Sliding tokens on block graphs

Duc A. Hoang [1] Eli Fox-Epstein [2] Ryuhei Uehara [1]

[1] JAIST, Japan

[2] Brown University, USA

Outline

1. Reconfiguration problems and moving tokens on graphs
2. Sliding tokens on block graphs in polynomial time
3. Open questions

Outline

1. Reconfiguration problems and moving tokens on graphs
2. Sliding tokens on block graphs in polynomial time
3. Open questions

General framework: Reconfiguration Problems

- INSTANCE:
  1. Collection of configurations.
  2. Allowed transformation rule(s).
- QUESTION: Decide if configuration A can be transformed to configuration B using the given rule(s), while maintaining a configuration throughout.

- INSTANCE:
  1. Collection of configurations. Labelled tokens on a 4×4 grid.
  2. Allowed transformation rule(s).
- QUESTION: Decide if configuration A can be transformed to configuration B using the given rule(s), while maintaining a configuration throughout.

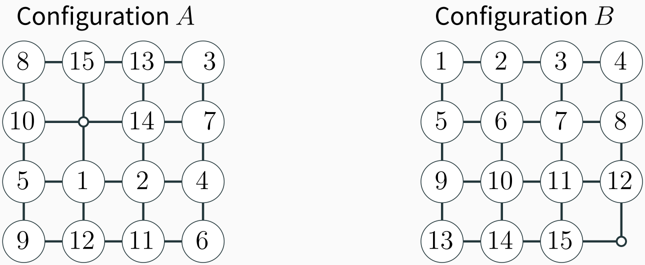


Figure 1: The 15-puzzles.

General framework: Reconfiguration Problems

- INSTANCE:
  1. Collection of configurations. Labelled tokens on a 4×4 grid.
  2. Allowed transformation rule(s). Token Sliding (TS).
- QUESTION: Decide if configuration A can be transformed to configuration B using the given rule(s), while maintaining a configuration throughout.

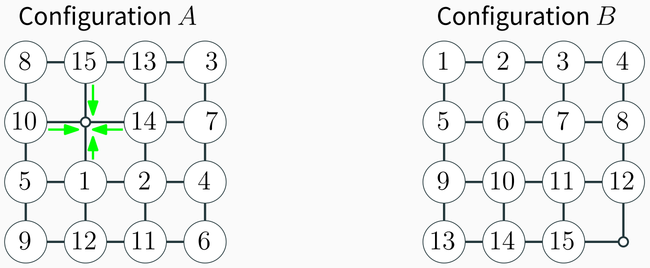


Figure 1: The 15-puzzles.

General framework: Reconfiguration Problems

- INSTANCE:
  1. Collection of configurations. Labelled tokens on a 4×4 grid.
  2. Allowed transformation rule(s). Token Sliding (TS).
- QUESTION: Decide if configuration A can be transformed to configuration B using the TS rule, while maintaining a configuration throughout.

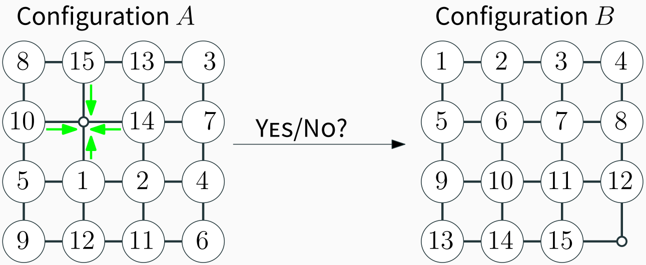
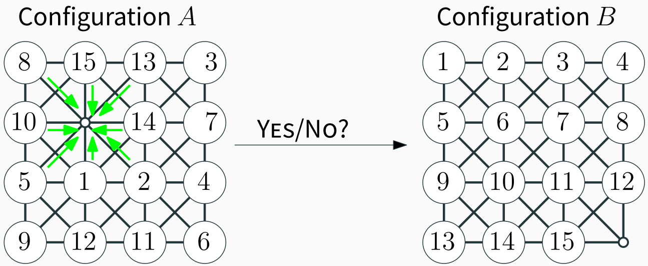


Figure 1: The 15-puzzles.

