

Planning and Optimization

D1. Cost Partitioning: Definition, Properties, and Abstractions

Malte Helmert and Gabriele Röger

Universität Basel

December 5, 2016

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.

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

- $cost_i : O \rightarrow \mathbb{R}_0^+$ 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

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



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.

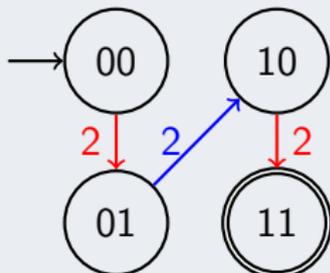
Let o be an operator that is applicable in state s .

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



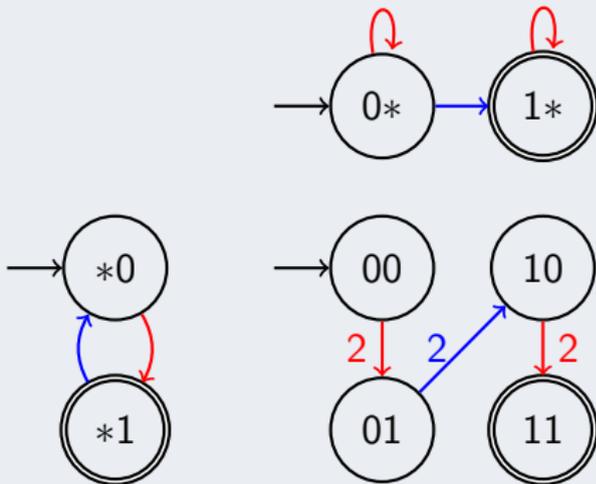
Cost Partitioning: Example

Example



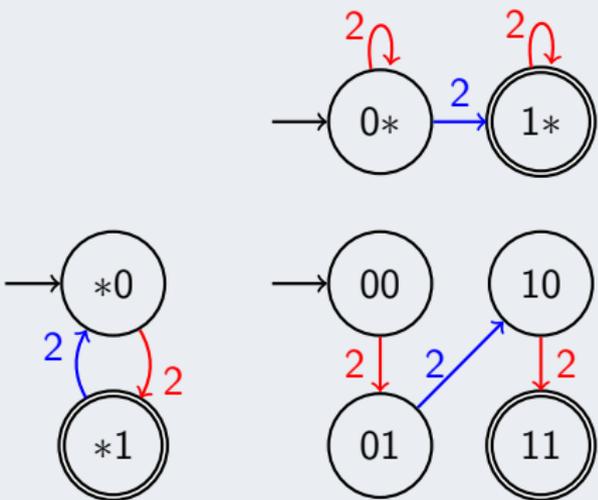
Cost Partitioning: Example

Example



Cost Partitioning: Example

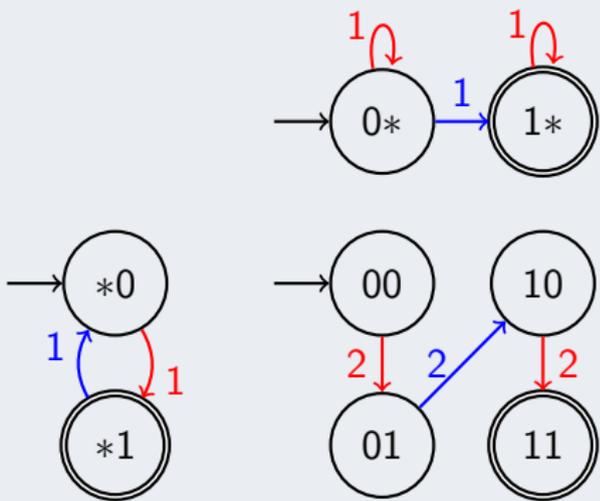
Example (No Cost Partitioning)



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

Cost Partitioning: Example

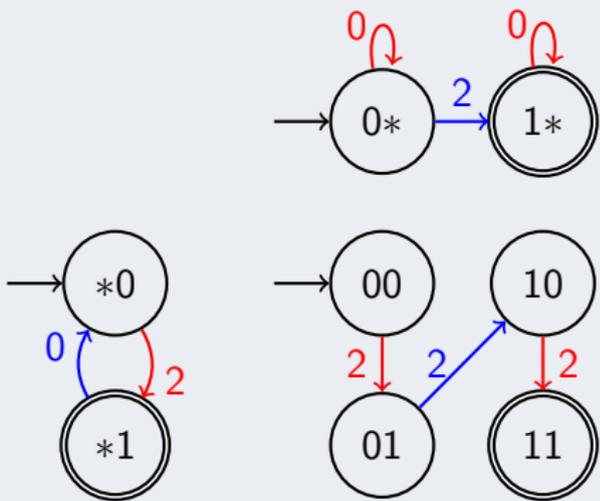
Example (Cost Partitioning 1)



