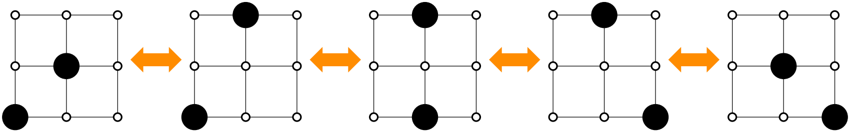
Independent Set Reconfiguration and Related Problems for Some Restricted Graphs



by Hoang Anh Duc
PhD Student (1520016, D3), Uehara Lab, JAIST
`hoanganhduc@jaist.ac.jp`
May 07, 2018

Supervisor: Ryuhei Uehara

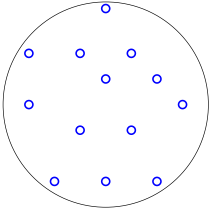
Outline

1. Background and Motivation
   - Reconfiguration Problems
   - Reconfigurability of Independent Set
2. Our Results
3. Future Works
4. Publications

Outline

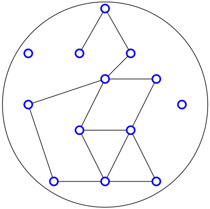
1. Background and Motivation
   - Reconfiguration Problems
   - Reconfigurability of Independent Set
2. Our Results
3. Future Works
4. Publications

Reconfiguration Problems ≡ The study of reconfiguration graphs



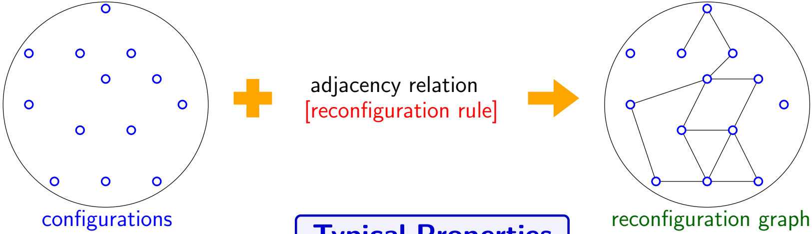
configurations

\+ adjacency relation [reconfiguration rule] →



reconfiguration graph

Reconfiguration Problems ≡ The study of reconfiguration graphs

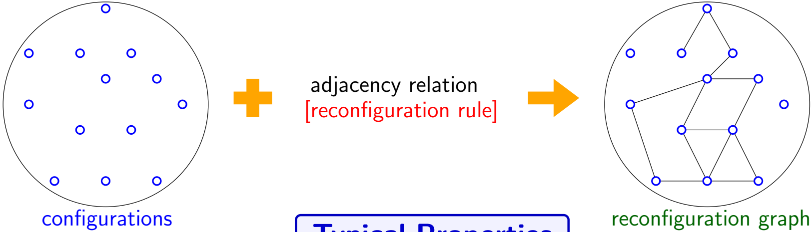


Typical Properties

- adjacency is polynomial testable
- reconfiguration graph is huge

Reconfiguration Problems

Reconfiguration Problems ≡ The study of reconfiguration graphs



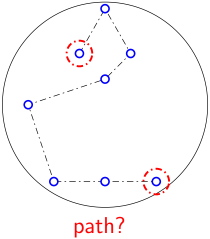
Typical Properties

- adjacency is polynomial testable
- reconfiguration graph is huge

Reconfiguration v.s. Solution Space

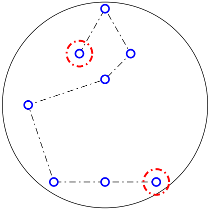
- configurations ≡ feasible solutions of a problem
- reconfiguration rule ≡ small change that preserves the "feasibility" of a solution

Fundamental Questions

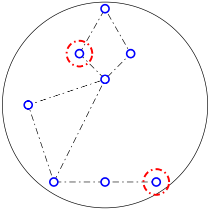


REACHABILITY

Fundamental Questions



path?

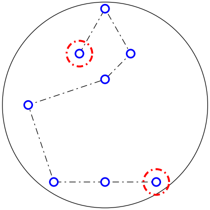


find shortest path?

REACHABILITY

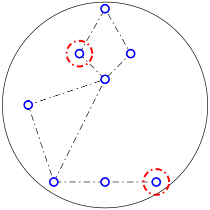
SHORTEST RECONF.

Fundamental Questions



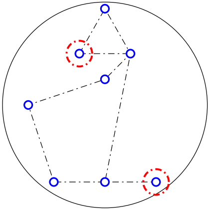
path?

REACHABILITY



find shortest path?

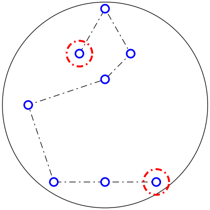
SHORTEST RECONF.



path of length $\leq \ell$?

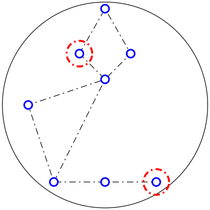
BOUNDED RECONF.

Fundamental Questions



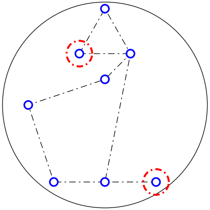
path?

REACHABILITY



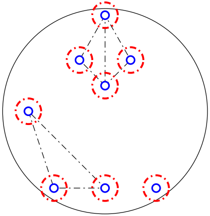
find shortest path?

SHORTEST RECONF.



path of length $\leq \ell$?

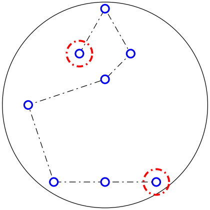
BOUNDED RECONF.



connected?

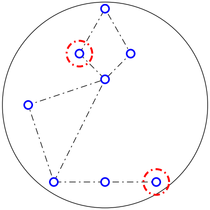
CONNECTIVITY

Fundamental Questions



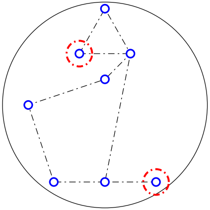
path?

REACHABILITY



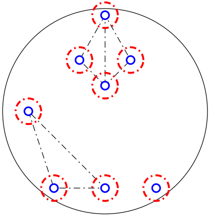
find shortest path?

SHORTEST RECONF.



