Planning and Optimization

F3. Post-hoc Optimization & Operator Counting

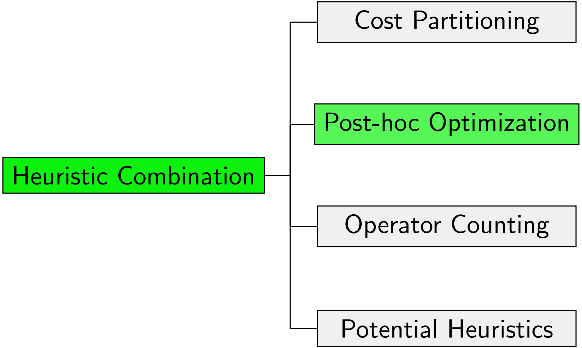
Malte Helmert and Gabriele Röger

Universität Basel

December 6, 2017

Post-hoc Optimization Heuristic

Content of this Course: Heuristic Combination



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 $v \in V$
- $I = \{A \mapsto 0, B \mapsto 0, C \mapsto 0\}$
- $O = \{inc_x^v \mid v \in V, x \in \{0, 1, 2\}\} \cup \{jump^v \mid v \in V\}$
    - $inc_x^v = \langle v = x, v := x + 1, 1 \rangle$
    - $jump^v = \langle \bigwedge_{v' \in V : v' \neq v} v' = 4, v := 3, 1 \rangle$
- $\gamma = A = 3 \wedge B = 3 \wedge C = 3$

- 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)

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

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

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

Post-hoc Optimization Heuristic: Idea

Consider the example task:

- *type-v* operator: operator modifying variable *v*

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.

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.

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

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

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

$$X_o \geq 0 \quad \text{for all } o \in O$$

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 $h_{\mathcal{C}}^{\mathrm{PhO}}$ for pattern collection $\mathcal{C}$ is the objective value of the following linear program:

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

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

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:

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

$$\sum_{P \in \mathcal{C} : o \text{ affects } P} Y_P \leq 1 \qquad \text{for all } [o] \in O/\!\sim$$

$$Y_P \geq 0 \qquad \text{for all } P \in \mathcal{C}.$$

Post-hoc Optimization Heuristic: Insight

Corresponding dual program to h^{PhO} LP:

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

$$\sum_{P \in \mathcal{C}: o \text{ affects } P} Y_P \leq 1 \qquad \text{for all } [o] \in O/{\sim}$$

$$Y_P \geq 0 \qquad \text{for all } P \in \mathcal{C}.$$

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 $h^{\mathcal{C}}(s)$.

Relation to Canonical Heuristic

Theorem

Consider the dual D of the LP solved by $h_{\mathcal{C}}^{\mathrm{PhO}}$ 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

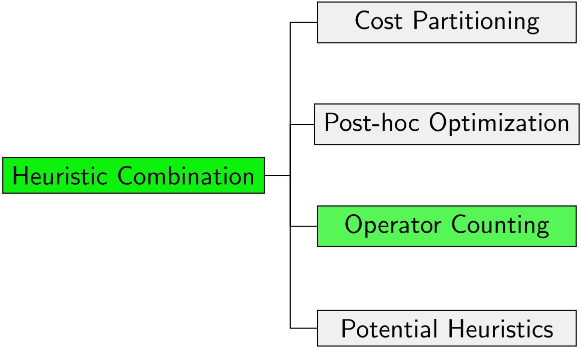
The post-hoc optimization heuristic $h_{\mathcal{C}}^{\mathrm{PhO}}$ dominates the canonical heuristic $h^{\mathcal{C}}$ for the same pattern collection $\mathcal{C}$.

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.

Operator-counting Framework

Content of this Course: Heuristic Combination



Reminder: Optimal Cost Partitioning for Landmarks

Variables

Count_o for each operator o

Objective

Minimize $\sum_o \text{Count}_o \cdot cost(o)$

Subject to

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

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

Numbers of operator occurrences in any plan satisfy constraints.
Minimizing the total plan cost gives an admissible estimate.
Can we apply this idea more generally?

