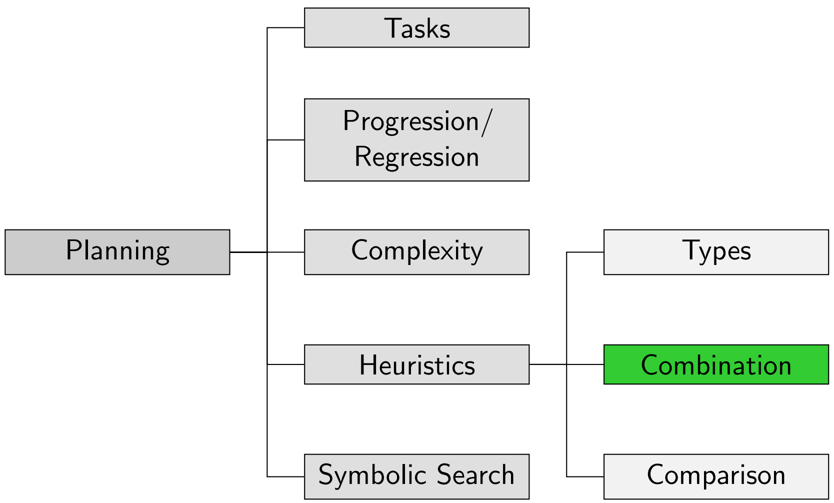
Planning and Optimization
F1. Cost Partitioning: Definition, Properties, and Abstractions

Malte Helmert and Gabriele Röger

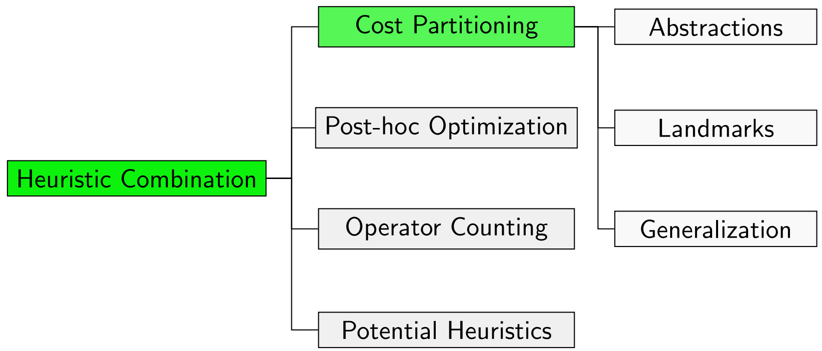
Universität Basel

December 4, 2017

Content of this Course



Content of this Course: Heuristic Combination



F1.1 Introduction

Exploiting Additivity

- Additivity allows to add up heuristic estimates admissibly. This gives better heuristic estimates than the maximum.
- For example, the canonical heuristic for PDBs sums up where addition is admissible (by an additivity criterion) and takes the maximum otherwise.
- Cost partitioning provides a more general additivity criterion, based on an adaption of the operator costs.

F1.2 Cost Partitioning

Cost Partitioning

Definition (Cost Partitioning)

Let Π be a planning task with operators O.

A cost partitioning for Π is a tuple $\langle cost_1, \dots, cost_n \rangle$, where

- $cost_i : O \to \mathbb{R}^+_0$ for $1 \leq i \leq n$ and
- $\sum_{i=1}^{n} cost_i(o) \leq cost(o)$ for all $o \in O$.

The cost partitioning induces a tuple $\langle \Pi_1, \dots, \Pi_n \rangle$ of planning tasks, where each Π_i is identical to Π except that the cost of each operator o is $cost_i(o)$.

Cost Partitioning: Admissibility (1)

Theorem (Sum of Solution Costs is Admissible)

Let Π be a planning task, $\langle cost_1, \dots, cost_n \rangle$ be a cost partitioning and $\langle \Pi_1, \dots, \Pi_n \rangle$ be the tuple of induced tasks.

Then the sum of the solution costs of the induced tasks is an admissible heuristic for Π, i.e., $\sum_{i=1}^{n} h^*_{\Pi_i} \leq h^*_{\Pi}$.

Cost Partitioning: Admissibility (2)

Proof of Theorem.

Let $\pi = \langle o_1, \dots, o_m \rangle$ be an optimal plan for state s of Π. Then

$$\begin{aligned}
\sum_{i=1}^{n} h^*_{\Pi_i}(s) &\leq \sum_{i=1}^{n} \sum_{j=1}^{m} cost_i(o_j) && (\pi \text{ plan in each } \Pi_i) \\
&= \sum_{j=1}^{m} \sum_{i=1}^{n} cost_i(o_j) && (\text{comm./ass. of sum}) \\
&\leq \sum_{j=1}^{m} cost(o_j) && (\text{cost partitioning}) \\
&= h^*_{\Pi}(s) && (\pi \text{ optimal plan in } \Pi)
\end{aligned}$$

□

Cost Partitioning Preserves Admissibility

In the rest of the chapter, we write h_{Π} to denote heuristic h evaluated on task Π.

