Shortest paths between shortest paths and independent sets

Marcin Kamiński

Université Libre de Brussels

Paul Medvedev

University of Toronto

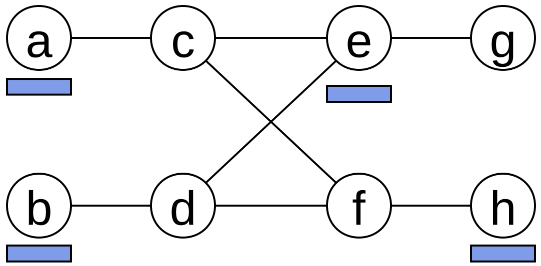
Martin Milanič

University of Primorska, Slovenia

International Workshop on Algorithms on Combinatorial Algorithms 2010

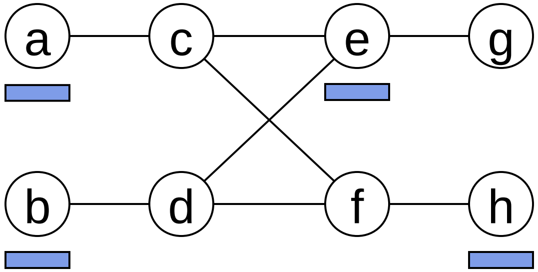
Reconfiguration Problems

- Usual world (`Prob`)
  - domain – graph
  - constraints – feasible solution
    - e.g. independent set
  - optimization problem – optimum solution
    - e.g. find ind.set. of maximum size
- Reconfiguration world (`ReconProb`)
  - you are given two solutions
  - reconfiguration rule
    - e.g. token jumping
  - each intermediate step must be a solution
  - problem: are the two solutions reconfigurable?

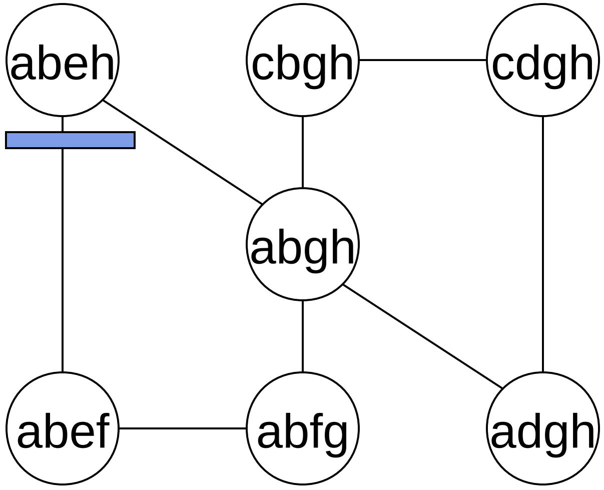


Reconfiguration Graph (R_G)

- Reconfiguration between two solutions is a path in the reconfiguration graph



Main graph G



R_G: Reconfiguration graph of G

Systematic study of ReconProb Complexity

- Complexity of Prob vs. ReconProb

		ReconProb	
Prob		poly-time	intractable
	poly-time	spanning tree matching matroid (Ito et al. 08)	empty???
	intractable	3-coloring (Cereceda et al 08)	independent set set cover integer program. (Ito et al 08) tight-SAT (Gopalan et al 09)

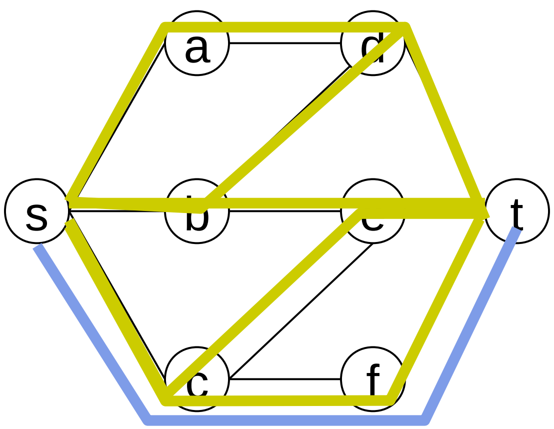
- Diameter of R_G vs. ReconProb Complexity
  - ReconProb is poly-time $\rightarrow$ polynomial diameter of R_G ???
    - 2-coloring, 3-coloring (Cereceda et al 08), tightSAT (Gopalan et al 09)

