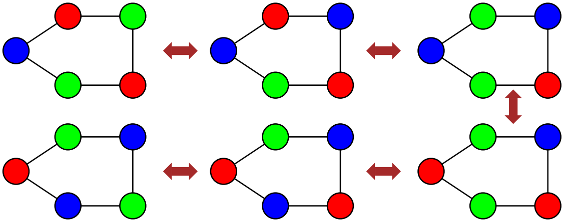
VIASM-KAIST Annual Meeting on Mathematics Sciences



Duc A. Hoang

hoanganhduc@hus.edu.vn

VNU University of Science
Hanoi, Vietnam

October 27–30, 2025

Distance Recoloring

in collaboration with
Niranka Banerjee (Kyoto University, Japan)
Christian Engels (National Institute of Informatics, Japan)

Contents

Introduction to Reconfiguration

Reconfiguration Setting

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

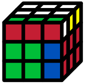
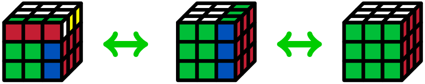
Reconfiguration Process	**Reconfiguration Graph**	**Example** RUBIK'S CUBE
state	node	
move *[reconfiguration rule]*	edge	
sequence of moves *[reconfiguration sequence]*	path	

Figure: Reconfiguration Setting

Introduction to Reconfiguration

Reconfiguration vs. Solution Space

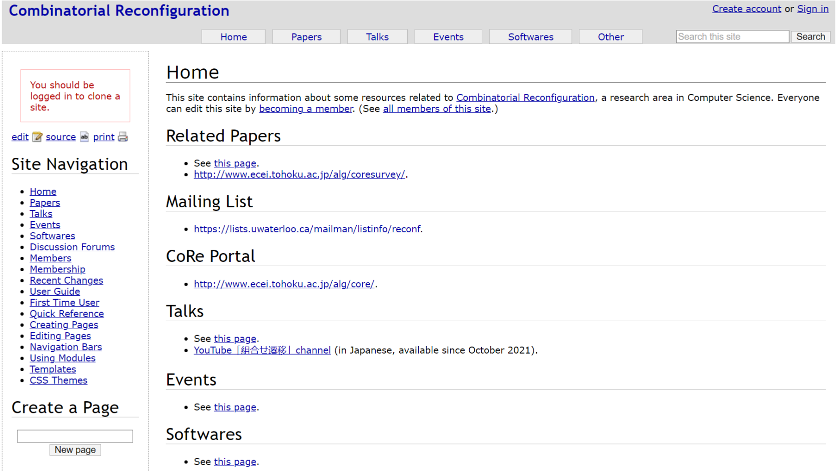
For a computational problem $\mathcal{P}$ (e.g., INDEPENDENT SET, DOMINATING SET, VERTEX-COLORING, etc.)

Reconfiguration	**Solution Space**
States/Configurations	Feasible solutions of $\mathcal{P}$
Allowed Moves	Slight modifications of a solution *without changing its feasibility*

Algorithmic Questions

- REACHABILITY: Given two states S and T, is there a sequence of moves that *transforms S into T*?
- SHORTEST TRANSFORMATION: Given two states S and T and a 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

Introduction to Reconfiguration


Combinatorial Reconfiguration

Create account or Sign in

Home | Papers | Talks | Events | Softwares | Other | Search this site | Search

You should be logged in to clone a site.

edit source print

Site Navigation

- Home
- Papers
- Talks
- Events
- Softwares
- Discussion Forums
- Members
- Membership
- Recent Changes
- User Guide
- First Time User
- Quick Reference
- Creating Pages
- Editing Pages
- Navigation Bars
- Using Modules
- Templates
- CSS Themes

Create a Page

New page

Home

This site contains information about some resources related to Combinatorial Reconfiguration, a research area in Computer Science. Everyone can edit this site by becoming a member. (See all members of this site.)

Related Papers

- See this page.
- http://www.ecei.tohoku.ac.jp/alg/coresurvey/.

Mailing List

- https://lists.uwaterloo.ca/mailman/listinfo/reconf.

CoRe Portal

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

Talks

- See this page.
- YouTube「組合せ遷移」channel (in Japanese, available since October 2021).

Events

- See this page.

Softwares

- See this page.

