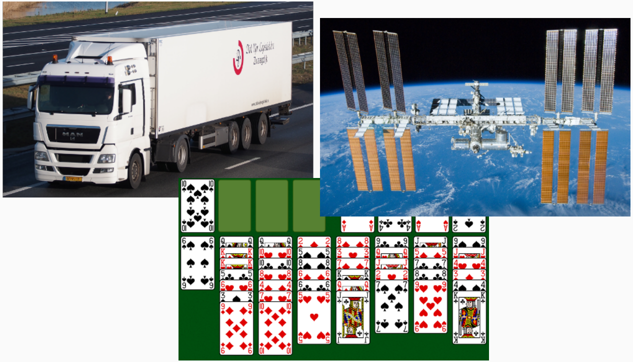
Sensitivity Analysis for Saturated Post-hoc Optimization

Paul Höft, David Speck, Jendrik Seipp

October 4, 2023

ECAI 2023

Optimal Classical Planning



- Find cost-optimal action sequence that achieves goal
- Deterministic, fully observable, domain-independent
- A* search with admissible heuristic

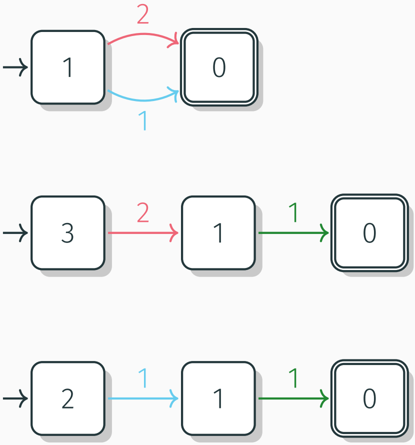
Abstractions

Abstraction

Simplified statespace from abstraction function $\alpha : S \to S^\alpha$

Use multiple different abstractions: $H = \langle h_{\alpha_1}, \ldots, h_{\alpha_n} \rangle$ as heuristic

Addition of abstraction heuristics not admissible

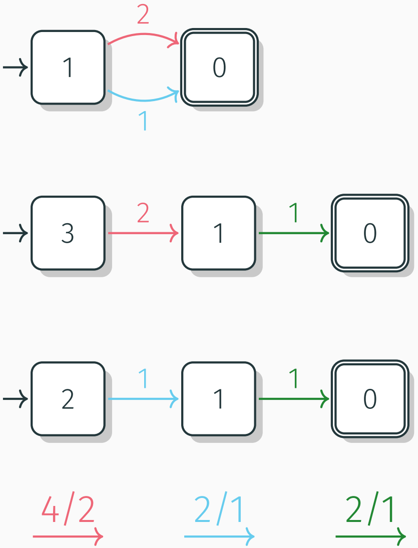


Cost Partitioning

Additive Abstractions

$H = \langle h_1, \ldots, h_n \rangle$ is additive if $\sum_i^n cost_i(o) \leq cost(o)$ for all $o \in O$

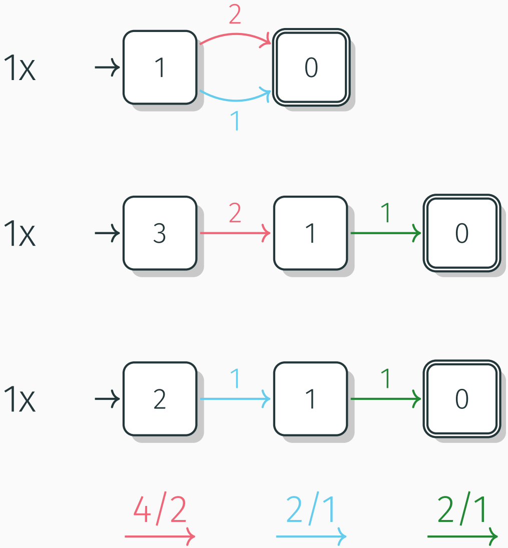
Redistribute costs for abstractions: Cost partitioning



PhO Heuristic

Scale abstractions to obey additivity constraints:

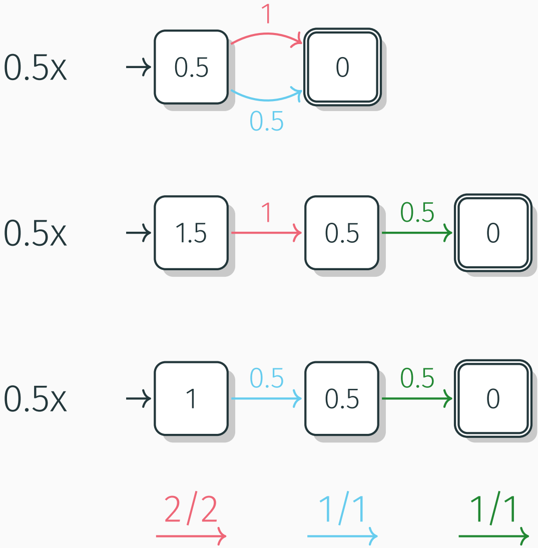
$\sum_i^n cost_i(o) \leq cost(o)$ for all $o \in O$



PhO Heuristic

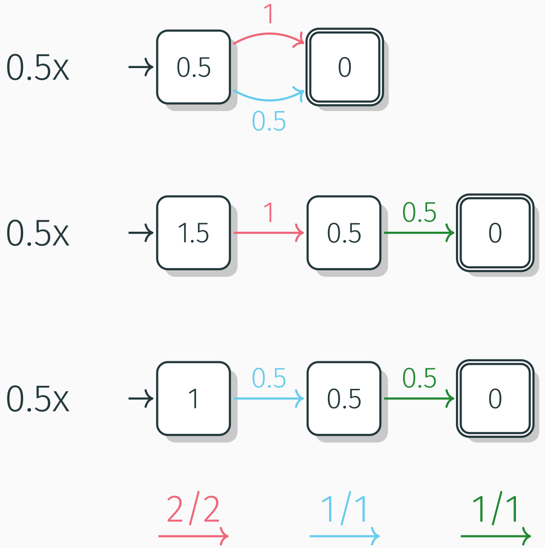
Scale abstractions to obey additivity constraints:

$\sum_i^n cost_i(o) \leq cost(o)$ for all $o \in O$



PhO Heuristic

H	a set of abstraction heuristics h
$h(s)$	abstract goal distance of state s in abstraction h
$cost_h(o)$	cost of operator o in heuristic h
o affects h	operator appears on a state-changing edge in h



PhO Heuristic

PhO LP for state s

$$\text{maximize} \sum_{h \in H} h_i(s) \cdot w_h \text{ s.t.}$$

$$\sum_{h \in H : o \text{ affects } h} cost(o) \cdot w_h \leq cost(o) \text{ for all } o \in O$$