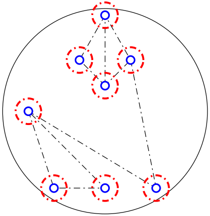
path of length $\leq \ell$?

BOUNDED RECONF.



connected?

CONNECTIVITY



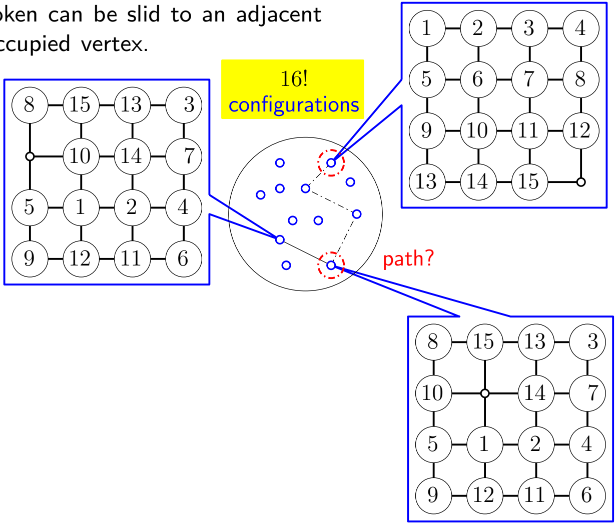
diameter?

DIAMETER

Example: 15-PUZZLE

Configuration Placement of 15 tokens labeled $1, 2, \ldots, 15$, on a 4×4 grid.

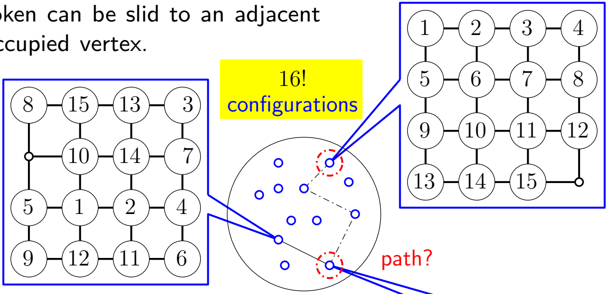
Adjacency [Token Sliding] A token can be slid to an adjacent unoccupied vertex.

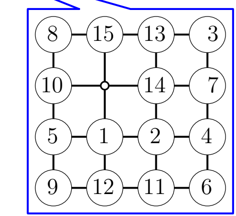


Example: 15-PUZZLE

Configuration Placement of 15 tokens labeled $1, 2, \ldots, 15$, on a 4×4 grid.

Adjacency [Token Sliding] A token can be slid to an adjacent unoccupied vertex.





Problem	Complexity	Reference
REACH.	P	[Johnson and Story 1879]
CONN.	P	[Wilson 1974]
SHORTEST RECONF.	NP-complete	[Ratner and Warmuth 1990]
BOUNDED RECONF.	NP-complete	[Goldreich 2011]
DIAM.	P	[Kornhauser et al. 1984]

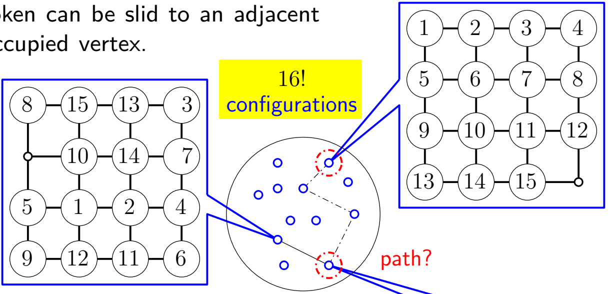
Example: 15-PUZZLE

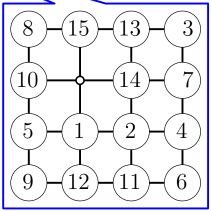
Configuration Placement of 15 tokens labeled $1, 2, \ldots, 15$, on a 4×4 grid.

Adjacency [Token Sliding] A token can be slid to an adjacent unoccupied vertex.

Generalization
[v.d. Heuvel, CoRe'17]

- $(n^2 - 1)$-PUZZLE
- SLIDING-BLOCK
- PEBBLE MOTION
- ROBOT MOTION
- and so on





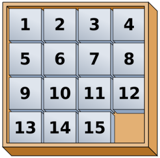
Problem	Complexity	Reference
REACH.	P	[Johnson and Story 1879]
CONN.	P	[Wilson 1974]
SHORTEST RECONF.	NP-complete	[Ratner and Warmuth 1990]
BOUNDED RECONF.	NP-complete	[Goldreich 2011]
DIAM.	P	[Kornhauser et al. 1984]

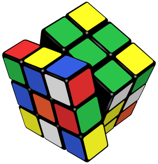
Motivation

Theory Understand the solution space of a problem.

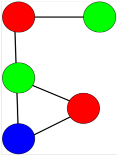
Application Model real-world situations involving movement and change.

- Games & Puzzles.

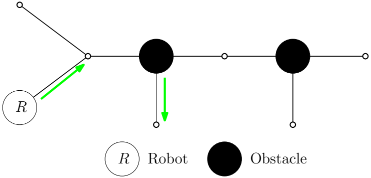




- Frequency Re-Assignment.



- Robot Motion.



Reconfiguration Problems

An Incomplete History [Nishimura, CanaDAM'17]

Before 2011 Lots of work not called reconfiguration

2011 Reconfiguration framework [Ito et al. 2011]
- ◦ Study REACHABILITY questions
- ◦ Several classic NP-complete problems have PSPACE-complete reconfiguration variants
- ◦ Several problems in P whose reconfiguration variants are also in P

Since 2011 Lots of work called reconfiguration (and also lots of work not called reconfiguration)

Reconfiguration Problems

An Incomplete History [Nishimura, CanaDAM'17]

Before 2011 Lots of work not called reconfiguration

2011 Reconfiguration framework [Ito et al. 2011]

◦ Study REACHABILITY questions

◦ Several classic NP-complete problems have PSPACE-complete reconfiguration variants

◦ Several problems in P whose reconfiguration variants are also in P

Since 2011 Lots of work called reconfiguration (and also lots of work not called reconfiguration)

Surveys on Reconfiguration:

Jan van den Heuvel (2013). "The complexity of change". In: *Surveys in Combinatorics 2013*. Ed. by Simon R. Blackburn et al. London Mathematical Society Lecture Note Series. Cambridge University Press, pp. 127–160. DOI: 10.1017/CBO9781139506748.005

