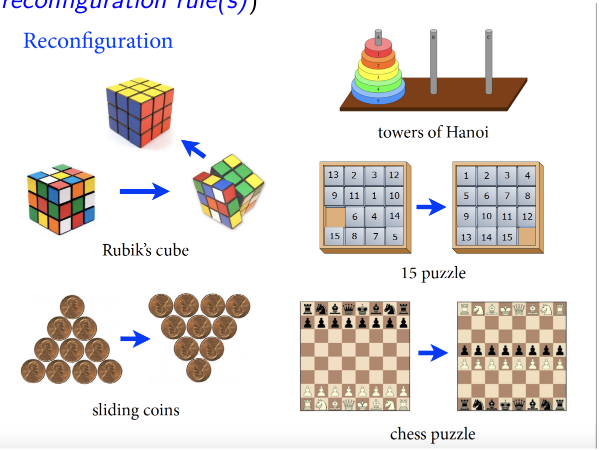


Figure: Reconfiguration [Anna Lubiw, CoRe2019]

A Brief Overview of Reconfiguration

Two major viewpoints

as a *process* or as a *graph*

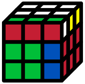
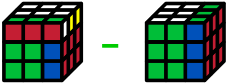
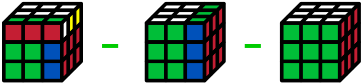
Reconf. Process	**Reconf. Graph**	**Example** RUBIK'S CUBE
state	node	
move *[reconf. rule]*	edge	
seq. of moves *[reconf. sequence]*	path	

Figure: Combinatorial Reconfiguration

A Brief Overview of Reconfiguration

Two major directions

Algorithmic and *Graph-Theoretic*

- **Algorithmic Questions**
  - REACHABILITY: Given two states S and T, is there a sequence of moves that *transforms S into T*? **[One of the most considered questions]**
  - SHORTEST TRANSFORMATION: Given two states S and T and some positive integer ℓ, is there a sequence of moves that *transforms S into T using at most ℓ moves*?
  - CONNECTIVITY: Is there a sequence of moves *between any pair of states*?
  - and so on
- **Graph-Theoretic Questions**
  - GRAPH PROPERTIES: Is the reconfiguration graph *connected*? *bipartite*? *Eulerian*? *Hamiltonian*?, and so on
  - GRAPH CLASSIFICATION: Does the reconfiguration graph *belong to some specific graph class* (e.g., planar graphs, perfect graphs, etc.)?
  - and so on

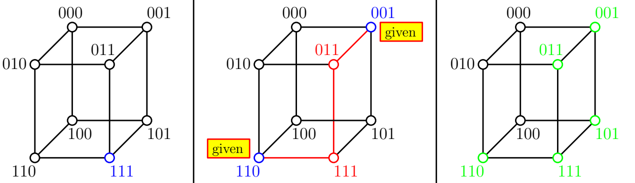
Understanding Solution Space and Complexity of Problems

Reconfiguration vs. Solution Space

- *States*: are *feasible solutions* of a computational problem
- *Reconfiguration rule*: describes *"small" changes* in a feasible solution

SAT formula $\varphi = (x \wedge y) \vee z$

State $\equiv$ **Feasible Solution** (assignment of variables (x, y, z) makes φ true)
Reconfiguration Rule: flip exactly one bit



Search Problem
Check if **at least one** feasible solution exists among 2^n candidates for n variables

Reconfiguration Problem
Find **a sequence** of adjacent feasible solutions among 2^n candidates for n variables

Enumeration Problem
List **all** feasible solutions among 2^n candidates for n variables

Note: In **Reconfiguration** Problem, *we do NOT know which ones among the other $2^n - 2$ candidates are feasible solutions*

Understanding Solution Space and Complexity of Problems

[Gopalan, Kolaitis, Maneva, and Papadimitriou 2009]

- ... studied *structural and connectivity-related properties of the space of solutions of Boolean satisfiability problems*

