AI Planning

16. Combining Heuristic Functions Part II - LP Heuristics

"Beyond Optimal Cost Partitioning"

Álvaro Torralba, Cosmina Croitoru

SAARLAND UNIVERSITY

COMPUTER SCIENCE

Winter Term 2018/2019

Thanks to Prof. Malte Helmert and Dr. Gabrielle Röger for slide sources

Agenda

Our Agenda for This Chapter

2. **Post-hoc optimization:** Combining heuristics with a subset of relevant actions.
3. **Operator Counting Heuristics:** Interpreting heuristics as constraints on the set of actions that must be applied.
4. **Potential Heuristics:** Deriving heuristics via linear programming.

Combining Estimates from Abstraction Heuristics

We already know two approaches to derive heuristic estimates from a collection of abstractions:

(Reminder **Chapter 12**): Canonical Heuristic Function $h^{\mathcal{C}}(s) = \max_{\mathcal{D} \in \text{cliques}(\mathcal{C})} \sum_{P \in \mathcal{D}} h^P(s)$.

(Reminder **Chapter 16**): Optimal Cost Partitioning: uses a state-specific LP to find the best possible cost partitioning, and sums up the heuristic estimates.
$\rightarrow$is very expensive to compute (recomputing the PDBs in every state).

Is there a way to combine multiple abstraction heuristics that does not need to re-compute the distances in every state and is more informative than the canonical heuristic?

Example Task (1)

Example (Example Task)

FDR task $\Pi = \langle V, I, O, \gamma \rangle$ with

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

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

Example Task (2)

- In projections on single variables we can reach the goal with a *jump* action: $h^{\{A\}}(I) = h^{\{B\}}(I) = h^{\{C\}}(I) = 1$.
- In projections on more variables, we need for each variable three applications of increment actions 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* action: action modifying variable v
- $h^{\{A,B\}} = 6$
  $\Rightarrow$ any plan contains at least 6 actions of type A or B.
- $h^{\{A,C\}} = 6$
  $\Rightarrow$ any plan contains at least 6 actions of type A or C.
- $h^{\{B,C\}} = 6$
  $\Rightarrow$ any plan contains at least 6 actions of type B or C.
- $\Rightarrow$ at least 9 actions in any plan

Can we generalize this kind of reasoning?

Post-hoc Optimization Heuristic: Linear Program

For pattern collection $\mathcal{C}$:

Variables

X_a for each action $a \in A$

Objective

Minimize $\sum_{a \in A} X_a$

Subject to

$$\sum_{a \in A : a \text{ affects } P} X_a \geq h^P(s) \quad \text{for all patterns } P \in \mathcal{C}$$

$$X_a \geq 0 \quad \text{for all } a \in A$$

→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 $c_\pi(A')$ be the cost incurred by actions from $A' \subseteq A$ in π.

Setting each $X_{[a]}$ to $c_\pi([a])$ is a feasible variable assignment: Constraints $X_{[a]} \geq 0$ 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.

Relation to Canonical Heuristic

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

Proof: Consider the dual D of the LP solved by $h_{\mathcal{C}}^{\mathsf{PhO}}$ in state s for a given pattern collection $\mathcal{C}$: Maximize $\sum_{P \in \mathcal{C}} Y_P h^P(s)$ 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 action costs within each heuristic by a factor Y_i.

If we restrict thef variables in D to integers, the objective value is the canonical heuristic value $h^{\mathcal{C}}(s)$.

→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.

