PARIS: Planning Algorithms for Reconfiguring Independent Sets

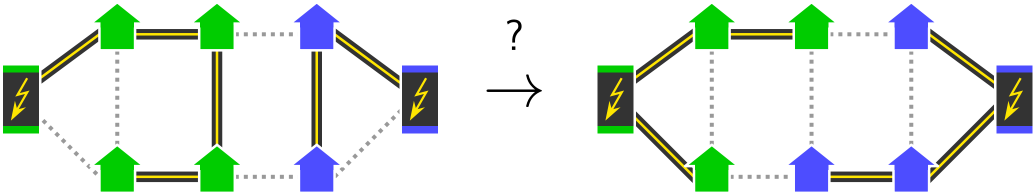
Remo Christen Salomé Eriksson Michael Katz
Christian Muise Alice Petrov Florian Pommerening
Jendrik Seipp Silvan Sievers **David Speck**

University of Basel IBM T.J. Watson Research Center Queen's University Linköping University

Combinatorial Reconfiguration

Reconfiguration Problem

Transform a solution into another solution so that **all intermediate steps are also solutions**.



Power Distribution

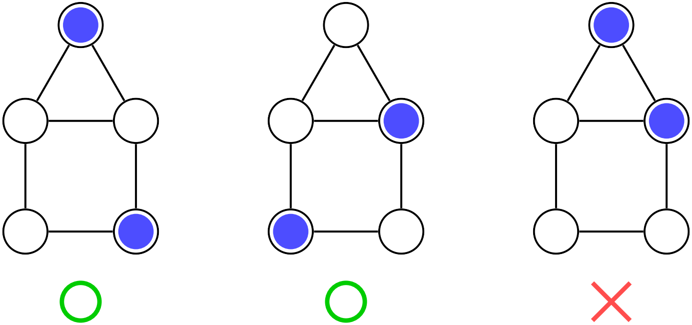
Reconfigure network while keeping all households connected.

Independent Sets

Many such **reconfiguration problems** can be cast to and analyzed using the **Independent Set Reconfiguration (ISR)** problem.

Independent Set (IS)

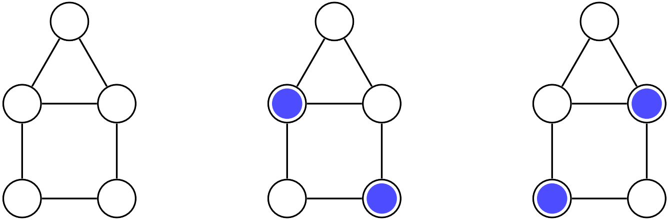
A set of vertices/nodes of a graph such that **no two are adjacent**.



Independent Set Reconfiguration

Input

- Graph
- Initial set
- Goal set

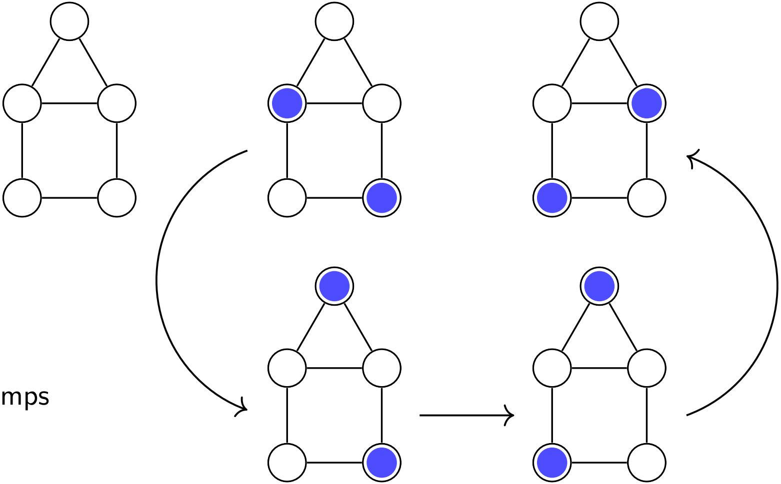


Output

- Sequence of token jumps

Input

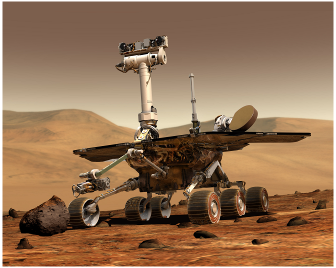
- Graph
- Initial set
- Goal set

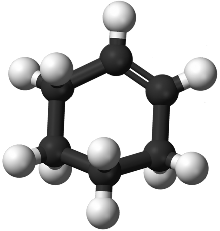


Output

- Sequence of token jumps

Classical Planning





Compact Description of State Spaces

- **State variables**: Describe the world
- **States**: Assignments to these variables
- **Actions**: Define transitions between states

$\rightsquigarrow$ **Objective**: Find a plan from an initial state to a goal state

Independent Set Reconfiguration as Classical Planning

Theoretical Contribution

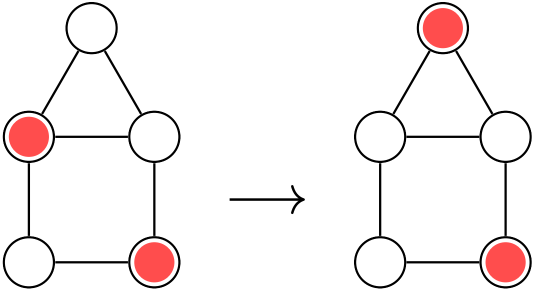
Sound, complete, and optimality preserving formulations of ISR as planning problem.