Operator Counting

Operator-counting Constraints

- linear constraints whose variables denote number of occurrences of a given operator
- must be satisfied by every plan

Examples:

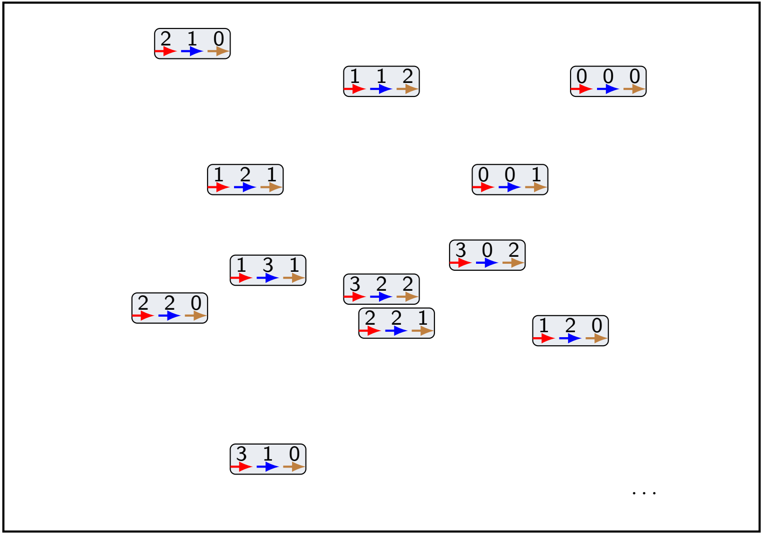
- $\text{Count}_{o_1} + \text{Count}_{o_2} \geq 1$ "must use o_1 or o_2 at least once"
- $\text{Count}_{o_1} - \text{Count}_{o_3} \leq 0$ "cannot use o_1 more often than o_3"

Motivation:

- declarative way to represent knowledge about solutions
- allows reasoning about solutions to derive heuristic estimates

