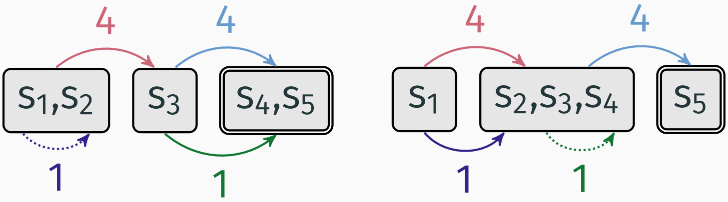
Cost Partitioning Algorithms for PDB Heuristics

Jendrik Seipp

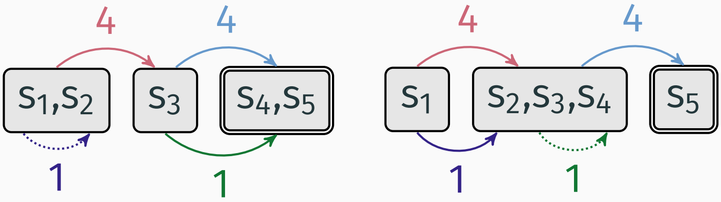
October 4, 2018

University of Basel

Example Projections



Cost Partitioning Algorithms



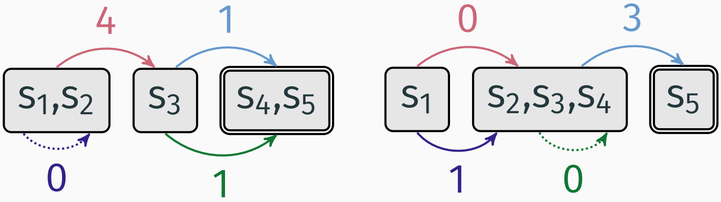
$$h(s_1) = ?$$

OCP

- cost partitioning with highest heuristic value for a given state among all cost partitionings

- computable in polynomial time for abstractions
- too expensive in practice

Optimal Cost Partitioning



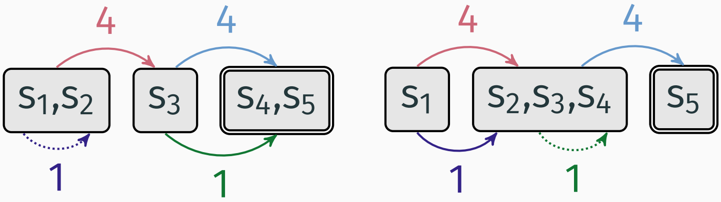
$$h(s_1) = 5 + 3 = 8$$

OCP

- cost partitioning with highest heuristic value for a given state among all cost partitionings

- computable in polynomial time for abstractions
- too expensive in practice

Post-hoc Optimization



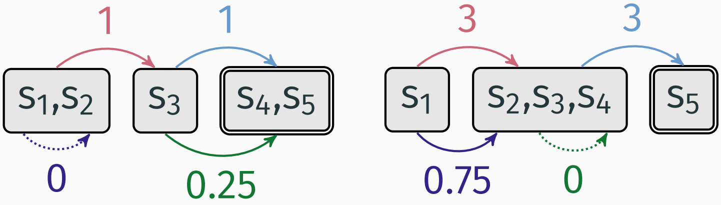
$$h(s_1) = ?$$

PhO

- compute weights $0 \leq w \leq 1$ by maximizing $\sum_{i=1}^{n} w_i \cdot h_i(s)$
- for each operator: sum of relevant heuristic factors ≤ 1

  $w_1 + w_2 \leq 1$, $w_1 + w_2 \leq 1$, $w_2 \leq 1$, $w_1 \leq 1$
- use costs $w_i \cdot cost(o)$ if o is relevant for h_i, otherwise 0

