The Complexity of Distance-r Dominating Set Reconfiguration

Niranka Banerjee[1] Duc A. Hoang[2,3]

[1]RIMS, Kyoto University, Japan

[2]VNU University of Science, Hanoi, Vietnam

[3]VIASM, Hanoi, Vietnam

August 23–25, 2024

Outline

Graph Reconfiguration

Reconfiguration Setting

In a *reconfiguration variant* of a computational problem (e.g., SAT, INDEPENDENT SET, DOMINATING SET, VERTEX-COLORING, etc.), two *feasible solutions* S and T are given along with a *reconfiguration rule* that describes how to slightly modify one feasible solution to obtain a new one.

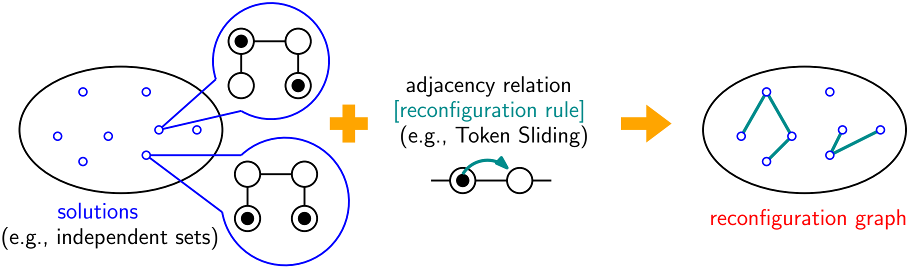


Figure: Reconfiguration.

Graph Reconfiguration

Main Questions

In the reconfiguration graph,

- REACHABILITY: Is there a path between two given solutions? Can we transform S into T via a sequence of feasible solutions?

  Such a sequence, if exists, is called a *reconfiguration sequence*.

- SHORTEST PATH: If REACHABILITY is yes, can we find a shortest path between S and T?

Graph Reconfiguration

Reconfiguration Rules

Tells us rules we need to follow to go from one feasible solution to another.

Graph Reconfiguration

Reconfiguration Rules

Tells us rules we need to follow to go from one feasible solution to another.

A few well known Reconfiguration rules: Token Sliding (TS), Token Jumping (TJ) and Token Addition/ Removal (TAR).

Tokens refer to the solution set. (No vertex has more than one token.)

Graph Reconfiguration

Reconfiguration Rules

Tells us rules we need to follow to go from one feasible solution to another.

A few well known Reconfiguration rules: Token Sliding (TS), Token Jumping (TJ) and Token Addition/ Removal (TAR).

Tokens refer to the solution set. (No vertex has more than one token.)

- **Token Sliding (TS):** one can move a token to one of its unoccupied neighbors as long as the resulting token-set forms a feasible solution.

Graph Reconfiguration

Reconfiguration Rules

Tells us rules we need to follow to go from one feasible solution to another.

A few well known Reconfiguration rules: Token Sliding (TS), Token Jumping (TJ) and Token Addition/ Removal (TAR).

Tokens refer to the solution set. (No vertex has more than one token.)

- **Token Sliding (TS):** one can move a token to one of its unoccupied neighbors as long as the resulting token-set forms a feasible solution.
- **Token Jumping (TJ):** one can move a token to any unoccupied vertex as long as the resulting token-set forms a feasible solution.

Graph Reconfiguration

Reconfiguration Rules

Tells us rules we need to follow to go from one feasible solution to another.

A few well known Reconfiguration rules: Token Sliding (TS), Token Jumping (TJ) and Token Addition/ Removal (TAR).

Tokens refer to the solution set. (No vertex has more than one token.)

