LP-based Heuristics for Cost-optimal Classical Planning

2. Cost Partitioning

Florian Pommerening   Gabriele Röger   Malte Helmert

ICAPS 2015 Tutorial

June 7, 2015

Cost Partitioning

Cost Partitioning

Idea 1: Cost Partitioning

- create copies $\Pi_1, \ldots, \Pi_n$ of planning task Π
- each has its own operator cost function $cost_i : \mathcal{O} \to \mathbb{R}_0^+$ (otherwise identical to Π)
- for all o: require $cost_1(o) + \cdots + cost_n(o) \leq cost(o)$

$\rightsquigarrow$ sum of solution costs in copies is admissible heuristic: $h^*_{\Pi_1} + \cdots + h^*_{\Pi_n} \leq h^*_{\Pi}$

Cost Partitioning

- for admissible heuristics $h_1, \ldots, h_n$,
  $h(s) = h_{1,\Pi_1}(s) + \cdots + h_{n,\Pi_n}(s)$
  is an admissible estimate
- $h(s)$ 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

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

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?
$\rightsquigarrow$ 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

$$\begin{aligned} \text{Distance}_{s_I} &= 0 && \text{for the initial state } s_I \\ \text{Distance}_{s'} &\leq \text{Distance}_s + cost(o) && \text{for all transition } s \xrightarrow{o} s' \\ \text{GoalDist} &\leq \text{Distance}_{s_\star} && \text{for all goal states } s_\star \end{aligned}$$

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

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

$$\text{Cost}^{\alpha}_{o} \geq 0 \qquad \text{for all abstractions } \alpha$$

and for all abstractions α

$$\text{Distance}^{\alpha}_{s_I} = 0 \qquad \text{for the abstract initial state } s_I$$

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

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

Optimal Cost Partitioning for Landmarks

Disjunctive action landmark

- Set of operators
- Every plan uses at least one of them
- Landmark cost = cost of cheapest operator

Optimal Cost Partitioning for Landmarks

Variables

Cost_L for each landmark L

Objective

Maximize $\sum_L \text{Cost}_L$

Subject to

$$\sum_{L: o \in L} \text{Cost}_L \leq cost(o) \qquad \text{for all operators } o$$

$$\text{Cost}_L \geq 0 \qquad \text{for all landmarks } L$$

Optimal Cost Partitioning for Landmarks (Dual view)

Variables

Occurrences$_o$ for each operator o

Objective

Minimize $\sum_o$ Occurrences$_o \cdot cost(o)$

Subject to

$$\sum_{o \in L} \text{Occurrences}_o \geq 1 \text{ for all landmarks } L$$

$$\text{Occurrences}_o \geq 0 \text{ for all operators } o$$

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.

General Cost Partitioning

General Cost Partitioning

Cost functions usually non-negative

- We tacitly also required this for task copies
- Makes intuitively sense: original costs are non-negative
- But: not necessary for cost-partitioning!

General Cost Partitioning

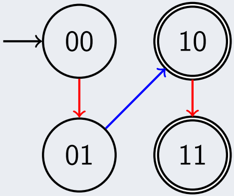
General Cost Partitioning

- Create copies $\Pi_1, \ldots, \Pi_n$ of planning task Π
- each has its own operator cost function $cost_i : \mathcal{O} \to \mathbb{R}$ (otherwise identical to Π)
- for all o: require $cost_1(o) + \cdots + cost_n(o) \leq cost(o)$

⇝ sum of solution costs in copies is admissible heuristic: $h^*_{\Pi_1} + \cdots + h^*_{\Pi_n} \leq h^*_{\Pi}$

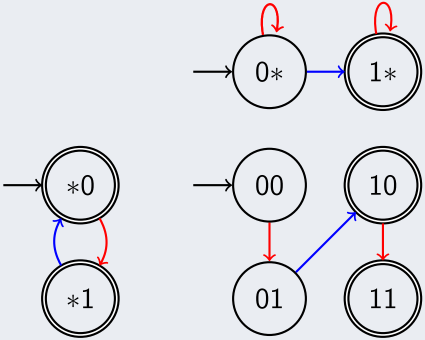
General Cost Partitioning: Example

Example



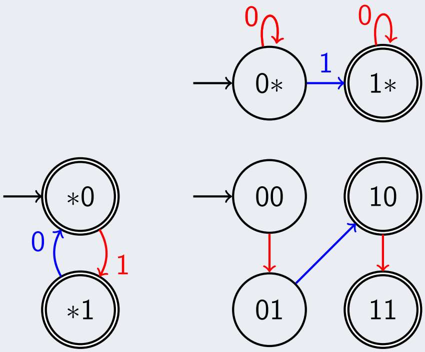
General Cost Partitioning: Example

Example



General Cost Partitioning: Example

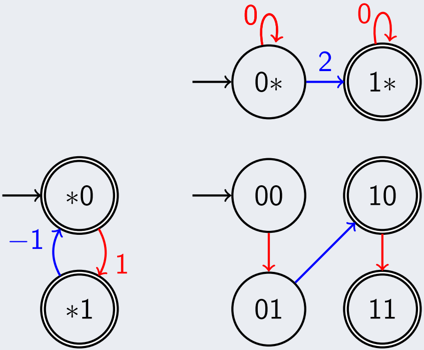
Example



Heuristic value: $0 + 1 = 1$

General Cost Partitioning: Example

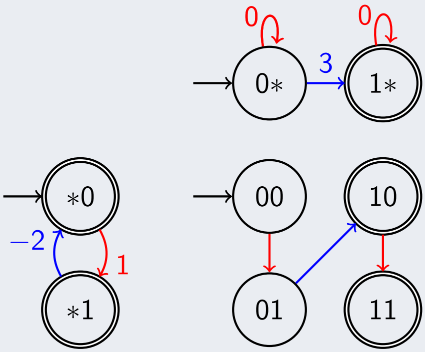
Example



Heuristic value: $0 + 2 = 2$

General Cost Partitioning: Example

Example



Heuristic value: $-\infty + 3 = -\infty$

General Cost Partitioning: Conclusion

- More powerful than non-negative cost partitioning
- Optimal general cost partitioning:
  omit constraints to non-negative cost variables
  - optimal cost partitioning maximizes objective value
  - removing constraints can only increase heuristic value
- Next section: connection to operator counting

Tutorial Structure

1. ~~Introduction and Overview~~
2. ~~Cost Partitioning~~
3. Operator Counting
4. Potential Heuristics