Planning and Optimization

D3. Post-hoc Optimization

Malte Helmert and Gabriele Röger

Universität Basel

December 8, 2016

Planning and Optimization

December 8, 2016 — D3. Post-hoc Optimization

D3.1 Introduction

Combining Estimates from Abstraction Heuristics

- Pattern databases grow exponentially with the number of variables in the pattern.
- Instead of one large pattern, planners use collections of multiple smaller patterns.
- We alrady know two approaches to derive heuristic estimates from a pattern collection:
  - Canonical heuristic
  - Optimal cost partitioning

Can we do better than these approaches?

Reminder: The Canonical Heuristic Function

If for a set of patterns no operator affects more than one pattern, the sum of the heuristic estimates is admissible.

Definition (Canonical Heuristic Function)

Let Π be an FDR planning task. Let $\mathcal{C}$ be a pattern collection for Π and let $cliques(\mathcal{C})$ denote the set of all maximal additive subsets of $\mathcal{C}$. The canonical heuristic $h^{\mathcal{C}}$ for $\mathcal{C}$ is defined as

$$h^{\mathcal{C}}(s) = \max_{\mathcal{D} \in cliques(\mathcal{C})} \sum_{P \in \mathcal{D}} h^P(s).$$

For a given pattern collection, the canonical heuristic is the best possible admissible heuristic not using cost partitioning.

Reminder: Optimal Cost Partitioning for Abstractions

Optimal cost partitioning for abstractions...

- ...uses a state-specific LP to find the best possible cost partitioning, and sums up the heuristic estimates.
- ...dominates the canonical heuristic, i.e.. for the same pattern collection, it never gives lower estimates than $h^{\mathcal{C}}$.
- ...is very expensive to compute (recomputing the PDBs in every state).

Example Task (1)

Example (Example Task)

SAS$^+$ task $\Pi = \langle V, I, O, \gamma \rangle$ with

- $V = \{A, B, C\}$ with $\mathrm{dom}(v) = \{0, 1, 2, 3, 4\}$ for all $v \in V$
- $I = \{A \mapsto 0, B \mapsto 0, C \mapsto 0\}$
- $O = \{inc_x^v \mid v \in V, x \in \{0, 1, 2\}\} \cup \{jump^v \mid v \in V\}$
  - $inc_x^v = \langle v = x, v := x + 1, 1 \rangle$
  - $jump^v = \langle \bigwedge_{v' \in V : v' \neq v} v' = 4, v := 3, 1 \rangle$
- $\gamma = A = 3 \land B = 3 \land C = 3$

- Each optimal plan consists of three increment operators for each variable $\rightsquigarrow h^*(I) = 9$
- Each operator affects only one variable.

Example Task (2)

- In projections on single variables we can reach the goal with a *jump* operator: $h^{\{A\}}(I) = h^{\{B\}}(I) = h^{\{C\}}(I) = 1$.
- In projections on more variables, we need for each variable three applications of increment operators to reach the abstract goal from the abstract initial state: $h^{\{A,B\}}(I) = h^{\{A,C\}}(I) = h^{\{B,C\}}(I) = 6$

Example (Canonical Heuristic)

$\mathcal{C} = \{\{A\}, \{B\}, \{C\}, \{A, B\}, \{A, C\}, \{B, C\}\}$

$$h^{\mathcal{C}}(s) = \max\{h^{\{A\}}(s) + h^{\{B\}}(s) + h^{\{C\}}(s), h^{\{A\}}(s) + h^{\{B,C\}}(s), h^{\{B\}}(s) + h^{\{A,C\}}(s), h^{\{C\}}(s) + h^{\{A,B\}}(s)\}$$

$h^{\mathcal{C}}(I) = 7$

D3.2 Post-hoc Optimization Heuristic

Post-hoc Optimization Heuristic: Idea

Consider the example task:

- *type-v* operator: operator modifying variable v
- $h^{\{A,B\}} = 6$
  $\Rightarrow$ any plan contains at least 6 operators of type A or B.
- $h^{\{A,C\}} = 6$
  $\Rightarrow$ any plan contains at least 6 operators of type A or C.
- $h^{\{B,C\}} = 6$
  $\Rightarrow$ any plan contains at least 6 operators of type B or C.
- $\Rightarrow$ at least 9 operators in any plan

Can we generalize this kind of reasoning?