Reminder: Optimal Cost Partitioning for Landmarks

Variables

Count_a for each action a

Objective

Minimize $\sum_a \text{Count}_a \cdot c(a)$

Subject to

$$\sum_{a \in L} \text{Count}_a \geq 1 \text{ for all landmarks } L$$

$$\text{Count}_a \geq 0 \text{ for all actions } a$$

Numbers of action 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 action
- must be satisfied by every plan

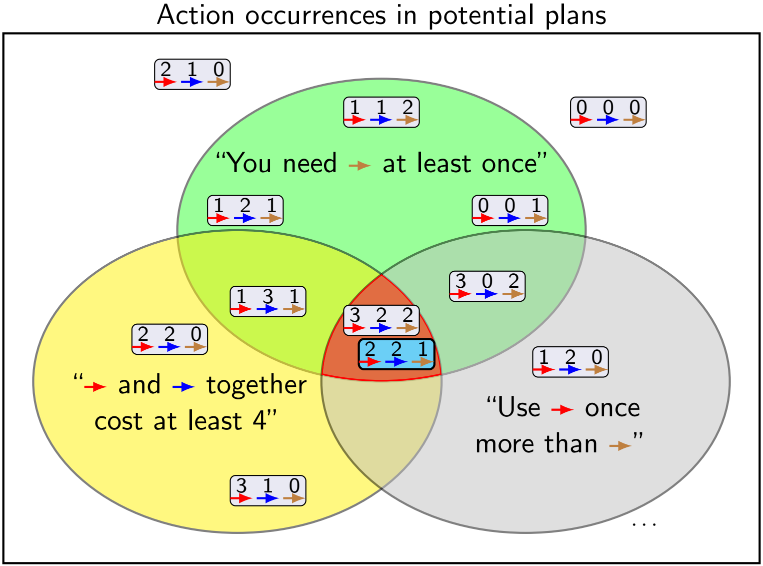
Examples:

- $\text{Count}_{a_1} + \text{Count}_{a_2} \geq 1$ "must use a_1 or a_2 at least once"
- $\text{Count}_{a_1} - \text{Count}_{a_3} \leq 0$ "cannot use a_1 more often than a_3"

Motivation:

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

Operator Counting Heuristics



Operator-counting Constraint

Definition (Operator-counting Constraints) *Let Π be an FDR task with actions A and let s be a state. Let $\mathcal{V}$ be the set of integer variables* Count_a *for each $a \in A$. A linear inequality over $\mathcal{V}$ is called an operator-counting constraint for s if for every plan π for s setting each* Count_a *to the number of occurrences of a in π is a feasible variable assignment.*

$\rightarrow$ An operator-counting constraint must be true in every plan!.

What constraints we know already?

- Landmark constraint for landmark L: $\sum_{a \in L} \text{Count}_a \geq 1$
- Post-hoc Optimization constraint for heuristic h

$$\sum_{a \text{ is relevant for } h} \text{Count}_a \cdot c(a) \geq h(s)$$

Flow Heuristic Constraints

Flow heuristic Use one flow constraint per fact (v, d)

$$\sum_{\substack{(v,d) \text{ produced} \\ \text{by } a}} \text{Count}_a - \sum_{\substack{(v,d) \text{ consumed} \\ \text{by } a}} \text{Count}_a \geq [G[v] = d] - [s[v] = d]$$

where:

- (v, d) consumed by a if $(v, d) \in pre(a)$ and $(v, d') \in eff(a)$ for some $d' \neq d$
- (v, d) produced by a if $(v, d) \in eff(a)$ and $((v, d) \notin pre(a))$
- $[G[v] = d] = 1$ if $G[v] = d$ and 0 otherwise
- $[s[v] = d] = 1$ if $s[v] = d$ and 0 otherwise

Remark: In Transition Normal Form tasks (see Conclusion section) we can make this constraints tighter by replacing $\geq$ by $=$.

Operator-counting IP/LP Heuristic

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*

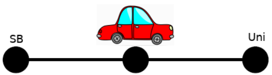
$$\textit{Minimize} \sum_{a \in A} \mathsf{Count}_a \cdot c(a) \textit{ subject to}$$

$$C \textit{ and } \mathsf{Count}_a \geq 0 \textit{ for all } a \in A,$$

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 ∞*.*

Questionnaire



- Variables: $at : \{SB, Home, Uni\}$; $v(SB) : \{T, F\}$, $v(Uni) : \{T, F\}$.
- Actions: $drive(x, y)$ where x, y have a road.
- Initial state: $at = Home, v(SB) = F, v(Uni) = F$.
- Goal: $at = Home, v(SB) = T, v(Uni) = T$.

