Planning and Optimization

D6. Comparison of Heuristic Families I

Malte Helmert and Gabriele Röger

Universität Basel

December 12, 2016

Comparing Heuristic Classes

Comparing Heuristic Quality

- We have seen many different heuristics.
  Can we compare their quality?
- For inadmissible heuristics, it is very hard to compare their quality theoretically (need a model of the search space).
- For admissible heuristics, if $h(s) \geq h'(s)$ for all states s then h is at least as good as h' in terms of heuristic quality.
- For example, we know that $h^m \geq h^{m'}$ for $m \geq m'$, so the heuristic quality of h^m cannot get worse with larger m.
- Only very few heuristics can be compared with this strong notion of dominance.

Heuristic Classes

- Many "heuristics" we have seen are actually heuristic classes of many different specific heuristics.
  - There is no single PDB heuristic but one such heuristic for each pattern.
  - Merge-and-shrink heuristics depend on the merge and shrinking strategies (and tie-breaking).
  - Different sets of landmarks lead to different landmark heuristics.
  - ...
- How can we compare such heuristic classes?

Comparing Heuristic Classes (1)

- Compare best cases: Given the best heuristic of class $\mathcal{H}$, can we find a heuristic of class $\mathcal{H}'$ that is at least as good?
- No need to talk about a specific best heuristic (which is hard to identify), we can consider arbitrary heuristics instead: Given an arbitrary heuristic of class $\mathcal{H}$, can we find a heuristic of class $\mathcal{H}'$ that is at least as good?
- It is only very rarely the case that there is a single heuristic that works globally for all states (as for example with PDB heuristics and merge-and-shrink heuristics).
- Focus on individual states instead: Given an arbitrary heuristic of class $\mathcal{H}$ and a state s, can we find a heuristic of class $\mathcal{H}'$ that is at least as good on state s?

Comparing Heuristic Classes (2)

- Cost partitioning allows to derive strong heuristic ensembles even from comparatively weak heuristics.
- We want to consider this in our comparison:
  Given an arbitrary additive set of heuristics of class $\mathcal{H}$ and a state s, can we find an additive set of heuristics of class $\mathcal{H}'$ that is at least as good on state s?
- Some classes cover the perfect heuristic. For example, exponential-size abstractions can always represent h^*.
- To prevent such trivial cases, we concentrate on heuristics that can be computed in polynomial time in the representation size of the task.

Compilability

Definition (Compilability)

A class of heuristics $\mathcal{H}$ is compilable to a class of heuristics $\mathcal{H}'$ if for every state s and every additive set of heuristics $h_1, \ldots, h_n$ of class $\mathcal{H}$ we can compute an additive set of heuristics $h'_1, \ldots, h'_m$ of class $\mathcal{H}'$ such that $\sum_{i=1}^{n} h_i(s) \leq \sum_{i=1}^{m} h'_i(s)$.

It is sufficient to consider $n = 1$. Why?

Analogy to reduction in theoretical computer science.

What to Compare?

Delete Relaxation

- h^{add} and h^{FF} are inadmissible.
- h^+ is NP-hard to compute.
- This leaves $h^{\max}$.
- Reminder: $h^{\max} \leq h^+$

Throughout this topic, we write

- o^+ for the delete-relaxation of operator o,
- for sets O of operators: O^+ for $\{o^+ \mid o \in O\}$, and
- Π^+ for the delete-relaxation of task Π.

Abstraction

- In this course: PDB and merge-and-shrink heuristics
- Both are admissible.
- Merge-and-shrink heuristics are at least as powerful as PDB heuristics because we can compute an equivalent merge-and-shrink heuristic for each PDB heuristic with only polynomial overhead.
- Merge-and-shrink heuristics can represent abstractions that are not projections, so merge-and-shrink heuristics are strictly more powerful than PDBs.
- Makes sense to compare other heuristic classes to both of these abstraction heuristic classes.

Landmarks (1)

- Have seen LM-Cut, LM-count and cost-partitioning for landmarks.
- LM-count is inadmissible.
- All admissible heuristics can be expressed by cost partitioning and heuristics that use the cost of the landmark as estimate.
- Most landmark generation methods only generate landmarks of the delete relaxation, which is a severe limitation.
- We therefore analyse such relaxation-based landmark heuristics.

