A Comparison of Cost Partitioning Algorithms for Optimal Classical Planning

Jendrik Seipp Thomas Keller Malte Helmert

University of Basel

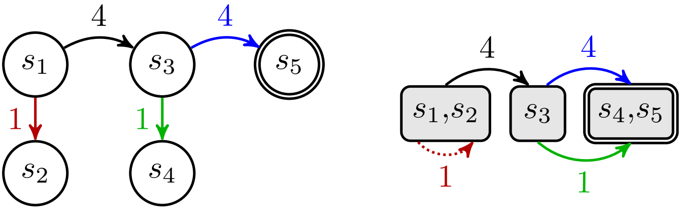
June 21, 2017

Setting

- optimal classical planning
- A^* search + admissible heuristic
- abstraction heuristics

Setting

- optimal classical planning
- A* search + admissible heuristic
- abstraction heuristics



Problem

- single heuristic unable to capture enough information

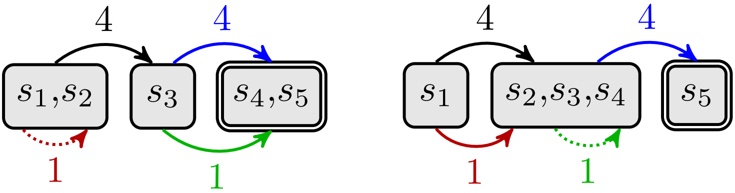
Problem

- single heuristic unable to capture enough information
  → use multiple heuristics

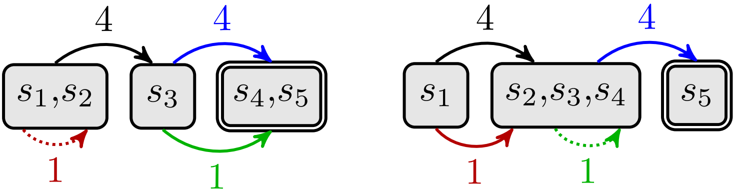
Problem

- single heuristic unable to capture enough information
  $\rightarrow$ use multiple heuristics
- how to combine multiple heuristics admissibly?

Multiple Heuristics

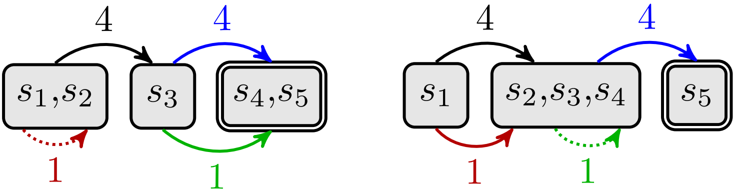


Multiple Heuristics



$h_1(s_1) = 5$ $h_2(s_1) = 5$

Multiple Heuristics



$h_1(s_1) = 5$ $h_2(s_1) = 5$

- maximizing only selects best heuristic $\rightarrow h(s_1) = 5$

Multiple Heuristics: Cost Partitioning

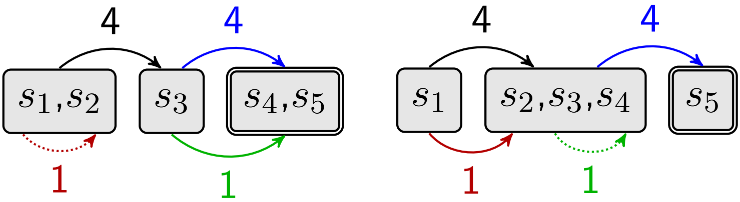
Cost Partitioning

- split operator costs among heuristics
- total costs must not exceed original costs