Post-hoc Optimization Heuristic: Linear Program (1)

Construct linear program for pattern collection $\mathcal{C}$:

- variable X_o for each operator $o \in O$
- intuitively X_o is cost incurred by operator o
- PDB heuristics are admissible

$$h^P(s) \leq \sum_{o \in O} X_o \text{ for each pattern } P \in \mathcal{C}$$

- can tighten these constraints to

$$h^P(s) \leq \sum_{o \in O : o \text{ affects } P} X_o$$

Post-hoc Optimization Heuristic: Linear Program (2)

For pattern collection $\mathcal{C}$:

Variables

X_o for each operator $o \in O$

Objective

Minimize $\sum_{o \in O} X_o$

Subject to

$$\begin{aligned} \sum_{o \in O : o \text{ affects } P} X_o &\geq h^P(s) && \text{for all patterns } P \in \mathcal{C} \\ X_o &\geq 0 && \text{for all } o \in O \end{aligned}$$

Post-hoc Optimization Heuristic: Simplifying the LP

- Reduce size of LP by aggregating variables which always occur together in constraints.
- Happens when several operators are relevant for exactly the same PDBs.
- Partitioning $O/\!\sim$ induced by this equivalence relation
- One variable $X_{[o]}$ for each $[o] \in O/\!\sim$

Post-hoc Optimization Heuristic: Definition

Definition (Post-hoc Optimization Heuristic)

The post-hoc optimization heuristic $h^{\mathrm{PhO}}_{\mathcal{C}}$ for pattern collection $\mathcal{C}$ is the objective value of the following linear program:

$$\text{Minimize} \sum_{[o] \in O/\sim} X_{[o]} \text{ subject to}$$

$$\begin{aligned} \textstyle\sum_{[o] \in O/\sim : o \text{ affects } P} X_{[o]} &\geq h^P(s) && \text{for all } P \in \mathcal{C} \\ X_{[o]} &\geq 0 && \text{for all } [o] \in O/\!\sim, \end{aligned}$$

where $o \sim o'$ iff o and o' affect the same patterns in $\mathcal{C}$.

- Precompute PDBs for all $P \in \mathcal{C}$.
- Create LP for initial state.
- For each new state, just change the bounds $h^P(s)$.

Post-hoc Optimization Heuristic: Admissibility

Theorem (Admissibility)

The post-hoc optimization heuristic is admissible.

Proof.

Let Π be a planning task and $\mathcal{C}$ be a pattern collection.
Let π be an optimal plan for state s and let $cost_\pi(O')$ be the cost incurred by operators from $O' \subseteq O$ in π.

Setting each $X_{[o]}$ to $cost_\pi([o])$ is a feasible variable assignment: Constraints $X_{[o]} \geq 0$ are satisfied. For each $P \in \mathcal{C}$, π is a solution in the abstract transition system and the sum in the corresponding constraint equals the cost of the "true" abstract state transitions (i.e.. not accounting for self-loops). As $h^P(s)$ corresponds to the cost of an optimal solution in the abstraction, the inequality holds.

For this assignment the objective function has value $h^*(s)$ (cost of π), so the objective value of the LP is admissible. □

Post-hoc Optimization Heuristic: Insight

Corresponding dual program to h^{PhO} LP:

$$\text{Maximize } \textstyle\sum_{P \in \mathcal{C}} Y_P h^P(s) \text{ subject to}$$

$$\begin{aligned} \sum_{P \in \mathcal{C} : o \text{ affects } P} Y_P &\leq 1 && \text{for all } [o] \in O/\!\sim \\ Y_P &\geq 0 && \text{for all } P \in \mathcal{C}. \end{aligned}$$

We compute a state-specific cost partitioning that can only scale the operator costs within each heuristic by a factor Y_i.

Relation to Canonical Heuristic

Theorem

Consider the dual D of the LP solved by $h^{\mathrm{PhO}}_{\mathcal{C}}$ in state s for a given pattern collection $\mathcal{C}$. If we restrict the variables in D to integers, the objective value is the canonical heuristic value $h^{\mathcal{C}}(s)$.

Corollary

The post-hoc optimization heuristic $h^{\mathrm{PhO}}_{\mathcal{C}}$ dominates the canonical heuristic $h^{\mathcal{C}}$ for the same pattern collection $\mathcal{C}$.

Post-hoc Optimization Heuristic: Remarks

- For the canonical heuristic, we need to find all maximal cliques, which is an NP-hard problem.
- The post-hoc optimization heuristic dominates the canonical heuristic and can be computed in polynomial time.
- With post-hoc optimization, we can handle much larger pattern collections than found with the iPDB procedure.
- For the approach it is better to use a large number of small patterns, e.g., all patterns up to size 2 that satisfy the same relevance criteria as used for the iPDB patterns.
- Post-hoc optimization is not limited to PDBs but there is a straightforward extension to any admissible heuristic for which we can determine the "relevant" operators.

D3.3 Summary

Summary

- Post-hoc optimization heuristic explores middle ground between canonical heuristic and optimal cost partitioning.
- For the same pattern collection the post-hoc optimization heuristic dominates the canonical heuristic.
- The computation can be done in polynomial time.

Literature

References on post-hoc optimization:

Florian Pommerening, Gabriele Röger and Malte Helmert.
Getting the Most Out of Pattern Databases for Classical Planning.
Proc. IJCAI 2013, pp. 2357–2364, 2013.
Introduces post-hoc optimization and points out relation to canonical heuristic.