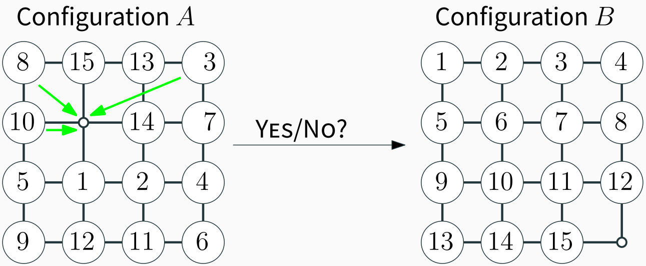
Generalization of 15-puzzles

- **Graphs:** grid, trees, block, planar, perfect, etc.
- **Rules:** Token Sliding, Token Jumping, Token Swapping, etc.
- **Labels:** distinct labels for all tokens, some tokens can be of the same label, no label, etc.
- **Restrictions:** no restriction, independent set, dominating set, etc.



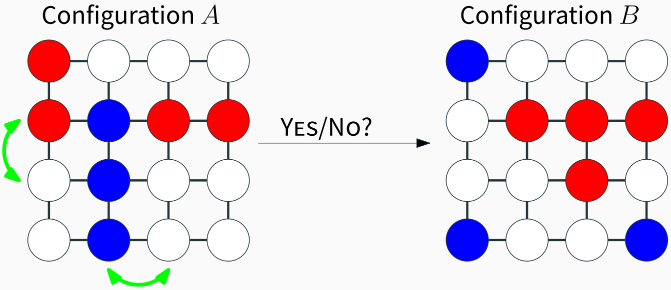
Generalization of 15-puzzles

- **Graphs:** grid, trees, block, planar, perfect, etc.
- **Rules:** Token Sliding, Token Jumping, Token Swapping, etc.
- **Labels:** distinct labels for all tokens, some tokens can be of the same label, no label, etc.
- **Restrictions:** no restriction, independent set, dominating set, etc.



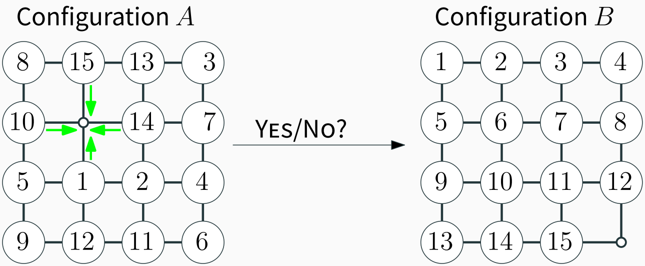
Generalization of 15-puzzles

- **Graphs:** grid, trees, block, planar, perfect, etc.
- **Rules:** Token Sliding, Token Jumping, Token Swapping, etc.
- **Labels:** distinct labels for all tokens, some tokens can be of the same label, no label, etc.
- **Restrictions:** no restriction, independent set, dominating set, etc.



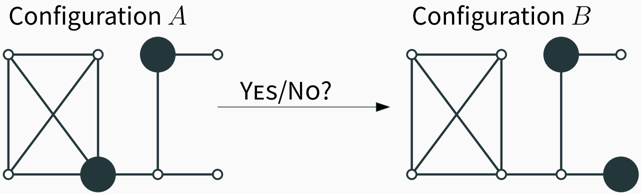
Generalization of 15-puzzles

- **Graphs:** grid, trees, block, planar, perfect, etc.
- **Rules:** Token Sliding, Token Jumping, Token Swapping, etc.
- **Labels:** distinct labels for all tokens, some tokens can be of the same label, no label, etc.
- **Restrictions:** no restriction, independent set, dominating set, etc.