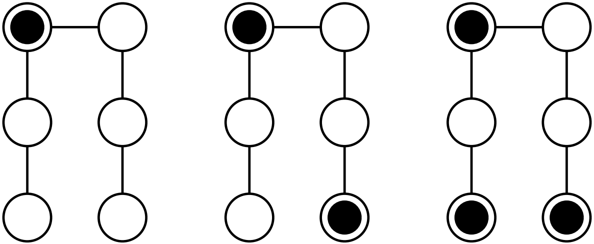
- **Token Sliding (TS):** one can move a token to one of its unoccupied neighbors as long as the resulting token-set forms a feasible solution.
- **Token Jumping (TJ):** one can move a token to any unoccupied vertex as long as the resulting token-set forms a feasible solution.
- **Token Addition/Removal ($TAR(k)$):** one can either add or remove a token as long as the resulting token-set forms a feasible solution of size at most some threshold $k \geq 0$.

Distance-r Dominating Set Reconfiguration (DrDSR)

Distance-r Dominating Set (DrDS) (Meir and Moon [PJM 1975])

Given a fixed integer $r \geq 1$, a *distance-r dominating set (DrDS)* of G is a vertex subset D where each vertex of G is within distance r from some member of D.

For $r = 1$, this is the classical dominating set concept.



D1DS: NO
D2DS: NO

D1DS: NO
D2DS: YES

D1DS: YES
D2DS: YES

Figure: Examples of DrDSs ($r = 1, 2$).

Distance-r Dominating Set Reconfiguration (DrDSR)

Goal

We study Reachability of DrDSs ($r \geq 1$) under TS and TJ.

Distance-r Dominating Set Reconfiguration (DrDSR)

Goal

We study REACHABILITY of DrDSs $(r \geq 1)$ under TS and TJ.

(G, D_s, D_t) under R

- **Input:** Two DrDSs $(r \geq 1)$ D_s and D_t of a graph G and a reconfiguration rule $R \in \{TS, TJ\}$.

Distance-r Dominating Set Reconfiguration (DrDSR)

Goal

We study REACHABILITY of DrDSs ($r \geq 1$) under TS and TJ.

(G, D_s, D_t) under R

- **Input:** Two DrDSs ($r \geq 1$) D_s and D_t of a graph G and a reconfiguration rule $R \in \{TS, TJ\}$.
- **Question:** Is there a reconfiguration sequence between D_s and D_t, i.e., a sequence $\langle D_0, D_1, \dots, D_\ell \rangle$ such that each D_i is a DrDS and D_{i+1} is obtained from D_i ($i \in \{0, \dots, \ell\}$) by applying R exactly once?

Distance-r Dominating Set Reconfiguration (DrDSR)

Haddadan et al. [TCS 2016] studied the problem for $r = 1$ under TAR. Later, Bonamy et al. [DAM 2021] observed that these results also hold under TJ.

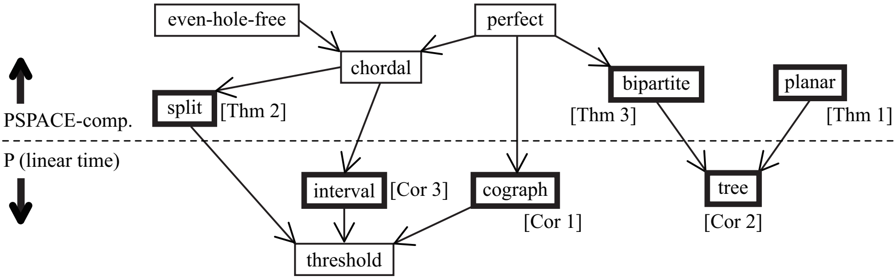


Figure: Complexity of D1DSR under TAR on some graphs, © Haddadan et al. [TCS 2016]. Arrows indicate inclusion.

Distance-r Dominating Set Reconfiguration (DrDSR)

Bonamy et al. [DAM 2021] showed that several results of Haddadan et al. [TCS 2016] also hold under TS.

