Getting the Most Out of Pattern Databases for Classical Planning

Florian Pommerening, Gabriele Röger, Malte Helmert
University of Basel

Motivation and Abstract

State of the art for computing additive abstraction heuristics:
iPDB procedure [Haslum et al., 2007]

- Combines multiple patterns in the canonical heuristic

Can we obtain stronger heuristic estimates?

Running Example

- Three variables $\{A, B, C\}$
- Each operator affects only one variable
- Pattern databases

$$h^{\{A\}}(s) = h^{\{B\}}(s) = h^{\{C\}}(s) = 1$$
$$h^{\{A,B\}}(s) = h^{\{A,C\}}(s) = h^{\{B,C\}}(s) = 6$$

- What is the best heuristic value we can get from this information?

Canonical Heuristic

Sums where possible, maximizes where necessary:

$$h^{\mathcal{C}}(s) = \max_{\mathcal{A} \in MAS(\mathcal{C})} \sum_{P \in \mathcal{A}} h^P(s)$$

Example

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

$$\begin{aligned} h^{\mathcal{C}}(s) = \max\{ & h^{\{A\}}(s) + h^{\{B\}}(s) + h^{\{C\}}(s), \\ & h^{\{A\}}(s) + h^{\{B,C\}}(s), \\ & h^{\{B\}}(s) + h^{\{A,C\}}(s), \\ & h^{\{C\}}(s) + h^{\{A,B\}}(s)\} = 7 \end{aligned}$$

Post-hoc Optimization Heuristic: Idea

Linear program for pattern collection $\mathcal{C}$:

- Variable X_o for cost incurred by operator o
- $X_o \geq 0$ for all variables
- PDB heuristics admissible

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

- Can tighten to

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

- Total cost of the plan is $\sum_{o \in \mathcal{O}} X_o$
- Minimizing total cost leads to admissible estimate
- Optimization: merge variables in equivalence classes

Example

$$\begin{array}{rcll} 6 & = h^{\{A,B\}}(s) \leq & \text{cost}_A + \text{cost}_B & \\ 6 & = h^{\{A,C\}}(s) \leq & \text{cost}_A + & \text{cost}_C \\ 6 & = h^{\{B,C\}}(s) \leq & \text{cost}_B + & \text{cost}_C \\ \hline 18 & \leq & 2\ \text{cost}_A + 2\ \text{cost}_B + 2\ \text{cost}_C & \\ 9 & \leq & \text{cost}_A + \text{cost}_B + \text{cost}_C & \end{array}$$

Post-hoc Optimization Heuristic: General

Minimize $\sum_{o \in \mathcal{O}} X_o$ subject to

$$\sum_{o \in \mathcal{O}: o \text{ affects } P} X_o \geq h^P(s) \quad \text{for each pattern } P$$
$$X_o \geq 0 \quad \text{for each operator } o \in \mathcal{O}.$$

Insights from Dual Program

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

$$\sum_{P \in \mathcal{C}: o \text{ affects } P} Y_P \leq 1 \quad \text{for each operator } o$$
$$Y_P \geq 0 \quad \text{for each pattern } P \in \mathcal{C}.$$

- State-specific cost partitioning: can only scale operator costs within each heuristic by a factor Y_P.
- Restriction to integer variables: canonical heuristic