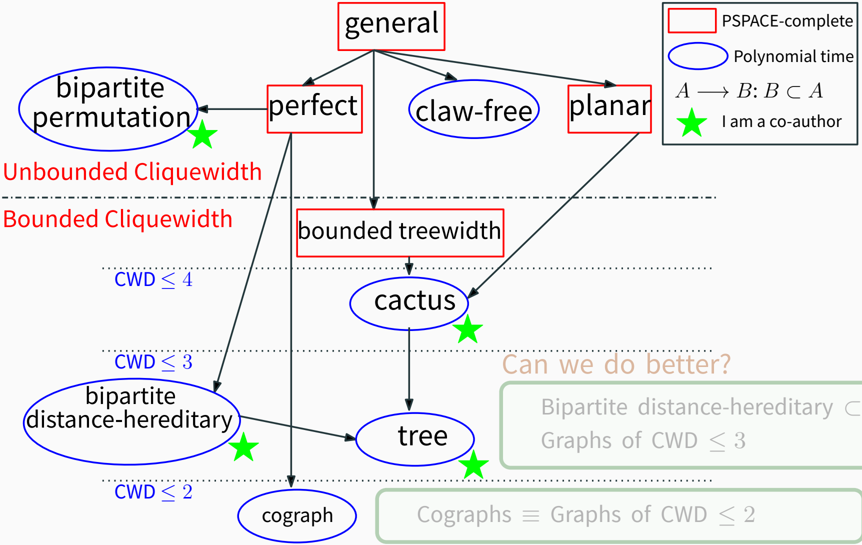
Our Problem: Sliding Token for block graphs

- **Graphs:** grid, trees, block, planar, perfect, etc.
- **Rules:** Token Sliding, Token Jumping, Token Swapping, etc.
- **Labels:** distinct labels for all tokens, some tokens can be of the same label, no label, etc.
- **Restrictions:** no restriction, independent set, dominating set, etc.



Block graphs: Every block (i.e., maximal 2-connected subgraph) is a clique.