Corollary (Sum of Admissible Estimates is Admissible)

Let Π be a planning task and let $\langle \Pi_1, \dots, \Pi_n \rangle$ be induced by a cost partitioning.

For admissible heuristics $h_1, \dots, h_n$, the sum $h(s) = \sum_{i=1}^{n} h_{i,\Pi_i}(s)$ is an admissible estimate for s in Π.

Cost Partitioning Preserves Consistency

Theorem (Cost Partitioning Preserves Consistency)

Let Π be a planning task and let $\langle \Pi_1, \dots, \Pi_n \rangle$ be induced by a cost partitioning $\langle cost_1, \dots, cost_n \rangle$.

If $h_1, \dots, h_n$ are consistent heuristics then $h = \sum_{i=1}^{n} h_{i,\Pi_i}$ is a consistent heuristic for Π.

Proof.

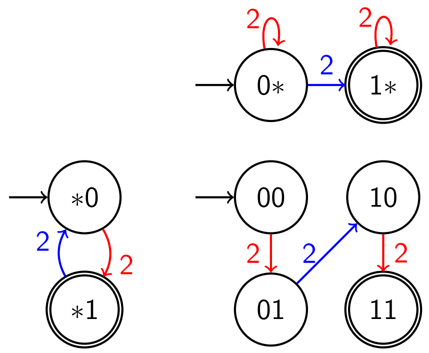
Let o be an operator that is applicable in state s.

$$\begin{aligned}
h(s) = \sum_{i=1}^{n} h_{i,\Pi_i}(s) &\leq \sum_{i=1}^{n} (cost_i(o) + h_{i,\Pi_i}(s[\![o]\!])) \\
&= \sum_{i=1}^{n} cost_i(o) + \sum_{i=1}^{n} h_{i,\Pi_i}(s[\![o]\!]) \leq cost(o) + h(s[\![o]\!])
\end{aligned}$$

□

Cost Partitioning: Example

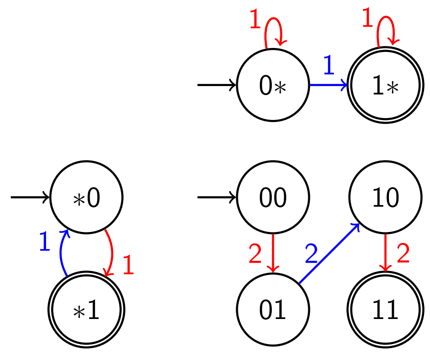
Example (No Cost Partitioning)



Heuristic value: $\max\{2, 2\} = 2$

Cost Partitioning: Example

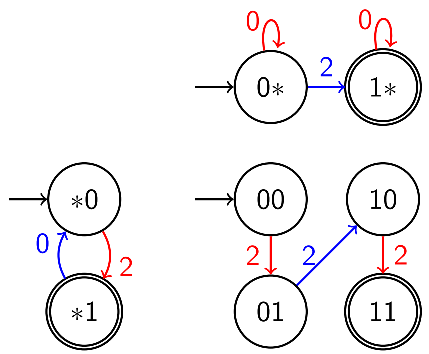
Example (Cost Partitioning 1)



Heuristic value: $1 + 1 = 2$

Cost Partitioning: Example

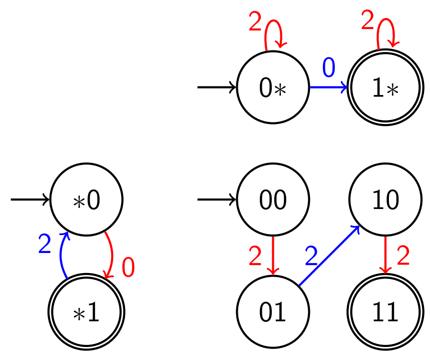
Example (Cost Partitioning 2)



Heuristic value: $2 + 2 = 4$

Cost Partitioning: Example

Example (Cost Partitioning 3)



Heuristic value: $0 + 0 = 0$

Cost Partitioning: Quality

- $h(s) = h_{1,\Pi_1}(s) + \cdots + h_{n,\Pi_n}(s)$ can be better or worse than any $h_{i,\Pi}(s)$
  $\rightarrow$ depending on cost partitioning
- strategies for defining cost-functions
  - uniform: $cost_i(o) = cost(o)/n$
  - zero-one: full operator cost in one copy, zero in all others
  - ...

Can we find an optimal cost partitioning?