Figure: Online Reconfiguration Wiki Page (`https://reconf.wikidot.com/`)

Introduction to Reconfiguration

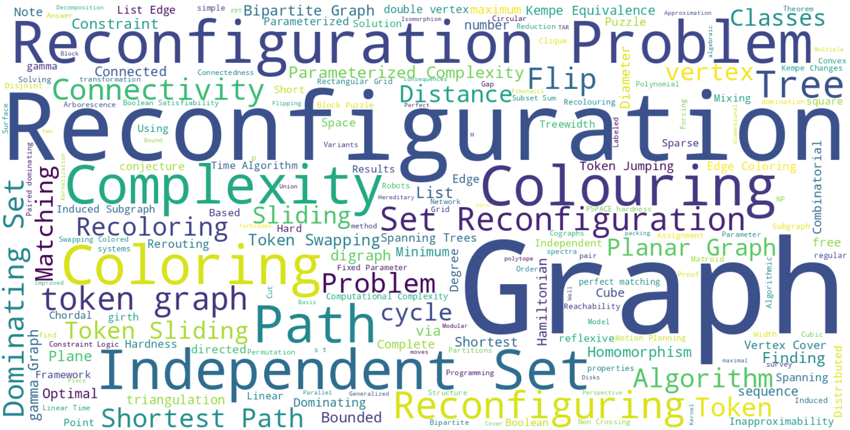
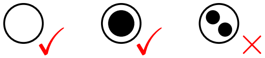


Figure: A word cloud of titles extracted from the current list of papers related to reconfiguration problems available at `https://reconf.wikidot.com/` (Accessed on October 26, 2025)

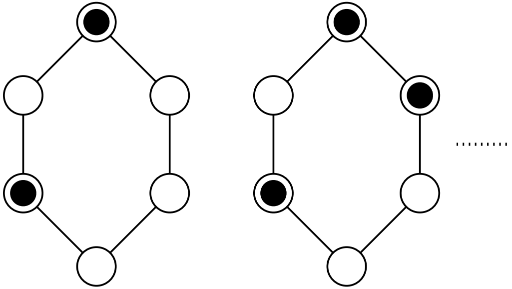
Some Examples

Token Reconfiguration

Each vertex has at most one token



Each state is a set of tokens satisfying certain property



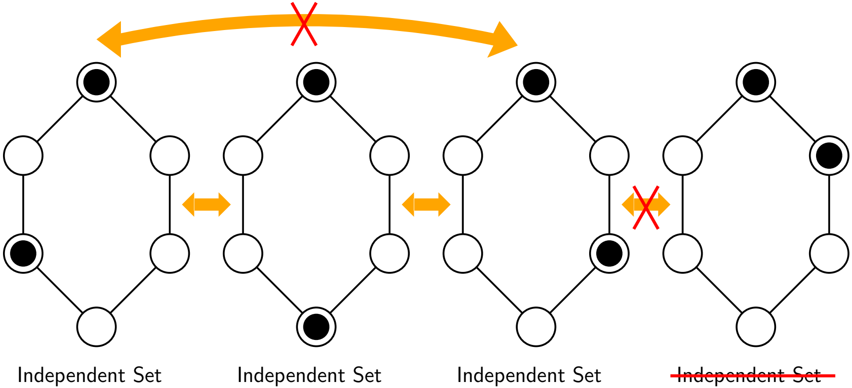
Independent Set   Dominating Set

Some Examples

[Hearn and Demaine 2005]

TOKEN RECONFIGURATION (Example: INDEPENDENT SET RECONFIGURATION)

Token Sliding (TS)



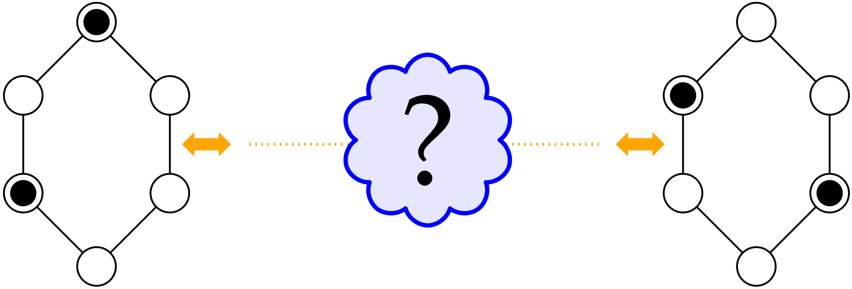
Some Examples

[Hearn and Demaine 2005]

