Online Saturated Cost Partitioning for Classical Planning

Jendrik Seipp

October 21, 2020

University of Basel

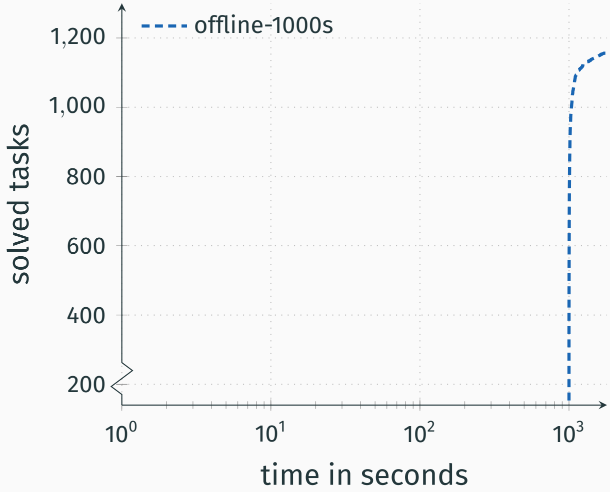
Setting

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

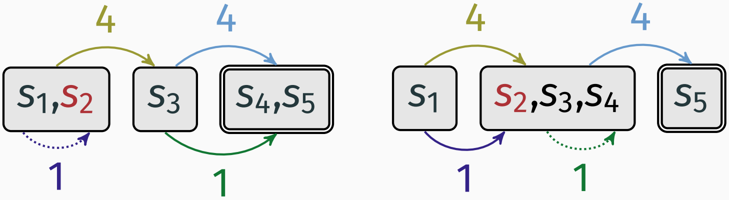
Setting

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

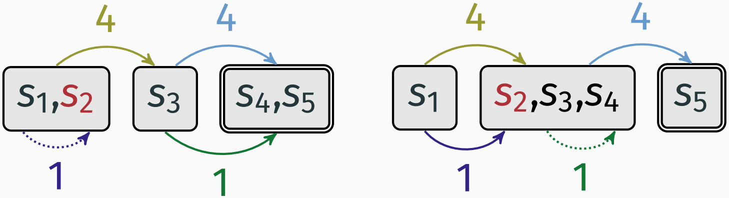
Coverage over time



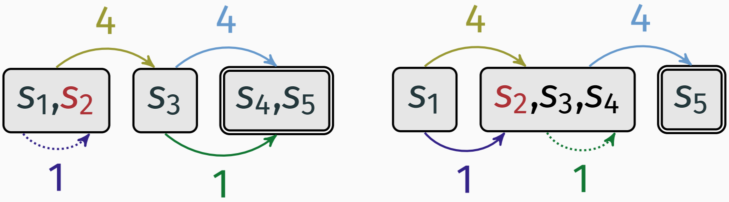
Background



Background



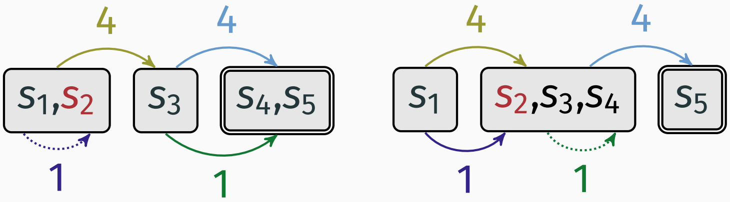
$h_1(s_2) = 5$ $h_2(s_2) = 4$



$h_1(s_2) = 5$ $\qquad$ $h_2(s_2) = 4$

maximize over estimates:

- $h(s_2) = 5$



$h_1(s_2) = 5$

$h_2(s_2) = 4$

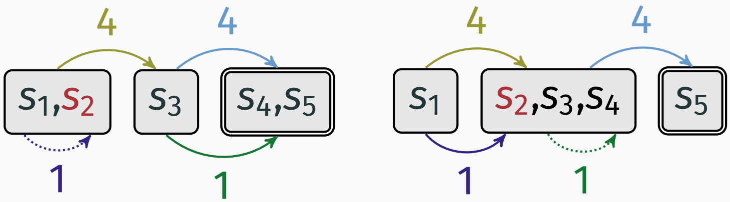
maximize over estimates:

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

Background

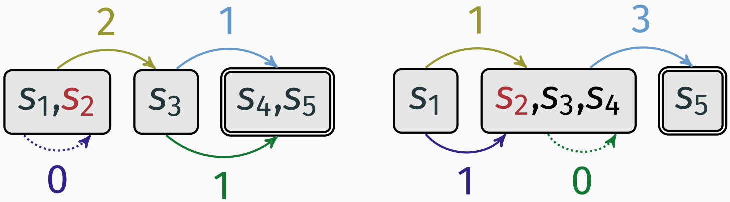
Cost partitioning

- split action costs among heuristics
- sum of costs $\leq$ original cost



Cost partitioning

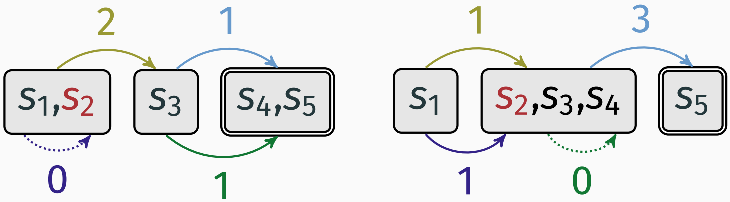
- split action costs among heuristics
- sum of costs $\leq$ original cost



Background

Cost partitioning

- split action costs among heuristics
- sum of costs $\leq$ original cost

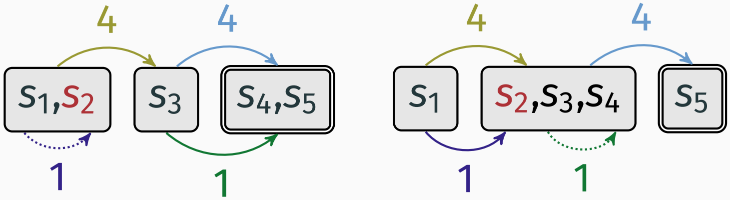


$$h(s_2) = 3 + 3 = 6$$

Background

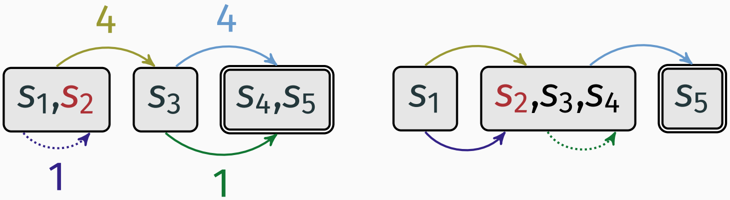
Saturated cost partitioning

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



Saturated cost partitioning

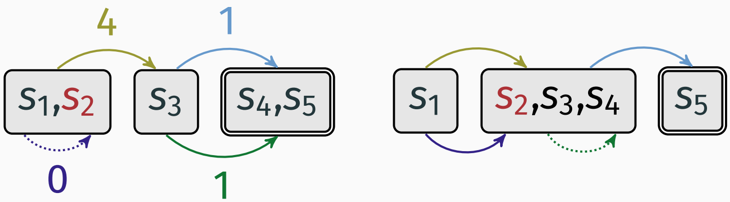
- order heuristics, then for each heuristic *h*:
  - use minimum costs preserving all estimates of *h*
  - use remaining costs for subsequent heuristics



