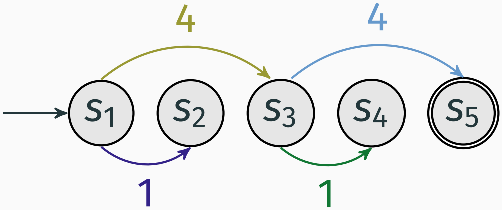
Saturated Post-hoc Optimization for Classical Planning

Jendrik Seipp,[1] Thomas Keller,[2] Malte Helmert[2]

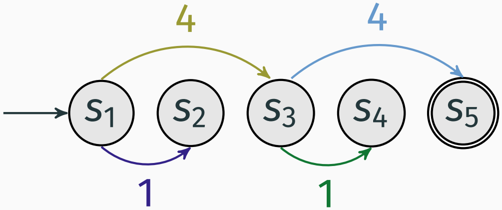
February, 2021

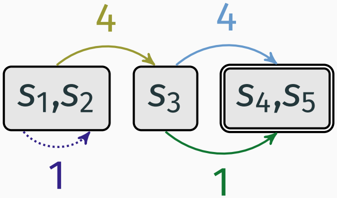
Linköping University (1) and University of Basel (2)

Optimal Classical Planning

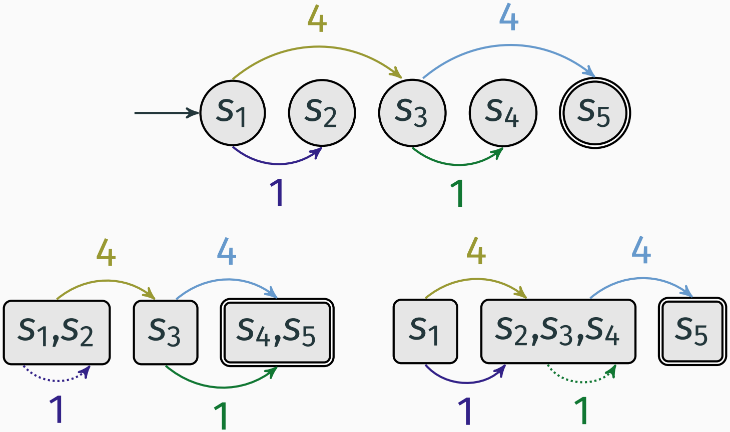


Abstraction Heuristics





Abstraction Heuristics

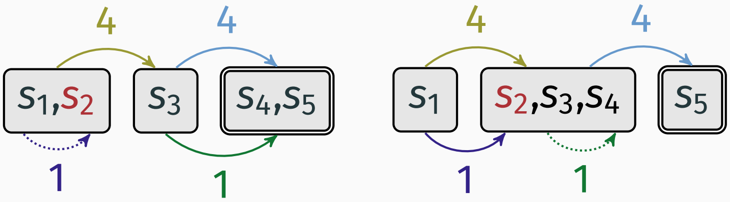


Multiple Heuristics

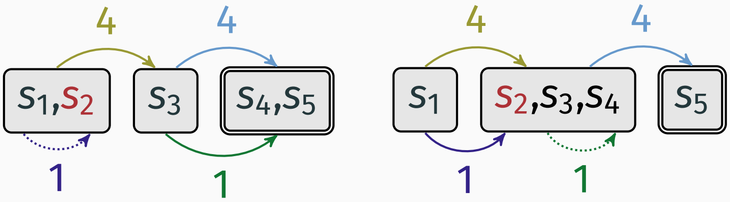
how to combine multiple heuristics?

Multiple Heuristics

how to combine multiple heuristics?



how to combine multiple heuristics?

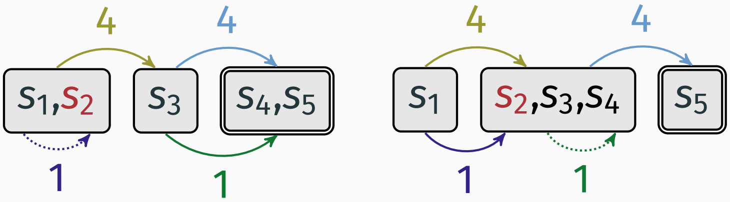


$h_1(s_2) = 5$

$h_2(s_2) = 4$

Multiple Heuristics

how to combine multiple heuristics?