Post-hoc Optimization

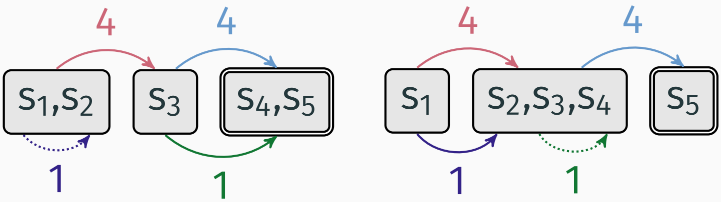


$w_1 = 0.25,\ w_2 = 0.75 \rightarrow h(s_1) = 1.25 + 3.75 = 5$

PhO

- compute weights $0 \leq w \leq 1$ by maximizing $\sum_{i=1}^{n} w_i \cdot h_i(s)$
- for each operator: sum of relevant heuristic factors ≤ 1
  $w_1 + w_2 \leq 1$, $w_1 + w_2 \leq 1$, $w_2 \leq 1$, $w_1 \leq 1$
- use costs $w_i \cdot cost(o)$ if o is relevant for h_i, otherwise 0

Greedy Zero-one Cost Partitioning

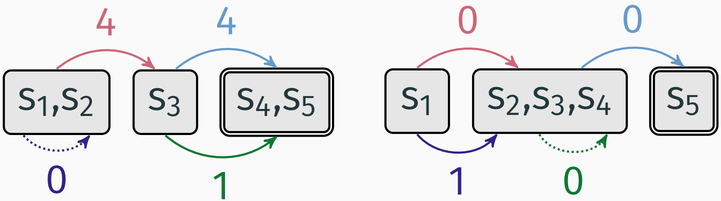


$$h(s_1) = ?$$

GZOCP

- order heuristics
- use full costs for the first relevant heuristic

Greedy Zero-one Cost Partitioning

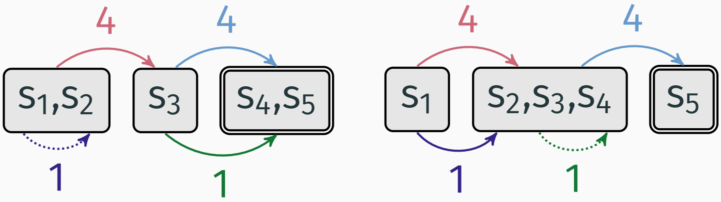


$$h(s_1) = 5 + 0 = 5$$

GZOCP

- order heuristics
- use full costs for the first relevant heuristic

Saturated Cost Partitioning

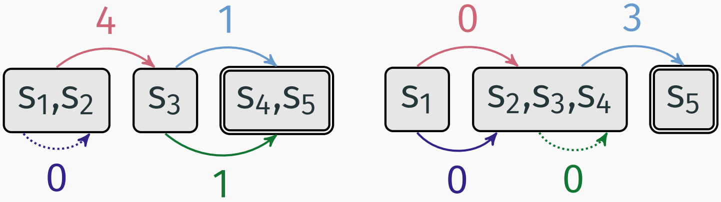


$h(s_1) = ?$

SCP

- order heuristics
- for each heuristic h:
    - use minimum costs preserving all estimates for h
    - use remaining costs for subsequent heuristics

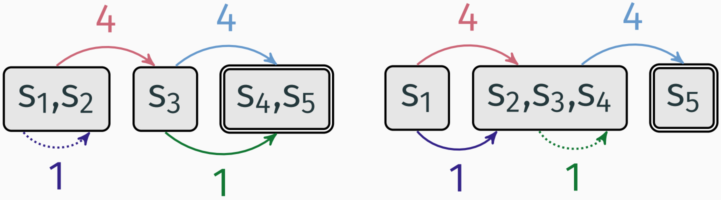
Saturated Cost Partitioning



$$h(s_1) = 5 + 3 = 8$$

SCP

- order heuristics
- for each heuristic h:
  - use minimum costs preserving all estimates for h
  - use remaining costs for subsequent heuristics