SIAM J. Comput.
Vol. 38, No. 6, pp. 2330–2355

THE CONNECTIVITY OF BOOLEAN SATISFIABILITY: COMPUTATIONAL AND STRUCTURAL DICHOTOMIES*

PARIKSHIT GOPALAN†, PHOKION G. KOLAITIS‡, ELITZA MANEVA‡, AND CHRISTOS H. PAPADIMITRIOU§

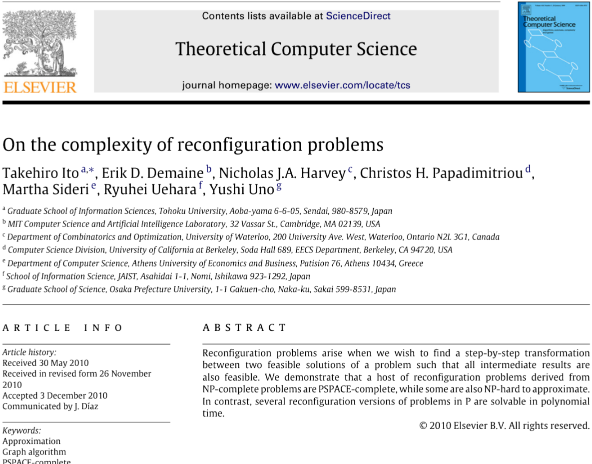
Abstract. Boolean satisfiability problems are an important benchmark for questions about complexity, algorithms, heuristics, and threshold phenomena. Recent work on heuristics and the satisfiability threshold has centered around the structure and connectivity of the solution space. Motivated by this work, we study structural and connectivity-related properties of the space of solutions of Boolean satisfiability problems and establish various dichotomies in Schaefer's framework. On the structural side, we obtain dichotomies for the kinds of subgraphs of the hypercube that can be induced by the solutions of Boolean formulas, as well as for the diameter of the connected components of the solution space. On the computational side, we establish dichotomy theorems for the complexity of the connectivity and *st*-connectivity questions for the graph of solutions of Boolean formulas. Our results assert that the intractable side of the computational dichotomies is PSPACE-complete, while the tractable side—which includes but is not limited to all problems with polynomial-time algorithms for satisfiability—is in P for the *st*-connectivity question, and in coNP for the connectivity question. The diameter of components can be exponential for the PSPACE-complete cases, whereas in all other cases it is linear; thus, diameter and complexity of the connectivity problems are remarkably aligned. The crux of our results is an expressibility theorem showing that in the tractable cases, the subgraphs induced by the solution space possess certain good structural properties, whereas in the intractable cases, the subgraphs can be arbitrary.

Key words. Boolean satisfiability, computational complexity, PSPACE, PSPACE-completeness, dichotomy theorems, graph connectivity

Understanding Solution Space and Complexity of Problems

[Ito et al. 2011]

- … *find a step-by-step transformation between two feasible solutions of a problem such that all intermediate results are also feasible*



Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

On the complexity of reconfiguration problems

Takehiro Ito [a,*], Erik D. Demaine [b], Nicholas J.A. Harvey [c], Christos H. Papadimitriou [d], Martha Sideri [e], Ryuhei Uehara [f], Yushi Uno [g]

[a] Graduate School of Information Sciences, Tohoku University, Aoba-yama 6-6-05, Sendai, 980-8579, Japan
[b] MIT Computer Science and Artificial Intelligence Laboratory, 32 Vassar St., Cambridge, MA 02139, USA
[c] Department of Combinatorics and Optimization, University of Waterloo, 200 University Ave. West, Waterloo, Ontario N2L 3G1, Canada
[d] Computer Science Division, University of California at Berkeley, Soda Hall 689, EECS Department, Berkeley, CA 94720, USA
[e] Department of Computer Science, Athens University of Economics and Business, Patision 76, Athens 10434, Greece
[f] School of Information Science, JAIST, Asahidai 1-1, Nomi, Ishikawa 923-1292, Japan
[g] Graduate School of Science, Osaka Prefecture University, 1-1 Gakuen-cho, Naka-ku, Sakai 599-8531, Japan

ARTICLE INFO

Article history:
Received 30 May 2010
Received in revised form 26 November 2010
Accepted 3 December 2010
Communicated by J. Díaz

Keywords:
Approximation
Graph algorithm
PSPACE-complete

ABSTRACT

Reconfiguration problems arise when we wish to find a step-by-step transformation between two feasible solutions of a problem such that all intermediate results are also feasible. We demonstrate that a host of reconfiguration problems derived from NP-complete problems are PSPACE-complete, while some are also NP-hard to approximate. In contrast, several reconfiguration versions of problems in P are solvable in polynomial time.

