Constraint Graphs

Machine = graph, red & blue edges
**Constraint Graphs**

Machine state
= orientation

constraint graph
Constraint Logic

Rule: at least 2 units incoming at a vertex

Move: reverse an edge, preserving Rule
Nondeterministic Constraint Logic (NCL)

can you reverse this edge?
PSPACE-complete
AND vertex

not your usual AND gate!

Rule: at least 2 units incoming at a vertex
**SPLIT vertex**

Rule: at least 2 units incoming at a vertex.
**OR vertex**

*Rule:* at least 2 units incoming at a vertex.

Not your usual OR gate!
CHOICE vertex
Wiring Vertices Together

AND wants red
OR wants blue
Red-Blue Conversion

assume an even number of conversions
Red-Blue Conversion

assume an even number of conversions
CNF Formula

\((w \lor x \lor y) \land (w \lor \overline{x} \lor z) \land (x \lor \overline{y} \lor z)\)
Wire Terminators

unconstrained blue terminator

forced-inward blue terminator

unconstrained red terminator
\((w \lor x \lor y) \land (w \lor \bar{x} \lor z) \land (x \lor \bar{y} \lor z)\)
Constraint Graph
Satisfaction is NP-complete
Nondeterministic Constraint Logic (NCL)

can you reverse this edge?

PSPACE-complete
Quantified Boolean Formulas (QBF/QSAT)

\[ \forall x \exists y \forall w \cdots \exists z [ (x \lor y) \land \cdots \land (\overline{z} \lor x \lor \overline{w}) ] \]
Quantifiers

existential

universal
Latch

unlocked
Existential Quantifier
Universal Quantifier
Quantified Boolean Formulas (QBF/QSAT)

$$\forall x \exists y \forall w \ldots \exists z \left[ (x \lor y) \land \ldots \land (\overline{z} \lor x \lor \overline{w}) \right]$$
Crossover Gadget
Vertex with 4 Red Edges
Grid Constraint Graphs: Straights & Turns

- straight
- turn
- OR
OR from Protected OR
Ma’s Puzzle

L’Ane Rouge

7. Sliding-Block Puzzles

Dad’s Puzzle (1926)

Sam Loyd’s 15 Puzzle
Sliding-Block Puzzles

Dad's Puzzler
Few Solve It
It Can Be Done

Place the blocks in box as per diagram. The puzzle is to move the big square block from corner A to corner C without jumping or raising any block from bottom of the box or turning any piece.

"Take One Home"

1926
83 moves
Sliding-Block Puzzles
[Hearn & Demaine 2002]

(a) AND

(b) OR

PSPACE-complete
Sliding-Block Puzzles
[Hearn & Demaine 2002]

(a) AND

(b) Protected OR

PSPACE-complete
Sliding Tokens
(Reconfiguration Independent Set)

(a) AND

(b) OR
Rush Hour is PSPACE-complete
[Flake & Baum 2002; Hearn & Demaine 2002]
Rush Hour is PSPACE-complete

[Flake & Baum 2002; Hearn & Demaine 2002]
Rush Hour

[Tromp & Cilibrasi 2008]

2 ANDs

2 protected ORs
Open: 1 × 1 Rush Hour
[Tromp & Cilibrasi 2008]

- P or PSPACE-complete or ...?
Triangular Rush Hour
[Hearn & Demaine 2009]

PSPACE-complete

(a) AND vertex
(b) Connector
(c) OR vertex
Hinged Dissection

[Dudeney 1902]
Polygons of equal area have a hinged dissection that folds continuously without self-intersection.
Hinged Dissection Motion
[Hearn & Demaine 2009]

(c) Hinged slider

PSPACE-complete
# Pushing $1 \times 1$ Blocks Complexity

<table>
<thead>
<tr>
<th>Name</th>
<th>Push</th>
<th>Fixed</th>
<th>Slide</th>
<th>Goal</th>
<th>Complexity</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Push-$k$</td>
<td>$k \geq 1$</td>
<td>no</td>
<td>min</td>
<td>path</td>
<td>NP-hard</td>
<td>D, D, O’Rourke 2000</td>
</tr>
<tr>
<td>Push-*</td>
<td>$\infty$</td>
<td>no</td>
<td>min</td>
<td>path</td>
<td>NP-hard</td>
<td>Hoffmann 2000</td>
</tr>
<tr>
<td>PushPush-$k$</td>
<td>$k \geq 1$</td>
<td>no</td>
<td>max</td>
<td>path</td>
<td>PSPACE-complete</td>
<td>D, Hoffmann, Holzer 2004</td>
</tr>
<tr>
<td>PushPush-*</td>
<td>$\infty$</td>
<td>no</td>
<td>max</td>
<td>path</td>
<td>NP-hard</td>
<td>Hoffmann 2000</td>
</tr>
<tr>
<td>Push-1F</td>
<td>1</td>
<td>yes</td>
<td>min</td>
<td>path</td>
<td>NP-hard</td>
<td>DDO 2000</td>
</tr>
<tr>
<td>Push-$kF$</td>
<td>$k \geq 2$</td>
<td>yes</td>
<td>min</td>
<td>path</td>
<td>PSPACE-complete</td>
<td>D, Hearn, Hoffmann 2002</td>
</tr>
<tr>
<td>Push-*F</td>
<td>$\infty$</td>
<td>yes</td>
<td>min</td>
<td>path</td>
<td>PSPACE-complete</td>
<td>Bremner, O’Rourke, Shermer 1994</td>
</tr>
<tr>
<td>Push-$kX$</td>
<td>$k \geq 1$</td>
<td>no</td>
<td>min</td>
<td>simple path</td>
<td>NP-complete</td>
<td>D, Hoffmann 2001</td>
</tr>
<tr>
<td>Push-*X</td>
<td>$\infty$</td>
<td>no</td>
<td>min</td>
<td>simple path</td>
<td>NP-complete</td>
<td>Hoffmann 2000</td>
</tr>
<tr>
<td>Sokoban</td>
<td>1</td>
<td>yes</td>
<td>min</td>
<td>storage</td>
<td>PSPACE-complete</td>
<td>Culberson 1998</td>
</tr>
</tbody>
</table>
Sokoban
[Thinking Rabbit, Hiroyuki Imabayashi, 1982]
Sokoban is PSPACE-complete

[Culberson 1998; Hearn & Demaine 2002]

(a) AND

(b) OR

(c) Utility gadgets
Push-2F
[Demaine, Hearn, Hoffmann 2002]
Rolling Block Mazes
Rolling Block Mazes PSPACE-complete

[Holzer & Jakobi 2012]

AND

OR

effectively immovable 1 × 1 × 2 blocks
Rolling Block Mazes PSPACE-complete

[Holzer & Jakobi 2012]

Effectively immovable 1 × 1 × 2 blocks

Rightward edge

Leftward edge

Shift
**Plank Puzzles**

[Andrea Gilbert; ThinkFun]

**HOW TO PLAY:**

1. **Hiker STARTS TO CROSS.**

2. **Hiker MOVES the medium plank TO A NEW POSITION.**

3. **Hiker MOVES the small plank TO A NEW POSITION.**

4. **Phew! I Made it!**

   - Hiker MOVES the medium plank and makes it across!
Plank Puzzles
[Hearn 2004]
Plank Puzzles
[Hearn 2004]
Dynamic Map Labeling
[Buchin & Gerrits 2013]
Partial Searchlight Scheduling
[Viglietta 2013]