$h_1(s_2) = 5$

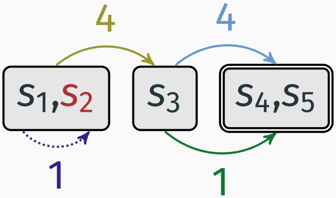
$h_2(s_2) = 4$

maximize over estimates:

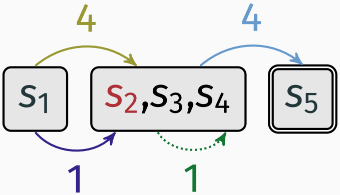
- $h(s_2) = 5$

Multiple Heuristics

how to combine multiple heuristics?



$h_1(s_2) = 5$



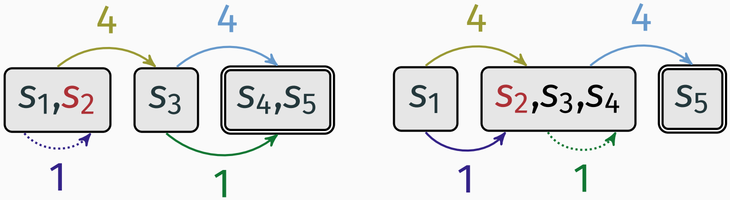
$h_2(s_2) = 4$

maximize over estimates:

- $h(s_2) = 5$
- only selects best heuristic
- does not combine heuristics

Cost Partitioning

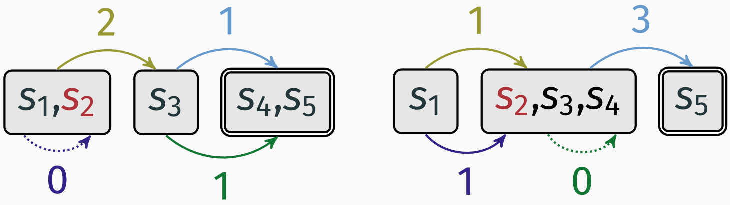
- split operator costs among heuristics
- sum of costs must not exceed original cost



Multiple Heuristics: Cost Partitioning

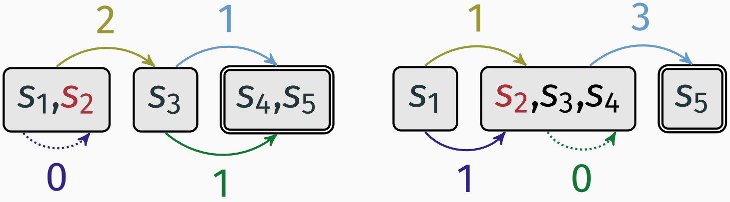
Cost Partitioning

- split operator costs among heuristics
- sum of costs must not exceed original cost



Cost Partitioning

- split operator costs among heuristics
- sum of costs must not exceed original cost

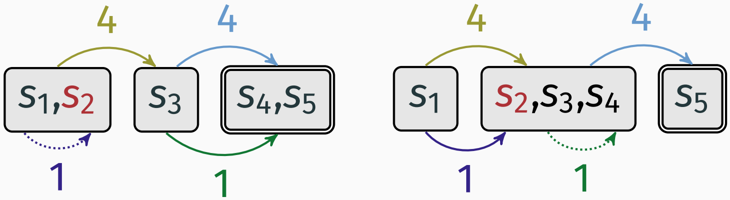


$$h(s_2) = 3 + 3 = 6$$

Saturated Cost Partitioning

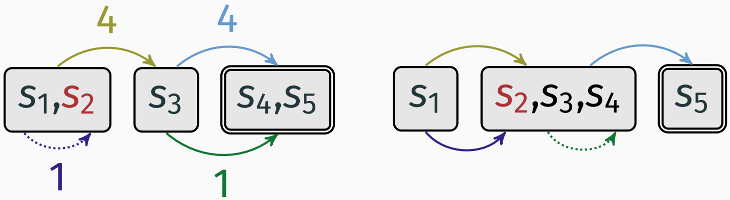
Saturated Cost Partitioning Algorithm

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



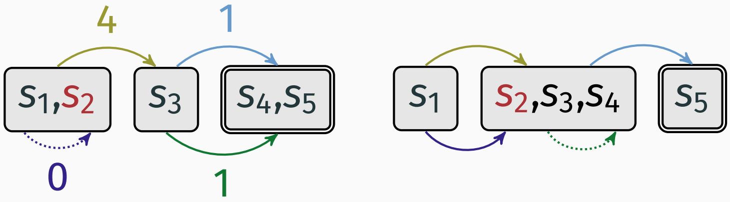
Saturated Cost Partitioning Algorithm

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



Saturated Cost Partitioning Algorithm

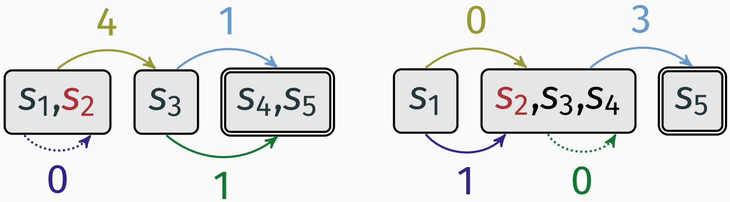
- order heuristics, then for each heuristic *h*:
  - use minimum costs preserving all estimates of *h*
  - use remaining costs for subsequent heuristics