Understanding Solution Space and Complexity of Problems

[Ito et al. 2011]

- ... *find a step-by-step transformation between two feasible solutions of a problem such that all intermediate results are also feasible*

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

On the complexity of reconfiguration problems

Takehiro Ito[a,*], Erik D. Demaine[b], Nicholas J.A. Harvey[c], Christos H. Papadimitriou[d], Martha Sideri[e], Ryuhei Uehara[f], Yushi Uno[g]

- Showed that several classic NP-complete problems have PSPACE-complete reconfiguration variants
  - Deciding the "reachability" between solutions of a "difficult" problem may sometimes be "more difficult" than the problem itself
- Named the area "Reconfiguration". Opened new research directions

Understanding Solution Space and Complexity of Problems

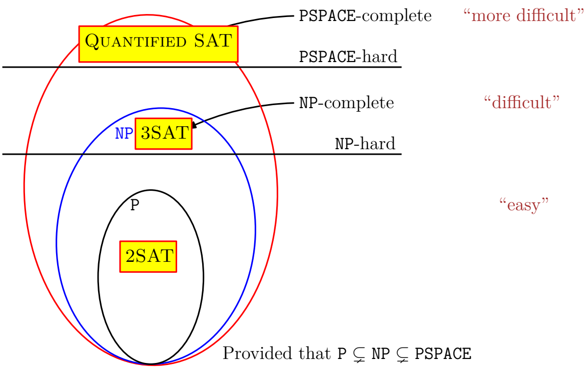


Figure: Using problems to "characterize" complexity classes

Understanding Solution Space and Complexity of Problems

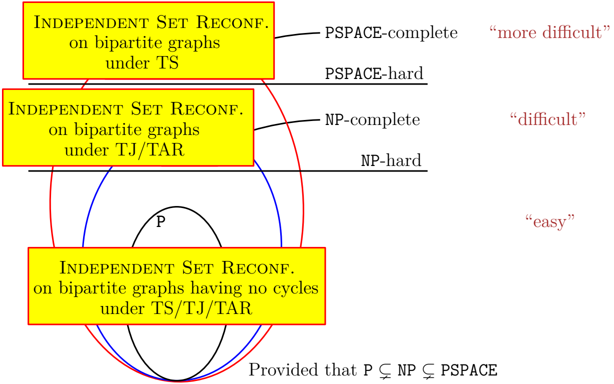


Figure: Using problems to "characterize" complexity classes

Understanding Solution Space and Complexity of Problems

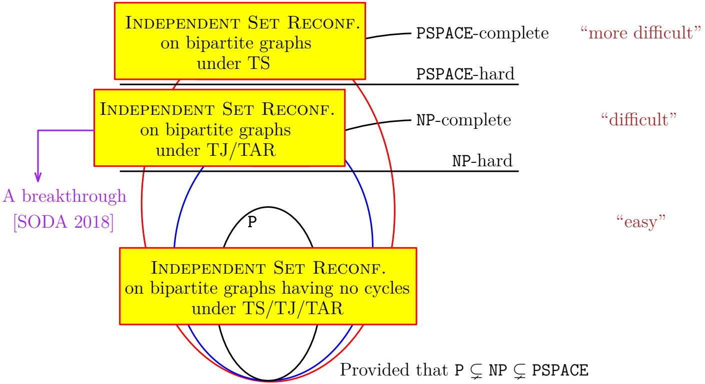


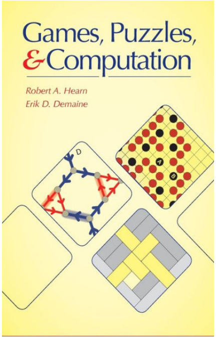
Figure: Using problems to "characterize" complexity classes

