# Optimal Planning with Global Numerical State Constraints


Franc Ivankovic, Patrik Haslum and Sylvie Thiebaux
Australian National University &
NICTA Optimisation Research Group

Vikas Shivashankhar and Dana Nau
University of Maryland at College Park

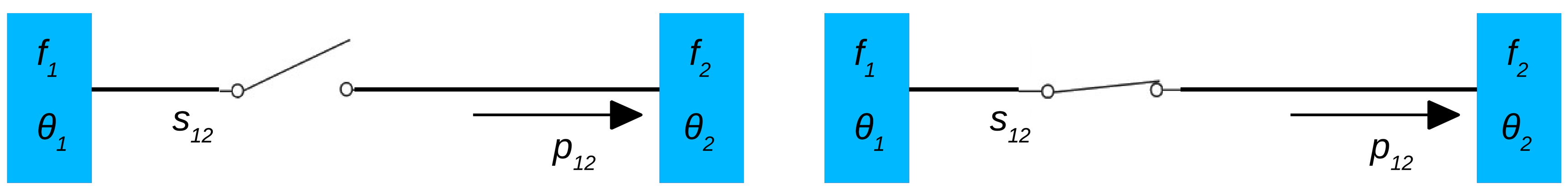

NICTA

## Planning with Global Constraints

- Classical planning problems with **non-classical state constraints**, e.g.,
  - Systems of (linear or non-linear) equations/inequalities;
  - PDDL axioms (stratified logic programs);
  - Qualitative spatial algebra; etc.
- Each constraint type is supported by effective solvers, but these are not always suited to dealing with the planning part of the problem.
- How to combine classical planning techniques with external constraint solvers?

## Formalism

- Variables: **primary** (finite domain) and **secondary** (used in constraints).
- **Actions** assign only primary variables.
- **Switched constraints**: $\varphi \to \gamma$
  - $\varphi$ (trigger) over primary and $\gamma$ over secondary variables.
  - $\varphi \to \gamma$ is **active** in $s$ iff $s \models \varphi$.



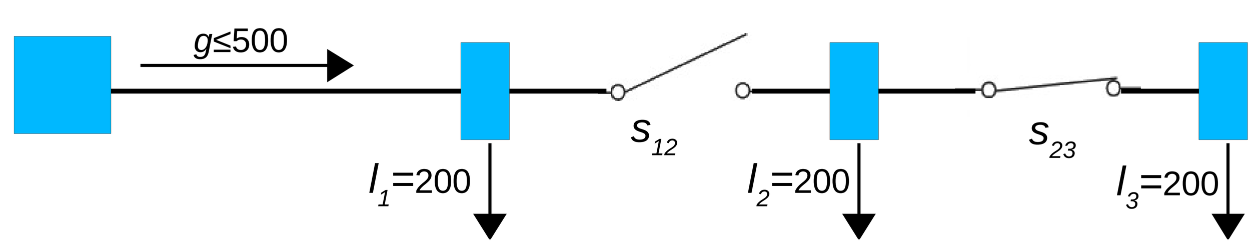

$s_{12} = \text{open} \to p_{12} = 0$

$s_{12} = \text{closed} \to p_{12} = -b_{12}(\theta_1 - \theta_2)$
$s_{12} = \text{closed} \to f_1 = f_2$

- Invariant constraints ($C_{\text{inv}}$)
  - A state is **valid** iff the set of active invariant constraints is satisfiable.
- Goal constraints ($G_S$)
  - Goal achieved only when jointly satisfiable with active invariant constraints.
- Secondary preconditions ($\text{pre}_S(a)$)
  - $a$ applicable only when jointly satisfiable with active invariant constraints.

## Extending the Delete Relaxation

- Value-accumlating relaxation of primary variables – equivalent to classical delete relaxation.
- $\varphi \to \gamma$ is active in relaxed state $s^+$ iff $\varphi$ cannot be False in $s^+$.




| $s_{12}$ | $= \{ \text{open} \}$ |
|---|---|
| $s_{23}$ | $= \{ \text{closed} \}$ |
| $s_{12} = \text{o}$ | $\to p_{12} = 0$ |
| $s_{12} = \text{c}$ | $\to p_{12} = -b_{12}(\theta_1 - \theta_2)$ |
| $s_{12} = \text{c}$ | $\to f_1 = f_2$ |
| $s_{23} = \text{o}$ | $\to p_{23} = 0$ |
| $s_{23} = \text{c}$ | $\to p_{23} = -b_{23}(\theta_2 - \theta_3)$ |
| $s_{23} = \text{c}$ | $\to f_2 = f_3$ |
| True | $\to f_1 = 1.0$ |
| True | $\to g \leq g_{max}$ |
| True | $\to g = l_1 f_1 + p_{12}$ |
| True | $\to p_{12} = l_2 f_2 + p_{23}$ |
| True | $\to p_{23} = l_3 f_3$ |
| True | $\to f_2 = 1.0$ |