Saturated Cost Partitioning

Saturated Cost Partitioning Algorithm

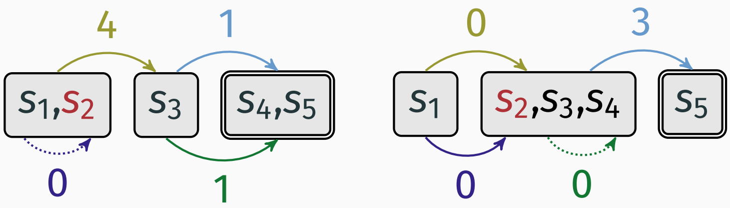
- order heuristics, then for each heuristic h:
  - use minimum costs preserving all estimates of h
  - use remaining costs for subsequent heuristics



Saturated Cost Partitioning

Saturated Cost Partitioning Algorithm

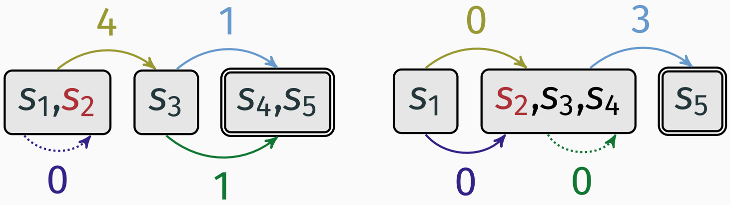
- order heuristics, then for each heuristic h:
  - use minimum costs preserving all estimates of h
  - use remaining costs for subsequent heuristics



Saturated Cost Partitioning

Saturated Cost Partitioning Algorithm

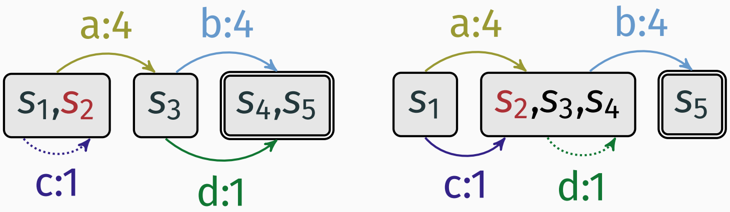
- order heuristics, then for each heuristic *h*:
  - use minimum costs preserving all estimates of *h*
  - use remaining costs for subsequent heuristics