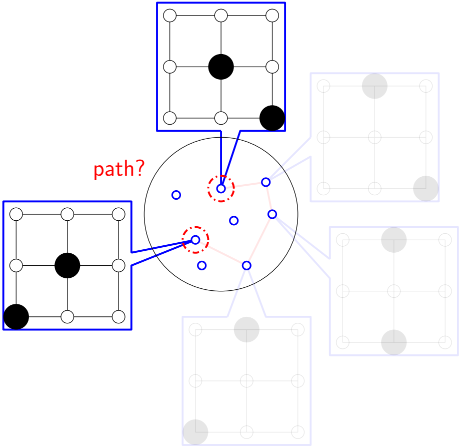
Naomi Nishimura (2018). "Introduction to Reconfiguration". In: *Algorithms* 11.4. DOI: 10.3390/a11040052

Reconfiguration Web Portal: http://www.ecei.tohoku.ac.jp/alg/core/

Reconfigurability of Independent Set

SLIDING TOKEN [Hearn and Demaine 2005]

Configuration	Independent set (viewed as a set of tokens) of a graph
Adjacency [Token Sliding]	A token can be slid to an adjacent unoccupied vertex
Problem	REACHABILITY



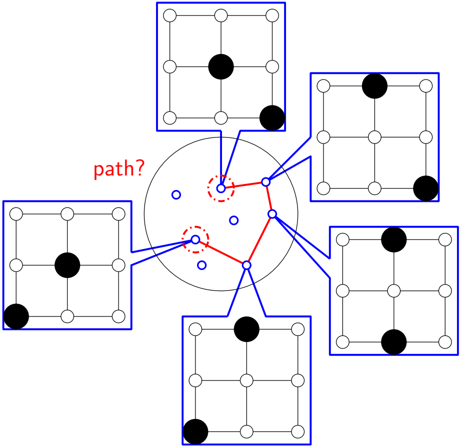
Reconfigurability of Independent Set

SLIDING TOKEN [Hearn and Demaine 2005]

Configuration Independent set (viewed as a set of tokens) of a graph

Adjacency [Token Sliding] A token can be slid to an adjacent unoccupied vertex

Problem REACHABILITY

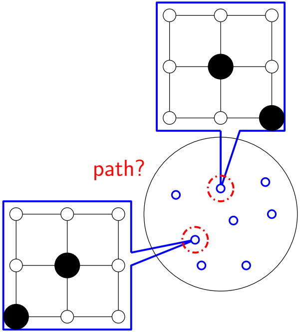


TOKEN JUMPING [Kamiński et al. 2012]

Configuration Independent set (viewed as a set of tokens) of a graph

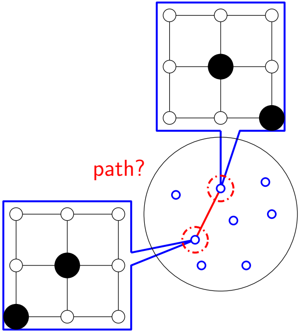
Adjacency [Token Jumping] A token can be moved to an unoccupied vertex

Problem REACHABILITY



TOKEN JUMPING [Kamiński et al. 2012]

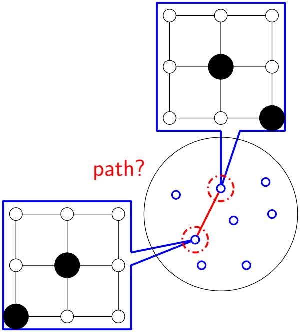
Configuration	Independent set (viewed as a set of tokens) of a graph
Adjacency [Token Jumping]	A token can be moved to an unoccupied vertex
Problem	REACHABILITY



Reconfigurability of Independent Set

Token Jumping [Kamiński et al. 2012]

Configuration Independent set (viewed as a set of tokens) of a graph
Adjacency [Token Jumping] A token can be moved to an unoccupied vertex
Problem Reachability



Multiple Token Jump [de Berg et al. 2016]

Configuration Independent set (viewed as a set of tokens) of a graph
Adjacency p tokens can be moved simultaneously to unoccupied vertices
Problem Find smallest p such that any two independent sets of equal size are connected by a path in the reconfiguration graph

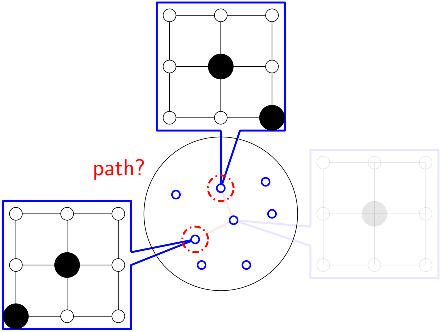
Reconfigurability of Independent Set

TOKEN ADDITION AND REMOVAL [Ito et al. 2011]

Configuration Independent set (viewed as a set of tokens) of a graph

Adjacency [Token Addition and Removal] A token can be added or removed ($\sharp$ remaining tokens $\geq k$)

Problem REACHABILITY



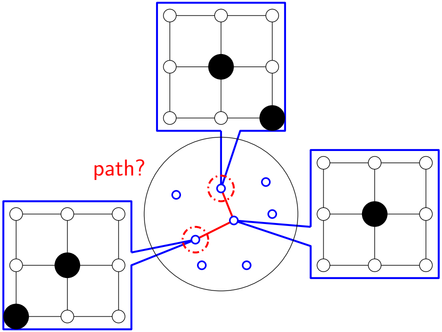
Reconfigurability of Independent Set

TOKEN ADDITION AND REMOVAL [Ito et al. 2011]

Configuration Independent set (viewed as a set of tokens) of a graph

Adjacency [Token Addition and Removal] A token can be added or removed ($\sharp$ remaining tokens $\geq k$)

Problem REACHABILITY



Reachability problems for Independent Set

Graph	Adjacency			Complexity	Reference
	TS	TAR	TJ		
planar ∩ maximum degree 3	◯	◯	◯	PSPACE-complete	[Hearn and Demaine 2005]
general line					
perfect even-hole-free cograph					
cograph					
bounded bandwidth					
claw-free (⊃ line)					
tree (⊂ even-hole-free)					
bipartite permutation bipartite distance-hereditary					
planar ∩ maximum degree 3 ∩ bounded bandwidth					
cactus					
cactus					
interval (⊂ even-hole-free)					
bipartite					