Landmarks (2)

Definition (Elementary Landmark Heuristic)

The elementary landmark heuristic for planning task $\Pi = \langle V, I, O, \gamma \rangle$ and operator subset $L \subseteq O$ is

$$h_L(s) = \begin{cases} \min_{o \in L} cost(o) & \text{if } L^+ \text{ is a landmark for } s \text{ in } \Pi^+ \\ 0 & \text{otherwise} \end{cases}$$

Additive sets of such heuristics cover all admissible relaxation-based landmark heuristics we have seen (on a specific state).

Critical Paths

- The h^m heuristic family is admissible.
- For $m \geq m'$, $h^m(s) \geq h^{m'}(s)$ for all states s.
- For $m > m'$, there are tasks and states s with $h^m(s) > h^{m'}(s)$.
- For large enough m (depending on the task), $h^m = h^*$.
- Computation is exponential in m.
- Polynomial-time compilations can only compile to critical path heuristics for fixed m.
- Reminder: $h^1 = h^{\max}$

Landmarks vs. Abstractions

Abstractions to Landmarks

Theorem

There is no compilation of PDB heuristics into elementary landmarks.

Proof.

The estimate of a PDB heuristic can exceed h^+ while elementary landmark heuristics are bounded by h^+. □

The result directly carries over to merge-and-shrink heuristics.

Landmarks to PDBs

Theorem (Landmarks to PDBs)

There is no polynomial-time compilation of elementary landmarks into PDB heuristics.

Proof.

Consider task family $(\Pi_n)_{n \in \mathbb{N}_1}$, where $\Pi_n = \langle V_n, I_n, O_n, g \rangle$ with $V_n = \{v_1, \dots, v_n, g\}$, $I_n(v) = \mathbf{F}$ for $v \in V_n$, and $O = \{\langle \top, v_i, 1 \rangle \mid 1 \leq i \leq n\} \cup \{\langle v_i, g, 0 \rangle \mid 1 \leq i \leq n\}$.

$L = \{\langle \top, v_i, 1 \rangle \mid 1 \leq i \leq n\}$ is a landmark for I, so $h_L(I) = 1$.

However, the initial estimate of every PDB heuristic that projects away at least one variable v is 0, as the abstract goal can be reached with $\langle v, g, 0 \rangle$. For large enough n, any polynomial-time compilation must project away a variable. □

Landmarks to Merge-and-Shrink Abstractions (1)

Theorem

Elementary landmarks can be compiled into merge-and-shrink abstractions in polynomial time.

Proof.

Let $\Pi = \langle V, I, O, \gamma \rangle$ be a STRIPS planning task and $L \subseteq O$.
Let U be the set of variables that cannot be reached from s in Π^+ without using an operator from L^+.
Consider abstraction

$$\alpha(s') = \begin{cases} s_u & s' \models \bigvee_{v \in U} v \\ s_r & \text{otherwise} \end{cases}$$

...

Landmarks to Merge-and-Shrink Abstractions (2)

Proof (continued).

The abstraction can be computed as merge-and-shrink abstraction in polynomial time by a linear merge strategy with arbitrary variable order. After each merge step, shrink all abstract states where all (already included) variables in U have value $\mathbf{F}$ to one state and all other states to a second state.

If L^+ is not a landmark for s in Π^+, then $h_L(s) = 0$ and trivially $h^\alpha(s) \geq h_L(s)$.

If L^+ is a landmark then $\gamma \models \bigvee_{v \in U} v$. So, for all goal states $s_\star$ it holds that $\alpha(s_\star) = s_u$, so s_u is the only abstract goal state. ...

Landmarks to Merge-and-Shrink Abstractions (3)

Proof (continued).

As all true variables in s are reachable from s in Π^+, $s \not\models \bigvee_{v \in U} v$ and $\alpha(s) = s_r$.

All abstract plans for s must contain a transition from s_r to s_u and $h^\alpha(s)$ is the minimal cost of all such transitions.

Assume that there is a transition from a state s_1 with $\alpha(s_1) = s_r$ to a state s_2 with $\alpha(s_2) = s_u$ by an operator $o \notin L$. Then o^+ is applicable in s_1 and leads to a state where a variable from U is true, contradicting the definition of U.

Therefore all abstract transitions from s_r to s_u are induced by an operator from L and have cost at least $\min_{o \in L} cost(o)$.
So $h^\alpha(s) \geq \min_{o \in L} cost(o) = h_L(s)$. □

Summary

Summary

- We can use compilability to compare the power of different classes of admissible heuristics.
- So far we have established that PDB heuristics are incomparable with landmark heuristics, and
- Merge-and-shrink heuristics strictly dominate landmark heuristics.