SLIDING TOKEN - **Complexity Status**

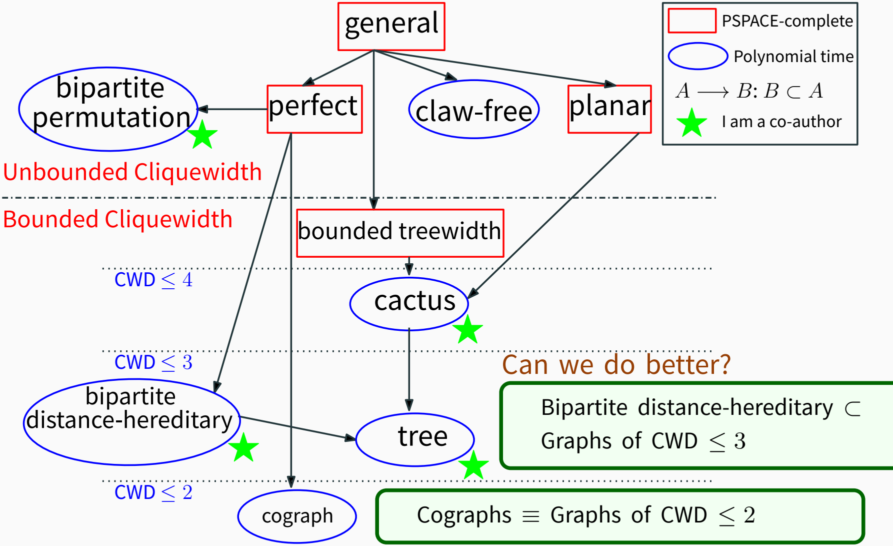


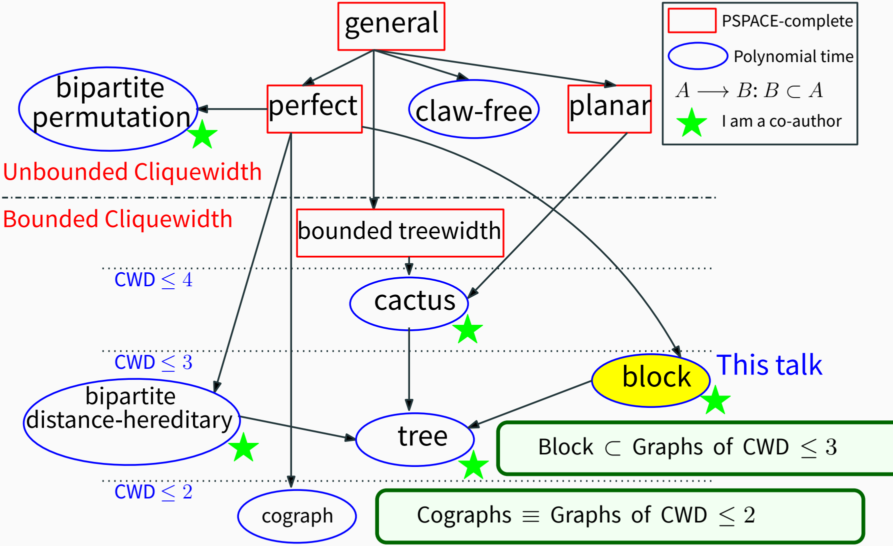
Can we do better?

Bipartite distance-hereditary $\subset$ Graphs of CWD ≤ 3

Cographs $\equiv$ Graphs of CWD ≤ 2

Sliding Token - **Complexity Status**



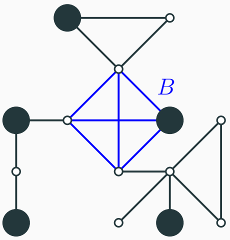

general
bipartite permutation
perfect
claw-free
planar
PSPACE-complete
Polynomial time
$A \longrightarrow B$: $B \subset A$
I am a co-author
Unbounded Cliquewidth
Bounded Cliquewidth
bounded treewidth
CWD ≤ 4
cactus
CWD ≤ 3
block
This talk
bipartite distance-hereditary
tree
Block $\subset$ Graphs of CWD ≤ 3
CWD ≤ 2
cograph
Cographs $\equiv$ Graphs of CWD ≤ 2

Outline

1. Reconfiguration problems and moving tokens on graphs
2. Sliding tokens on block graphs in polynomial time
3. Open questions

Key structure: (G, I)-confined clique

(G, I)-confined clique: The "inside" token cannot be slid "out."



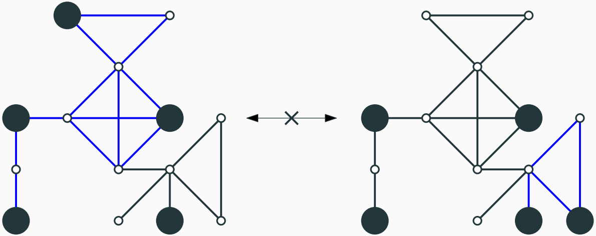
Lemma 1: One can find all (G, I)-confined cliques in time $O(m^2)$, where $m = |E(G)|$.

Lemma 2: For two independent sets I, J, if the set of confined cliques for I and J are different, then I cannot be reconfigured to J (and vice versa).

Lemma 3: If there are no confined cliques for both I and J, then I can be reconfigured to J iff $|I| = |J|$.

Key structure: (G, I)-confined clique

(G, I)**-confined clique:** The "inside" token cannot be slid "out."



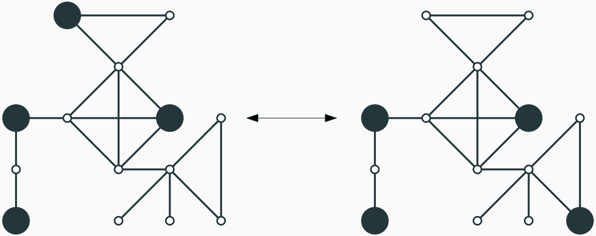
Lemma 1: One can find all (G, I)-confined cliques in time $O(m^2)$, where $m = |E(G)|$.