→ combines heuristics

→ allows summing heuristic values admissibly

Cost Partitioning Algorithms



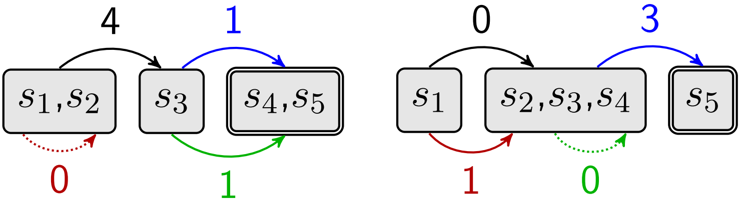
$$h(s_1) = ?$$

Optimal Cost Partitioning

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

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

Cost Partitioning Algorithms



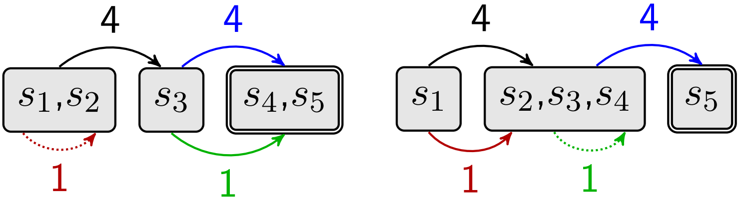
$$h(s_1) = 5 + 3 = 8$$

Optimal Cost Partitioning

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

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

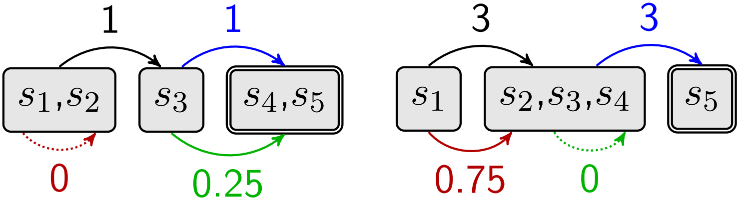
Cost Partitioning Algorithms



$$h(s_1) = ?$$

Post-hoc Optimization

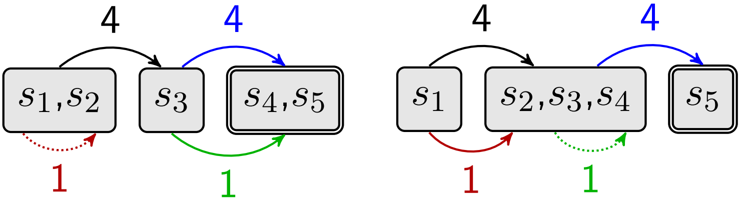
- compute best factor $0 \leq w \leq 1$ for each heuristic
- for each operator: sum of relevant heuristic factors ≤ 1
  e.g., $w_1 + w_2 \leq 1$, $w_2 \leq 1$
- use costs $w \cdot \mathit{cost}(o)$ if o is relevant for h, otherwise 0



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

Post-hoc Optimization

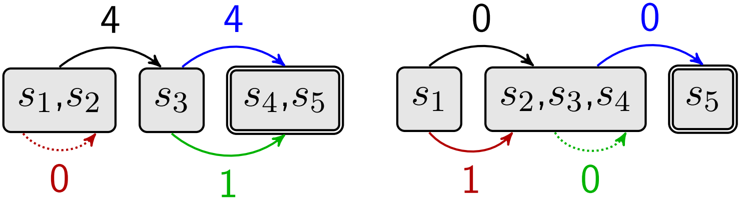
- compute best factor $0 \leq w \leq 1$ for each heuristic
- for each operator: sum of relevant heuristic factors ≤ 1
  e.g., $w_1 + w_2 \leq 1$, $w_2 \leq 1$
- use costs $w \cdot \mathit{cost}(o)$ if o is relevant for h, otherwise 0



$$h(s_1) = ?$$

Greedy Zero-one Cost Partitioning

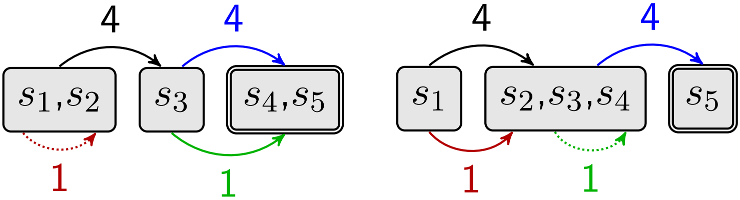
- order heuristics
- use full costs for the first relevant heuristic