Outline

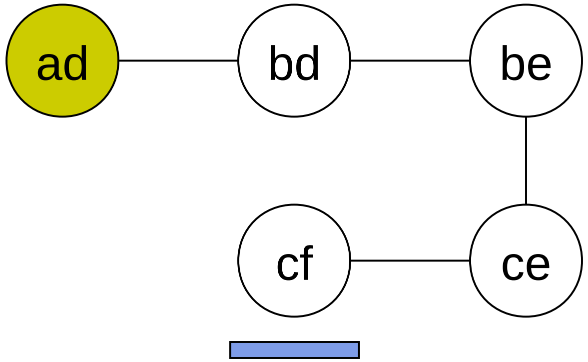
- Shortest path reconfiguration
  - SP reconfiguration has exponential diameter
  - finding minimum reconfiguration distance is NP-hard
- Independent set reconfiguration
  - graph classes with linear reconfigurability algorithm
  - graph classes where reconfigurability remains hard

Shortest Path Reconfiguration

- feasible solution: shortest s-t path
- reconfiguration rule: swap one vertex

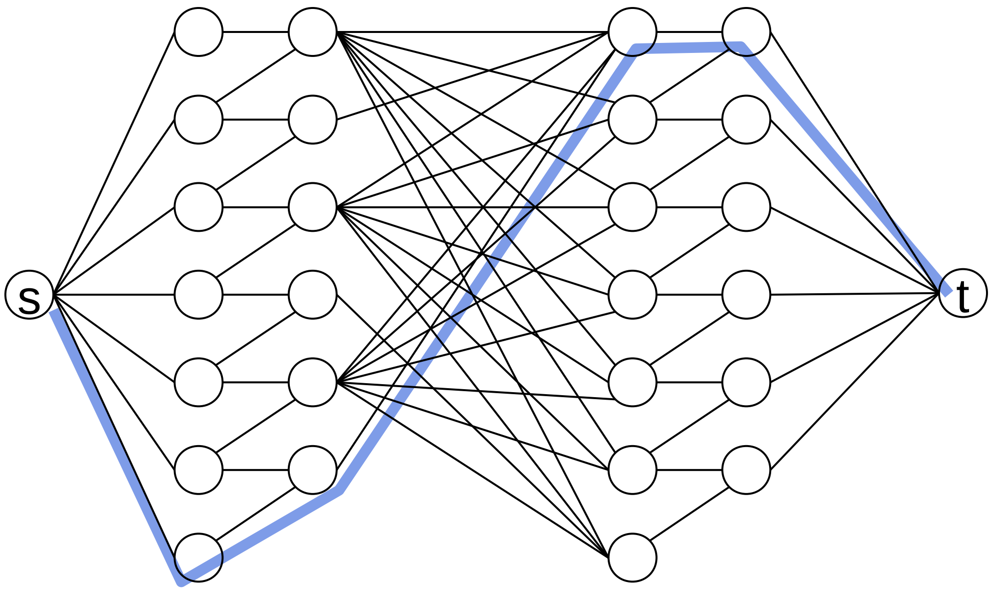


Main graph G