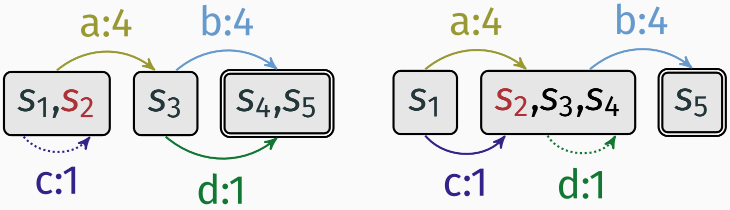
$$h(s_2) = 5 + 3 = 8$$

Post-hoc Optimization

Post-hoc Optimization

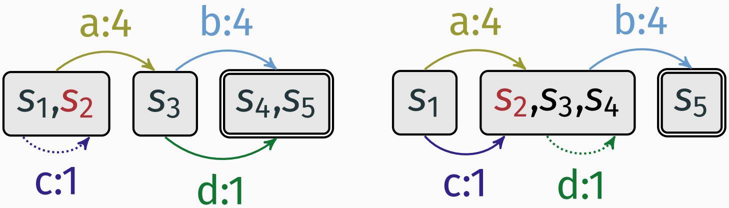


Post-hoc Optimization



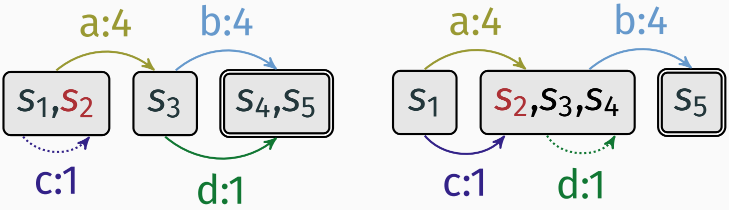
- A, B, D active $\quad h_1(s_2) = 5$

Post-hoc Optimization



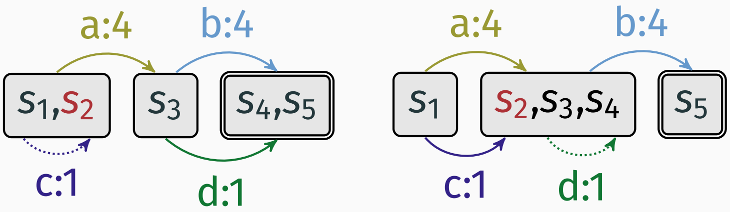
- A, B, D active $\quad h_1(s_2) = 5 \quad \rightarrow \quad 4A + 4B + 1D \geq 5$

Post-hoc Optimization



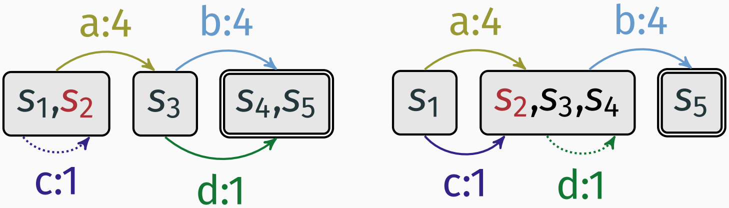
- A, B, D active $\quad h_1(s_2) = 5 \quad \rightarrow \quad 4A + 4B + 1D \geq 5$
- A, B, C active $\quad h_2(s_2) = 4$

Post-hoc Optimization



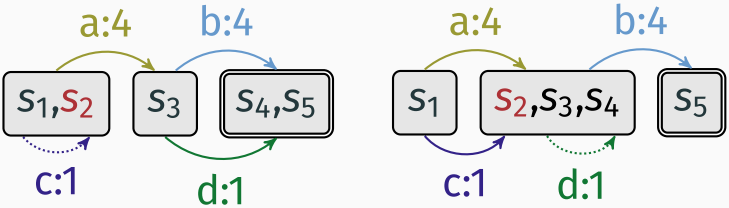
- A, B, D active $\quad h_1(s_2) = 5 \quad \rightarrow \quad 4A + 4B + 1D \geq 5$
- A, B, C active $\quad h_2(s_2) = 4 \quad \rightarrow \quad 4A + 4B + 1C \geq 4$

Post-hoc Optimization



- A, B, D active $\quad h_1(s_2) = 5 \quad \rightarrow \quad 4A + 4B + 1D \geq 5$
- A, B, C active $\quad h_2(s_2) = 4 \quad \rightarrow \quad 4A + 4B + 1C \geq 4$
- $A \geq 0$, $B \geq 0$, $C \geq 0$, $D \geq 0$

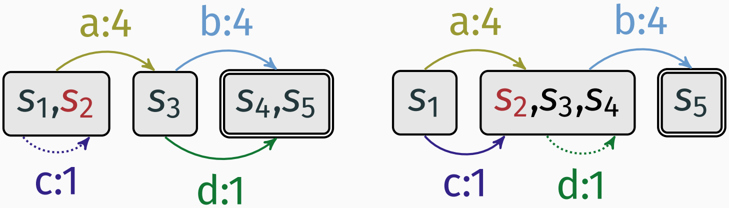
Post-hoc Optimization



minimize $4A + 4B + 1C + 1D$ **such that**

- A, B, D active $\quad h_1(s_2) = 5 \quad \rightarrow \quad 4A + 4B + 1D \geq 5$
- A, B, C active $\quad h_2(s_2) = 4 \quad \rightarrow \quad 4A + 4B + 1C \geq 4$
- $A \geq 0$, $B \geq 0$, $C \geq 0$, $D \geq 0$

Post-hoc Optimization



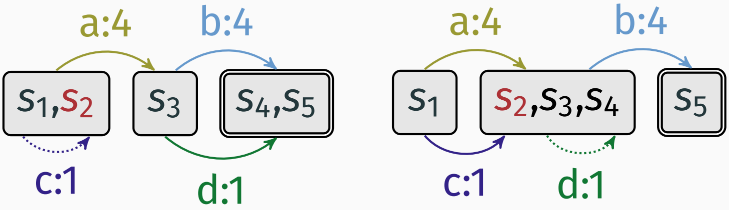
minimize $4A + 4B + 1C + 1D$ **such that**