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

Greedy Zero-one Cost Partitioning

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

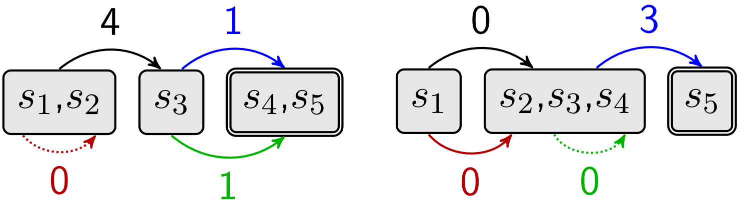


$$h(s_1) = ?$$

Saturated Cost Partitioning

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

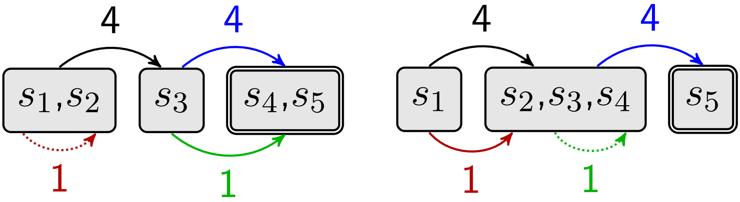
Cost Partitioning Algorithms



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

Saturated Cost Partitioning

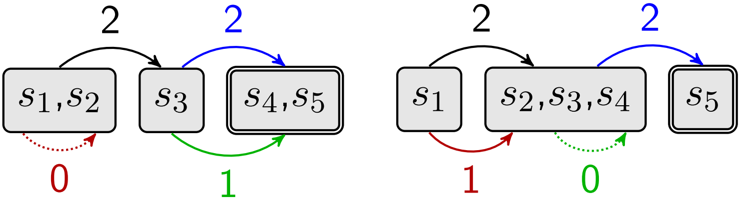
- order heuristics
- for each heuristic h:
  - use minimum costs preserving all heuristic estimates for h
  - use remaining costs for subsequent heuristics



$$h(s_1) = ?$$

Uniform Cost Partitioning

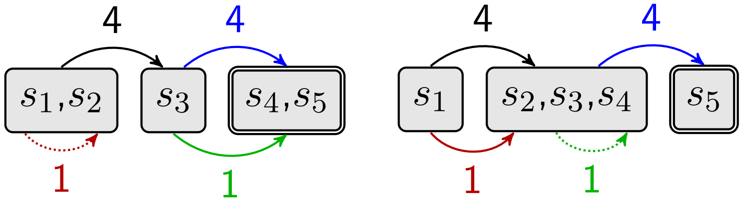
- distribute costs uniformly among relevant heuristics



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

Uniform Cost Partitioning

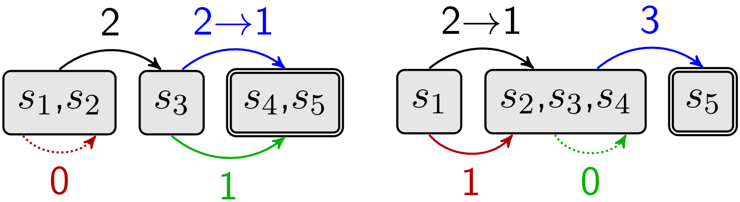
- distribute costs uniformly among relevant heuristics



$$h(s_1) = ?$$

Opportunistic Uniform Cost Partitioning (New)

- 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

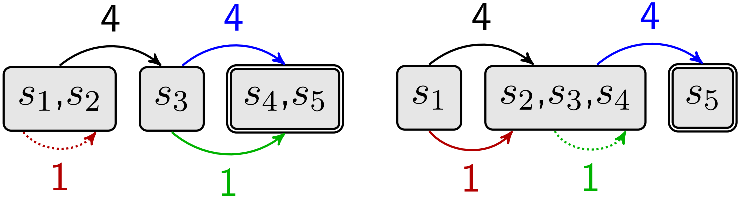


$$h(s_1) = 3 + 4 = 7$$

Opportunistic Uniform Cost Partitioning (New)