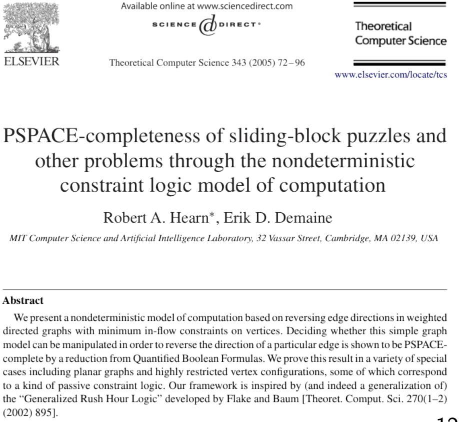
Daniel Lokshtanov and Amer E. Mouawad (2019). "The Complexity of Independent Set Reconfiguration on Bipartite Graphs". In: *ACM Transactions on Algorithms* 15.1, 7:1–7:19. DOI: 10.1145/3280825

Understanding Solution Space and Complexity of Problems

Reconfiguration *provides new powerful tools for proving the hardness of a problem*

- One of such tools is the *Nondeterministic Constraint Logic (NCL)*, *first introduced* in [Hearn and Demaine 2005]





ELSEVIER

Available online at www.sciencedirect.com

SCIENCE DIRECT®

Theoretical Computer Science 343 (2005) 72–96

Theoretical Computer Science

www.elsevier.com/locate/tcs

PSPACE-completeness of sliding-block puzzles and other problems through the nondeterministic constraint logic model of computation

Robert A. Hearn*, Erik D. Demaine

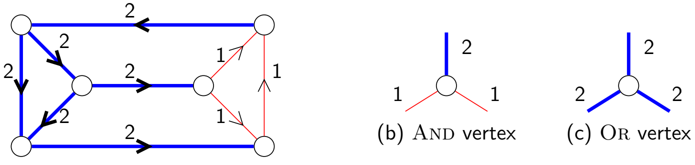
MIT Computer Science and Artificial Intelligence Laboratory, 32 Vassar Street, Cambridge, MA 02139, USA

Abstract

We present a nondeterministic model of computation based on reversing edge directions in weighted directed graphs with minimum in-flow constraints on vertices. Deciding whether this simple graph model can be manipulated in order to reverse the direction of a particular edge is shown to be PSPACE-complete by a reduction from Quantified Boolean Formulas. We prove this result in a variety of special cases including planar graphs and highly restricted vertex configurations, some of which correspond to a kind of passive constraint logic. Our framework is inspired by (and indeed a generalization of) the “Generalized Rush Hour Logic” developed by Flake and Baum [Theoret. Comput. Sci. 270(1–2) (2002) 895].

Understanding Solution Space and Complexity of Problems

- **Input:**
  - Each *state/configuration* involves a graph having red (weight 1) and blue (weight 2) edges where each edge is oriented such that *(*) the sum of weights of in-coming arcs at each vertex is at least 2*
  - **Reconfiguration Rule:** Each *move* involves re-orienting an edge such that (*) is satisfied
- **Question:** Is there a sequence of moves that transforms one given configuration into another? (PSPACE-complete even on *planar graphs* having only *two types of vertices*)

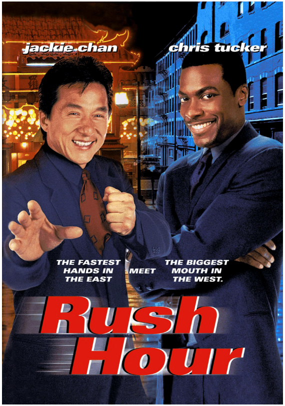


(a) An NCL configuration (b) AND vertex (c) OR vertex

Understanding Solution Space and Complexity of Problems

An Application of NCL

Rush Hour (the puzzle, not the movie) *is* PSPACE*-complete*



Understanding Solution Space and Complexity of Problems

[Flake and Baum 2002]

Reduce from QUANTIFIED SAT. *Use 3 "primitive devices" and more complicated "gadgets" built from the "devices"*

ELSEVIER

Theoretical Computer Science 270 (2002) 895–911

Theoretical Computer Science

www.elsevier.com/locate/tcs

Mathematical Games

Rush Hour is PSPACE-complete, or "Why you should generously tip parking lot attendants"

Gary William Flake *, Eric B. Baum

NEC Research Institute, 4 Independence Way, Princeton, NJ 08540, USA

Received June 1999; revised February 2001; accepted February 2001
Communicated by A. Fraenkel

Abstract

Rush Hour is a children's game that consists of a grid board, several cars that are restricted to move either vertically or horizontally (but not both), a special target car, and a single exit on the perimeter of the grid. The goal of the game is to find a sequence of legal moves that allows the target car to exit the grid. We consider a slightly generalized version of the game that uses an $n \times n$ grid and assume that we can place the single exit and target car at any location we choose on initialization of the game.