For other problems, see [Nishimura 2018, Section 4]

REACHABILITY problems for Independent Set

Graph	Adjacency			Complexity	Reference
	TS	TAR	TJ		
planar ∩ maximum degree 3	◯	◯	◯	PSPACE-complete	[Hearn and Demaine 2005]
general line ⇐ MATCHING RECONF.	◯	◯ ◯	◯ ◯	PSPACE-complete P	[Ito et al. 2011]
perfect even-hole-free cograph					
cograph					
bounded bandwidth					
claw-free (⊃ line)					
tree (⊂ even-hole-free)					
bipartite permutation bipartite distance-hereditary					
planar ∩ maximum degree 3 ∩ bounded bandwidth					
cactus					
cactus					
interval (⊂ even-hole-free)					
bipartite					

For other problems, see [Nishimura 2018, Section 4]

REACHABILITY problems for Independent Set

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	
cograph					
bounded bandwidth					
claw-free (⊃ line)					
tree (⊂ even-hole-free)		◯	◯	P	
bipartite permutation bipartite distance-hereditary					
planar ∩ maximum degree 3 ∩ bounded bandwidth					
cactus					
cactus					
interval (⊂ even-hole-free)		◯	◯	P	
bipartite					

For other problems, see [Nishimura 2018, Section 4]

Reachability problems for Independent Set

Graph	Adjacency TS	Adjacency TAR	Adjacency TJ	Complexity	Reference
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	
cograph		◯	◯	P	[Bonsma 2014]
bounded bandwidth					
claw-free (⊃ line)					
tree (⊂ even-hole-free)		◯	◯	P	
bipartite permutation bipartite distance-hereditary					
planar ∩ maximum degree 3 ∩ bounded bandwidth					
cactus					
cactus					
interval (⊂ even-hole-free)		◯	◯	P	
bipartite					

For other problems, see [Nishimura 2018, Section 4]

Reachability problems for Independent Set

Graph	Adjacency TS	Adjacency TAR	Adjacency TJ	Complexity	Reference
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	
cograph		○	○	P	[Bonsma 2014]
bounded bandwidth	○	○	○	PSPACE-complete	[Wrochna 2014]
claw-free (⊃ line)					
tree (⊂ even-hole-free)		○	○	P	
bipartite permutation bipartite distance-hereditary					
planar ∩ maximum degree 3 ∩ bounded bandwidth					
cactus					
cactus					
interval (⊂ even-hole-free)		○	○	P	
bipartite					

For other problems, see [Nishimura 2018, Section 4]

REACHABILITY problems for Independent Set

Graph	**Adjacency**			**Complexity**	**Reference**
	TS	TAR	TJ		
planar ∩ maximum degree 3	○	○	○	PSPACE-complete	[Hearn and Demaine 2005]
general line ⇐ MATCHING RECONF.	○	○ ○	○ ○	PSPACE-complete P	[Ito et al. 2011]
perfect even-hole-free cograph	○ ○	○ ○	○ ○	PSPACE-complete P P	[Kamiński et al. 2012]
cograph		○	○	P	[Bonsma 2014]
bounded bandwidth	○	○	○	PSPACE-complete	[Wrochna 2014]
claw-free (⊃ line)	○	○	○	P	[Bonsma et al. 2014]
tree (⊂ even-hole-free)		○	○	P	
bipartite permutation bipartite distance-hereditary					
planar ∩ maximum degree 3 ∩ bounded bandwidth					
cactus					
cactus					
interval (⊂ even-hole-free)		○	○	P	
bipartite					

For other problems, see [Nishimura 2018, Section 4]

Reachability problems for Independent Set

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	
cograph		○	○	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 bipartite distance-hereditary					
planar ∩ maximum degree 3 ∩ bounded bandwidth					
cactus					
cactus					
interval (⊂ even-hole-free)		○	○	P	
bipartite					

For other problems, see [Nishimura 2018, Section 4]

REACHABILITY problems for Independent Set

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	
cograph		◯	◯	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					
cactus					
cactus					
interval (⊂ even-hole-free)		◯	◯	P	
bipartite					

For other problems, see [Nishimura 2018, Section 4]

REACHABILITY problems for Independent Set

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	
cograph		○	○	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					
cactus					
interval (⊂ even-hole-free)		○	○	P	
bipartite					

For other problems, see [Nishimura 2018, Section 4]

REACHABILITY problems for Independent Set

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	
cograph		◯	◯	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	
bipartite					

For other problems, see [Nishimura 2018, Section 4]

REACHABILITY problems for Independent Set

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	
cograph		◯	◯	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					

For other problems, see [Nishimura 2018, Section 4]

Reachability problems for Independent Set

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	
cograph		○	○	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	○			P	
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]
bipartite		○	○	NP-complete	

For other problems, see [Nishimura 2018, Section 4]

REACHABILITY problems for Independent Set

Graph	Adjacency TS	Adjacency TAR	Adjacency TJ	Complexity	Reference
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	
cograph		○	○	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	

* Main Parts of Our Results

For other problems, see [Nishimura 2018, Section 4]

Outline

Our Results for SLIDING TOKEN

Graph	REACHABILITY	DIAMETER	Reference
trees	$O(n)$	$O(n^2)$	ISAAC 2014 Theor. Comp. Sci. 600, 132–142
cactus graphs	$O(n^2)$	$O(n^2)$	ISAAC 2016

* Here, n is the number of vertices of the corresponding graph.

Our Results for Sliding Token

Main Results

Graph	Reachability	Diameter	Reference
trees	$O(n)$	$O(n^2)$	ISAAC 2014 Theor. Comp. Sci. 600, 132–142
cactus graphs	$O(n^2)$	$O(n^2)$	ISAAC 2016

* Here, n is the number of vertices of the corresponding graph.

Our Results for SLIDING TOKEN

Main Results → Consequences

Graph	REACHABILITY	DIAMETER	Reference
trees	$O(n)$	$O(n^2)$	ISAAC 2014 Theor. Comp. Sci. 600, 132–142
cactus graphs	$O(n^2)$	$O(n^2)$	ISAAC 2016

* Here, n is the number of vertices of the corresponding graph.