- 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

Cost Partitioning Algorithms

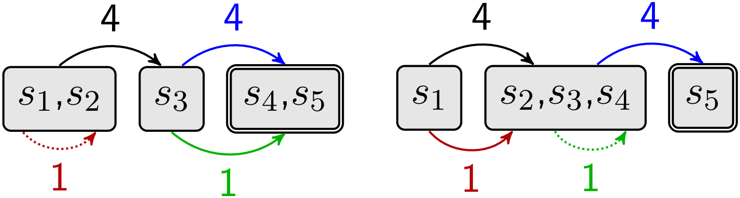


$$h(s_1) = ?$$

Canonical Heuristic

- compute independent heuristic subsets
- compute maximum over sums

Cost Partitioning Algorithms

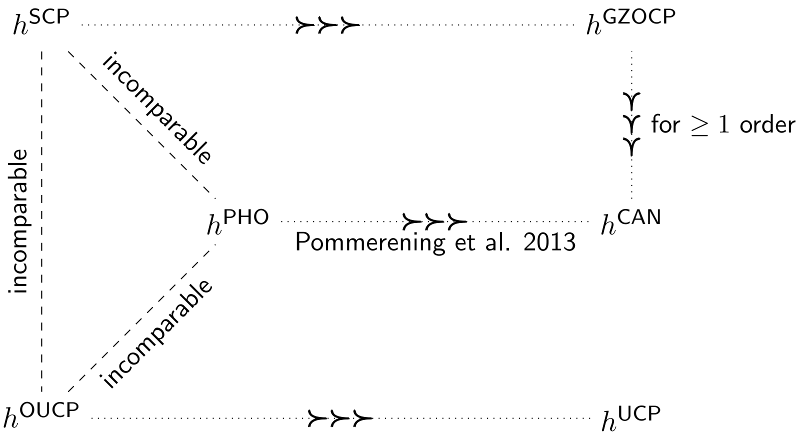


$$h(s_1) = \max(5, 5) = 5$$

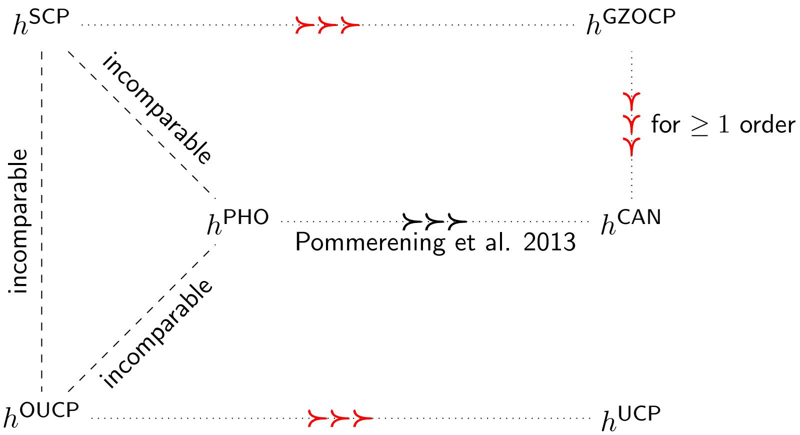
Canonical Heuristic

- compute independent heuristic subsets
- compute maximum over sums

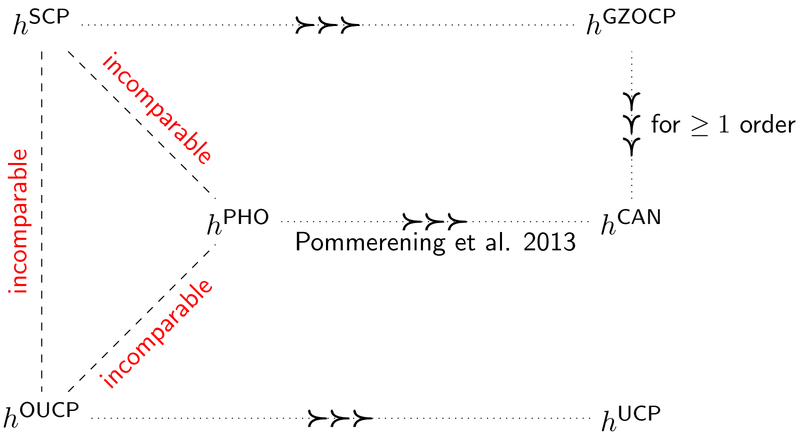
Theoretical Comparison



Theoretical Comparison



Theoretical Comparison



Empirical Comparison

Heuristics:

- hill climbing pattern databases
- systematic pattern databases
- Cartesian abstractions
- landmark heuristics

Empirical Comparison

Heuristics:

- hill climbing pattern databases
- systematic pattern databases
- Cartesian abstractions
- landmark heuristics