Token Reconfiguration (Example: Independent Set Reconfiguration)

Token Sliding (TS)

PSPACE-complete on general graphs



Independent Set

Independent Set

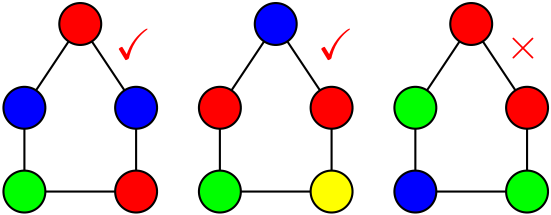
Some Examples

[Cereceda, van den Heuvel, and Johnson 2008]

VERTEX-COLORING RECONFIGURATION

Each vertex is colored by one of the k given colors

Each state is a k-coloring of all vertices such that
no two adjacent vertices share the same color



3-coloring $\quad$ 4-coloring $\quad$ ~~3-coloring~~

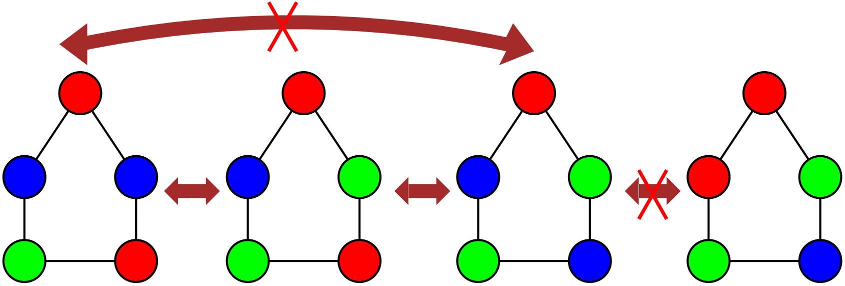
Some Examples

[Cereceda, van den Heuvel, and Johnson 2008]

Vertex-Coloring Reconfiguration

Recoloring using $\leq k$ colors

Example: $k = 3$



3-coloring   3-coloring   3-coloring   ~~3-coloring~~

Some Examples

[Cereceda, van den Heuvel, and Johnson 2008]

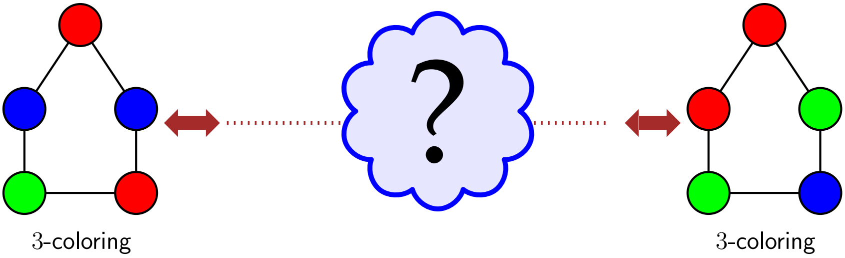
VERTEX-COLORING RECONFIGURATION

Recoloring using $\leq k$ colors

Example: $k = 3$

PSPACE-complete on general graphs for $k \geq 4$

P on general graphs for $k \leq 3$



Some Examples

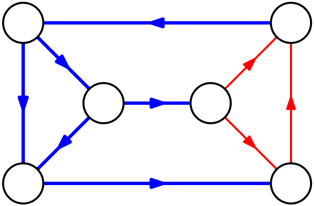
[Hearn and Demaine 2005]

Nondeterministic Constraint Logic (NCL)

Each state is a weighted, oriented graph

weight 2 weight 1

Total incoming weight at each vertex ≥ 2



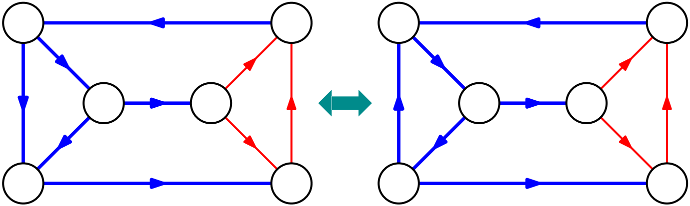
Some Examples

[Hearn and Demaine 2005]

NONDETERMINISTIC CONSTRAINT LOGIC (NCL)