Our Results for Sliding Token

Main Results → Consequences

Graph	Reachability	Diameter	Reference
trees	$O(n)$	$O(n^2)$	ISAAC 2014 Theor. Comp. Sci. 600, 132–142
cactus graphs	$O(n^2)$	$O(n^2)$	ISAAC 2016

* Here, n is the number of vertices of the corresponding graph.

For an instance (G, I, J) of Sliding Token, where G is one of the graphs above and I, J are independent sets of G.

1. One can identify all structures that forbid the existence of a path between I and J in the corresponding reconfiguration graph in polynomial time.
2. Without these forbidden structures, such a path exists iff $|I| = |J|$.

Our Results for SLIDING TOKEN

Main Results → Consequences

Graph	REACHABILITY	DIAMETER	Reference
trees	$O(n)$	$O(n^2)$	ISAAC 2014 Theor. Comp. Sci. 600, 132–142
cactus graphs	$O(n^2)$	$O(n^2)$	ISAAC 2016

* Here, n is the number of vertices of the corresponding graph.

For an instance (G, I, J) of SLIDING TOKEN, where G is one of the graphs above and I, J are independent sets of G.

1. One can identify all structures that forbid the existence of a path between I and J in the corresponding reconfiguration graph in polynomial time.
2. Without these forbidden structures, such a path exists iff $|I| = |J|$.

(G, I) ⟷ Structure S(G, I)

(G, J) ⟷ Structure S(G, J)

Our Results for Sliding Token

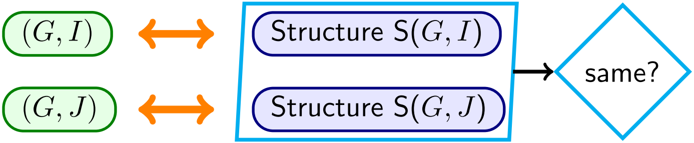
Main Results → Consequences

Graph	Reachability	Diameter	Reference
trees	$O(n)$	$O(n^2)$	ISAAC 2014 Theor. Comp. Sci. 600, 132–142
cactus graphs	$O(n^2)$	$O(n^2)$	ISAAC 2016

* Here, n is the number of vertices of the corresponding graph.

For an instance (G, I, J) of Sliding Token, where G is one of the graphs above and I, J are independent sets of G.

1. One can identify all structures that forbid the existence of a path between I and J in the corresponding reconfiguration graph in polynomial time.
2. Without these forbidden structures, such a path exists iff $|I| = |J|$.



Our Results for SLIDING TOKEN

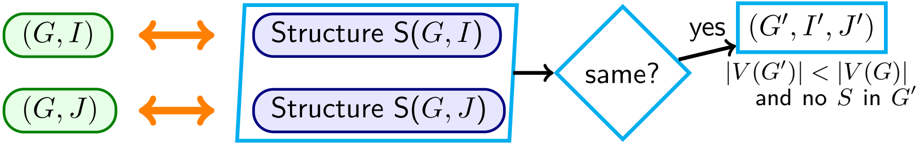
Hereditary classes — Main Results → Consequences

Graph	REACHABILITY	DIAMETER	Reference
trees	$O(n)$	$O(n^2)$	ISAAC 2014 Theor. Comp. Sci. 600, 132–142
cactus graphs	$O(n^2)$	$O(n^2)$	ISAAC 2016

* Here, n is the number of vertices of the corresponding graph.

For an instance (G, I, J) of SLIDING TOKEN, where G is one of the graphs above and I, J are independent sets of G.

1. One can identify all structures that forbid the existence of a path between I and J in the corresponding reconfiguration graph in polynomial time.
2. Without these forbidden structures, such a path exists iff $|I| = |J|$.



Our Results for SLIDING TOKEN

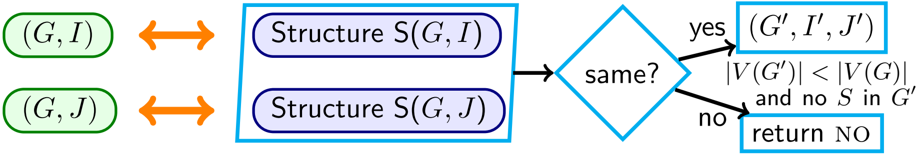
Hereditary classes — Main Results → Consequences

Graph	REACHABILITY	DIAMETER	Reference
trees	$O(n)$	$O(n^2)$	ISAAC 2014 Theor. Comp. Sci. 600, 132–142
cactus graphs	$O(n^2)$	$O(n^2)$	ISAAC 2016

* Here, n is the number of vertices of the corresponding graph.

For an instance (G, I, J) of SLIDING TOKEN, where G is one of the graphs above and I, J are independent sets of G.

1. One can identify all structures that forbid the existence of a path between I and J in the corresponding reconfiguration graph in polynomial time.
2. Without these forbidden structures, such a path exists iff $|I| = |J|$.



What next?

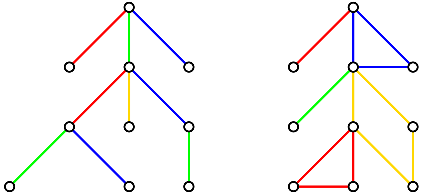
In the rest of this section, we shall present:

1. How to apply the general framework for trees.
2. High-level idea regarding how to identify forbidden structures for cactus graphs.

What are trees, cactus graphs?

A block of a graph is either an edge or a maximal 2-connected subgraph. A graph can be decomposed into a collection of blocks, where any two blocks share at most one common vertex. Intuitively, one can view

- a tree as a graph whose block is an edge;
- a cactus graph as a graph whose block is either an edge or a cycle.



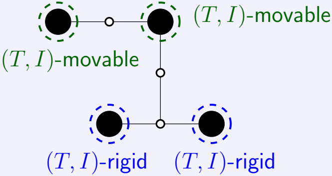
(a) A tree (b) A cactus graph

Figure: An example of (a) a tree, (b) and a cactus graph. Two blocks sharing the same vertex are colored by distinct colors.

For an instance (T, I, J) of Sliding Token, where T is a tree and I, J are independent sets of T.

Forbidden Structure: Rigid Tokens

Intuitively, a token t placed on vertex $u \in I$ is (T, I)-rigid if it cannot be moved at all. If t is not (T, I)-rigid, we say that it is (T, I)-movable.