$$w_h \geq 0 \text{ for all } h \in H$$

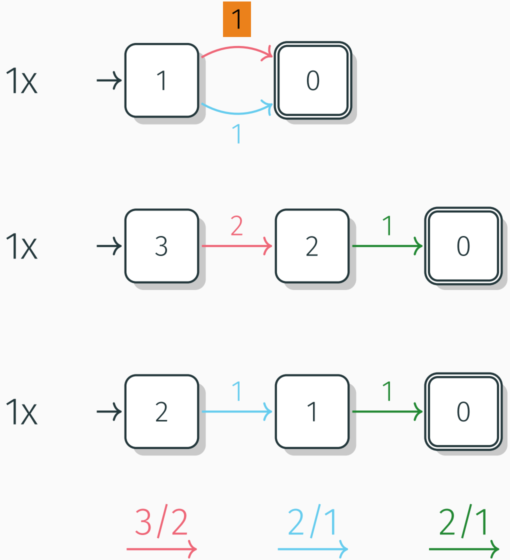
SPhO Heuristic

SPhO Linear Program

$$\text{minimize} \sum_{o \in O} cost(o) \cdot Y_o \text{ s.t.}$$

$$\sum_{o \in O} mscf_h(o) \cdot Y_o \geq h(s) \text{ for all } h \in H$$

$$Y_o \geq 0 \text{ for all } o \in O$$



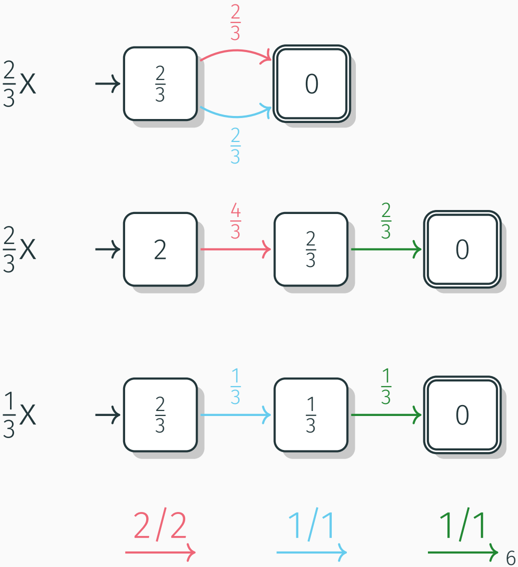
SPhO Heuristic

SPhO Linear Program

$$\text{minimize} \sum_{o \in O} cost(o) \cdot Y_o \text{ s.t.}$$

$$\sum_{o \in O} mscf_h(o) \cdot Y_o \geq h(s) \text{ for all } h \in H$$

$$Y_o \geq 0 \text{ for all } o \in O$$



Lazy SPhO

- LP computations expensive
- cost scaling cheap to evaluate

$\rightarrow$ reduce LP computations by re-using cost partitionings

Lazy SPhO

h_s^{SPhO} only optimized for state s

$h_s^{SPhO}(s') <= h_{s'}^{SPhO}(s')$

Cover Rule

h_s^{SPhO} covers s' if $h_s^{SPhO}(s') = h_{s'}^{SPhO}(s')$

no approximation

$Sols \leftarrow \emptyset$

function LAZY_SPHO(s)

  if COVERS(sol, s) any $sol \in Sols$ **then**

    return ADAPT(sol, s)

  $\langle sol, h \rangle \leftarrow$ solve SPhO LP for s

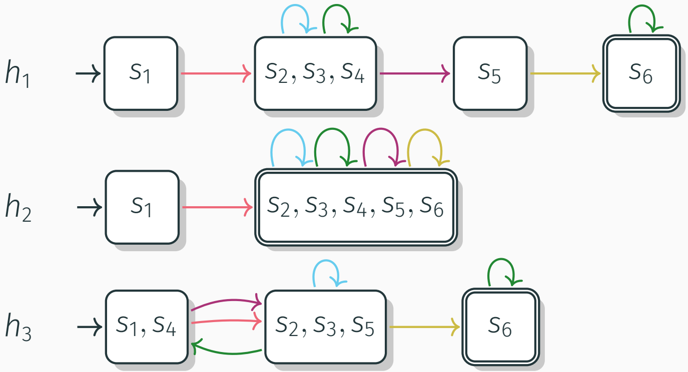
  $Sols \leftarrow Sols \cup \{sol\}$

  return h

Example

SPhO Constraints:

$$Y_{o1} + Y_{o4} + Y_{o5} \geq h_1(s)$$
$$Y_{o1} \geq h_2(s)$$
$$Y_{o1} + Y_{o4} + Y_{o5} \geq h_3(s)$$



Cover Rule: Grouped

$$Y_{o1} + Y_{o4} + Y_{o5} \geq \max(h_1(s), h_3(s))$$
$$Y_{o1} \geq h_2(s)$$

	s_1	s_2	s_3	s_4	s_5
$\max(h_1, h_3)$	3	2	2	2	1
h_2	1	0	0	0	0

Cover Rule: Grouped

h_s^{SPhO} covers s' if $\langle H_1(s), \ldots, H_m(s)\rangle = \langle H_1(s'), \ldots, H_m(s')\rangle$

$$
\begin{aligned}
\text{minimize } & Y_{o1} + Y_{o3} + Y_{o4} + Y_{o5} \text{ st} \\
Y_{o1} + Y_{o4} + Y_{o5} &\geq \max(h_1(s), h_3(s)) \\
Y_{o1} &\geq h_2(s)
\end{aligned}
$$