- A, B, D active $\quad h_1(s_2) = 5 \quad \rightarrow \quad 4A + 4B + 1D \geq 5$
- A, B, C active $\quad h_2(s_2) = 4 \quad \rightarrow \quad 4A + 4B + 1C \geq 4$
- $A \geq 0, B \geq 0, C \geq 0, D \geq 0$

$$h(s_2) = 5$$

Saturated Post-hoc Optimization

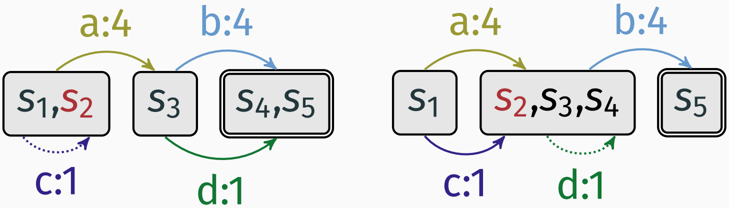
Saturated Post-hoc Optimization



minimize $4A + 4B + 1C + 1D$ **such that**

- $4A + 4B + 1D \geq 5$
- $4A + 4B + 1C \geq 4$
- $A \geq 0,\ B \geq 0,\ C \geq 0,\ D \geq 0$

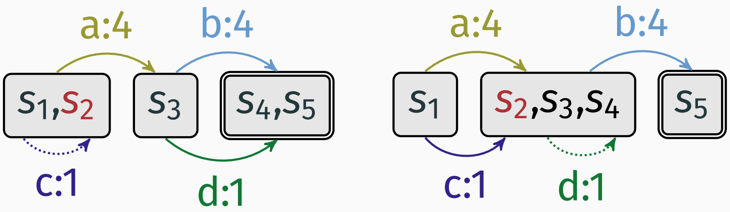
Saturated Post-hoc Optimization



minimize $4A + 4B + 1C + 1D$ **such that**

- $4A + 4B + 1D \geq 5$
- $4A + 4B + 1C \geq 4$
- $A \geq 0$, $B \geq 0$, $C \geq 0$, $D \geq 0$

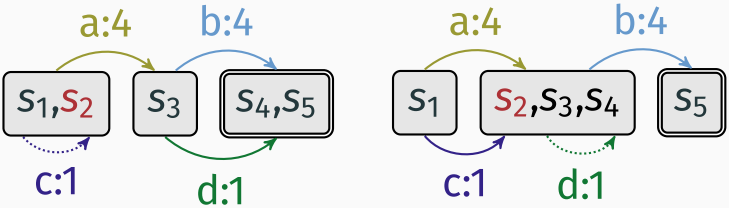
Saturated Post-hoc Optimization



minimize $4A + 4B + 1C + 1D$ **such that**

- $4A + 1B + 1D \geq 5$
- $4A + 4B + 1C \geq 4$
- $A \geq 0,\ B \geq 0,\ C \geq 0,\ D \geq 0$

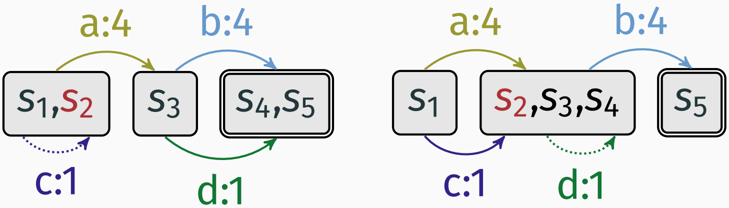
Saturated Post-hoc Optimization



minimize $4A + 4B + 1C + 1D$ **such that**

- $4A + 1B + 1D \geq 5$
- $4A + 4B + 1C \geq 4$
- $A \geq 0, B \geq 0, C \geq 0, D \geq 0$

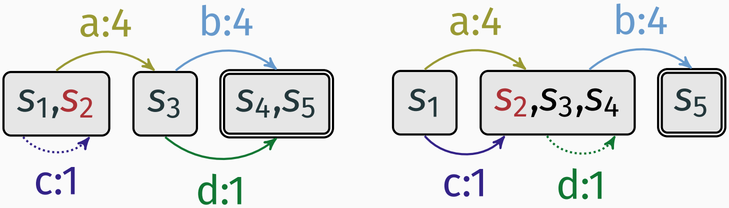
Saturated Post-hoc Optimization



minimize $4A + 4B + 1C + 1D$ **such that**

- $4A + 1B + 1D \geq 5$
- $4A + 4B + 1C \geq 4$
- $A \geq 0, B \geq 0, C \geq 0, D \geq 0$