Reversing edge direction

Total incoming weight at each vertex ≥ 2



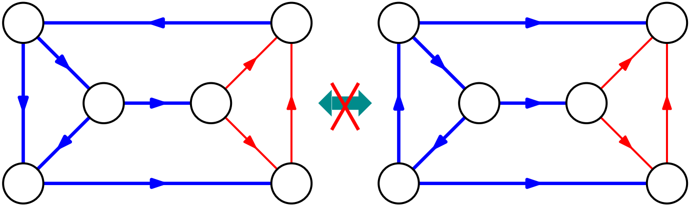
Some Examples

[Hearn and Demaine 2005]

NONDETERMINISTIC CONSTRAINT LOGIC (NCL)

Reversing edge direction

Total incoming weight at each vertex ≥ 2



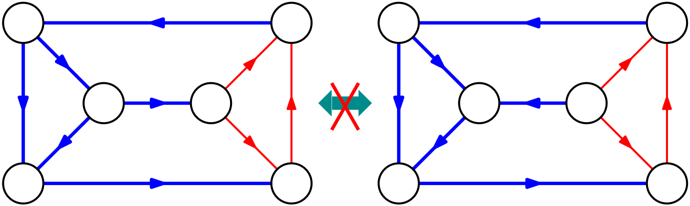
Some Examples

[Hearn and Demaine 2005]

NONDETERMINISTIC CONSTRAINT LOGIC (NCL)

Reversing edge direction

Total incoming weight at each vertex ≥ 2

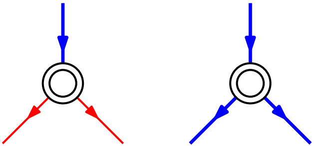


Some Examples

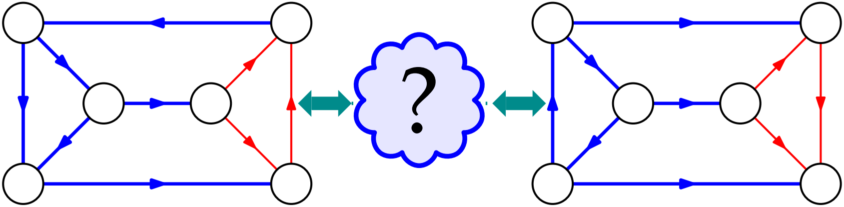
[van der Zanden 2015]

NONDETERMINISTIC CONSTRAINT LOGIC (NCL)

PSPACE-complete on
- planar graphs
- having only two types of vertices
- max degree 3
- bounded bandwidth



AND vertex   OR vertex

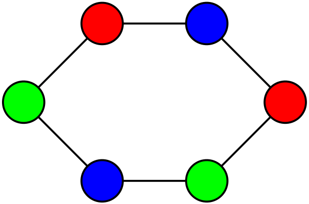


Distance Recoloring

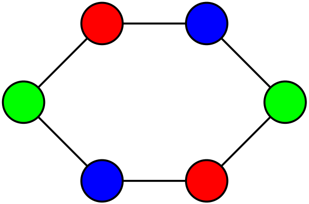
Distance constraints for *vertex colorings*

	k-Coloring	(d,k)-Coloring
Distance between two vertices having the same color	≥ 2	$\geq d+1$

- (d,k)-coloring was first studied in [F. Kramer and H. Kramer 1969]
- Has applications in *frequency assignment problem* (or radio channel assignment) [F. Kramer and H. Kramer 2008]



3-coloring
~~$(2,3)$-coloring~~

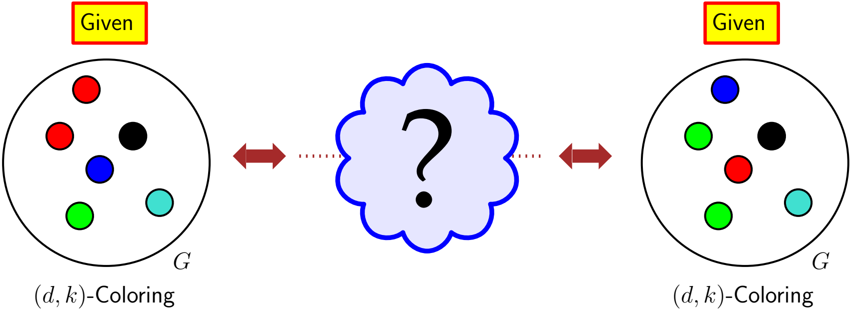


3-coloring
$(2,3)$-coloring

Distance Recoloring

(d, k)-Coloring Reconfiguration ((d, k)-CR)

