Subset-Saturated Cost Partitioning for Optimal Classical Planning

Jendrik Seipp, Malte Helmert

July 14, 2019

University of Basel, Switzerland

In a nutshell

- optimal classical planning
- A^* search + admissible heuristic
- multiple heuristics
- cost partitioning

In a nutshell

- optimal classical planning
- A^* search + admissible heuristic
- multiple heuristics
- saturated cost partitioning

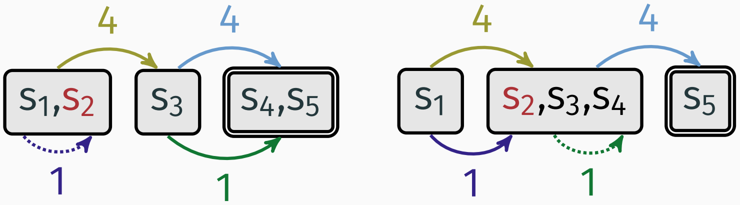
In a nutshell

- optimal classical planning
- A^* search + admissible heuristic
- multiple heuristics
- subset-saturated cost partitioning

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

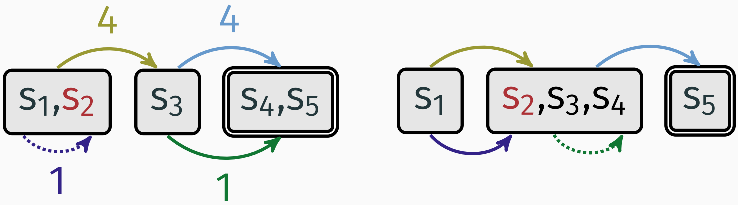


$\max(h_1(s_2), h_2(s_2)) = \max(5, 4) = 5$

Saturated cost partitioning

Saturated cost partitioning algorithm

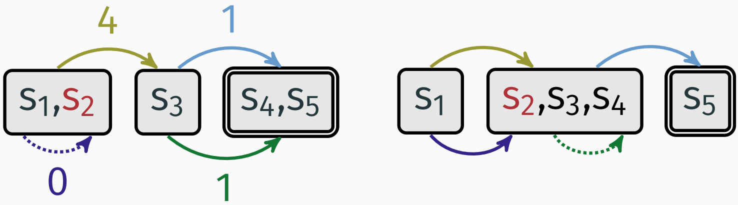
- order heuristics, then for each heuristic h:
  - use minimum costs preserving all estimates of h
  - use remaining costs for subsequent heuristics



Saturated cost partitioning

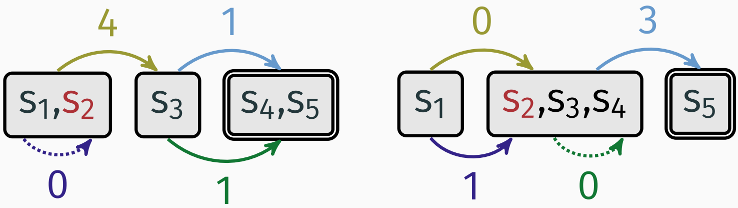
Saturated cost partitioning algorithm

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



Saturated cost partitioning algorithm

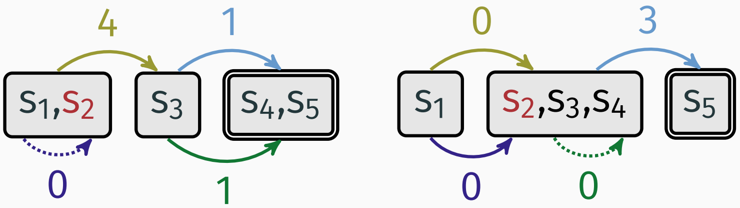
- order heuristics, then for each heuristic h:
  - use minimum costs preserving all estimates of h
  - use remaining costs for subsequent heuristics



Saturated cost partitioning

Saturated cost partitioning algorithm

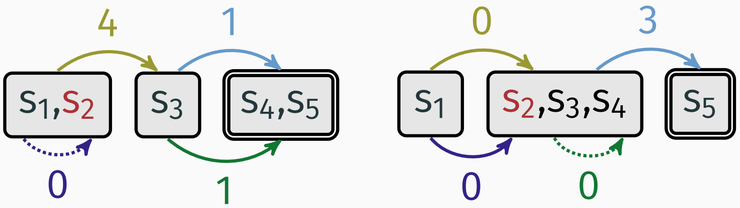
- order heuristics, then for each heuristic h:
  - use minimum costs preserving all estimates of h
  - use remaining costs for subsequent heuristics