Open($s_{23}$) $\to$

| $s_{12}$ | $= \{ \text{open} \}$ |
|---|---|
| $s_{23}$ | $= \{ \text{closed, open} \}$ |
| $s_{12} = \text{o}$ | $\to p_{12} = 0$ |
| $s_{12} = \text{c}$ | $\to p_{12} = -b_{12}(\theta_1 - \theta_2)$ |
| $s_{12} = \text{c}$ | $\to f_1 = f_2$ |
| $s_{23} = \text{o}$ | $\to p_{23} = 0$ |
| $s_{23} = \text{c}$ | $\to p_{23} = -b_{23}(\theta_2 - \theta_3)$ |
| $s_{23} = \text{c}$ | $\to f_2 = f_3$ |
| True | $\to f_1 = 1.0$ |
| True | $\to g \leq g_{max}$ |
| True | $\to g = l_1 f_1 + p_{12}$ |
| True | $\to p_{12} = l_2 f_2 + p_{23}$ |
| True | $\to p_{23} = l_3 f_3$ |
| True | $\to f_2 = 1.0$ |

Close($s_{12}$) $\to$

| $s_{12}$ | $= \{ \text{open, closed} \}$ |
|---|---|
| $s_{23}$ | $= \{ \text{closed, open} \}$ |
| $s_{12} = \text{o}$ | $\to p_{12} = 0$ |
| $s_{12} = \text{c}$ | $\to p_{12} = -b_{12}(\theta_1 - \theta_2)$ |
| $s_{12} = \text{c}$ | $\to f_1 = f_2$ |
| $s_{23} = \text{o}$ | $\to p_{23} = 0$ |
| $s_{23} = \text{c}$ | $\to p_{23} = -b_{23}(\theta_2 - \theta_3)$ |
| $s_{23} = \text{c}$ | $\to f_2 = f_3$ |
| True | $\to f_1 = 1.0$ |
| True | $\to g \leq g_{max}$ |
| True | $\to g = l_1 f_1 + p_{12}$ |
| True | $\to p_{12} = l_2 f_2 + p_{23}$ |
| True | $\to p_{23} = l_3 f_3$ |
| True | $\to f_2 = 1.0$ |

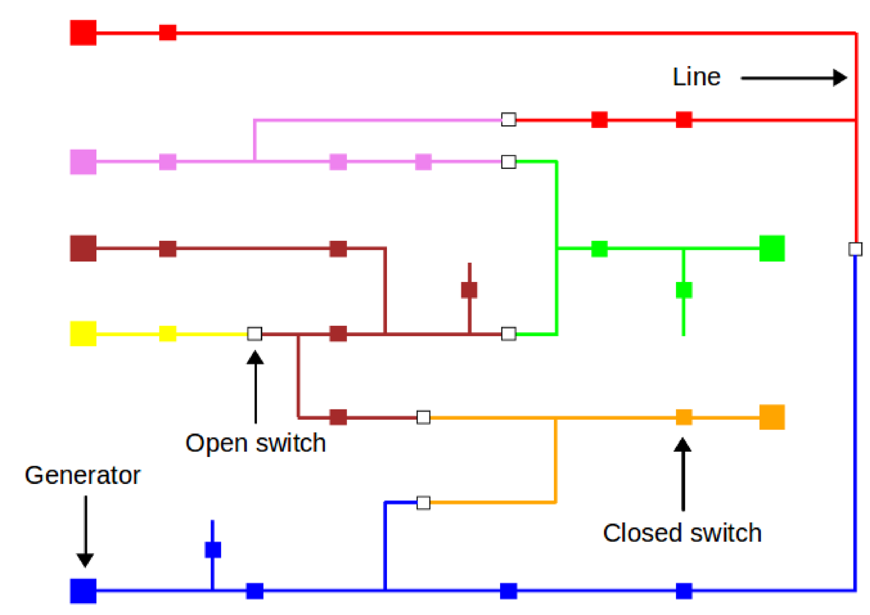
- RPG-like construction provides a (polynomial) relaxed reachability test.
- Combined with landmark-based algorithm (Haslum, Slaney & Thiébaux, 2012) to compute $h^+$ heuristic.