Reconfiguration Rule: Recoloring a vertex



- The case $d = 1$ (k-Coloring Reconfiguration (k-CR)) has been well-studied [Mynhardt and Nasserasr 2019]; [Heuvel 2013]
- We focus on the case $d \geq 2$

Distance Recoloring

Graph	k-CR $(= (1,k)$-CR)	(d,k)-CR $(d \geq 2)$
general	PSPACE-C $(k \geq 3)$ [Cereceda, Heuvel, and Johnson 2011]	PSPACE-C $(k = \Omega(d^2))$
planar	PSPACE-C $(4 \leq k \leq 6)$ P $(k \geq 7)$ [Bonsma and Cereceda 2009]	
bipartite	PSPACE-C $(k \geq 4)$ [Bonsma and Cereceda 2009]	
planar ∩ bipartite	PSPACE-C $(k = 4)$ P $(k \geq 5)$ [Bonsma and Cereceda 2009]	
2-degenerate	P [Hatanaka, Ito, and Zhou 2019]	
planar ∩ bipartite ∩ 2-degenerate	P (⊆ 2-degenerate)	
path	P (⊆ planar ∩ bipartite ∩ 2-degenerate)	P $(k \geq d+1)$
split	P [Hatanaka, Ito, and Zhou 2019]	PSPACE-C $(d = 2$, large $k)$ P $(d \geq 3)$

Table: Our Results for $d \geq 2$. We provide the status for $d = 1$ for comparison. Here PSPACE-C stands for PSPACE-complete [Banerjee, Engels, and **Hoang** 2024]

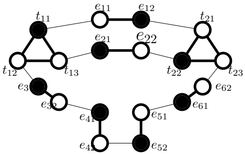
Distance Recoloring

Theorem (Banerjee, Engels, and Hoang 2024)

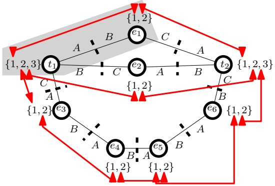
(d,k)-CR is PSPACE*-complete for $d \geq 2$ and $k = \Omega(d^2)$ on graphs which are planar, bipartite, and 2-degenerate.*

Proof Sketch.

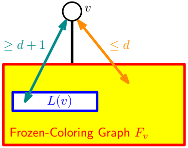
Restricted PSPACE-Complete Variant of ISR under TS (Restricted Sliding Tokens)



List $(d, \Omega(d))$-Coloring Reconfiguration



$(d, \Omega(d^2))$-Coloring Reconfiguration



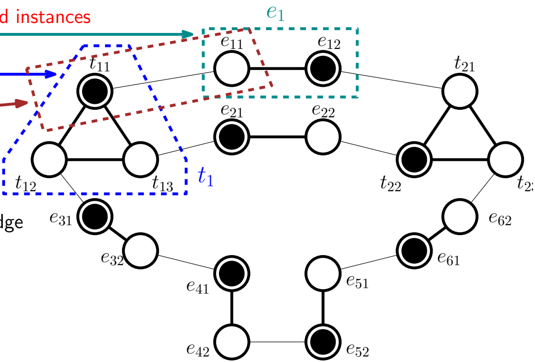
Distance Recoloring

First Phase

RESTRICTED SLIDING TOKENS $\Rightarrow$ LIST $(d, \Omega(d))$-COLORING RECONFIGURATION

RESTRICTED SLIDING TOKENS

- PSPACE-complete on very restricted instances
  - Three types of gadgets
    - Token triangle
    - Token edge
    - Link edge
  - Planar
  - Max degree 3, min degree 2
  - No two token triangles are directly joined by a link edge
  - Each token triangle/token edge has exactly one token