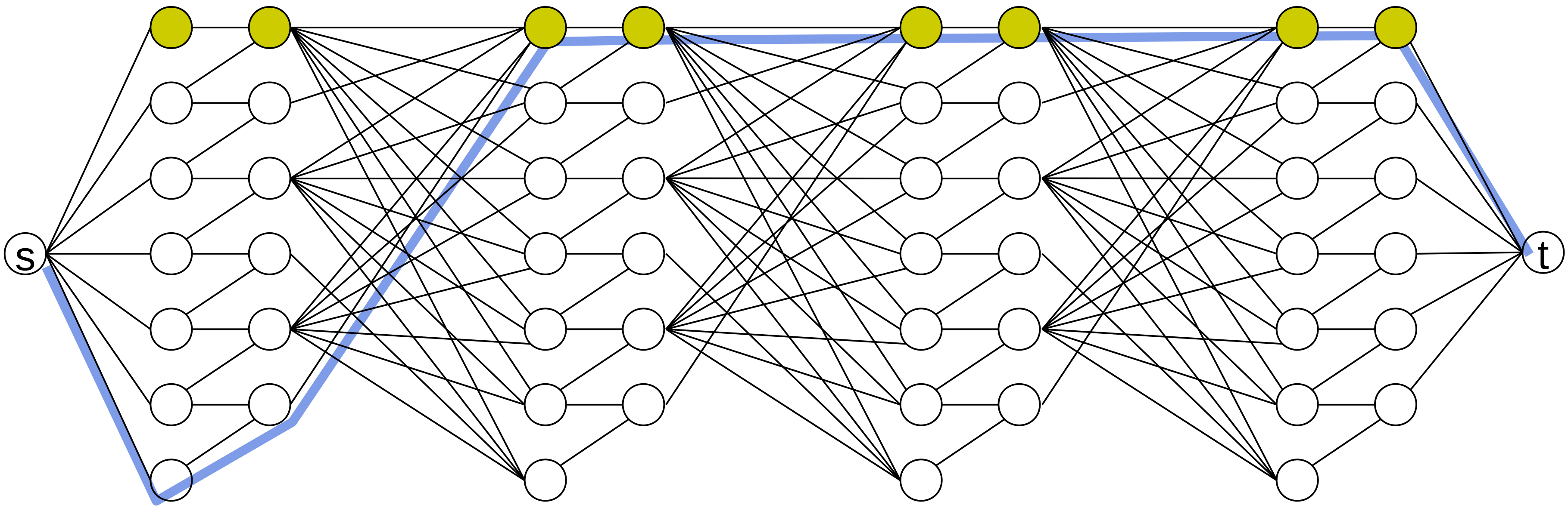
R_G: Reconfiguration graph of G

SP Reconfiguration Graph has Exponential Diameter

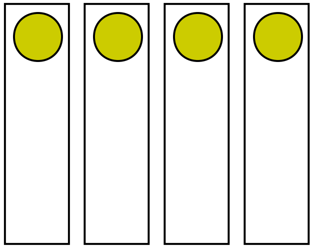


- In order to propogate the 1st layer down, the 2nd layer must propogate both down and up

SP Reconfiguration Graph has Exponential Diameter

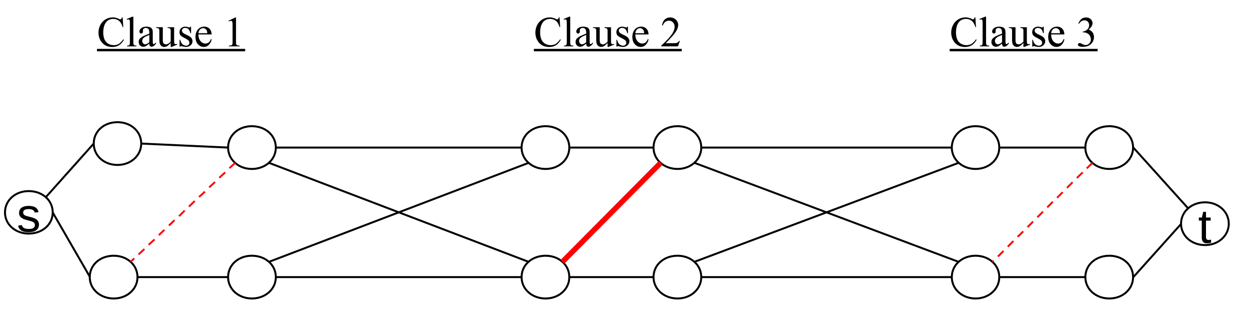


- The reconfiguration distance is $\Theta(2^k)$ for k layers

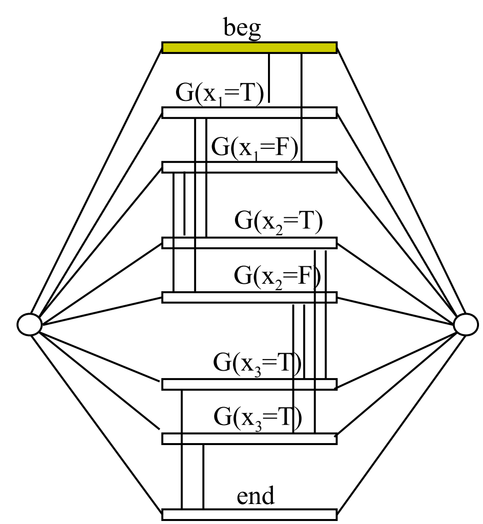


SP min-Reconfiguration is NP-hard (via SAT)