Background

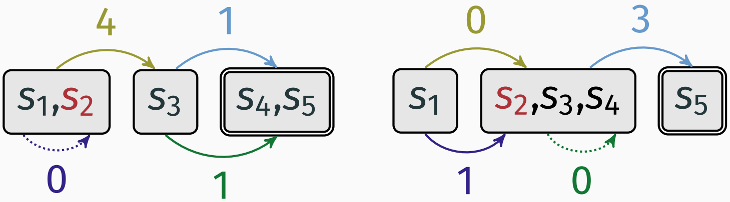
Saturated cost partitioning

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



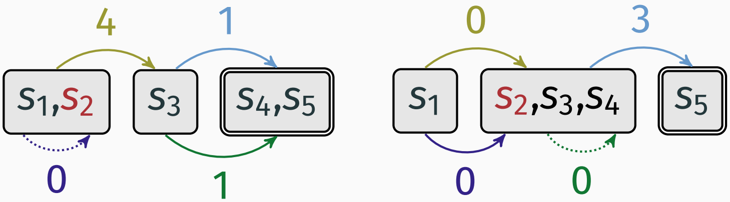
Saturated cost partitioning

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



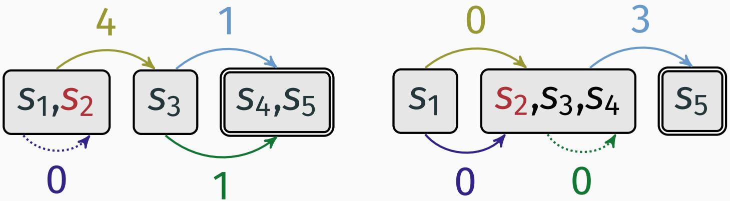
Saturated cost partitioning

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



Saturated cost partitioning

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



$$h^{\text{SCP}}(s_2) = 5 + 3 = 8$$

Background

Order matters:

- $h^{\text{SCP}}_{\langle h_1, h_2 \rangle}(s_2) = 8$
- $h^{\text{SCP}}_{\langle h_2, h_1 \rangle}(s_2) = 7$

Order matters:

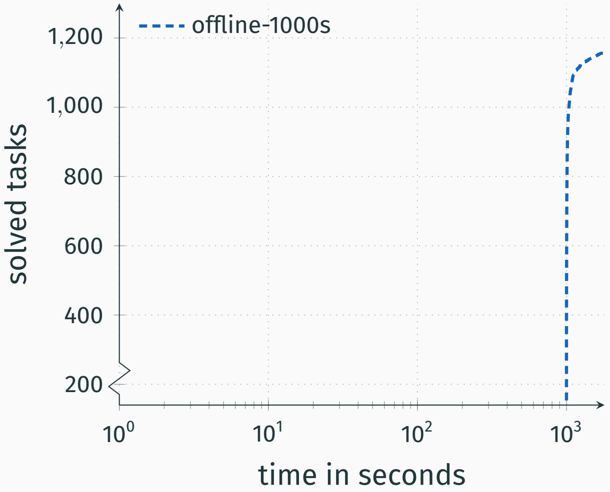
- $h^{\text{SCP}}_{\langle h_1,h_2\rangle}(s_2) = 8$
- $h^{\text{SCP}}_{\langle h_2,h_1\rangle}(s_2) = 7$

$\rightarrow$ use multiple orders and maximize over estimates:
$\max(h^{\text{SCP}}_{\langle h_1,h_2\rangle}(s_2), h^{\text{SCP}}_{\langle h_2,h_1\rangle}(s_2))$

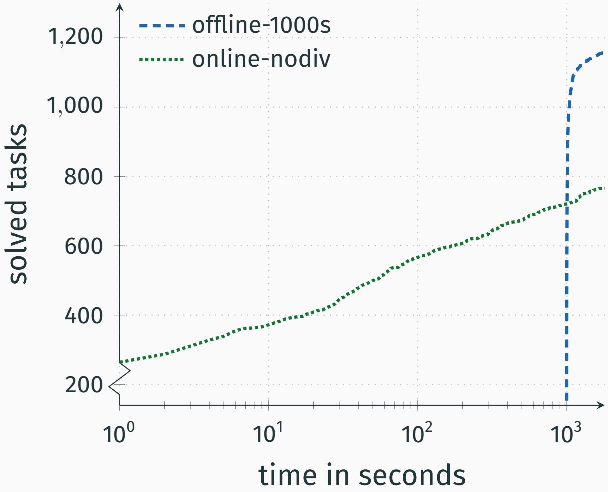
Offline diversification

- sample 1000 states
- start with empty set of orders
- until time limit is reached:
  - compute order for new sample
  - store order if a sample profits from it