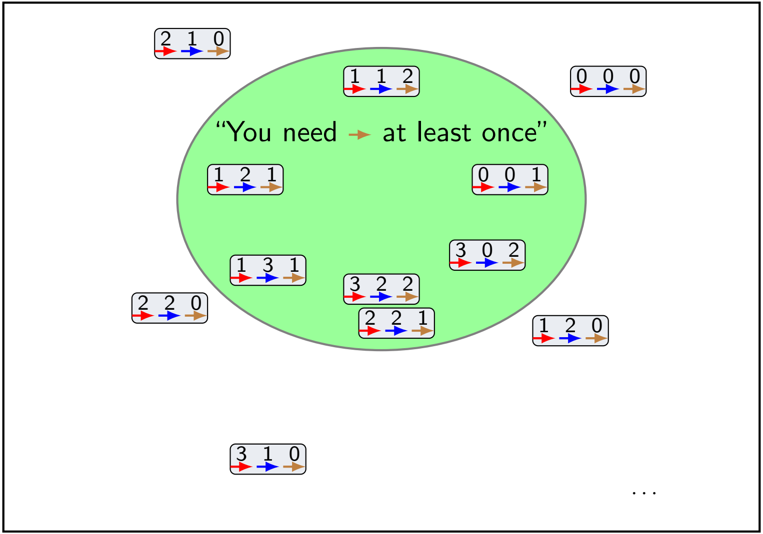
Operator Counting Heuristics

Operator occurrences in potential plans



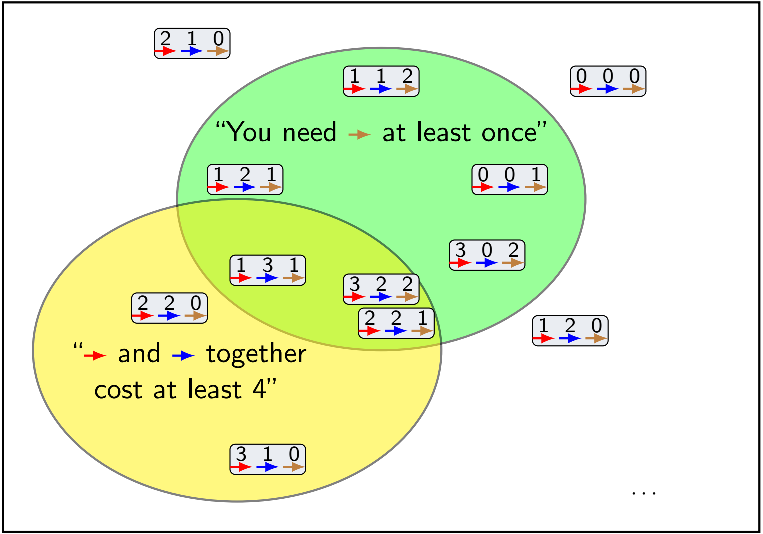
Operator Counting Heuristics

Operator occurrences in potential plans



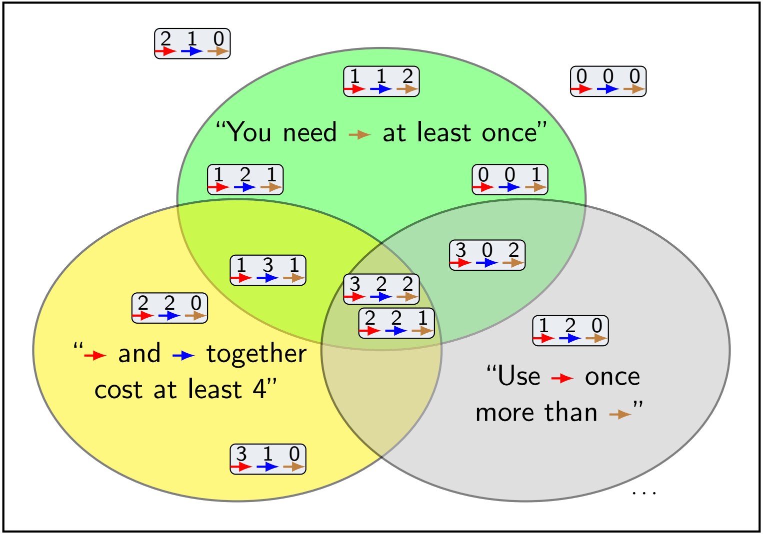
Operator Counting Heuristics

Operator occurrences in potential plans



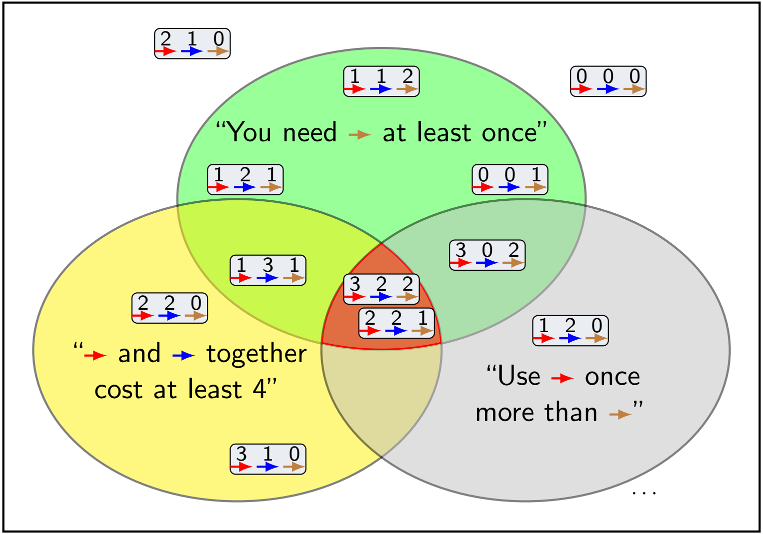
Operator Counting Heuristics

Operator occurrences in potential plans



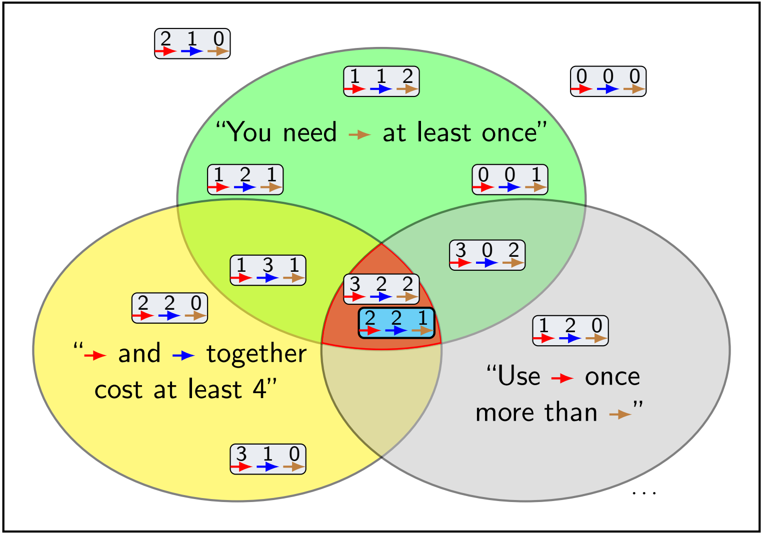
Operator Counting Heuristics

Operator occurrences in potential plans



Operator Counting Heuristics

Operator occurrences in potential plans



Operator-counting Constraint

Definition (Operator-counting Constraints)

Let Π be a planning task with operators O and let s be a state. Let $\mathcal{V}$ be the set of integer variables Count_o for each $o \in O$.