## Related work

- Combining a temporal planner with AC powerflow solver for power balancing (Piacentini et al, 2013).
- Planning Modulo Theories (Gregory et al, 2012).
- Domain-predictive control (Löhr et al, 2012).

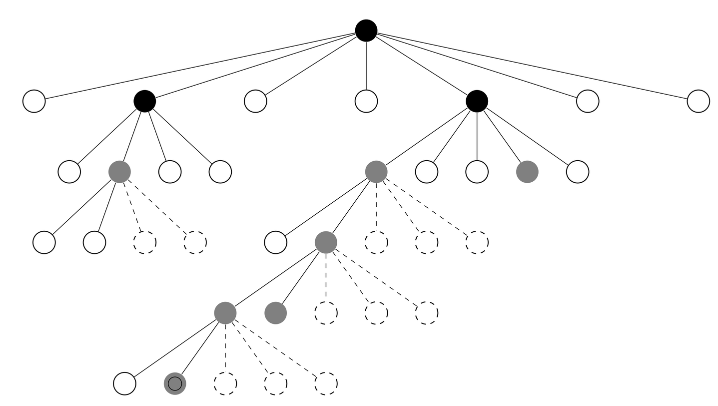
## Example - power network reconfiguration

- Actions control:
  - Switch states (discrete); and
  - Generator output (continuous).
- Subject to constraints:
  - Physical laws governing power flow.
    - AC model (most accurate, computationally expensive).
    - Linearised DC model.
    - Intermediate approxiations.
  - Generator and line capacity limits.
  - Not feeding faults.
- A single discrete action can affect power flow across the entire network.
- Before taking any action, we need to ensure the resulting state does not violate any constraint.




## Preferred actions in A* search

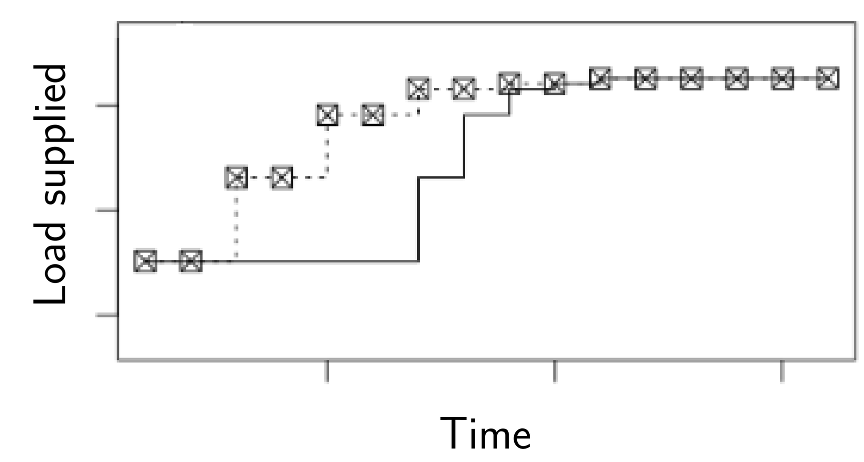
- **Preferred actions** are widely used in non-optimal planning as a way of gaining more information from computationally expensive heuristics – such as $h^+$.
- Combined with **partial expansion A***.
- Initally generate only preferred successors, but keep parent state on queue.
- Tie-break in favour of preferred states when $f$-values are equal.
- Can **avoid generating** a (substantial) fraction of nodes in the $f^*$ layer.



- Effective when heuristic is computationally **expensive** but **accurate**, and states have **many successors**.
  - In our experiments: 42.8% reduction in state evaluations.

## State-dependent action costs

- The objective in power restoration tasks is to resupply as much load as quickly as possible – maximise the "area under the curve".
- Each action's cost equals currently unsupplied load.




- Sum of conditional constant costs:

$$\text{cost}(a, s) = c_0 + \begin{cases} c_1 & \text{if } s \models \varphi_1 \\ 0 \end{cases} + \ldots + \begin{cases} c_k & \text{if } s \models \varphi_k \\ 0 \end{cases}$$

- Little, if any, work on optimal planning with state-dependent action costs.
- Straightforward, but exponential-size, compilation to constant action costs.
- Adapted $h^+$ heuristic, using an incremental compilation scheme.
  - Not effective: $\times 6.85$ average slow-down, and reduced heuristic accuracy.

## Future work

- Apply framework to other types of constraints (e.g., PDDL axioms).
- Integrate with other classical planning heuristics and techniques.
- Evaluate usefulness of PrefPEA* in classical planning (e.g., with LM-Cut, landmark heuristic with optimal cost partitioning, or LP-based heuristics).