Question!

What heuristic we obtain from the operator counting heuristic with the flow constraints?

1. For $v(Uni)$: $\text{Count}_{dr(H,Uni)} \geq 1$
2. For v(SB): $\text{Count}_{dr(H,SB)} \geq 1$
3. For at(Home):

$$(\text{Count}_{dr(H,SB)} + \text{Count}_{dr(H,Uni)}) - (\text{Count}_{dr(Uni,H)} + \text{Count}_{dr(SB,H)}) \geq 0$$

$\rightarrow h = 4$ (We have similar constraints for at(SB) and at(Uni) but they are not relevant for h in this case so we ommit them)

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 action 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.

Theorem: Combining operator-counting heuristics in one LP is equivalent to computing their optimal general cost partitioning.

Proof idea: The linear programs are each others duals.

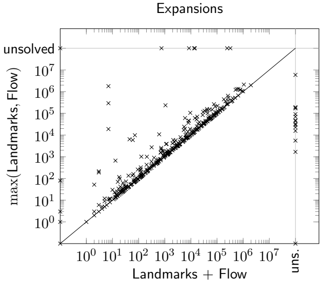
Even better combination of heuristics with IP heuristic

- Considers that actions cannot be used 1.5 times
- But computation is no longer polynomial

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

Motivation

- Operator-counting heuristics solve an LP to compute the heuristic estimate for a single state.
- Can we also define an entire heuristic function solving only one LP?
- Axiomatic approach for defining heuristics:
  - What should a heuristic look like mathematically?
  - Which properties should it have?
- Define a space of interesting heuristics.
- Use optimization to pick a good representative.

Potential Heuristics

Potential Heuristics: Idea

Heuristic design as an optimization problem:

- Define simple numerical state features $f_1, \dots, f_n$.
- Consider heuristics that are linear combinations of features:

$$h(s) = w_1 f_1(s) + \cdots + w_n f_n(s)$$

with weights (potentials) $w_i \in \mathbb{R}$

- Find potentials for which h is admissible and well-informed.

Motivation:

- declarative approach to heuristic design
- heuristic very fast to compute if features are

Potential Heuristics

Definition (feature)

A (state) feature for a planning task is a numerical function defined on the states of the task: $f : S \to \mathbb{R}$.

Definition (potential heuristic)

A potential heuristic for a set of features $\mathcal{F} = \{f_1, \ldots, f_n\}$ is a heuristic function h defined as a linear combination of the features:

$$h(s) = w_1 f_1(s) + \cdots + w_n f_n(s)$$

with weights (potentials) $w_i \in \mathbb{R}$.

$\rightsquigarrow$ cf. evaluation functions for board games like chess

Atomic Potential Heuristics

Atomic features test if some atom is true in a state:

Definition (atomic feature)

Let $X = x$ be an atom of a FDR planning task.

The atomic feature $f_{X=x}$ is defined as:

$$f_{X=x}(s) = \begin{cases} 1 & \text{if variable } X \text{ has value } x \text{ in state } s \\ 0 & \text{otherwise} \end{cases}$$

- We only consider atomic potential heuristics, which are based on the set of all atomic features.
- Example for a task with state variables X and Y:

$$h(s) = 3f_{X=a} + \tfrac{1}{2}f_{X=b} - 2f_{X=c} + \tfrac{5}{2}f_{Y=d}$$

How to Set the Weights?

We want to find good atomic potential heuristics:

- admissible
- consistent
- well-informed

How to achieve this? Linear programming to the rescue!