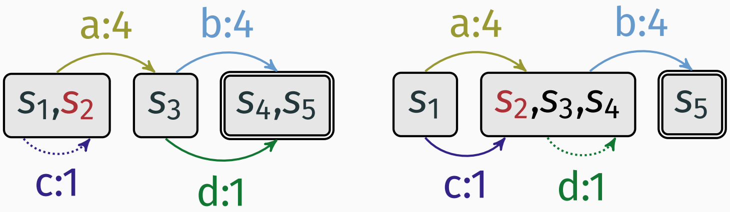
Saturated Post-hoc Optimization



minimize $4A + 4B + 1C + 1D$ **such that**

- $4A + 1B + 1D \geq 5$
- $1A + 4B + 1C \geq 4$
- $A \geq 0,\ B \geq 0,\ C \geq 0,\ D \geq 0$

Saturated Post-hoc Optimization



minimize $4A + 4B + 1C + 1D$ **such that**

- $4A + 1B + 1D \geq 5$
- $1A + 4B + 1C \geq 4$
- $A \geq 0,\ B \geq 0,\ C \geq 0,\ D \geq 0$

$$h(s_2) = 7.2$$

Saturated Post-hoc Optimization

Properties

- admissible
- dominates post-hoc optimization

Relation to Other Cost Partitioning Algorithms

UCP

Uniform Cost Partitioning

distribute costs evenly among relevant heuristics

Cost Partitioning Algorithms

GZOCP

UCP

Greedy Zero-one Cost Partitioning

order heuristics and give full cost to first relevant heuristic

Cost Partitioning Algorithms

GZOCP

PhO

UCP

Post-hoc Optimization

Cost Partitioning Algorithms

GZOCP

PhO

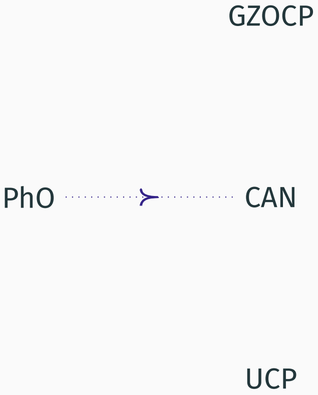
CAN

UCP

Canonical Heuristic

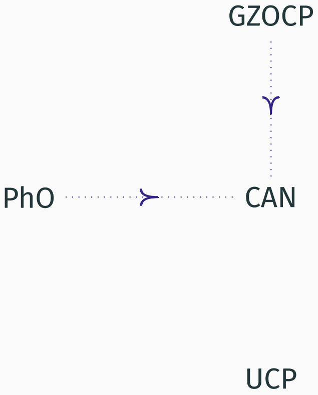
maximum over sums of independent heuristic subsets