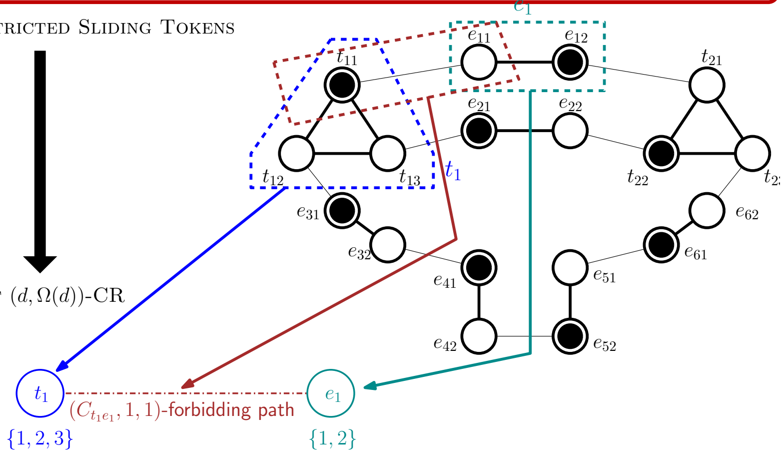
Distance Recoloring

First Phase

Restricted Sliding Tokens $\Rightarrow$ List $(d, \Omega(d))$-Coloring Reconfiguration

Restricted Sliding Tokens

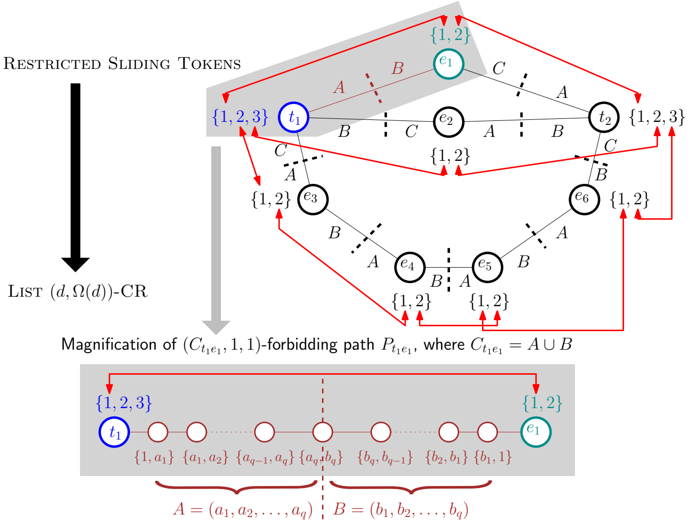
List $(d, \Omega(d))$-CR



Distance Recoloring

First Phase

Restricted Sliding Tokens $\Rightarrow$ List $(d, \Omega(d))$-Coloring Reconfiguration



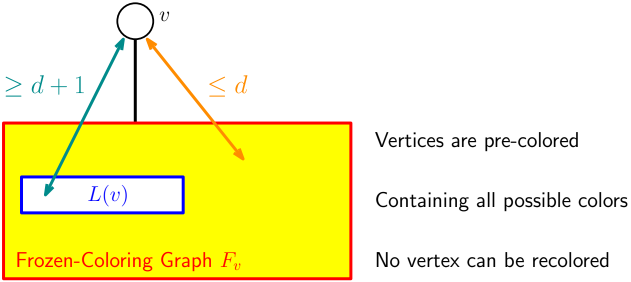
Distance Recoloring

Second Phase

LIST $(d, \Omega(d))$-COLORING RECONFIGURATION $\Rightarrow$ $(d, \Omega(d^2))$-COLORING RECONFIGURATION

Key Ideas

1. List Coloring $\equiv$ Coloring with constraints on which colors can be used for each vertex
2. *Frozen-Coloring Graphs:* Pre-colored graphs where *no vertex can be recolored*



Distance Recoloring

Second Phase

List $(d, \Omega(d))$-Coloring Reconfiguration $\Rightarrow$ $(d, \Omega(d^2))$-Coloring Reconfiguration

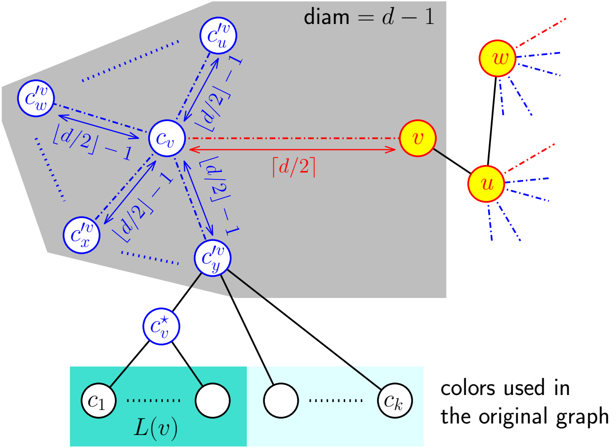


Figure: Construction of the frozen-coloring subgraph F_v for a vertex v. The colors used in this figure are just for illustration of paths.