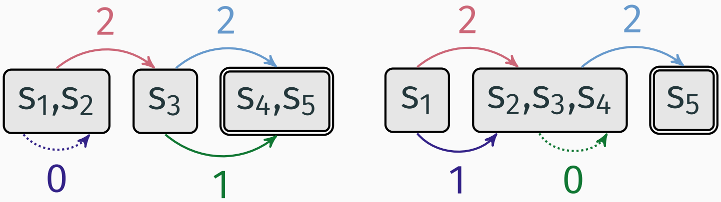


$$h(s_1) = ?$$

UCP

- distribute costs uniformly among relevant heuristics

Uniform Cost Partitioning

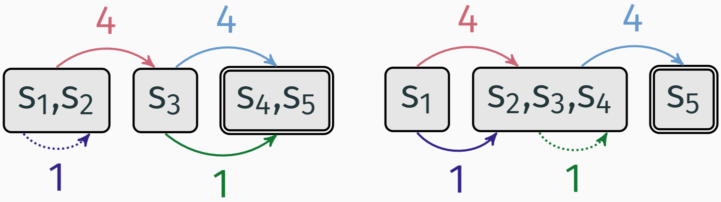


$$h(s_1) = 3 + 3 = 6$$

UCP

- distribute costs uniformly among relevant heuristics

Opportunistic Uniform Cost Partitioning

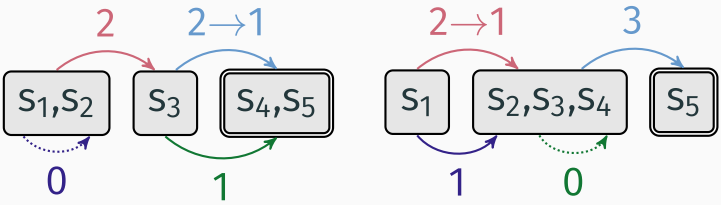


$$h(s_1) = ?$$

OUCP

- order heuristics
- for each heuristic h:
  - distribute costs uniformly among h and other relevant remaining heuristics
  - use saturated costs for h
  - use remaining costs for subsequent heuristics

Opportunistic Uniform Cost Partitioning

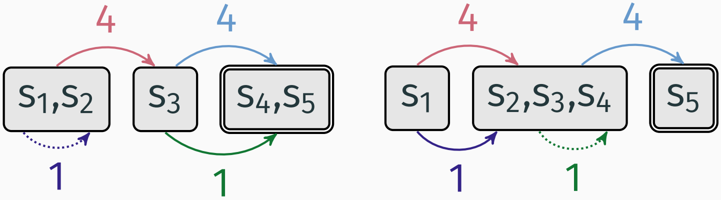


$$h(s_1) = 3 + 4 = 7$$

OUCP

- order heuristics
- for each heuristic h:
    - distribute costs uniformly among h and other relevant remaining heuristics
    - use saturated costs for h
    - use remaining costs for subsequent heuristics