- Theorem: finding the minimum reconfiguration distance is NP-hard
- widget for each variable assignment, e.g. x_1 = true
  - red edge is present if clause *i* is satisfied by x_1 = true
- the two x_i widgets are connected to the two x_{i+1} widgets



$G(x_1 = T)$



Shortest Path Summary

- Our results
  - SP reconfiguration has exponential diameter
  - finding minimum reconfiguration distance is NP-hard
- If SP reconfig is...
  - intractable
    - it is the first poly-time `Prob` with intractable `ReconProb`
  - poly-time

    ~~"`ReconProb` is poly time → polynomial diameter of R_G"~~

		ReconProb	
Prob		poly	intractable
	poly	spanning tree matching matroid (Ito et al. 08)	SP???
	intractable	3-coloring (cite)	independent set set cover integer program. (Ito et al 08) tight-SAT (Gopalan et al 09)

- Open problem
  - what is the complexity of deciding if two shortest paths are reconfigurable?

Outline

- Shortest path reconfiguration
  - SP reconfiguration has exponential diameter
  - finding minimum reconfiguration distance is NP-hard
- Independent set reconfiguration
  - graph classes with linear reconfigurability algorithm
  - graph classes where reconfigurability remains hard

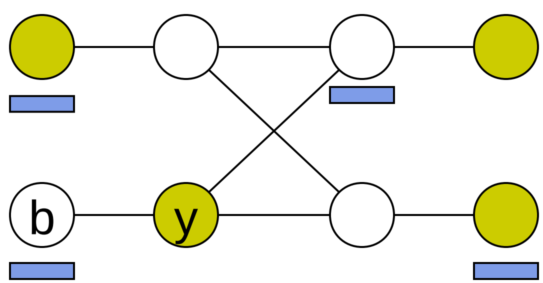
Linear Time Algorithm For Independent Set Reconfiguration

<u>Theorem</u>: If G has no induced even cycles, and suppose we have 2 independent sets of the same cardinality. Then:

1. They can always be reconfigured
2. The shortest reconfiguration sequence can be found in linear time

```
Recon(YEL, BLUE) //YEL and BLUE are sets of vertices

1. Identify subgraph G' induced by YEL / BLUE (sym. dif.)
2. Note G' is bipartite, and is a forest
3. Find a blue vertex b in G' with at most one yellow neighbor y in G'
4. If b is independent, pick y arbitrarily.
5. YEL = YEL + b - y
6. Repeat
```



Other Independent Set Results

Theorem: Min-reconfigurability of independent sets remains hard even for perfect graphs.

Theorem: Reconfigurability of independent sets can be determined in poly-time for

- P_4-free graphs
- line graphs

Theorem: The token jumping model is equivalent to the token addition and removal (TAR) model.

Shortest paths between shortest paths and independent sets

The End

Marcin Kamiński
Université Libre de Brussels

Paul Medvedev
University of Toronto

Martin Milanič
University of Primorska, Slovenia

International Workshop on Algorithms on Combinatorial Algorithms 2010

Reconfiguration Rules

- Token view: the vertices in your independent set are “tokens”
  - Token Sliding (TS)
    - tokens can be moved to their neighbours
  - Token Addition and Removal (TAR)
    - tokens can be added and tokens can be removed, but there must always be at least *t* tokens.
  - `ReconIndependentSet` is PSPACE-complete under both rules

SP min-Reconfiguration is NP-hard (via SAT)

- Theorem: finding the minimum reconfiguration distance is NP-hard
- widget for each variable assignment, e.g. x_1 = true
  - red edge is present if clause *i* is satisfied by x_1 = true
- the two x_i widgets are connected to the two x_{i+1} widgets