In this work, we show that deciding if the target car can legally exit the grid is PSPACE-complete. Our constructive proof uses a lazy form of dual-rail reversible logic such that movement of "output" cars can only occur if logical combinations of "input" cars can also move. Emulating this logic only requires three types of devices (two switches and one crossover); thus, our proof technique can be easily generalized to other games and planning problems in which the same three primitive devices can be constructed.

Keywords: Games; PSPACE-completeness; Reversible logic; Motion planning; Dual-rail logic

[Hearn and Demaine 2005]

Reduce from NCL. *Use 2 "gadgets"*

90 R.A. Hearn, E.D. Demaine / Theoretical Computer Science 343 (2005) 72–96

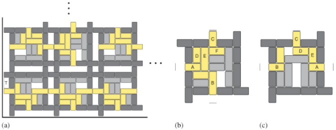


Fig. 14. Rush Hour layout and vertex gadgets. (a) Layout. (b) AND. (c) Protected OR.

generic crossover construction (Section 3.2), we do not need a crossover gadget. (We also do not need the miscellaneous wiring gadgets used in [4].)

Rush Hour layout. We tile the grid with our vertex gadgets, as shown in Fig. 14(a). One block (T) is the target, which must be moved to the bottom left corner; it is released when a particular port block slides into a vertex.

Dark-colored blocks represent the "cell walls", which unlike in our sliding-blocks construction are not shared. They are arranged so that they may not move at all. Light-colored blocks are "trigger" blocks, whose motion serves to satisfy the vertex constraints. Medium-colored blocks are fillers; some of them may move, but they do not disrupt the vertices' operation.

As in the sliding-blocks construction, edges are directed inward by sliding blocks out of the vertex gadgets; edges are directed outward by sliding blocks in. The layout ensures that no port block may ever slide out into an adjacent vertex; this helps keep the cell walls fixed.

Real-World Applications

Robot Motion Planning

[Murata, Kurokawa, and Kokaji 1994]

Self-Assembling Machine

Satoshi Murata, Haruhisa Kurokawa, Shigeru Kokaji
Mechanical Engineering Laboratory, AIST, MITI
1-2 Namiki, Tsukuba, 305 JAPAN

Abstract - The design of a machine which is composed of homogeneous mechanical units is described. We show the design of both hardware and control software of the unit. Each unit can connect with other units and change the connection by itself. In spite of its simple mechanism, a set of these units realizes various mechanical functions. We developed control software of the u
"self-assembly," one of the basic
machine. A set of these units can form
whole system by themselves. The
information about local geome
communication, and cooperate to for
through a diffusion-like process. Ther
controller to supervise these units, a
each unit is completely the same. Thre
been built to test the basic movements,
self-assembly has been verified by com

homogeneous hardware because of geometrical constraints. Therefore, only a few examples of this kind exist. Kokaji [1] made a link locomotion mechanism called a "fractal machine" by using homogeneous link units. It has a recursive structure like Sherpinski's gasket, and the size of the machine can be changed by adding/subtracting units.