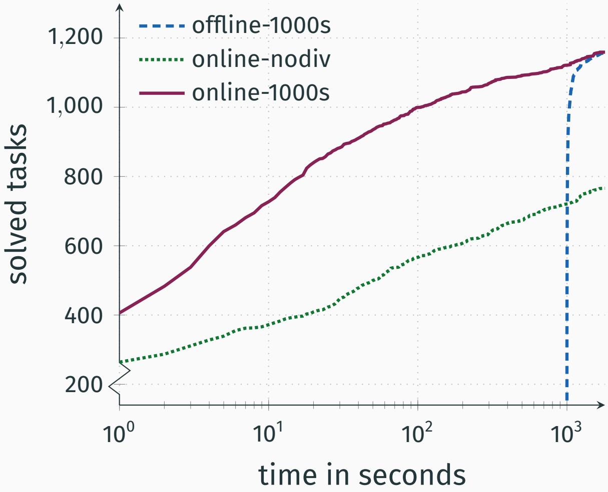
Coverage over time



Coverage over time



Coverage over time



ComputeHeuristic(s)

- if Select(s) and not time limit reached
  - compute order for s
  - store order if s profits from it
- return maximum over all stored orders for s

Offline vs. online diversification

Offline

- compute orders for samples for T seconds
- store order if one of 1000 samples profits from it

Online

- compute orders for subset of evaluated states for at most T seconds
- store order if single evaluated state profits from it

Selection strategies

SELECT

- Interval
- Novelty (Lipovetzky and Geffner 2012)
- Bellman (Eifler and Fickert 2018): $h(s) \geq \min_{s \xrightarrow{a} t \in T} (h(t) + cost(a))$

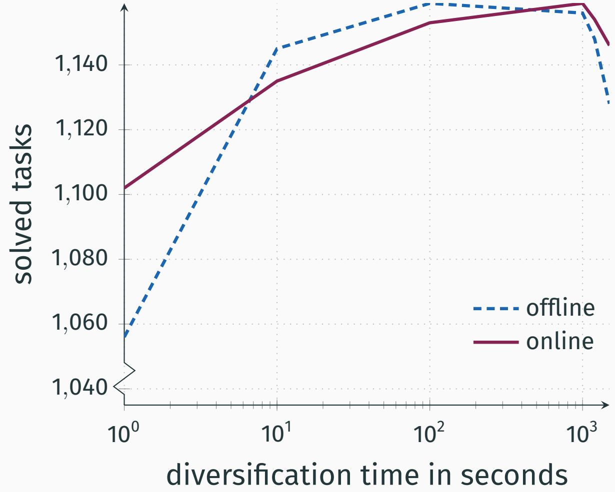
Selection strategies

SELECT

- Interval
- Novelty (Lipovetzky and Geffner 2012)
- Bellman (Eifler and Fickert 2018): $h(s) \geq \min_{s \xrightarrow{a} t \in T}(h(t) + cost(a))$

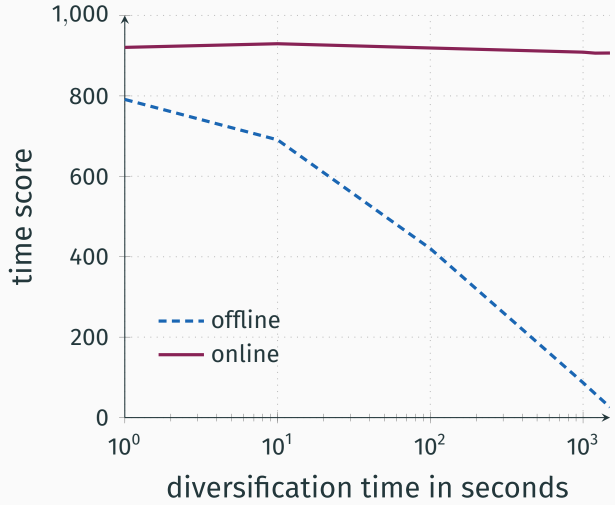
	Bellman	Novelty 1	Novelty 2	Interval 1–100K
Coverage	1145	1153	1157	1153–**1159**