Distance Recoloring

Theorem (Banerjee, Engels, and Hoang 2024)

On split graphs,

	(d,k)-*Coloring Reconfiguration*
$d = 1$	$\mathtt{P}$ *for any* k *[Hatanaka, Ito, and Zhou 2019]*
$d = 2$	$\mathtt{PSPACE}$-*complete for large* k
$d \geq 3$	$\mathtt{P}$ *for any* k

Proof Sketch.

$d \geq 3$ Trivial. (Any connected split graph has diameter $\leq 3 \Rightarrow$ Reconfiguration is easy!)

$d = 2$ Reduction from the problem for $d = 1$ and $k \geq 4$ on general graphs (which is known to be $\mathtt{PSPACE}$-complete [Bonsma and Cereceda 2009])

$\square$

Concluding Remarks

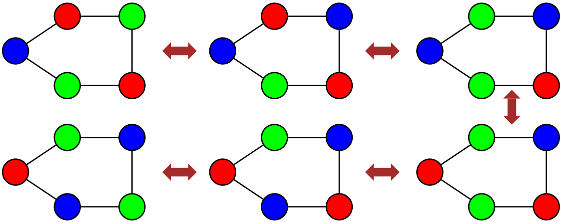
Take-Home Messages

1. *Reconfiguration* studies the "solution space" of a problem
   - Moving from one solution to another *without violating feasibility*
2. Under certain *distance constraints*,
   - *Reconfiguration problems* can be *hard* for very restricted graph classes
   - *Problems on graphs whose diameters are bounded by some constant c* (e.g., split graphs) are interesting when *restricted to distances close to c*
3. *Nondeterministic Constraint Logic* is a powerful tool for *hardness reductions*

Open Problems

The complexities of the following problem remain *open* for *trees*:

- (d, k)-CR $(d \geq 2)$



Partially supported by

Came to my talk, you did.
Thank you, I must!

References I

Banerjee, Niranka, Christian Engels, and **Duc A. Hoang** (2024). "Distance Recoloring". In: *arXiv preprint*. arXiv: 2402.12705.

Hatanaka, Tatsuhiko, Takehiro Ito, and Xiao Zhou (2019). "The Coloring Reconfiguration Problem on Specific Graph Classes". In: *IEICE Transactions on Information and Systems* E102.D.3, pp. 423–429. DOI: 10.1587/transinf.2018FCP0005.

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.

van der Zanden, Tom C. (2015). "Parameterized complexity of graph constraint logic". In: *Proceedings of IPEC 2015*. Vol. 43. LIPIcs. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, pp. 282–293. DOI: 10.4230/LIPIcs.IPEC.2015.282.

References II

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.

Cereceda, Luis, Jan van den Heuvel, and Matthew Johnson (2011). “Finding Paths Between 3-Colorings”. In: *Journal of Graph Theory* 67.1, pp. 69–82. DOI: 10.1002/jgt.20514.

Bonsma, Paul S. and Luis Cereceda (2009). “Finding Paths Between Graph Colourings: PSPACE-Completeness and Superpolynomial Distances”. In: *Theoretical Computer Science* 410.50, pp. 5215–5226. DOI: 10.1016/j.tcs.2009.08.023.

Cereceda, Luis, Jan van den Heuvel, and Matthew Johnson (2008). “Connectedness of the Graph of Vertex-Colourings”. In: *Discrete Mathematics* 308.5-6, pp. 913–919. DOI: 10.1016/j.disc.2007.07.028.

Kramer, Florica and Horst Kramer (2008). “A Survey on the Distance-Colouring of Graphs”. In: *Discrete mathematics* 308.2-3, pp. 422–426. DOI: 10.1016/j.disc.2006.11.059.

References III

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.

Kramer, Florica and Horst Kramer (1969). "Ein Färbungsproblem der Knotenpunkte eines Graphen bezüglich der Distanz p". In: *Rev. Roumaine Math. Pures Appl* 14.2, pp. 1031–1038.