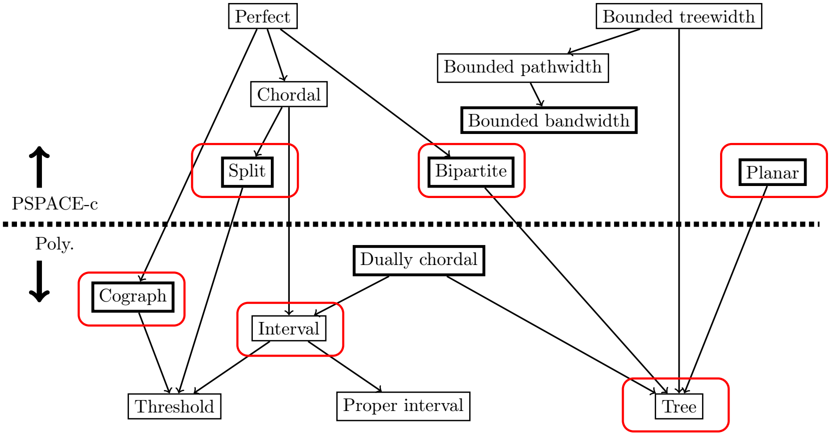


Figure: Complexity of D1DSR under TS on some graphs, © Bonamy et al. [DAM 2021]. Arrows indicate inclusion.

Distance-r Dominating Set Reconfiguration (DrDSR)

More results for $r = 1$ under TS:

- Bousquet and Joffard [FCT 2021]: PSPACE-complete on circle graphs, P on circular-arc graphs.
- Křišt'an and Svoboda [FCT 2023]: Polynomial-time algorithms for SHORTEST PATH variants on trees and interval graphs.

Distance-r Dominating Set Reconfiguration (DrDSR)

For $r = 1$, under TAR, from the parameterized complexity viewpoint:

Distance-r Dominating Set Reconfiguration (DrDSR)

For $r = 1$, under TAR, from the parameterized complexity viewpoint:

- Two natural parameterizations: the number of tokens k and the length of a reconfiguration sequence ℓ.

Distance-r Dominating Set Reconfiguration (DrDSR)

For $r = 1$, under TAR, from the parameterized complexity viewpoint:

- Two natural parameterizations: the number of tokens k and the length of a reconfiguration sequence ℓ.
- Mouawad et al. [Alg. 2017]: W[1]-hard parameterized by k, W[2]-hard by $k + \ell$.

Distance-r Dominating Set Reconfiguration (DrDSR)

For $r = 1$, under TAR, from the parameterized complexity viewpoint:

- Two natural parameterizations: the number of tokens k and the length of a reconfiguration sequence ℓ.
- Mouawad et al. [Alg. 2017]: W[1]-hard parameterized by k, W[2]-hard by $k + \ell$.
- Lokshtanov et al. [JCSS 2018]: FPT parameterized by k on graphs excluding $K_{d,d}$ as a subgraph for any constant d.

Distance-r Dominating Set Reconfiguration (DrDSR)

For $r = 1$, under TAR, from the parameterized complexity viewpoint:

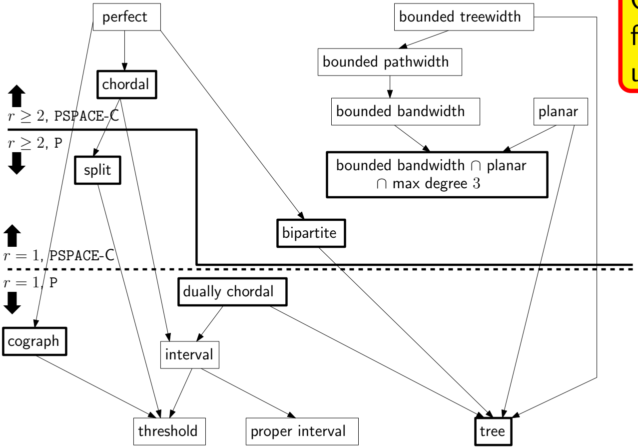
- Two natural parameterizations: the number of tokens k and the length of a reconfiguration sequence ℓ.
- Mouawad et al. [Alg. 2017]: W[1]-hard parameterized by k, W[2]-hard by $k + \ell$.
- Lokshtanov et al. [JCSS 2018]: FPT parameterized by k on graphs excluding $K_{d,d}$ as a subgraph for any constant d.
- Bousquet et al. [CSR 2024]: FPT parameterized by ℓ on any graph class where first-order model-checking is in FPT.

