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

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
Compact Description of State Spaces

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

- **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 ∼ more natural and more compact!
Independent Set Reconfiguration as Classical Planning

Alternative Formulation

Split action representation $\Rightarrow$ 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
- Single-engine
- Portfolio

- Solutions are submitted
- All teams used different resources
Competition Results

<table>
<thead>
<tr>
<th></th>
<th>existent</th>
<th>shortest</th>
<th>longest</th>
</tr>
</thead>
<tbody>
<tr>
<td>single-engine</td>
<td>1</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>portfolio</td>
<td>1</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

Existant Track

- Any solution + Unsolvability
- similar to agile IPC track + Unsolvability IPC
## Competition Results

<table>
<thead>
<tr>
<th></th>
<th>existent</th>
<th>shortest</th>
<th>longest</th>
</tr>
</thead>
<tbody>
<tr>
<td>single-engine</td>
<td><img src="#" alt="1" /></td>
<td><img src="#" alt="3" /></td>
<td><img src="#" alt="1" /></td>
</tr>
<tr>
<td>portfolio</td>
<td><img src="#" alt="1" /></td>
<td><img src="#" alt="3" /></td>
<td><img src="#" alt="2" /></td>
</tr>
</tbody>
</table>

### 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

<table>
<thead>
<tr>
<th>Single-engine</th>
<th>Existent</th>
<th>Shortest</th>
<th>Longest</th>
</tr>
</thead>
<tbody>
<tr>
<td>Existence</td>
<td>1</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Portfolio</td>
<td>1</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

### Existent Track

- Any solution + Unsolvability
- Similar to agile IPC track + Unsolvability IPC

### PARIS

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

### Competitors

- IDA* + Breadth-first search
**Competition Results**

<table>
<thead>
<tr>
<th></th>
<th>existent</th>
<th>shortest</th>
<th>longest</th>
</tr>
</thead>
<tbody>
<tr>
<td>single-engine</td>
<td>1</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>portfolio</td>
<td>1</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

**Shortest Track**

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

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
- Answer Set Programming

Competitors-Portfolio
- IDA* + Breadth-first search
## Competition Results

<table>
<thead>
<tr>
<th>single-engine portfolio</th>
<th>existent</th>
<th>shortest</th>
<th>longest</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

**Longest Track**

- Longest loopless solution among competitors
- No IPC equivalent
Competition Results

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
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: –

<table>
<thead>
<tr>
<th></th>
<th>existent</th>
<th>shortest</th>
<th>longest</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>+</td>
<td>–</td>
<td></td>
</tr>
<tr>
<td>JUNKA.</td>
<td>168  21</td>
<td>158   13</td>
<td>217  13</td>
</tr>
<tr>
<td>RECONF.</td>
<td>76   0</td>
<td>61    0</td>
<td>205  1</td>
</tr>
<tr>
<td>RECONGO</td>
<td>82   0</td>
<td>39    0</td>
<td>210  3</td>
</tr>
<tr>
<td>TELEMATIK</td>
<td>19   0</td>
<td>17    9</td>
<td>202  1</td>
</tr>
<tr>
<td>TODA</td>
<td>111   0</td>
<td>198   0</td>
<td>210  47</td>
</tr>
</tbody>
</table>
Conclusions

Summary:
• Combinatorial reconfiguration problem $\leadsto$ 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