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



$$h^{\mathrm{SCP}}_{\langle h_1,h_2\rangle}(s_2) = 5 + 3 = 8$$

Saturated cost function

1. $\text{scf}(o) \leq \text{cost}(o)$ for all operators $o \in \mathcal{O}$
2. $h(\text{scf}, s) = h(\text{cost}, s)$ for all states $s \in S$

Saturated cost function

1. $\mathrm{scf}(o) \leq \mathrm{cost}(o)$ for all operators $o \in \mathcal{O}$
2. $h(\mathrm{scf}, s) = h(\mathrm{cost}, s)$ for all states $s \in S' \subseteq S$

Saturated cost function

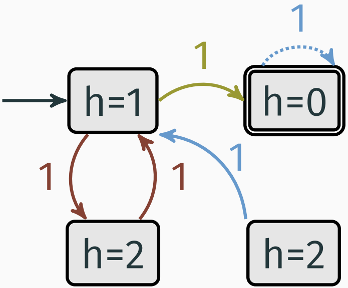
1. $\mathrm{scf}(o) \leq \mathrm{cost}(o)$ for all operators $o \in \mathcal{O}$
2. $h(\mathrm{scf}, s) = h(\mathrm{cost}, s)$ for all states $s \in S' \subseteq S$

$\rightarrow$ 4 saturators

Saturate for all states

Saturator all

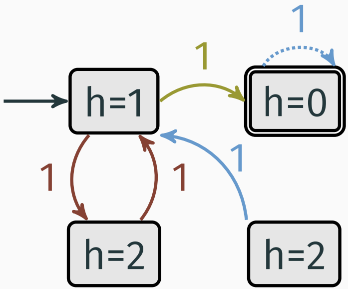
$$\mathrm{scf}(o) = \max_{a \xrightarrow{o} b \in T} (h(a) - h(b))$$



Saturate for reachable states

Saturator reach

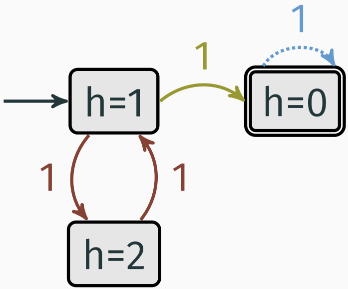
- remove unreachable states and transitions
- use all saturator for remaining states



Saturate for reachable states

Saturator **reach**

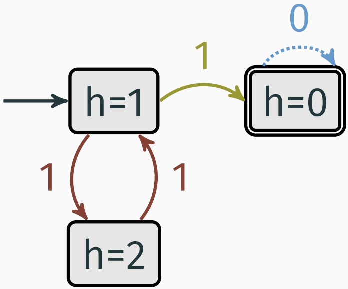
- remove unreachable states and transitions
- use all saturator for remaining states



Saturate for reachable states

Saturator reach

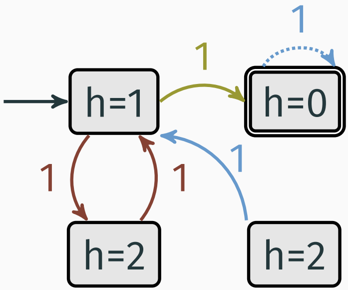
- remove unreachable states and transitions
- use all saturator for remaining states



Saturate for a perimeter

Saturator perim

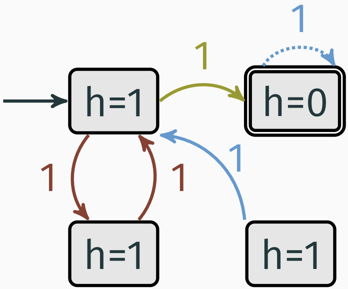
- use perimeter k around goal states ($k = h(s)$)
- cap all distances at k
- use all saturator



Saturate for a perimeter

Saturator perim

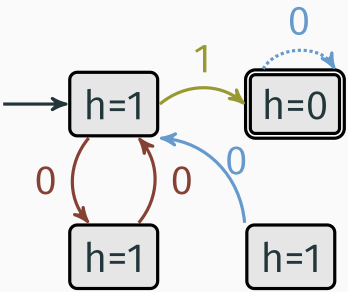
- use perimeter k around goal states ($k = h(s)$)
- cap all distances at k
- use all saturator