Coverage

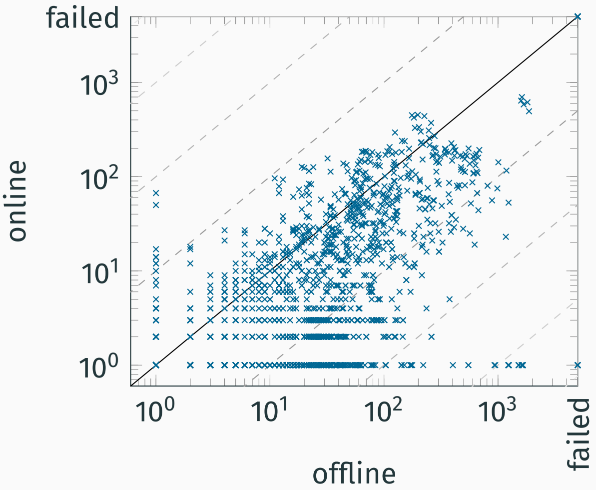


solved tasks
1,140
1,120
1,100
1,080
1,060
1,040
10^0
10^1
10^2
10^3
diversification time in seconds
offline
online

Time score


1,000
800
600
400
200
0
time score
offline
online
10^0
10^1
10^2
10^3
diversification time in seconds

Stored orders



Before the A^* search can start

Build abstractions

- e.g., Cartesian abstractions and symbolic PDBs

Compute orders and cost partitionings

- e.g., saturated cost partitioning

Before the A^* search can start

Build abstractions

- e.g., Cartesian abstractions and symbolic PDBs

$\rightarrow$ online refinement (e.g., Eifler and Fickert 2018, Franco and Torralba 2019)

Compute orders and cost partitionings

- e.g., saturated cost partitioning

Before the A^* search can start

Build abstractions

- e.g., Cartesian abstractions and symbolic PDBs

$\rightarrow$ online refinement (e.g., Eifler and Fickert 2018, Franco and Torralba 2019)

Compute orders and cost partitionings

- e.g., saturated cost partitioning

$\rightarrow$ Bellman, novelty, interval

Before the A^* search can start

Choose which abstractions to build

- e.g., patterns for PDBs

Build abstractions

- e.g., Cartesian abstractions and symbolic PDBs

$\rightarrow$ online refinement (e.g., Eifler and Fickert 2018, Franco and Torralba 2019)

Compute orders and cost partitionings

- e.g., saturated cost partitioning

$\rightarrow$ Bellman, novelty, interval

Before the A^* search can start

Choose which abstractions to build

- e.g., patterns for PDBs

$\rightarrow$ future work

Build abstractions

- e.g., Cartesian abstractions and symbolic PDBs

$\rightarrow$ online refinement (e.g., Eifler and Fickert 2018, Franco and Torralba 2019)

Compute orders and cost partitionings

- e.g., saturated cost partitioning

$\rightarrow$ Bellman, novelty, interval

Summary

Offline diversification	Online computation	Online diversification
long precomputation	no precomputation	no precomputation
samples	states	states
fast evaluations	slow evaluations	fast evaluations
high coverage	low coverage	high coverage

Offline vs. online diversification

		1s	10s	100s	1000s	1200s	1500s
Coverage	offline	1056	1145	**1159**	1156	1148	1128
	online	1102	1135	1153	**1159**	1154	1146

Offline vs. online diversification

		1s	10s	100s	1000s	1200s	1500s
Coverage	offline	1056	1145	**1159**	1156	1148	1128
	online	1102	1135	1153	**1159**	1154	1146
Time Score	offline	**791.2**	690.7	420.3	86.8	59.2	25.9
	online	920.6	**929.7**	919.1	908.7	906.3	906.6