Lemma 2: For two independent sets I, J, if the set of confined cliques for I and J are different, then I cannot be reconfigured to J (and vice versa).

Lemma 3: If there are no confined cliques for both I and J, then I can be reconfigured to J iff $|I| = |J|$.

Key structure: (G, I)-confined clique

(G, I)-confined clique: The "inside" token cannot be slid "out."



Lemma 1: One can find all (G, I)-confined cliques in time $O(m^2)$, where $m = |E(G)|$.

Lemma 2: For two independent sets I, J, if the set of confined cliques for I and J are different, then I cannot be reconfigured to J (and vice versa).

Lemma 3: If there are no confined cliques for both I and J, then I can be reconfigured to J iff $|I| = |J|$.

Our Algorithm

Given an instance (G, I, J) of Sliding Token, where I, J are two independent sets of a block graph G.

1. Find all confined cliques for both I and J. If the set of confined cliques for I and J are different, return NO.
   Otherwise, remove all confined cliques for I and J (they are the same). Let G' be the resulting graph.
2. For each component F of G', if $|I \cap F| \neq |J \cap F|$, return NO.
   Otherwise, return YES.

Running time: $O(m^2 + n)$, where $m = |E(G)|$ and $n = |V(G)|$.

Outline

1. Reconfiguration problems and moving tokens on graphs
2. Sliding tokens on block graphs in polynomial time
3. Open questions

Open questions

- Whether one can solve SLIDING TOKEN for block graphs in linear time.
- When considering graphs of cliquewidth at most 3, distance-hereditary graphs is more general than block graphs. SLIDING TOKEN remains open for distance-hereditary graphs.

SLIDING TOKEN is also polynomial-time solvable for bipartite distance-hereditary graphs [Fox-Epstein, Hoang, Otachi, and Uehara 2015] and cographs [Kamiński, Medvedev, and Milanič 2012].

Open questions

- Whether one can solve SLIDING TOKEN for block graphs in linear time.
- When considering graphs of cliquewidth at most 3, distance-hereditary graphs is more general than block graphs. SLIDING TOKEN remains open for distance-hereditary graphs.

SLIDING TOKEN is also polynomial-time solvable for bipartite distance-hereditary graphs [Fox-Epstein, Hoang, Otachi, and Uehara 2015] and cographs [Kamiński, Medvedev, and Milanič 2012].

Appendix

- Recent results on studying ISReconf
- Cliquewidth

Recent results on studying ISRECONF

Graph	Rule(s)	Complexity	Paper(s)
planar	TS, TJ, TAR	PSPACE-complete	Hearn and Demaine 2005
general	TS, TJ, TAR	PSPACE-complete	Ito et al. 2011
line	TJ, TAR	P	
perfect	TS, TJ, TAR	PSPACE-complete	Kamiński, Medvedev, and Milanič 2012
even-hole-free	TJ, TAR	P	
cograph (P_4-free)	TS	P	
cograph (P_4-free)	TJ, TAR	P	Bonsma 2016
bounded bandwidth	TS, TJ, TAR	PSPACE-complete	Wrochna 2014
claw-free	TS, TJ	P	Bonsma, Kamiński, and Wrochna 2014
tree	TS	P	Demaine et al. 2015
bipartite permutation	TS	P	Fox-Epstein, Hoang, Otachi, and Uehara 2015
bipartite distance-hereditary	TS	P	
cactus	TS	P	Hoang and Uehara 2016
block	TS	P	Hoang, Fox-Epstein, and Uehara 2017

Table 1: Recent results on studying ISRECONF under Token Sliding (TS), Token Jumping (TJ), and Token Addition and Removal (TAR).