Distance-r Dominating Set Reconfiguration (DrDSR)

Our Results

We prove several (classic) complexity results for $r \geq 2$ under TJ and TS on different graph classes. (Arrows indicate inclusion.)

Our Results for $r \geq 2$ under TJ

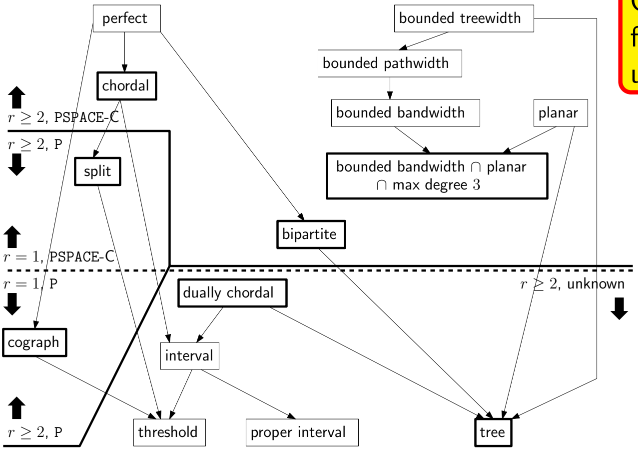


Distance-r Dominating Set Reconfiguration (DrDSR)

Our Results

We prove several (classic) complexity results for $r \geq 2$ under TJ and TS on different graph classes. (Arrows indicate inclusion.)

Our Results for $r \geq 2$ under TS



Distance-r Dominating Set Reconfiguration (DrDSR)

Main Result 1

DrDSR ($r \geq 1$) is PSPACE-complete on planar graphs of maximum degree 3 and bounded bandwidth.

- Previously known results ($r = 1$) are only for "maximum degree 6".
- We improve the known results and extend for $r \geq 2$.

Distance-r Dominating Set Reconfiguration (DrDSR)

Main Result 1

DrDSR ($r \geq 1$) is PSPACE-complete on planar graphs of maximum degree 3 and bounded bandwidth.

- Previously known results ($r = 1$) are only for "maximum degree 6".
- We improve the known results and extend for $r \geq 2$.

Main Result 2

DrDSR ($r \geq 1$) on split graphs: PSPACE-complete when $r = 1$ (which is already known) but in P when $r \geq 2$ (which we prove).

- An interesting complexity dichotomy.
- We further establish some non-trivial bounds on the length of a shortest reconfiguration sequence when $r = 2$. (The case $r \geq 3$ is trivial and boring.)

Planar Graphs

Main Result 1

DrDSR ($r \geq 1$) is PSPACE-complete on planar graphs of maximum degree 3 and bounded bandwidth.

Planar Graphs

Main Result 1

DrDSR ($r \geq 1$) is PSPACE-complete on planar graphs of maximum degree 3 and bounded bandwidth.

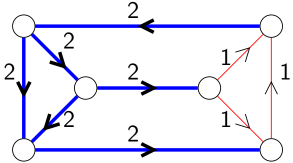
Idea

Reduction from Nondeterministic Constraint Logic (NCL), a powerful tool introduced by Hearn and Demaine [TCS 2008]

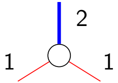
Planar Graphs

Nondeterministic Constraint Logic (NCL)

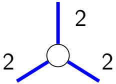
- **Input:**
  - Each *state/configuration* involves a graph having red (weight 1) and blue (weight 2) edges where each edge is oriented such that $(\star)$ the sum of weights of in-coming arcs at each vertex is at least 2.
  - **Reconfiguration Rule:** Each *move* involves re-orienting an edge such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



(a) An NCL configuration.



(b) AND vertex.

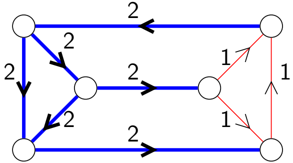


(c) OR vertex.