Optimal Cost Partitioning

Optimal Cost Partitioning with LPs

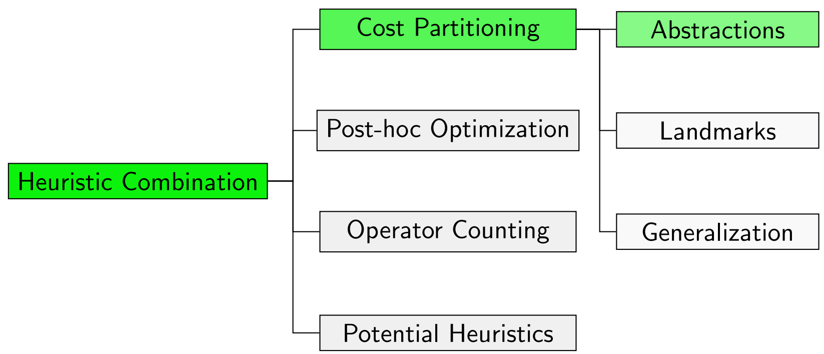
- Use variables for cost of each operator in each task copy
- Express heuristic values with linear constraints
- Maximize sum of heuristic values subject to these constraints

LPs known for

- abstraction heuristics
- landmark heuristic

F1.3 Optimal Cost Partitioning for Abstractions

Content of this Course: Heuristic Combination



Optimal Cost Partitioning for Abstractions

Abstractions

- Simplified versions of the planning task, e.g. projections
- Cost of optimal abstract plan is admissible estimate

How to express the heuristic value as linear constraints?
$\rightsquigarrow$ Shortest path problem in abstract transition system

LP for Shortest Path in State Space

Variables
Distance_s for each state s,
GoalDist

Objective
Maximize GoalDist

Subject to

$\text{Distance}_{s_I} = 0$ for the initial state s_I

$\text{Distance}_{s'} \leq \text{Distance}_s + cost(o)$ for all transitions $s \xrightarrow{o} s'$

$\text{GoalDist} \leq \text{Distance}_{s_\star}$ for all goal states $s_\star$

Optimal Cost Partitioning for Abstractions I

Variables
For each abstraction α:
Distance^α_s for each abstract state s,
cost^α_o for each operator o,
GoalDist^α

Objective
Maximize $\sum_\alpha \text{GoalDist}^\alpha$

...

Optimal Cost Partitioning for Abstractions II

Subject to
for all operators o

$\sum_\alpha \text{Cost}^\alpha_o \leq cost(o)$

$\text{Cost}^\alpha_o \geq 0$ for all abstractions α

and for all abstractions α

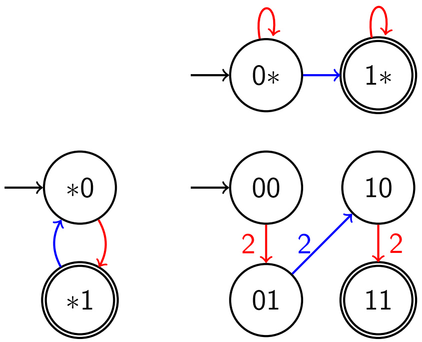
$\text{Distance}^\alpha_{s_I} = 0$ for the abstract initial state s_I

$\text{Distance}^\alpha_{s'} \leq \text{Distance}^\alpha_s + \text{Cost}^\alpha_o$ for all transition $s \xrightarrow{o} s'$

$\text{GoalDist}^\alpha \leq \text{Distance}^\alpha_{s_\star}$ for all abstract goal states $s_\star$

Example (1)

Example



Example (2)

Maximize $\text{GoalDist}^1 + \text{GoalDist}^2$ subject to

$$\begin{aligned}
\text{Cost}^1_{\text{red}} + \text{Cost}^2_{\text{red}} &\leq 2 \\
\text{Cost}^1_{\text{blue}} + \text{Cost}^2_{\text{blue}} &\leq 2 \\
\text{Cost}^1_{\text{red}} &\geq 0 \\
\text{Cost}^2_{\text{red}} &\geq 0 \\
\text{Cost}^1_{\text{blue}} &\geq 0 \\
\text{Cost}^2_{\text{blue}} &\geq 0 \qquad \ldots
\end{aligned}$$

Example (3)

. . . and . . .

$$\begin{aligned}
\text{Distance}^1_0 &= 0 \\
\text{Distance}^1_0 &\leq \text{Distance}^1_0 + \text{Cost}^1_{\text{red}} \\
\text{Distance}^1_1 &\leq \text{Distance}^1_0 + \text{Cost}^1_{\text{blue}} \\
\text{Distance}^1_1 &\leq \text{Distance}^1_1 + \text{Cost}^1_{\text{red}} \\
\text{GoalDist}^1 &\leq \text{Distance}^1_1
\end{aligned}$$

$$\begin{aligned}
\text{Distance}^2_0 &= 0 \\
\text{Distance}^2_1 &\leq \text{Distance}^2_0 + \text{Cost}^2_{\text{red}} \\
\text{Distance}^2_0 &\leq \text{Distance}^2_1 + \text{Cost}^2_{\text{blue}} \\
\text{GoalDist}^2 &\leq \text{Distance}^2_1
\end{aligned}$$

Caution

A word of warning

- optimization for every state gives best-possible cost partitioning
- but takes time

Better heuristic guidance often does not outweigh the overhead.

F1.4 Summary

Summary

- Cost partitioning allows to admissibly add up estimates of several heuristics.
- This can be better or worse than the best individual heuristic on the original problem, depending on the cost partitioning.
- For some heuristic classes, we know how to determine an optimal cost partitioning, using linear programming.
- Although solving a linear program is possible in polynomial time, the better heuristic guidance often does not outweigh the overhead.