A linear inequality over $\mathcal{V}$ is called an operator-counting constraint for s if for every plan π for s setting each Count_o to the number of occurrences of o in π is a feasible variable assignment.

Operator-counting Heuristics

Definition (Operator-counting IP/LP Heuristic)

The operator-counting integer program IP_C for a set C of operator-counting constraints for state s is

$$\text{Minimize} \sum_{o \in O} \mathrm{Count}_o \cdot cost(o) \text{ subject to}$$

$$C \text{ and } \mathrm{Count}_o \geq 0 \text{ for all } o \in O,$$

where O is the set of operators.

The IP heuristic h_C^{IP} is the objective value of IP_C,
the LP heuristic h_C^{LP} is the objective value of its LP-relaxation.

If the IP/LP is infeasible, the heuristic estimate is ∞.

Admissibility

Theorem (Operator-counting Heuristics are Admissible)

The IP and the LP heuristic are admissible.

Proof.

Let C be a set of operator-counting constraints for state s and π be an optimal plan for s. The number of operator occurrences of π are a feasible solution for C. As the IP/LP minimizes the total plan cost, the objective value cannot exceed the cost of π and is therefore an admissible estimate. □

Dominance

Theorem

Let C and C' be operator-counting constraints for s and let $C \subseteq C'$. Then $\mathrm{IP}_C \leq \mathrm{IP}_{C'}$ and $\mathrm{LP}_C \leq \mathrm{LP}_{C'}$.

Proof.

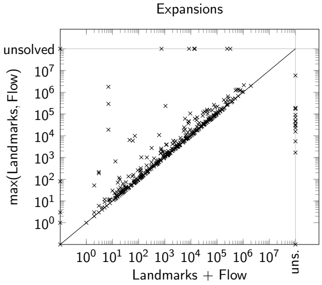
Every feasible solution of C' is also feasible for C. As the LP/IP is a minimization problem, the objective value subject to C can therefore not be larger than the one subject to C'. □

Adding more constraints can only improve the heuristic estimate.

Combining Heuristics

Combination of two heuristics

- Use both operator-counting constraints
- Combination always dominates individual heuristics
- Positive interaction between constraints



Combination often better than best individual heuristic

Constraints from Disjunctive Action Landmarks

Optimal cost partitioning for disjunctive action landmarks

- Use one landmark constraint per landmark

Landmark constraint for landmark L

$$\sum_{o \in L} \text{Count}_o \geq 1$$

Constraints from Flow Heuristic

Flow heuristic

- Use one flow constraint per atom

Flow Constraint for atom a

$$[a \in s] + \sum_{o \in O : a \in \mathit{eff}(o)} \text{Count}_o = [a \in \gamma] + \sum_{o \in O : a \in \mathit{pre}(o)} \text{Count}_o$$

Remark: Assumes transition normal form (not a limitation)

Constraints from Post-hoc Optimization Heuristic

Post-hoc optimization heuristic

- X_o for cost incurred by operator o
- Replace each such variable with $\text{Count}_o \cdot cost(o)$ to fit the operator-counting framework.
- Use one post-hoc optimization constraint per sub-heuristic

Post-hoc optimization constraint for heuristic h

$$\sum_{o \text{ is relevant for } h} \text{Count}_o \cdot cost(o) \geq h(s)$$

Further Examples?

- The definition of operator-counting constraints can be extended to groups of constraints and auxiliary variables.
- With this extended definition we could also cover
  - optimal cost partitioning for abstractions, and
  - the perfect relaxation heuristic h^+.

Summary

Summary

- Post-hoc optimization heuristic explores the 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.

- Many heuristics can be formulated in terms of operator-counting constraints.
- The operator-counting heuristic framework allows to combine the constraints and to reason on the entire encoded declarative knowledge.
- The heuristic estimate for the combined constraints can be better than the one of the best ingredient heuristic but never worse.

Literature (1)

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.

Blai Bonet.
An Admissible Heuristic for SAS+ Planning Obtained from the State Equation.
Proc. IJCAI 2013, pp. 2268–2274, 2013.
Suggests combination of flow constraints and landmark constraints.

Literature (2)

Tatsuya Imai and Alex Fukunaga.
A Practical, Integer-linear Programming Model for the Delete-relaxation in Cost-optimal Planning.
Proc. ECAI 2014, pp. 459–464, 2014.
IP formulation of h^+.

Florian Pommerening, Gabriele Röger, Malte Helmert and Blai Bonet.
LP-based Heuristics for Cost-optimal Planning.
Proc. ICAPS 2014, pp. 226–234, 2014.
Systematic introduction of operator-counting framework.