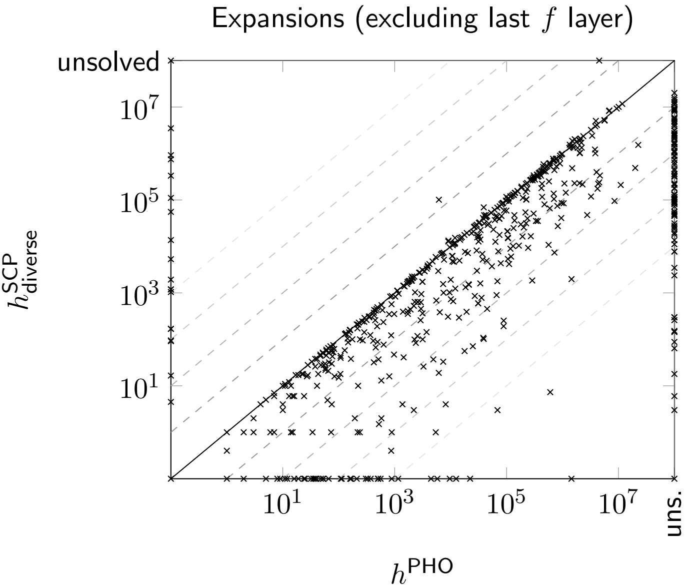
Orders:

- for order-dependent algorithms: single order and diverse orders

Empirical Comparison: Systematic PDBs

	h^{UCP}	h^{OUCP}_{single}	$h^{OUCP}_{diverse}$	h^{GZOCP}_{single}	$h^{GZOCP}_{diverse}$	h^{SCP}_{single}	$h^{SCP}_{diverse}$	h^{CAN}	h^{PHO}	h^{OCP}	coverage
h^{UCP}	–	0	3	**15**	3	4	0	**11**	**10**	**30**	709.0
h^{OUCP}_{single}	**14**	–	**9**	**22**	8	6	0	**14**	**13**	**31**	744.9
$h^{OUCP}_{diverse}$	**13**	8	–	**22**	7	6	0	**14**	**14**	**31**	734.6
h^{GZOCP}_{single}	3	1	4	–	3	0	0	9	11	**29**	694.0
$h^{GZOCP}_{diverse}$	**15**	**12**	**14**	**20**	–	9	0	**13**	**13**	**30**	749.9
h^{SCP}_{single}	**20**	**19**	**17**	**23**	**16**	–	0	**18**	**21**	**32**	775.7
$h^{SCP}_{diverse}$	**27**	**26**	**24**	**28**	**22**	**22**	–	**23**	**26**	**33**	**854.9**
h^{CAN}	8	7	7	**17**	5	8	2	–	**13**	**28**	656.0
h^{PHO}	9	7	7	**15**	7	6	3	10	–	**31**	737.0
h^{OCP}	4	4	4	4	4	4	3	5	3	–	471.0

Empirical Comparison: Systematic PDBs



Discussion of Results

In each setting:

- reuse unused costs
- assign costs greedily
- use multiple orders

$\rightarrow$ saturated cost partitioning

Comparison to State of the Art (Using h^2 Mutexes)

	HC+$h^{\text{SCP}}_{\text{diverse}}$	Sys2+$h^{\text{SCP}}_{\text{diverse}}$	Cart.+$h^{\text{SCP}}_{\text{diverse}}$	LM+$h^{\text{SCP}}_{\text{single}}$	SymBA$^*_2$	coverage
HC+$h^{\text{SCP}}_{\text{diverse}}$	–	7	9	**19**	17	845.0
Sys2+$h^{\text{SCP}}_{\text{diverse}}$	**10**	–	11	**18**	16	878.5
Cart.+$h^{\text{SCP}}_{\text{diverse}}$	**19**	**14**	–	**24**	**17**	**1017.9**
LM+$h^{\text{SCP}}_{\text{single}}$	8	9	4	–	9	934.0
SymBA$^*_2$	**20**	**18**	16	**23**	–	1008.0

Related SoCS 2017 Paper

Better Orders for Saturated Cost Partitioning in Optimal Classical Planning

- combination of three types of abstraction heuristics
- better method for finding heuristic orders
- significantly higher coverage

Conclusion

- new dominance relations
- new cost partitioning algorithm
- saturated cost partitioning preferable in all tested settings