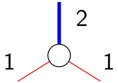
Planar Graphs

Nondeterministic Constraint Logic (NCL)

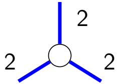
- **Input:**
  - Each *state/configuration* involves a graph having red (weight 1) and blue (weight 2) edges where each edge is oriented such that $(\star)$ the sum of weights of in-coming arcs at each vertex is at least 2.
  - **Reconfiguration Rule:** Each *move* involves re-orienting an edge such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



(a) An NCL configuration.



(b) AND vertex.



(c) OR vertex.

PSPACE-complete even on *planar graphs* having only *two types of vertices*.

Planar Graphs

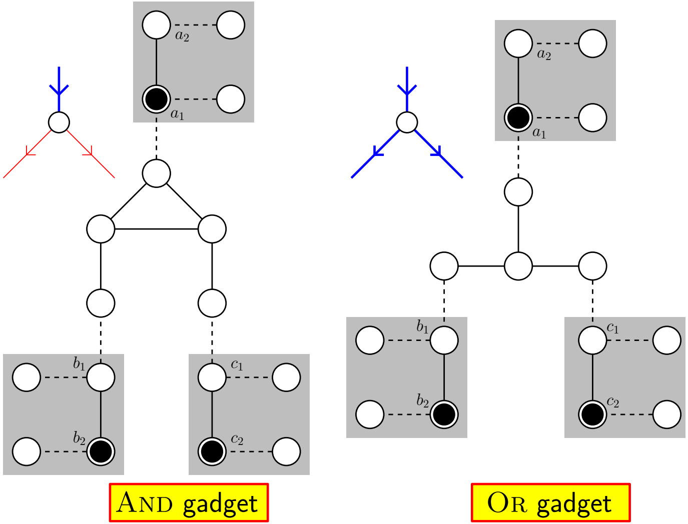
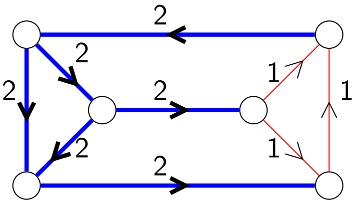


Figure: Our gadgets for DrDSR. Each dashed edge represents a path of length $r-1$. The gray boxes indicate the *link components* in a gadget.

Planar Graphs

Gadgets are joined together at their link components.



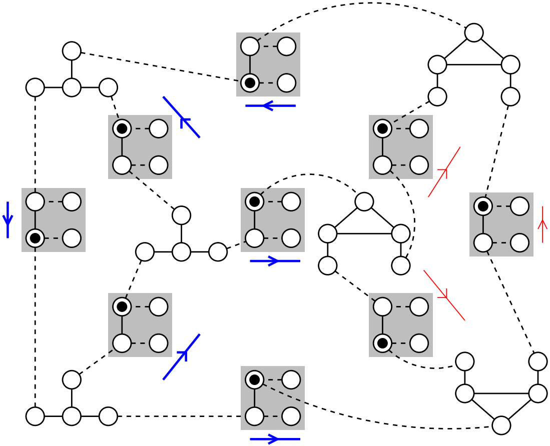


Figure: A NCL AND/OR constraint graph and its configuration (Left) and the corresponding graph and token-set (Right). Each dashed edge represents a path of length $r-1$. The gray boxes indicate the *link components* in a gadget.

Split Graphs

Main Result 2

DrDSR ($r \geq 1$) on split graphs: PSPACE-complete when $r = 1$ (which is already known) but in P when $r \geq 2$ (which we prove).

Split Graphs

Main Result 2

DrDSR ($r \geq 1$) on split graphs: PSPACE-complete when $r = 1$ (which is already known) but in P when $r \geq 2$ (which we prove).

Idea (for $r \geq 2$)

- $r \geq 3$: trivial. (Any connected split graph has diameter $\leq 3 \Rightarrow$ Any non-empty token-set is a DrDS $\Rightarrow$ Reconfiguration is easy!)
- $r = 2$: when doing reconfiguration, *always keep at least one token in the clique side* of the split graph.