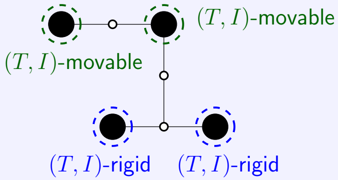


SLIDING TOKEN for trees

For an instance (T, I, J) of SLIDING TOKEN, where T is a tree and I, J are independent sets of T.

Forbidden Structure: Rigid Tokens

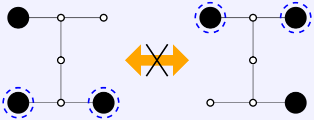
Intuitively, a token t placed on vertex $u \in I$ is (T, I)-rigid if it cannot be moved at all. If t is not (T, I)-rigid, we say that it is (T, I)-movable.



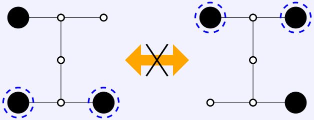
Lemma: *One can find the set* $\mathsf{R}(T, I)$ *of all* (T, I)*-rigid tokens in* $O(n)$ *time, where* $n = |V(T)|$.

Proof Sketch. Among (T, I)-movable tokens, there must be a token that can immediately be moved to one of its neighbors. The removal of such a token does not change the rigidity of other tokens.

Lemma: *If* $\mathsf{R}(T, I) \neq \mathsf{R}(T, J)$ *then* (T, I, J) *is a* NO*-instance.*



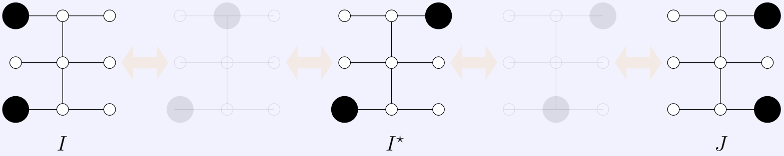
Lemma: *If* $\mathsf{R}(T, I) \neq \mathsf{R}(T, J)$ *then* (T, I, J) *is a* NO*-instance.*



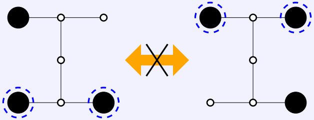
Lemma: *If* $\mathsf{R}(T, I) = \mathsf{R}(T, J) = \emptyset$ *then* (T, I, J) *is a* YES*-instance iff* $|I| = |J|$.

Proof Sketch. Construct an intermediate independent set I^*.

- Pick a safe vertex v, i.e., a vertex of degree-1 whose unique neighbor u has at most one neighbor of degree ≥ 2, and add v to I^*.
- Remove v, u, and the resulting isolated vertices. Repeat the process.



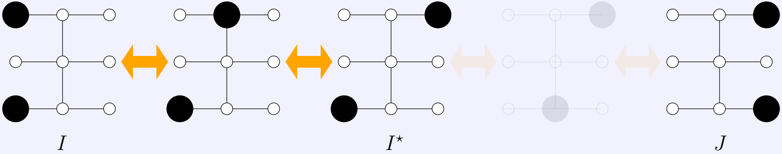
Lemma: *If* $\mathsf{R}(T, I) \neq \mathsf{R}(T, J)$ *then* (T, I, J) *is a* NO*-instance.*



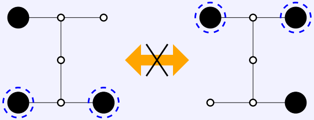
Lemma: *If* $\mathsf{R}(T, I) = \mathsf{R}(T, J) = \emptyset$ *then* (T, I, J) *is a* YES*-instance iff* $|I| = |J|$.

Proof Sketch. Construct an intermediate independent set I^*.

- Pick a safe vertex v, i.e., a vertex of degree-1 whose unique neighbor u has at most one neighbor of degree ≥ 2, and add v to I^*.
- Remove v, u, and the resulting isolated vertices. Repeat the process.



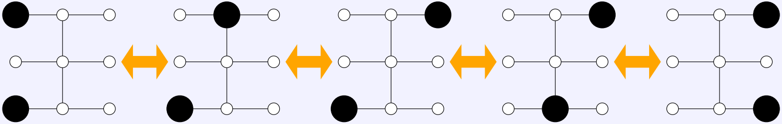
Lemma: *If* $\mathsf{R}(T, I) \neq \mathsf{R}(T, J)$ *then* (T, I, J) *is a* NO*-instance.*



Lemma: *If* $\mathsf{R}(T, I) = \mathsf{R}(T, J) = \emptyset$ *then* (T, I, J) *is a* YES*-instance iff* $|I| = |J|$*.*

Proof Sketch. Construct an intermediate independent set I^*.

- Pick a safe vertex v, i.e., a vertex of degree-1 whose unique neighbor u has at most one neighbor of degree ≥ 2, and add v to I^*.
- Remove v, u, and the resulting isolated vertices. Repeat the process.



Input: (T, I, J).
Output: YES if there is a path between I, J in the corresponding reconfiguration graph; and NO otherwise.

Input: (T, I, J).

Output: YES if there is a path between I, J in the corresponding reconfiguration graph; and NO otherwise.

- **Step 1:** If $\mathsf{R}(T, I) \neq \mathsf{R}(T, J)$, return NO. Otherwise, go to **Step 2**.
- **Step 2:** Let F be the forest obtained by removing all vertices in $N_T[\mathsf{R}(T, I)] = N_T[\mathsf{R}(T, J)]$. If $|I \cap V(F')| = |J \cap V(F')|$ for every component (tree) F' of F then return YES. Otherwise, return NO.

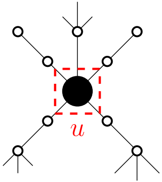
Input: (T, I, J).
Output: YES if there is a path between I, J in the corresponding reconfiguration graph; and NO otherwise.

- **Step 1:** If $\mathsf{R}(T, I) \neq \mathsf{R}(T, J)$, return NO. Otherwise, go to **Step 2**.
- **Step 2:** Let F be the forest obtained by removing all vertices in $N_T[\mathsf{R}(T, I)] = N_T[\mathsf{R}(T, J)]$. If $|I \cap V(F')| = |J \cap V(F')|$ for every component (tree) F' of F then return YES. Otherwise, return NO.

Running Time: $O(n)$, where $n = |V(T)|$.