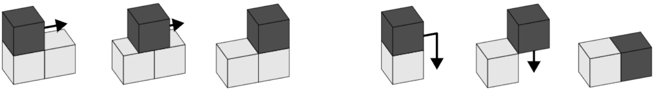
Real-World Applications

Robot Motion Planning

A recent research presented at SWAT (Scandinavian Symposium and Workshops on Algorithm Theory) 2024

Sliding (Hyper-)Cubes [Kostitsyna et al. 2024]

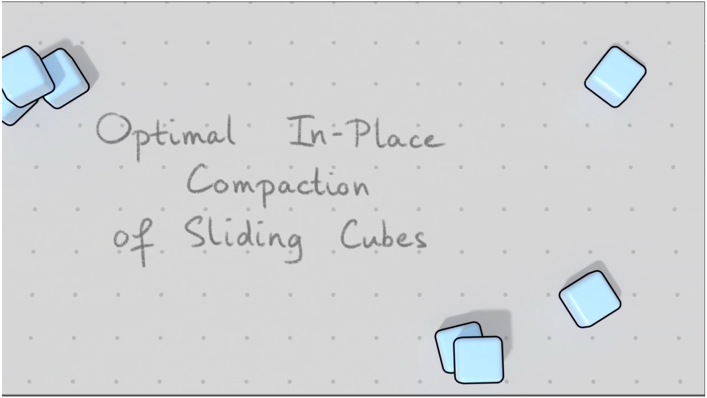
- Each configuration is a connected collection of n robot units ($=$ lattice-aligned unit (hyper-)cubes)
- **Reconfiguration rule:** Slide or Rotation



- *Given two configurations, is there a sequence of moves that transforms one configuration into the other?*
- First universal ($=$ "always yes") result for 2D sliding model is in [Dumitrescu and Pach 2006]
- Other variants
  - Change shapes
  - Change rules ($=$ types of allowed movements)

Real-World Applications

Robot Motion Planning



https://www.youtube.com/watch?v=cRn-ZRu0Z18

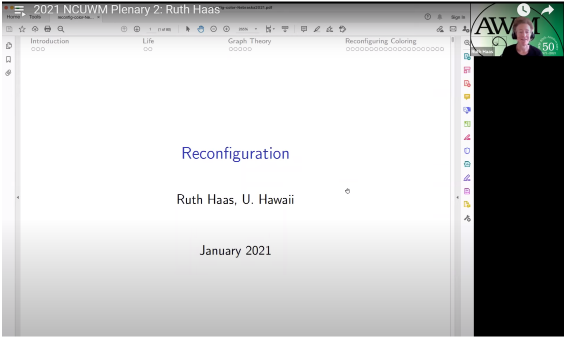
Real-World Applications

Robot Motion Planning

https://www.youtube.com/watch?v=6aZbJS6LZbs

For Motivating You Further

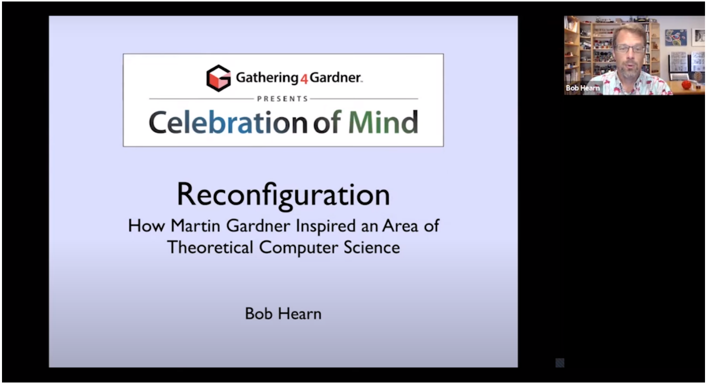
A nice and inspiring introduction to Reconfiguration in Graph Coloring (and other contexts) by Prof. Ruth Haas (U. Hawaii) at the NCUWM (Nebraska Conference for Undergraduate Women in Mathematics) 2021



https://www.youtube.com/watch?v=gApwRCEC89Q

For Motivating You Further

An inspiring talk in 2021 by Robert A. Hearn—one of the authors who introduced NCL [Hearn and Demaine 2005]



https://www.youtube.com/watch?v=4cWVjhBTDSY

For Motivating You Further

A more technical introduction at WALCOM (International Conference and Workshops on Algorithms and Computation) 2022 about Reconfiguration by Prof. Takehiro Ito (Tohoku Univ.)—one of the leading experts in this area

https://youtu.be/gwrIyuT3F8w?t=21308

For Motivating You Further

Mathematics and Art: Unifying Perspectives 18

Heather M. Russell and Radmila Sazdanovic

Heather M. Russell and Radmila Sazdanovic (2021). "Mathematics and Art: Unifying Perspectives". In: *Handbook of the Mathematics of the Arts and Sciences*. Ed. by Bharath Sriraman. Springer, pp. 497–525. DOI: 10.1007/978-3-319-57072-3_125