Admissible and Consistent Potential Heuristics

Constraints on potentials characterize (= are necessary and sufficient for) admissible and consistent atomic potential heuristics:

Goal-awareness

$$\sum_{\text{goal facts } f} w_a = 0$$

Consistency

$$\sum_{\substack{f \text{ consumed} \\ \text{by } a}} w_a - \sum_{\substack{f \text{ produced} \\ \text{by } a}} w_a \leq c(a) \quad \text{for all actions } a$$

Remarks:

- assumes transition normal form (not a limitation)
- goal-aware and consistent = admissible and consistent

Well-Informed Potential Heuristics

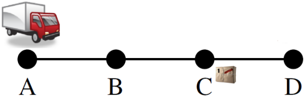
How to find a well-informed potential heuristic?

$\rightsquigarrow$ encode quality metric in the objective function
and use LP solver to find a heuristic maximizing it

Examples:

- maximize heuristic value of a given state (e.g., initial state)
- maximize average heuristic value of all states (including unreachable ones)
- maximize average heuristic value of some sample states
- minimize estimated search effort

Example: Potential Heuristic



- Initial state I: $t(A), p(C)$.
- Goal G: $t(A), p(D)$.
- Actions A: $dr(X, Y), lo(X), ul(X)$.

Goal Awareness:

$$w_{t(A)} + w_{p(D)} = 0$$

Consistency:

$$w_{at(X)} - w_{at(Y)} \leq 1 \quad \text{(drive (X, Y))}$$

$$w_{p(X)} - w_{p(T)} \leq 1 \quad \text{(load (X))}$$

$$w_{p(T)} - w_{p(X)} \leq 1 \quad \text{(unload (X))}$$

→Some Solution: $w_{p(T)} = 1$, $w_{p(A)} = w_{p(B)} = w_{p(C)} = 2$, others $= 0$

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.

Summary

- The combination into one operator-counting heuristic corresponds to the computation of the optimal general cost partitioning for the ingredient heuristics.
- General cost partitioning, operator-counting constraints and potential heuristics are facets of the same phenomenon.
- Study of each reinforces understanding of the others.

Transition Normal Form

Definition (Transition Normal Form) An FDR planning task $\Pi = (V, A, c, I, G)$ is in transition normal form (TNF) if for all $a \in A$, $V[pre(a)] = V[\mathit{eff}(a)]$, and $V[G] = V$.

$\rightarrow$An operator in TNF must mention the same variables in the precondition and effect, and a goal in TNF must mention all variables (= specify exactly one goal state).

Any FDR task can be transformed into TNF:

1. For every variable v, add a new auxiliary value u to its domain.
2. For every variable v and value $d \in D_v \setminus \{u\}$, add a new operator to change the value of v from d to u with no precondition at no cost.
3. For all actions, if $v = d$ is a precondition with no effect on v, just add the effect $v := d$.
4. For all actions a and all variables $v \in V[\mathit{eff}(a)] \setminus V[pre(a)]$, add the precondition $v = u$ to $pre(a)$.
5. For every variable $v \in V \setminus V[G]$, add the condition $v = u$ to G.

Reading

- *Getting the Most Out of Pattern Databases for Classical Planning* [Pommerening *et al.* (2013)].
  Available at:

  https://ai.dmi.unibas.ch/papers/pommerening-et-al-ijcai2013.pdf

  Content: Introduces the post-hoc optimization heuristics.

- *LP-Based Heuristics for Cost-Optimal Planning.*[Pommerening *et al.* (2014)] Available at:

  https://ai.dmi.unibas.ch/papers/pommerening-et-al-icaps-14.pdf

  Content: Introduces the operator counting heuristics.

References I

Florian Pommerening, Gabriele Röger, and Malte Helmert. Getting the most out of pattern databases for classical planning. In Francesca Rossi, editor, *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI'13)*. AAAI Press/IJCAI, 2013.

Florian Pommerening, Gabriele Röger, Malte Helmert, and Blai Bonet. LP-based heuristics for cost-optimal planning. In Steve Chien, Minh Do, Alan Fern, and Wheeler Ruml, editors, *Proceedings of the 24th International Conference on Automated Planning and Scheduling (ICAPS'14)*, pages 226–234. AAAI Press, 2014.