Saturate for a perimeter

Saturator perim

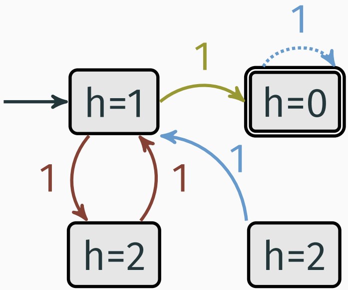
- use perimeter k around goal states ($k = h(s)$)
- cap all distances at k
- use all saturator



Saturate for a single state

Saturator lp

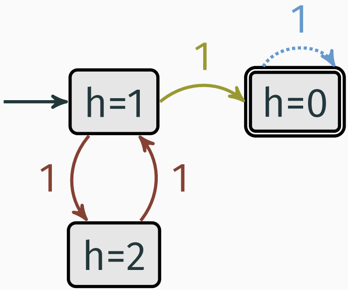
$$
\begin{aligned}
H_a &\leq 0 && \text{for all } a \in S_\star \\
H_a &\leq C_o + H_b && \text{for all } a \xrightarrow{o} b \in T \\
C_o &\leq \text{cost}(o) && \text{for all } o \in \mathcal{O} \\
H_{\alpha(s)} &= h(s)
\end{aligned}
$$



Saturate for a single state

Saturator lp

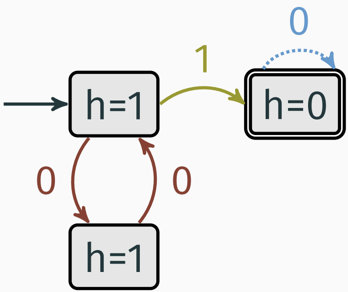
$$
\begin{aligned}
H_a &\leq 0 && \text{for all } a \in S_\star \\
H_a &\leq C_o + H_b && \text{for all } a \xrightarrow{o} b \in T \\
C_o &\leq \text{cost}(o) && \text{for all } o \in \mathcal{O} \\
H_{\alpha(s)} &= h(s)
\end{aligned}
$$



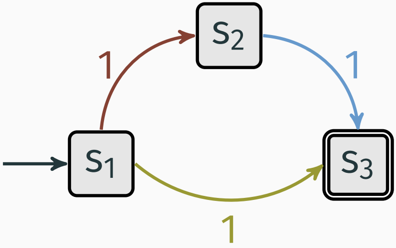
Saturate for a single state

Saturator lp

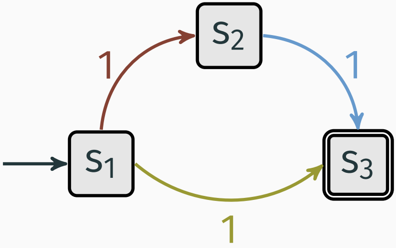
$$
\begin{aligned}
H_a &\leq 0 && \text{for all } a \in S_\star \\
H_a &\leq C_o + H_b && \text{for all } a \xrightarrow{o} b \in T \\
C_o &\leq \mathrm{cost}(o) && \text{for all } o \in \mathcal{O} \\
H_{\alpha(s)} &= h(s)
\end{aligned}
$$



No unique minimum saturated cost function



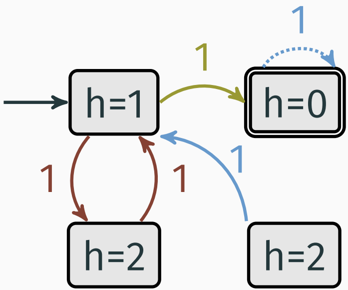
No unique minimum saturated cost function



- large Pareto frontier
- $\rightarrow$ chain saturators

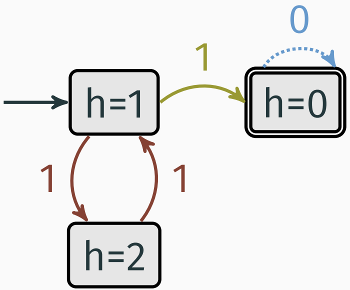
Chaining saturators

reach, perim:



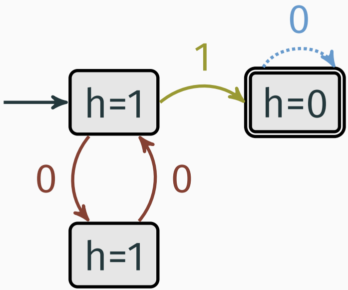
Chaining saturators

reach, perim:



Chaining saturators

reach, perim:



Experiments

Setup

- 30 minutes, 3.5 GiB
- hill climbing and systematic PDBs, Cartesian abstractions
- online and offline
- non-negative and general costs