Optimal basis still optimal after changes
$\rightarrow$ Sensitivity Analysis

Cover Rule: 100%

	s_1	s_2	s_3	s_4	s_5
$\max(h_1, h_3)$	3	2	2	2	1
h_2	1	0	0	0	0

	SA	
$h(s_1)$	L	U
$\max(h_1, h_3) = 3$	1	∞
$h_1 = 1$	$-\infty$	3

Cover Rule: 100%

h_s^{SPhO} covers s' if the $\sum_i^m \Delta H_j \leq 1$

Generalization of all other rules

Results

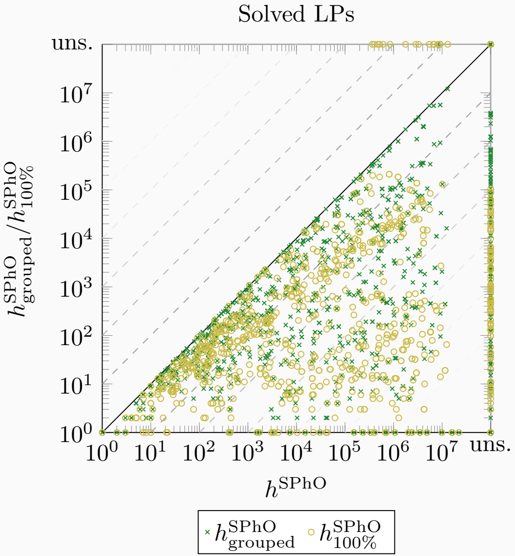
Tasks optimal IPC tasks from 1998-2018

Time 30min

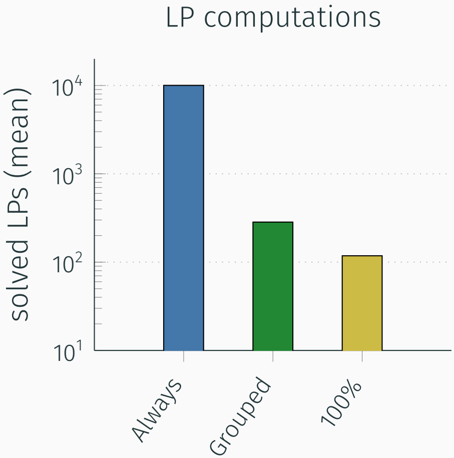
Memory 8G

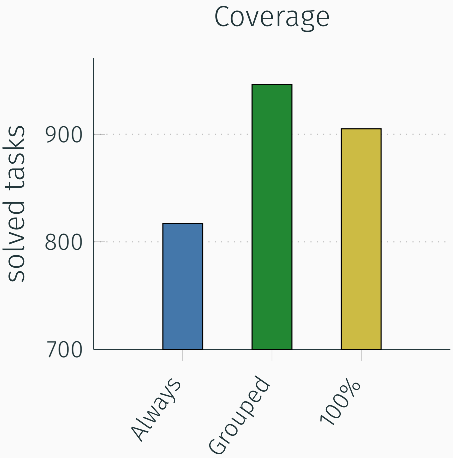
Abstractions systematic PDBs size 2

LP Solver CPLEX



Results





Grouped: $O(1)$, Percent: $O(n)$

Future Work

Lazy SPhO:

- Bound storage size for LPs $\rightarrow$ only consider most recent or most useful

Sensitivity Analysis:

- Apply to other structurally fixed heuristics: Potential Heuristics
- Sensitivity Analysis possible on rows or columns of the constraint matrix

LP solutions give other useful information: Reduced Cost, basic constraints, ...

- Identify useful patterns
- Abstraction refinement