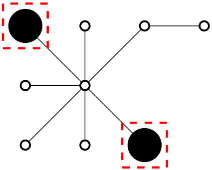
Generalization of rigid tokens

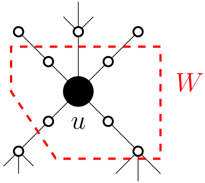
The concept of rigid tokens can be generalized.



(G, I)-rigid token
cannot be moved “out of” $G[\{u\}]$

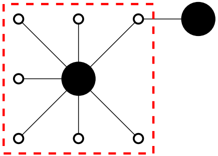
example





(G, I, W)-confined token
cannot be moved “out of” $G[W]$

example



Generalization: trees $\Rightarrow$ cactus graphs

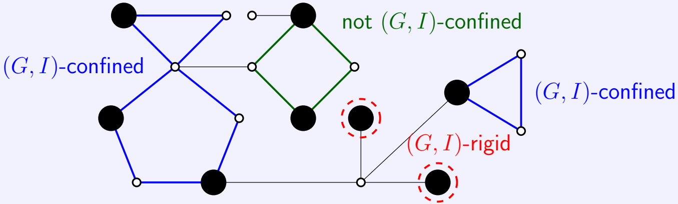
For an instance (G, I, J) of SLIDING TOKEN, where G is a cactus graph and I, J are independent sets of G.

Forbidden Structure

Rigid Tokens — **similar to trees**

Confined Cycles — **new structure**

Intuitively, an induced cycle C of G is (G, I)-confined if $I \cap V(C)$ forms a maximum independent set of C (of size $\lfloor |V(C)|/2 \rfloor$) and all tokens in $I \cap V(C)$ are $(G, I, V(C))$-confined.



Generalization: trees $\Rightarrow$ cactus graphs

Lemma: *One can respectively find the sets* $\mathsf{R}(G, I)$ *and* $\mathsf{C}(G, I)$ *of all* (G, I)*-rigid tokens and* (G, I)*-confined cycles in* $O(n^2)$ *time, where* $n = |V(G)|$*.*

Lemma: *If either* $\mathsf{R}(G, I) \neq \mathsf{R}(G, J)$ *or* $\mathsf{C}(G, I) \neq \mathsf{C}(G, J)$ *then* (G, I, J) *is a* NO*-instance.*

Lemma: *If* $\mathsf{R}(G, I) = \mathsf{R}(G, J) = \emptyset$ *and* $\mathsf{C}(G, I) = \mathsf{C}(G, J) = \emptyset$ *then* (G, I, J) *is a* YES*-instance iff* $|I| = |J|$*.*

Outline

1. Background and Motivation
   - Reconfiguration Problems
   - Reconfigurability of Independent Set

2. Our Results

3. Future Works

4. Publications

It is well-known that

> **Theorem ([van der Zanden 2015]):** *There exists a constant c such that* Independent Set Reconfiguration *(under* TS, TJ, *or* TAR*) is* PSPACE*-complete even for planar graphs of maximum degree 3 and of bandwidth/treewidth/pathwidth/cliquewidth at most c.*

An interesting open question is whether there exists efficient algorithms for solving the problem when the input graph is of small bandwidth/treewidth/pathwidth/cliquewidth. Interesting target graphs are:

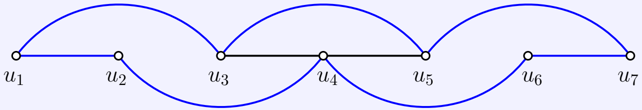
1. Series-parallel graphs ($\equiv$ graphs of treewidth ≤ 2). Cactus graphs is one of its subclasses.
2. Distance-hereditary graphs (whose cliquewidth ≤ 3).
3. Bandwidth-2 graphs ($\equiv$ graphs of bandwidth ≤ 2).

Future Works: Bandwidth-2 graphs

A graph $G = (V, E)$ is a bandwidth-2 (bw2) graph if there exists a one-to-one function $f : V \to \{1, 2, \ldots, |V|\}$ (called a bw2 layout of G) such that its bandwidth $\text{bw}(f) = \max_{uv \in E} |f(u) - f(v)|$ is at most 2. For a layout f, we use the notation $f = (u_1, u_2, \ldots, u_n)$, where $n = |V|$ and $u_i = f^{-1}(u_i)$.

Conjecture: Sliding Token *for bandwidth-2 graphs is in* P.

Lemma ([Makedon et al. 1993]): *Let $G = (V, E)$, $|V| = n \geq 3$, be a bw2 biconnected graph. Let $f = (u_1, u_2, \ldots, u_n)$ be a bw2 layout of G. Then, the pairs (u_1, u_2), (u_i, u_{i+2}), $1 \leq i \leq n-2$, and (u_{n-1}, u_n) are all edges of G.*



- cycles ⇒ bw2 biconnected graphs
- cactus graphs ⇒ bw2 graphs

In this thesis, we've claimed that

> **Theorem:** INDEPENDENT SET RECONFIGURATION *for trees (under* TS, TJ, TAR*) is in* P.

Moreover,

> **Theorem ([Kamiński et al. 2012]):** SHORTEST RECONFIGURATION *for* INDEPENDENT SET RECONFIGURATION *under* TJ *and* TAR *rules are in* P *for even-hole-free graphs* ($\supset$ *trees).*

Interestingly, the following question remains open for trees.

SHORTEST SLIDING TOKEN [Yamada and Uehara 2016]

Configuration	Independent set (viewed as a set of tokens) of a graph
Adjacency [Token Sliding]	A token can be slid to an adjacent unoccupied vertex
Problem	SHORTEST RECONFIGURATION

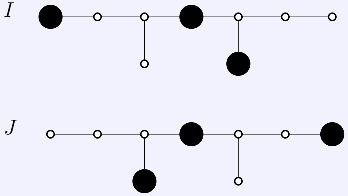
Theorem ([Yamada and Uehara 2016]): Shortest Sliding Token *for caterpillars* ($\subset$ *trees) is in* P.

Theorem (in discussion with Amanj Khorramian and Ryuhei Uehara [unpublished]): Shortest Sliding Token *for spiders (i.e., trees having exactly one vertex of degree* ≥ 3*) is in* P.

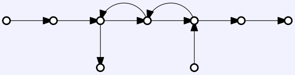
Conjecture: SHORTEST SLIDING TOKEN *for trees is in* P.