Heuristic value for initial state – non-negative costs (1506 tasks)

	all	reach	perim	lp	all, lp	reach, lp	perim, lp
all	–	2	2	88	2	2	2
reach	**15**	–	5	88	2	2	2
perim	**433**	**430**	–	142	15	15	6
lp	**493**	**490**	**229**	–	79	79	84
all, lp	**507**	**505**	**219**	**179**	–	**0**	12
reach, lp	**507**	**505**	**219**	**179**	**0**	–	12
perim, lp	**512**	**510**	**217**	**185**	**16**	**16**	–

Heuristic value for initial state – non-negative costs (1506 tasks)

	all	reach	perim	lp	all, lp	reach, lp	perim, lp
all	–	2	2	88	2	2	2
reach	**15**	–	5	88	2	2	2
perim	**433**	**430**	–	142	15	15	6
lp	**493**	**490**	**229**	–	79	79	84
all, lp	**507**	**505**	**219**	**179**	–	**0**	12
reach, lp	**507**	**505**	**219**	**179**	**0**	–	12
perim, lp	**512**	**510**	**217**	**185**	**16**	**16**	–

Heuristic value for initial state – non-negative costs (1506 tasks)

	all	reach	perim	lp	all, lp	reach, lp	perim, lp
all	–	2	2	88	2	2	2
reach	**15**	–	5	88	2	2	2
perim	**433**	**430**	–	142	15	15	6
lp	**493**	**490**	**229**	–	79	79	84
all, lp	**507**	**505**	**219**	**179**	–	**0**	12
reach, lp	**507**	**505**	**219**	**179**	**0**	–	12
perim, lp	**512**	**510**	**217**	**185**	**16**	**16**	–

Heuristic value for initial state – non-negative costs (1506 tasks)

	all	reach	perim	lp	all, lp	reach, lp	perim, lp
all	–	2	2	88	2	2	2
reach	**15**	–	5	88	2	2	2
perim	**433**	**430**	–	142	15	15	6
lp	**493**	**490**	**229**	–	79	79	84
all, lp	**507**	**505**	**219**	**179**	–	**0**	12
reach, lp	**507**	**505**	**219**	**179**	**0**	–	12
perim, lp	**512**	**510**	**217**	**185**	**16**	**16**	–

Heuristic value for initial state – non-negative costs (1506 tasks)

	all	reach	perim	lp	all, lp	reach, lp	perim, lp
all	–	2	2	88	2	2	2
reach	**15**	–	5	88	2	2	2
perim	**433**	**430**	–	142	15	15	6
lp	**493**	**490**	**229**	–	79	79	84
all, lp	**507**	**505**	**219**	**179**	–	**0**	12
reach, lp	**507**	**505**	**219**	**179**	**0**	–	12
perim, lp	**512**	**510**	**217**	**185**	**16**	**16**	–

Heuristic value for initial state – non-negative costs (1506 tasks)

	all	reach	perim	lp	all, lp	reach, lp	perim, lp
all	–	2	2	88	2	2	2
reach	**15**	–	5	88	2	2	2
perim	**433**	**430**	–	142	15	15	6
lp	**493**	**490**	**229**	–	79	79	84
all, lp	**507**	**505**	**219**	**179**	–	**0**	12
reach, lp	**507**	**505**	**219**	**179**	**0**	–	12
perim, lp	**512**	**510**	**217**	**185**	**16**	**16**	–

Online – non-negative (nn) vs. general costs (gen)

	Coverage		Evals/sec		$h(s_0)$ higher	
	nn	gen	nn	gen	nn	gen
all	680	**703**	940.9	**946.2**	7	**195**
reach	657	**679**	501.4	**514.1**	5	**222**
perim	722	**726**	897.7	**906.6**	4	**99**
lp	**360**	320	**8.7**	6.9	**493**	144
all, lp	**385**	354	**10.0**	8.1	138	**180**
reach, lp	**384**	355	**9.9**	8.1	120	**181**
perim, lp	**392**	367	**11.0**	8.8	31	**134**

Offline subset-saturated cost partitioning

	all	reach	perim	lp	Coverage
all	–	**1**	**9**	**36**	1136
reach	0	–	**8**	**36**	1134
perim	1	2	–	**36**	1117
lp	0	0	0	–	694

Offline subset-saturated cost partitioning

	all	reach	perim	lp	perim*	Coverage
all	–	**1**	**9**	**36**	1	1136
reach	0	–	**8**	**36**	1	1134
perim	1	2	–	**36**	0	1117
lp	0	0	0	–	0	694
perim*	**4**	**4**	**11**	**36**	–	**1144**

Summary

- generalization of saturated cost partitioning
- three new saturators
- stronger heuristics