Cliquewidth I

The *cliquewidth* of a graph G, denoted by $cwd(G)$, is the minimum number of labels needed to construct G using the following four operations:

1. Creation of a new vertex v with label i (denoted by $i(v)$).
2. Disjoint union of two labelled graphs G and H (denoted by $G \oplus H$).
3. Joining by an edge each vertex with label i to each vertex with label j ($i \neq j$, denoted by $\eta_{i,j}$).
4. Renaming label i to j (denoted by $\rho_{i \to j}$)

Cliquewidth II

Every graph can be defined by an algebraic expression using these four operations. For instance, a chordless path on five consecutive vertices a, b, c, d, e can be defined as follows:

$$\eta_{2,3}(\rho_{3\to 1}(\eta_{2,3}(\rho_{2\to 1}(\eta_{2,3}(\eta_{1,2}(1(a)\oplus 2(b))\oplus 3(c)))\oplus 2(d)))\oplus 3(e))$$

Such an expression is called a k-*expression* if it uses at most k different labels. Thus the cliquewidth of G is the minimum k for which there exists a k-expression defining G. For instance, from the above example we conclude that $cwd(P_5) \leq 3$.

Cliquewidth of some well-known graphs

- Cographs (graphs having no P_4 as induced subgraph) are exactly the graphs of cliquewidth at most 2.
- A complete graph K_n is of cliquewidth at most 2.
- A tree (and hence a forest) is of cliquewidth at most 3.

Theorem (González-Ruiz, Marcial-Romero, and Hernández-Servín 2016)

The cliquewidth of a cactus is at most 4.

Theorem (Golumbic and Rotics 2000)

The cliquewidth of a distance-hereditary graph is at most 3. Consequently, any subclass of distance-hereditary graphs is of cliquewidth at most 3.

Bibliography I

Bonsma, Paul (2016). "Independent Set Reconfiguration in Cographs and their Generalizations". In: *Journal of Graph Theory* 83.2, pp. 164–195. DOI: 10.1002/jgt.21992.

Bonsma, Paul, Marcin Kamiński, and Marcin Wrochna (2014). "Reconfiguring Independent Sets in Claw-Free Graphs". In: *Proceedings of SWAT 2014*. Ed. by R. Ravi and IngeLi Gørtz. Vol. 8503. LNCS. 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 (2015). "Linear-time algorithm for sliding tokens on trees". In: *Theoretical Computer Science* 600, pp. 132–142. DOI: 10.1016/j.tcs.2015.07.037.

Bibliography II

Fox-Epstein, Eli, Duc A. Hoang, Yota Otachi, and Ryuhei Uehara (2015). "Sliding Token on Bipartite Permutation Graphs". In: *Proceedings of ISAAC 2015*. Ed. by Khaled Elbassioni and Kazuhisa Makino. Vol. 9472. LNCS. Springer, pp. 237–247. DOI: 10.1007/978-3-662-48971-0_21.

Golumbic, Martin Charles and Udi Rotics (2000). "On the clique-width of some perfect graph classes". In: *International Journal of Foundations of Computer Science* 11.03, pp. 423–443. DOI: 10.1142/S0129054100000260.

González-Ruiz, J. Leonardo, J. Raymundo Marcial-Romero, and J.A. Hernández-Servín (2016). "Computing the Clique-width of Cactus Graphs". In: *Electronic Notes in Theoretical Computer Science* 328, pp. 47–57. DOI: 10.1016/j.entcs.2016.11.005.

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, pp. 72–96. DOI: 10.1016/j.tcs.2005.05.008.

Bibliography III

Hoang, Duc A., Eli Fox-Epstein, and Ryuhei Uehara (2017). "Sliding token on block graphs". In: *Proceedings of WALCOM 2017*. Ed. by Sheung-Hung Poon, Md. Saidur Rahman, and Hsu-Chun Yen. Vol. 10167. LNCS. Springer, pp. 460–471. DOI: 10.1007/978-3-319-53925-6_36.

Hoang, Duc A. and Ryuhei Uehara (2016). "Sliding Tokens on a Cactus". In: *Proceedings of ISAAC 2016*. Ed. by Seok-Hee Hong. Vol. 64. LIPIcs. 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, 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.

Bibliography IV

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