Key Structure: Auxiliary Graph $A(T, I, J)$

For an instance (T, I, J), we define the corresponding (directed) auxiliary graph $A(T, I, J)$ such that $V(A(T, I, J)) = V(T)$ and $E(A(T, I, J)) = \left\{(x, y) : xy \in E(T) \text{ and } \left|I \cap V(T^x_y)\right| \leq \left|J \cap V(T^x_y)\right|\right\}$, where T^x_y is the subtree of T induced by y and its descendants when regarding x as the root of T.



(a) An instance (T, I, J)



(b) The auxiliary graph $A(T, I, J)$

In general, one may study the structure of the corresponding reconfiguration graph of INDEPENDENT SET RECONFIGURATION (under TS, TJ, TAR). Some interesting problems are:

1. Which graph can be a reconfiguration graph of INDEPENDENT SET RECONFIGURATION?
   - Under TJ/TAR: First studied in [Fatehi et al. 2017].
   - Under TS: Open.
2. Whether the reconfiguration graph and its corresponding original graph belong to the same graph classes?
3. and so on.

Outline

Publications

- Journal
  - Erik D. Demaine, Martin L. Demaine, Eli Fox-Epstein, Duc A. Hoang, Takehiro Ito, Hirotaka Ono, Yota Otachi, Ryuhei Uehara, and Takeshi Yamada: "Linear-Time Algorithm for Sliding Tokens on Trees," *Theoretical Computer Science*, Vol. 600, pp. 132–142 (Jul. 2015).
- International Conference
  - Erik D. Demaine, Martin L. Demaine, Eli Fox-Epstein, Duc A. Hoang, Takehiro Ito, Hirotaka Ono, Yota Otachi, Ryuhei Uehara, and Takeshi Yamada: "Polynomial-Time Algorithm for Sliding Tokens on Trees," Proceedings of the 25th International Symposium on Algorithms and Computation (ISAAC 2014), LNCS 8889, pp. 389–400 (Dec. 2014).
  - Duc A. Hoang, and Ryuhei Uehara: "Sliding Tokens on a Cactus," Proceedings of the 27th International Symposium on Algorithms and Computation (ISAAC 2016), LIPIcs 64, pp. 37:1–37:26 (Dec. 2016).
- Others (not included in this thesis)
  - Eli Fox-Epstein, Duc A. Hoang, Yota Otachi, and Ryuhei Uehara: "Sliding Token on Bipartite Permutation Graphs," Proceedings of the 26th International Symposium on Algorithms and Computation (ISAAC 2015), LNCS 9472, pp. 237–247 (Dec. 2015).
  - Duc A. Hoang, Eli Fox-Epstein, and Ryuhei Uehara: "Sliding Tokens on Block Graphs," Proceedings of the 11th International Conference and Workshops on Algorithms and Computation (WALCOM 2017), LNCS 10167, pp. 460–471 (Mar. 2017).

Bibliography

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.

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.

Bibliography

de Berg, Mark, Bart M.P. Jansen, and Debankur Mukherjee (2016). "Independent-Set Reconfiguration Thresholds of Hereditary Graph Classes". In: *Proceedings of the 36th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2016*. Ed. by Akash Lal, S. Akshay, Saket Saurabh, and Sandeep Sen. Vol. 65. Leibniz International Proceedings in Informatics. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 34:1–34:15. DOI: 10.4230/LIPIcs.FSTTCS.2016.34.

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.

Fatehi, Davood, Saeid Alikhania, and Abdul Jalil M. Khalaf (2017). "The k-independent graph of a graph". In: *Advances and Applications in Discrete Mathematics* 18.1, pp. 45–56. DOI: 10.17654/DM018010045.

Bibliography

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.

Goldreich, Oded (2011). "Finding the Shortest Move-Sequence in the Graph-Generalized 15-Puzzle Is NP-Hard". In: *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation*. Ed. by Oded Goldreich. Vol. 6650. Lecture Notes in Computer Science. Springer, pp. 1–5. DOI: 10.1007/978-3-642-22670-0_1.

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.

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.

Johnson, Wm. Woolsey and William E. Story (1879). "Notes on the "15" Puzzle". In: *American Journal of Mathematics* 2.4, pp. 397–404. DOI: 10.2307/2369492.

Bibliography

- 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.
- Kornhauser, D., G. Miller, and P. Spirakis (1984). "Coordinating Pebble Motion On Graphs, The Diameter Of Permutation Groups, And Applications". In: *Proceedings of the 25th Annual Symposium on Foundations of Computer Science, SFCS 1984*. IEEE Computer Society, pp. 241–250. DOI: 10.1109/SFCS.1984.715921.
- 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.
- Makedon, Fillia, Dafna Sheinwald, and Yaron Wolfsthal (1993). "A simple linear-time algorithm for the recognition of bandwidth-2 biconnected graphs". In: *Information Processing Letters* 46.2, pp. 103–107. DOI: 10.1016/0020-0190(93)90206-O.

Bibliography

- 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. DOI: 10.3390/a11040052.
- Ratner, Daniel and Manfred Warmuth (1990). "The $(n^2 - 1)$-puzzle and related relocation problems". In: *Journal of Symbolic Computation* 10.2, pp. 111–137. DOI: 10.1016/S0747-7171(08)80001-6.
- 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.
- van den Heuvel, Jan (2013). "The complexity of change". In: *Surveys in Combinatorics 2013*. Ed. by Simon R. Blackburn, Stefanie Gerke, and Mark Wildon. London Mathematical Society Lecture Note Series. Cambridge University Press, pp. 127–160. DOI: 10.1017/CBO9781139506748.005.

Bibliography

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.

Wilson, Richard M. (1974). "Graph puzzles, homotopy, and the alternating group". In: *Journal of Combinatorial Theory, Series B* 16.1, pp. 86–96. DOI: 10.1016/0095-8956(74)90098-7.

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

Yamada, Takeshi and Ryuhei Uehara (2016). "Shortest reconfiguration of sliding tokens on a caterpillar". In: *The 10th International Workshop on Algorithms and Computation, WALCOM 2016*. Ed. by Mohammad Kaykobad and Rossella Petreschi. Vol. 9627. Lecture Notes in Computer Science. Springer, pp. 236–248. DOI: 10.1007/978-3-319-30139-6_19.