Contents

Abstract

In this chapter, we explore the interconnection of mathematics and art. We discuss mathematics as a lens to understand artwork and investigate how mathematical thinking and mathematical tools contribute to the process of creating art. Turning then to the manifestation of art within mathematics, we introduce ideas and constructions from mathematical graph theory that can be appreciated

Surveys and Wiki Page

- **General Surveys**
  - 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, p. 52. DOI: 10.3390/a11040052
- **Surveys on Specific Problems**
  - C.M. Mynhardt and S. Nasserasr (2019). "Reconfiguration of Colourings and Dominating Sets in Graphs". In: *50 years of Combinatorics, Graph Theory, and Computing*. Ed. by Fan Chung et al. 1st. CRC Press, pp. 171–191. DOI: 10.1201/9780429280092-10
  - Nicolas Bousquet, Amer E. Mouawad, Naomi Nishimura, and Sebastian Siebertz (2024). "A survey on the parameterized complexity of reconfiguration problems". In: *Computer Science Review* 53. (article 100663). DOI: 10.1016/j.cosrev.2024.100663
- **Online Wiki:** `http://reconf.wikidot.com/`

Thanks for your attention!

References I

Bousquet, Nicolas, Amer E. Mouawad, Naomi Nishimura, and Sebastian Siebertz (2024). "A survey on the parameterized complexity of reconfiguration problems". In: *Computer Science Review* 53. (article 100663). DOI: 10.1016/j.cosrev.2024.100663.

Kostitsyna, Irina, Tim Ophelders, Irene Parada, Tom Peters, Willem Sonke, and Bettina Speckmann (2024). "Optimal In-Place Compaction of Sliding Cubes". In: *Proceedings of SWAT 2024*. Ed. by Hans L. Bodlaender. Vol. 294. LIPIcs. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 31:1–31:14. DOI: 10.4230/LIPIcs.SWAT.2024.31.

Russell, Heather M. and Radmila Sazdanovic (2021). "Mathematics and Art: Unifying Perspectives". In: *Handbook of the Mathematics of the Arts and Sciences*. Ed. by Bharath Sriraman. Springer, pp. 497–525. DOI: 10.1007/978-3-319-57072-3_125.

References II

Lokshtanov, Daniel and Amer E. Mouawad (2019). "The Complexity of Independent Set Reconfiguration on Bipartite Graphs". In: *ACM Transactions on Algorithms* 15.1, 7:1–7:19. DOI: 10.1145/3280825.

Mynhardt, C.M. and S. Nasserasr (2019). "Reconfiguration of Colourings and Dominating Sets in Graphs". In: *50 years of Combinatorics, Graph Theory, and Computing*. Ed. by Fan Chung, Ron Graham, Frederick Hoffman, Ronald C. Mullin, Leslie Hogben, and Douglas B. West. 1st. CRC Press, pp. 171–191. DOI: 10.1201/9780429280092-10.

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

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.

References III

- 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.
- Gopalan, Parikshit, Phokion G. Kolaitis, Elitza N. Maneva, and Christos H. Papadimitriou (2009). "The Connectivity of Boolean Satisfiability: Computational and Structural Dichotomies". In: *SIAM Journal on Computing* 38.6, pp. 2330–2355. DOI: 10.1137/07070440x.
- Dumitrescu, Adrian and János Pach (2006). "Pushing Squares Around". In: *Graphs and Combinatorics* 22, pp. 37–50. DOI: 10.1007/s00373-005-0640-1.

References IV

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.

Flake, Gary William and Eric B. Baum (2002). "Rush Hour is PSPACE-complete, or "Why you should generously tip parking lot attendants"". In: *Theoretical Computer Science* 270.1-2, pp. 895–911. DOI: 10.1016/S0304-3975(01)00173-6.

Murata, S., H. Kurokawa, and S. Kokaji (1994). "Self-assembling machine". In: *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, 441–448 vol.1. DOI: 10.1109/ROBOT.1994.351257.