D1. Cost Partitioning: Definition, Properties, and Abstractions

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.
D1.2 Cost Partitioning

Cost Partitioning: Admissibility (1)

Theorem (Sum of Solution Costs is Admissible)
Let \( \Pi \) be a planning task, \( \langle \text{cost}_1, \ldots, \text{cost}_n \rangle \) be a cost partitioning and \( \langle \Pi_1, \ldots, \Pi_n \rangle \) be the tuple of induced tasks. Then the sum of the solution costs of the induced tasks is an admissible heuristic for \( \Pi \), i.e., \( \sum_{i=1}^{n} h^*_i(s) \leq h^*_\Pi \).

Cost Partitioning: Admissibility (2)

Proof of Theorem.
Let \( \pi = \langle o_1, \ldots, o_m \rangle \) be an optimal plan for state \( s \) of \( \Pi \). Then
\[
\sum_{i=1}^{n} h^*_i(s) \leq \sum_{i=1}^{n} \sum_{j=1}^{m} \text{cost}_i(o_j) \quad (\pi \text{ plan in each } \Pi_i)
\]
\[
= \sum_{j=1}^{m} \sum_{i=1}^{n} \text{cost}_i(o_j) \quad (\text{comm./ass. of sum})
\]
\[
\leq \sum_{j=1}^{m} \text{cost}(o_j) \quad (\text{cost partitioning})
\]
\[
= h^*_\Pi(s) \quad (\pi \text{ optimal plan in } \Pi)
\]
Cost Partitioning Preserves Admissibility

In the rest of the chapter, we write \( h_\Pi \) to denote heuristic \( h \) evaluated on task \( \Pi \).

Corollary (Sum of Admissible Estimates is Admissible)

Let \( \Pi \) be a planning task and let \( \langle \Pi_1, \ldots, \Pi_n \rangle \) be induced by a cost partitioning.

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

Cost Partitioning Preserves Consistency

Theorem (Cost Partitioning Preserves Consistency)

Let \( \Pi \) be a planning task and let \( \langle \Pi_1, \ldots, \Pi_n \rangle \) be induced by a cost partitioning \( \langle \text{cost}_1, \ldots, \text{cost}_n \rangle \).

If \( h_1, \ldots, h_n \) are consistent heuristics then \( h = \sum_{i=1}^{n} h_i,\Pi_i \) is a consistent heuristic for \( \Pi \).

Proof.

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

\[
    h(s) = \sum_{i=1}^{n} h_i,\Pi_i(s) \leq \sum_{i=1}^{n} (\text{cost}_i(o) + h_i,\Pi_i(s\|o))) \\
    = \sum_{i=1}^{n} \text{cost}_i(o) + \sum_{i=1}^{n} h_i,\Pi_i(s\|o)) \leq \text{cost}(o) + h(s\|o)
\]

Cost Partitioning: Example

Example (No Cost Partitioning)

Heuristic value: \( \max\{2, 2\} = 2 \)

Example (Cost Partitioning 1)

Heuristic value: \( 1 + 1 = 2 \)
**Cost Partitioning: Example**

Example (Cost Partitioning 2)

```
0 0
+0 0 0
10 1
```

Heuristic value: $2 + 2 = 4$

Example (Cost Partitioning 3)

```
2 0
+0 0 0
10 1
```

Heuristic value: $0 + 0 = 0$

---

**Cost Partitioning: Quality**

- $h(s) = h_{1,\Pi_1}(s) + \ldots + h_{n,\Pi_n}(s)$
  - can be better or worse than any $h_{i,\Pi}(s)$
  - depending on cost partitioning
- strategies for defining cost-functions
  - uniform: $\text{cost}(o) = \text{cost}(o)/n$
  - 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
D1.3 Optimal Cost Partitioning for Abstractions

Optimal Cost Partitioning for 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
- Distance$_{s_i} = 0$ for the initial state \( s_i \)
- Distance$_{s'} \leq \text{Distance}_s + \text{cost}(o)$ for all transition \( s \xrightarrow{o} s' \)
- GoalDist \( \leq \text{Distance}_s \) for all goal states \( s_\star \)

Optimal Cost Partitioning for Abstractions I

Variables
For each abstraction \( \alpha \):
- Distance$_s^\alpha$ for each abstract state \( s \),
- \( \text{cost}^\alpha_o \) for each operator \( o \),
- GoalDist$^\alpha$

Objective
Maximize \( \sum_\alpha \text{GoalDist}^\alpha \)

\ldots
D1. Cost Partitioning: Definition, Properties, and Abstractions

Optimal Cost Partitioning for Abstractions II

Subject to

for all operators $o$

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

$$\text{Cost}_\alpha^o \geq 0$$

for all abstractions $\alpha$

and for all abstractions $\alpha$

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

for the abstract initial state $s_I$

$$\text{Distance}_\alpha^s \leq \text{Distance}_\alpha^{s'} + \text{Cost}_\alpha^o$$

for all transition $s \rightarrow s'$

$$\text{GoalDist}^\alpha \leq \text{Distance}_\alpha^s$$

for all abstract goal states $s^\star$

Example (1)

Example

Example (2)

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

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

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

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

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

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

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

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

▶ 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.