Cost Partitioning Algorithms



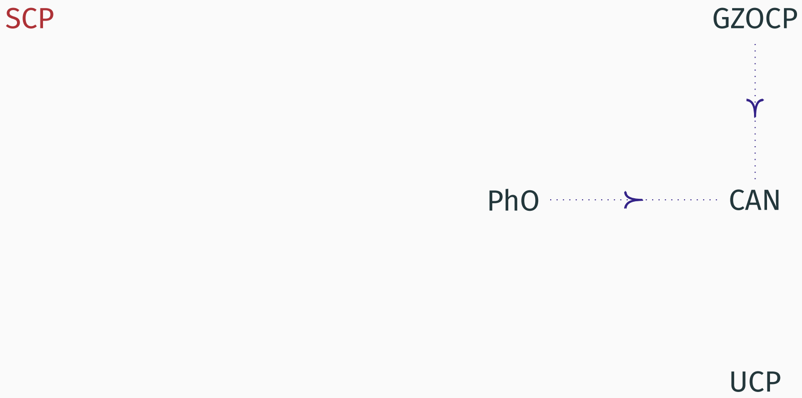
Pommerening et al. 2013

Cost Partitioning Algorithms

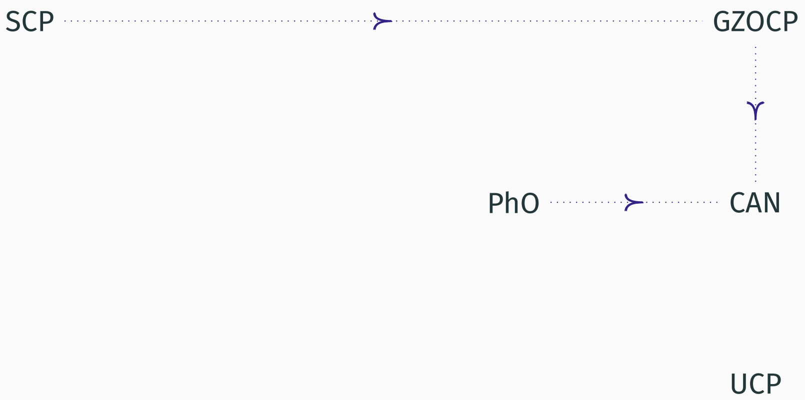


Seipp et al. 2017

Cost Partitioning Algorithms

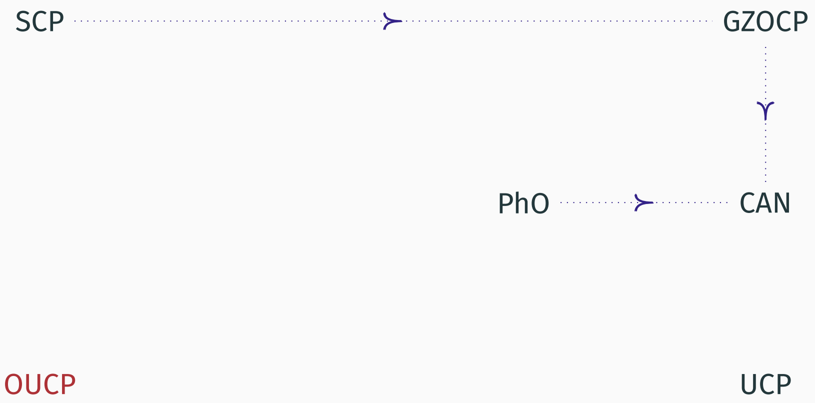


Cost Partitioning Algorithms



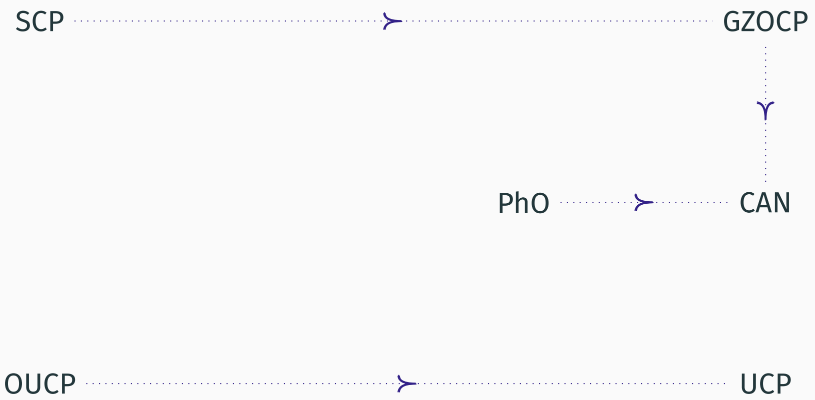
Seipp et al. 2017

Cost Partitioning Algorithms



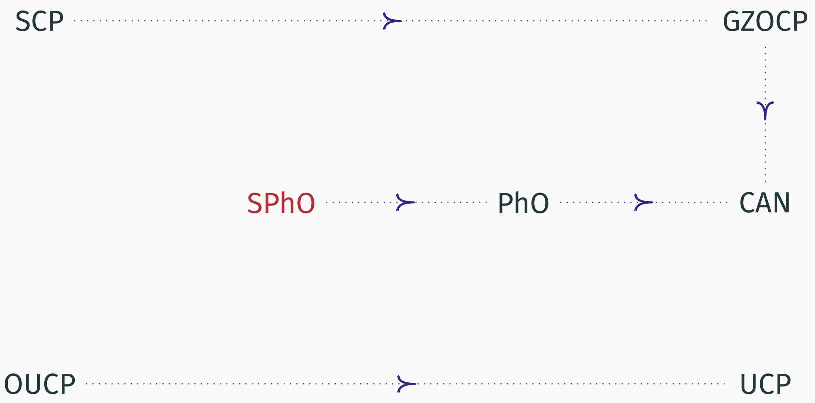
Seipp et al. 2017

Cost Partitioning Algorithms

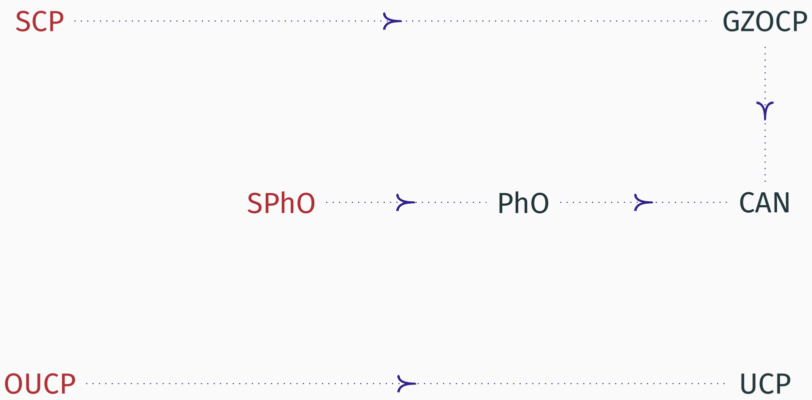


Seipp et al. 2017

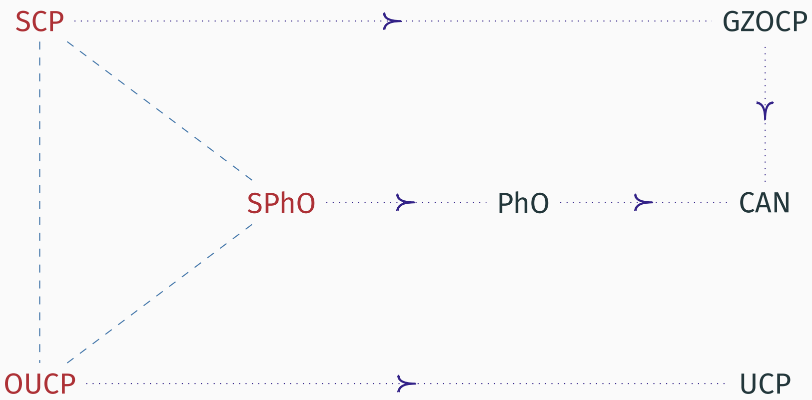
Cost Partitioning Algorithms



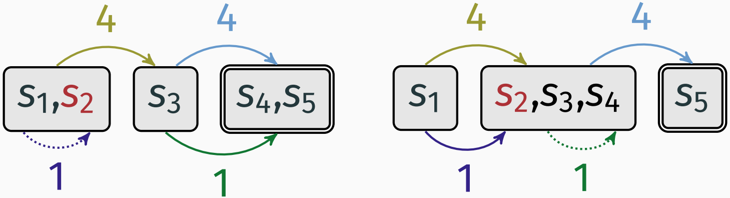
Cost Partitioning Algorithms



Cost Partitioning Algorithms



SPhO vs. SCP