Heuristic value: $1 + 1 = 2$

Cost Partitioning: Example

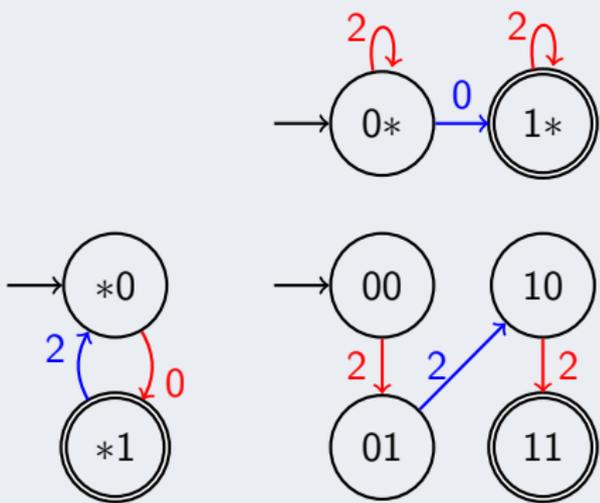
Example (Cost Partitioning 2)



Heuristic value: $2 + 2 = 4$

Cost Partitioning: Example

Example (Cost Partitioning 3)



Heuristic value: $0 + 0 = 0$

Cost Partitioning: Quality

- $h(s) = h_{1,\Pi_1}(s) + \dots + h_{n,\Pi_n}(s)$
can be **better or worse** than any $h_{i,\Pi}(s)$
→ 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

Optimal Cost Partitioning for Abstractions

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?

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?

↪ 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

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

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

Optimal Cost Partitioning for Abstractions I

Variables

For each abstraction α :

Distance_s^α for each abstract state s ,

cost_o^α 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

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

$$\text{Cost}_o^{\alpha} \geq 0$$

for all abstractions α

and for all abstractions α

$$\text{Distance}_{s_I}^{\alpha} = 0$$

for the abstract initial state s_I

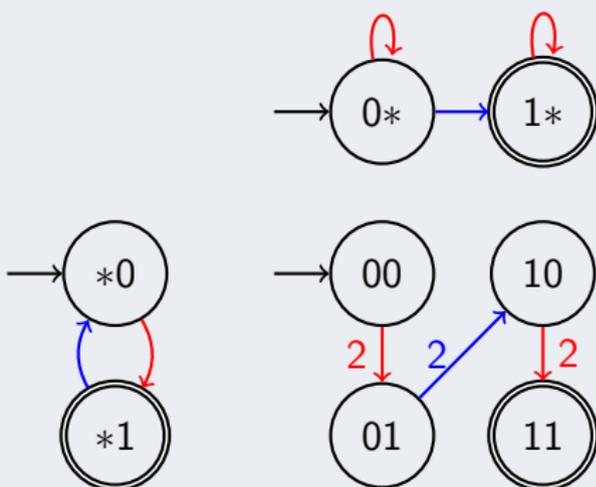
$$\text{Distance}_{s'}^{\alpha} \leq \text{Distance}_s^{\alpha} + \text{Cost}_o^{\alpha} \text{ for all transition } s \xrightarrow{o} s'$$

$$\text{GoalDist}^{\alpha} \leq \text{Distance}_{s_{\star}}^{\alpha}$$

for all abstract goal states s_{\star}

Example (1)

Example



Example (2)

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

$$\text{Cost}_{\text{red}}^1 + \text{Cost}_{\text{red}}^2 \leq 2$$

$$\text{Cost}_{\text{blue}}^1 + \text{Cost}_{\text{blue}}^2 \leq 2$$

$$\text{Cost}_{\text{red}}^1 \geq 0$$

$$\text{Cost}_{\text{red}}^2 \geq 0$$

$$\text{Cost}_{\text{blue}}^1 \geq 0$$

$$\text{Cost}_{\text{blue}}^2 \geq 0 \quad \dots$$

Example (3)

... and ...

$$\text{Distance}_0^1 = 0$$

$$\text{Distance}_0^1 \leq \text{Distance}_0^1 + \text{Cost}_{\text{red}}^1$$

$$\text{Distance}_1^1 \leq \text{Distance}_0^1 + \text{Cost}_{\text{blue}}^1$$

$$\text{Distance}_1^1 \leq \text{Distance}_1^1 + \text{Cost}_{\text{red}}^1$$

$$\text{GoalDist}^1 \leq \text{Distance}_1^1$$

$$\text{Distance}_0^2 = 0$$

$$\text{Distance}_1^2 \leq \text{Distance}_0^2 + \text{Cost}_{\text{red}}^2$$

$$\text{Distance}_0^2 \leq \text{Distance}_1^2 + \text{Cost}_{\text{blue}}^2$$

$$\text{GoalDist}^2 \leq \text{Distance}_1^2$$

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.

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.