Theorem

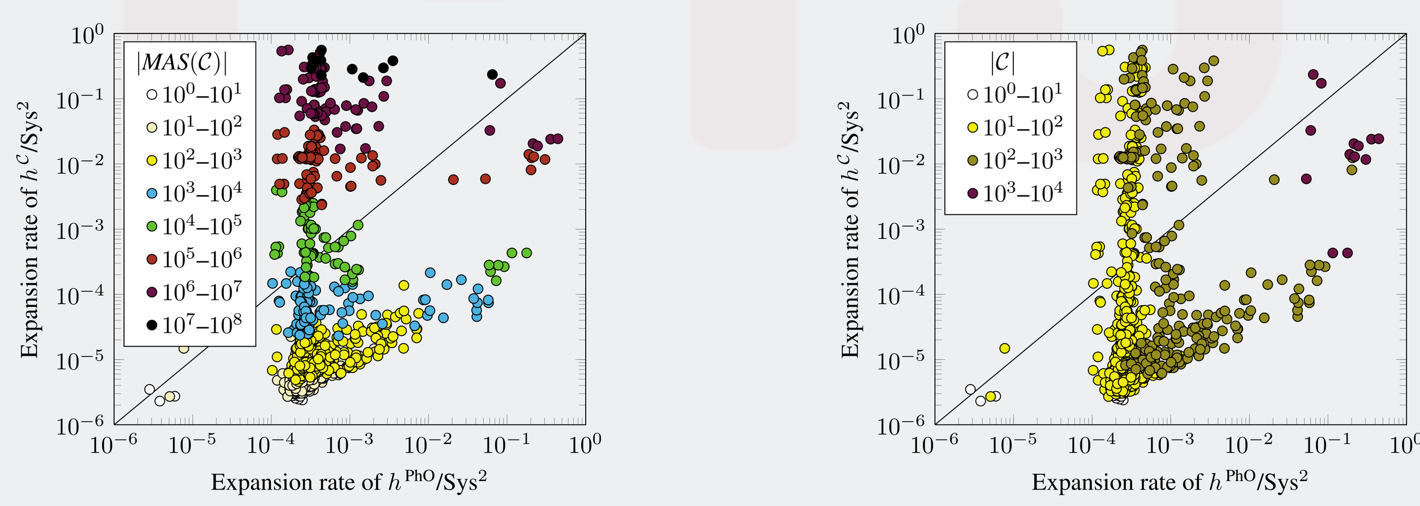
The post-hoc optimization heuristic is admissible and dominates the canonical heuristic for the same pattern collection.

Experimental Results

	HC $h^{\mathcal{C}}$	HC h^{PhO}	HC h^{OCP}	Sys[1] $h^{\mathcal{C}}$	Sys[1] h^{PhO}	Sys[2] $h^{\mathcal{C}}$	Sys[2] h^{PhO}	Sys[2] h^{OCP}	Sys* h^{PhO}	$h^{\text{LM-Cut}}$
barman (20)	4	4	0	4	4	4	4	0	4 (1–2)	4
elevators (20)	16	16	0	9	9	16	15	0	15 (2)	18
floortile (20)	2	2	0	2	2	2	2	0	2 (1–3)	7
nomystery (20)	16	16	3	12	12	18	18	6	19 (3–4)	14
openstacks (20)	14	14	5	14	14	5	14	0	14 (1–2)	14
parcprinter (20)	8	8	8	11	11	7	13	15	20 (4)	13
parking (20)	5	5	1	5	5	0	1	0	5 (1)	3
pegsol (20)	0	0	0	17	17	5	17	1	17 (1–2)	17
scanalyzer (20)	10	10	1	10	10	10	8	1	10 (1)	12
sokoban (20)	20	20	18	19	19	20	20	2	20 (2)	20
tidybot (20)	14	14	6	13	13	14	14	6	14 (2–3)	14
transport (20)	6	6	2	6	6	6	6	0	8 (3)	6
visitall (20)	16	16	10	16	16	16	16	10	16 (1–3)	11
woodworking (20)	2	2	2	5	5	3	10	2	10 (2)	12
Sum IPC 2011 (280)	133	133	56	143	143	126	158	43	174	165
IPC 1998–2008 (1116)	456	459	241	449	449	446	475	231	501	598
Sum (1396)	589	592	297	592	592	572	633	274	675	763

Additional evaluation on Sys^2 pattern collections:

- Theoretical dominance of the post-hoc optimization heuristic translates into better guidance on only a few domains.
- On tasks solved by both approaches, the *canonical heuristic computations tend to be faster.*



- Tasks solved by h^{PhO} but not by $h^{\mathcal{C}}$ ran out of memory during generation of maximal additive pattern sets.

Conclusion

Post-hoc optimization heuristic explores middle ground between "trivial cost partitioning" of canonical heuristic and optimal cost partitioning.