$$h^{\mathrm{SCP}}_{\langle h_1,h_2\rangle}(s_2) = 8$$

$$h^{\mathrm{SPhO}}(s_2) = 7.2$$

$$h^{\mathrm{SCP}}_{\langle h_2,h_1\rangle}(s_2) = 7$$

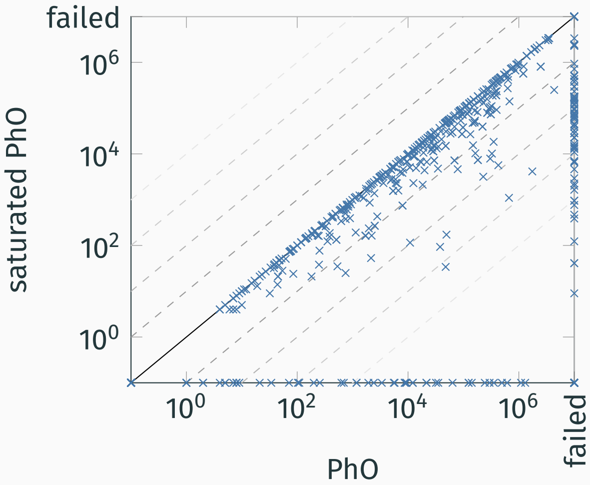
Experiments

Experiments: Setup

- saturated PhO vs. PhO
- compute for each state
- hill-climbing PDBs, systematic PDBs, Cartesian Abstractions
- 30 minutes, 3.5 GiB

Experiments: Coverage

	HillClimbing	Systematic	Cartesian	Combined
Domains ↑ (48)	**6**	**16**	**18**	**19**
Domains ↓ (48)	1	0	2	0
Tasks (1827)	823 **+10**	759 **+51**	657 **+169**	806 **+169**


failed
10^6
10^4
10^2
10^0
saturated PhO
10^0
10^2
10^4
10^6
failed
PhO

Conclusions

Saturated Post-hoc Optimization

- saturates costs
- dominates original
- admissible
- much stronger heuristics