Independent Set Reconfiguration as Classical Planning

Theoretical Contribution

Sound, complete, and optimality preserving formulations of ISR as planning problem.

- **State variables**: Binary variable for each node to represent token presence
- **Single action**: Move token (IS condition encoded as precondition)



Independent Set Reconfiguration as Classical Planning

Alternative Formulation

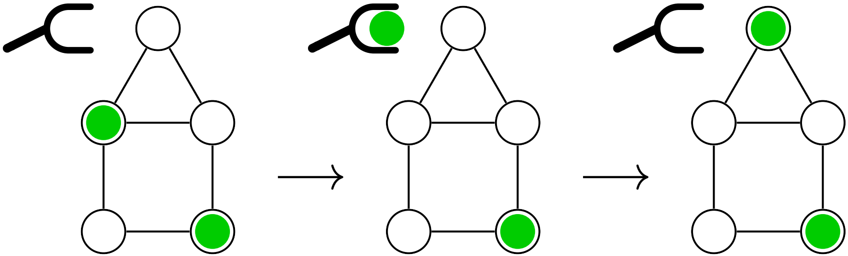
Split action representation $\rightsquigarrow$ more natural and **more compact**!

Independent Set Reconfiguration as Classical Planning

Alternative Formulation

Split action representation $\rightsquigarrow$ more natural and **more compact**!

- **State variables**: Binary variable for each node to represent token presence + **a binary variable for token holding**
- **Pick-up action**: Pick-up a token (precondition: gripper is empty)
- **Place action**: Place a token respecting IS condition



Combinatorial Reconfiguration Competition

- Toolbox of classical planning applicable
    - Search algorithms, heuristics, pruning techniques, . . .
- **1st Combinatorial Reconfiguration Competition in 2022**

Combinatorial Reconfiguration Competition

- Toolbox of classical planning applicable
  - Search algorithms, heuristics, pruning techniques, ...
- **1st Combinatorial Reconfiguration Competition in 2022**

Solver Tracks

- Existent
- Shortest
- Longest

$\times$

- Single-engine
- Portfolio

- Solutions are submitted
- All teams used **different resources**

Competition Results

	existent	shortest	longest
single-engine	1	3	1
portfolio	1 3	2	1 3

Existent Track

- Any solution + Unsolvability
- similar to *agile* IPC track + Unsolvability IPC

Competition Results

	existent	shortest	longest
single-engine	1	3	1
portfolio	1 3	2	1 3

Existent Track

- Any solution + Unsolvability
- similar to *agile* IPC track + Unsolvability IPC

PARIS

- GBFS + Landmarks (70min)

Competitors

2 Answer Set Programming

3 Greedy heuristic search + Bounded Model Checking

Competition Results

	existent	shortest	longest
single-engine	1	3	1
portfolio	1 3	2	1 3

Existent Track

- Any solution + Unsolvability
- similar to *agile* IPC track + Unsolvability IPC

PARIS

1. Counter abstraction (10s)
2. Symbolic search (70min)
3. A* + Landmarks (70min)
4. GBFS + Landmarks (70min)
5. Counter abstraction (14h)

Competitors

2 IDA* + Breadth-first search

Competition Results

	existent	shortest	longest
single-engine	1	3	1
portfolio	1 3	2	1 3

Shortest Track

- Shortest solution among competitors
- similar to *satisficing* IPC track

Competition Results

	existent	shortest	longest
single-engine	1	3	1
portfolio	1 3	2	1 3

Shortest Track

- Shortest solution among competitors
- similar to *satisficing* IPC track

PARIS

- GBFS + Landmarks (70min)

PARIS-Portfolio

1. Counter abstraction (10s)
2. Symbolic search (70min)
3. A* + Landmarks (70min)
4. GBFS + Landmarks (70min)
5. Counter abstraction (14h)

Competitors-Single

1 Answer Set Programming

Competitors-Portfolio

1 IDA* + Breadth-first search

Competition Results

	existent	shortest	longest
single-engine	1	3	1
portfolio	1 3	2	1 3

Longest Track

- Longest loopless solution among competitors
- no IPC equivalent

Competition Results

	existent	shortest	longest
single-engine	1	3	1
portfolio	1 3	2	1 3

Longest Track

- Longest loopless solution among competitors
- no IPC equivalent

PARIS

- *Symbolic top-k search* (70min)

PARIS-Portfolio

1. GBFS + Landmarks (5min)
2. *Symbolic top-k search* (65min)

Competitors

2 Answer Set Programming

Experiments with Fair Resource Allocation

How meaningful are these results since each team/approach was able to use different resources?

Experiments with Fair Resource Allocation

- Per-task comparison
- PARIS vs. other competitors:
  - Better: +
  - Worse: −

	existent		shortest		longest	
	+	−	+	−	+	−
JunKa.	168	21	158	13	217	13
Reconf.	76	0	61	0	205	1
Recongo	82	0	39	0	210	3
Telematik	19	0	17	9	202	1
Toda	111	0	198	0	210	47

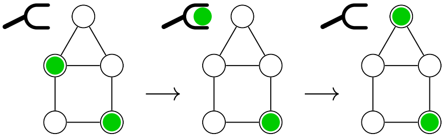
Conclusions

Summary:

- Combinatorial reconfiguration problem

$\rightsquigarrow$ Independent set reconfiguration (ISR)

- Formulations of ISR as classical planning
  - Complete, sound, optimality preserving
- State-of-the-art empirical performance



Future Work:

- **Planning techniques**: Improving our understanding of effective planning techniques for ISR
- **Strengthening Synergy**: Drawing stronger connections between the fields of reconfiguration and planning