Split Graphs

Main Result 2

DrDSR ($r \geq 1$) on split graphs: PSPACE-complete when $r = 1$ (which is already known) but in P when $r \geq 2$ (which we prove).

Idea (for $r \geq 2$)

- $r \geq 3$: trivial. (Any connected split graph has diameter ≤ 3 $\Rightarrow$ Any non-empty token-set is a DrDS $\Rightarrow$ Reconfiguration is easy!)
- $r = 2$: when doing reconfiguration, *always keep at least one token in the clique side* of the split graph.

Bounds on Length ℓ_R ($R \in \{TS, TJ\}$) of a Shortest R-Sequence Between Two D2DSs D_s and D_t of a Split Graph

$$\min_{\substack{\text{bijection}\\ f: D_s \to D_t}} \sum_{s \in D_s} \mathsf{dist}(s, f(s)) \underset{\text{trivial}}{\leq} \ell_{TS} \underset{\text{non-trivial}}{\leq} \min_{\substack{\text{bijection}\\ f: D_s \to D_t}} \sum_{s \in D_s} \mathsf{dist}(s, f(s)) + 2.$$

$$\frac{|D_s \Delta D_t|}{2} \underset{\text{trivial}}{\leq} \ell_{TJ} \underset{\text{non-trivial}}{\leq} \frac{|D_s \Delta D_t|}{2} + 1.$$

Split Graphs

(G, D_s, D_t) under TS with $\ell_{TS} = \min_{\substack{\text{bijection} \\ f:D_s \to D_t}} \sum_{s \in D_s} \text{dist}(s, f(s)) + 2$.

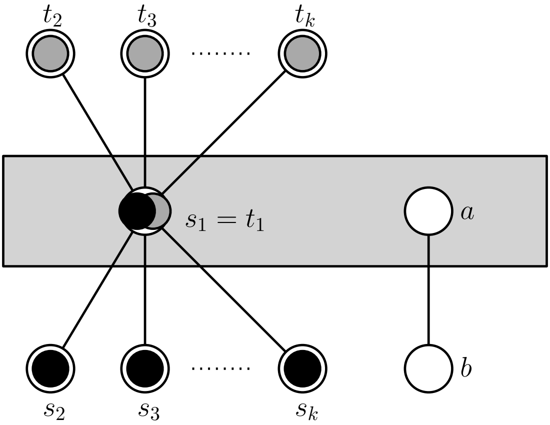


Figure: A D2DSR's instance on split graphs where ℓ_{TS} achieves the (non-trivial) upper bound. Black/gray tokens are respectively in D_s/D_t.

Split Graphs

(G, D_s, D_t) under TJ with $\ell_{TJ} = \dfrac{|D_s \Delta D_t|}{2} + 1$.

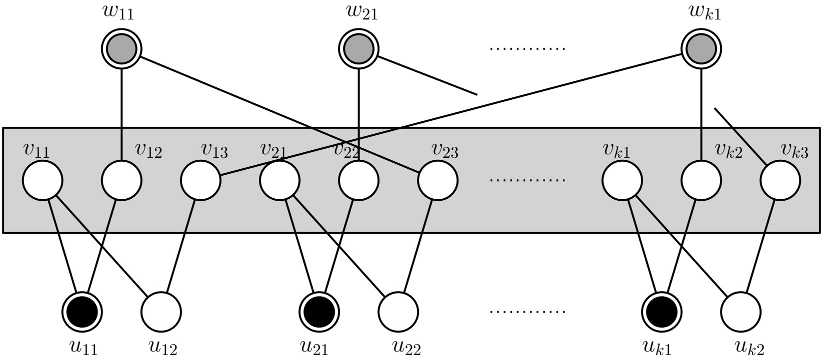


Figure: A D2DSR's instance on split graphs where ℓ_{TJ} achieves the (non-trivial) upper bound. Black/gray tokens are respectively in D_s/D_t.

Open Questions

Question 1

What is the complexity of DrDSR ($r \geq 2$) on *trees* under TS?

Question 2

What is the complexity of DrDSR ($r \geq 2$) on *interval graphs* under TS?