Canonical Heuristic

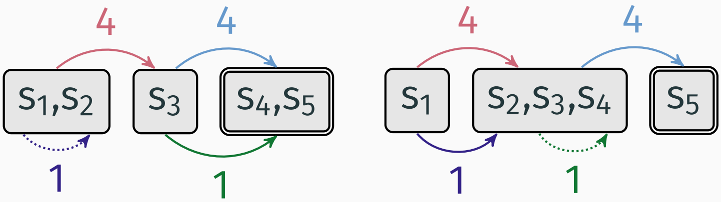


$$h(s_1) = ?$$

CAN

- compute independent PDB subsets
- compute maximum over sums

Canonical Heuristic



$$h(s_1) = \max(5, 5) = 5$$

CAN

- compute independent PDB subsets
- compute maximum over sums

Theoretical Comparison

Dominances and Non-Dominances

SCP

GZOCP

PhO

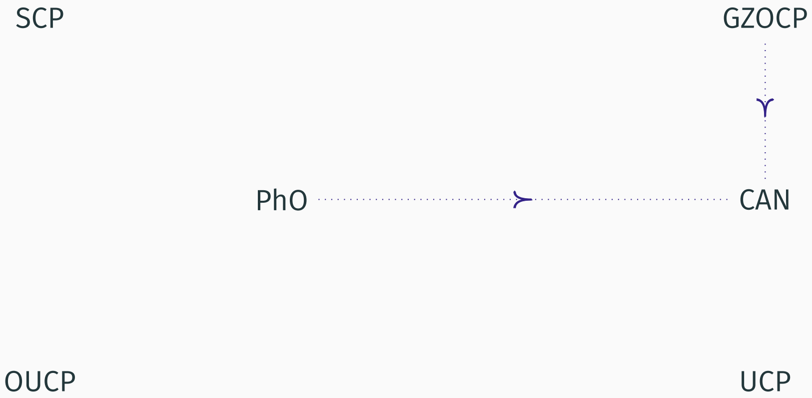
CAN

OUCP

UCP

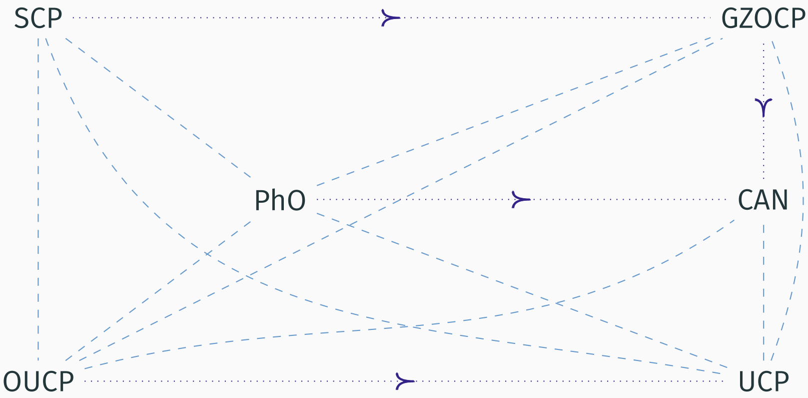
Dominances and Non-Dominances

Dominances and Non-Dominances



Dominances and Non-Dominances

Dominances and Non-Dominances



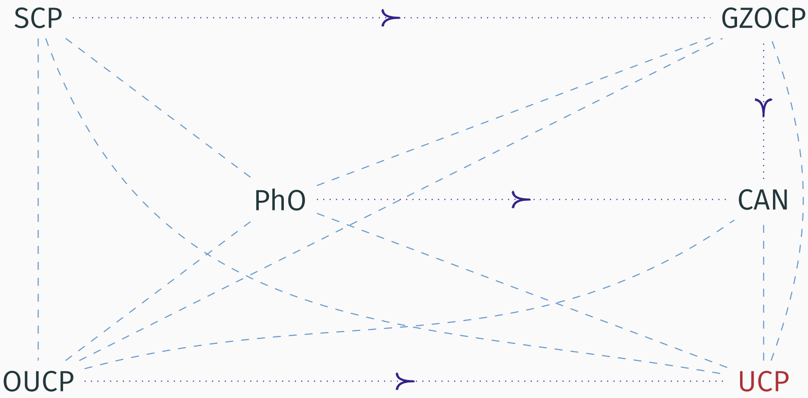
Experimental Comparison

Experimental Comparison: Systematic PDBs

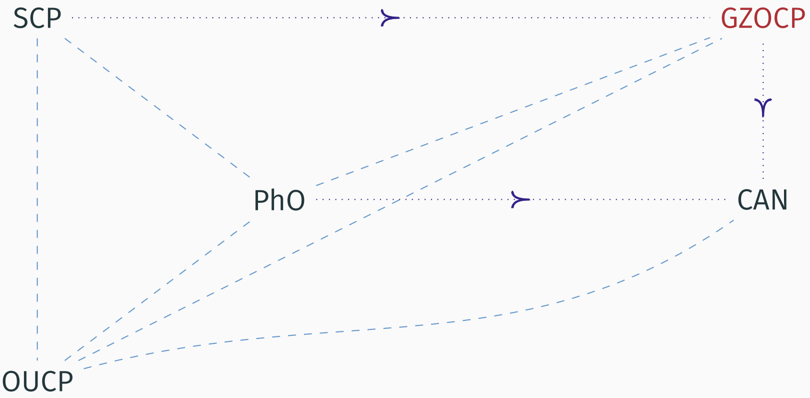
Systematic PDBs:

- PDBs for all interesting patterns of sizes 1 and 2

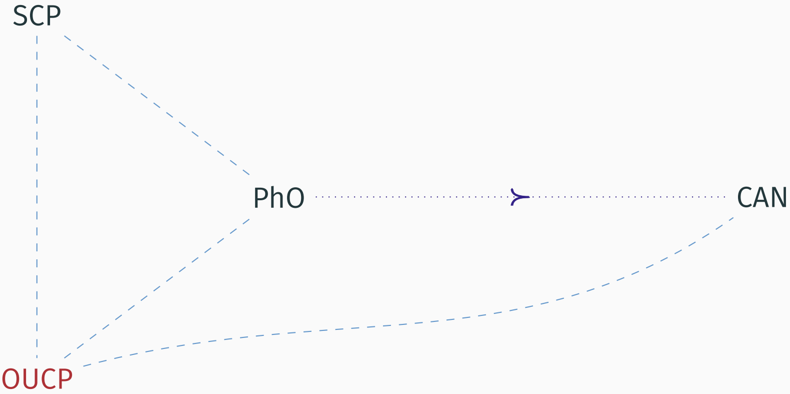
Experimental Comparison: Systematic PDBs



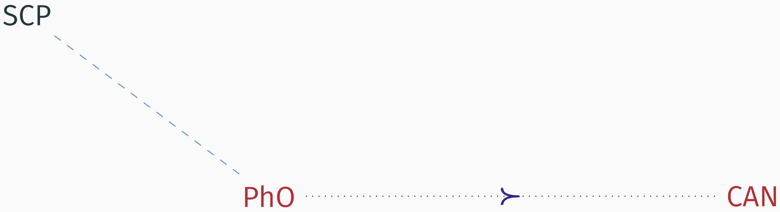
Experimental Comparison: Systematic PDBs



Experimental Comparison: Systematic PDBs



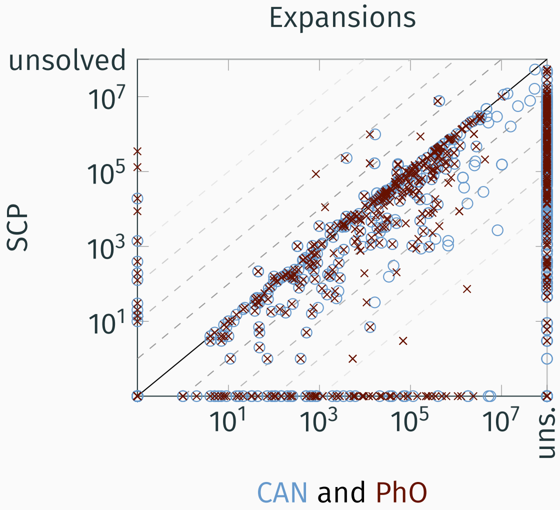
Experimental Comparison: Systematic PDBs



Experimental Comparison: Systematic PDBs

SCP

Experimental Comparison: Systematic PDBs



Discussion of Results

To obtain strongest cost-partitioned heuristic:

- reuse unused costs
- assign costs greedily

$\rightarrow$ saturated cost partitioning

Conclusion

- many cost partitioning algorithms
- dominance and non-dominance relations
- saturated cost partitioning preferable

ICAPS 2017 paper

Jendrik Seipp, Thomas Keller and Malte Helmert

A Comparison of Cost Partitioning Algorithms for Optimal Classical Planning