Planning and Optimization

F4. Potential Heuristics & Connections

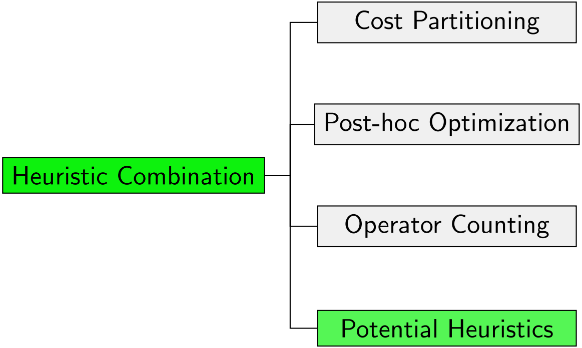
Malte Helmert and Gabriele Röger

Universität Basel

December 6, 2017

Potential Heuristics

Content of this Course: Heuristic Combination



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:

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

  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

Features

Definition (feature)

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

Potential Heuristics

Definition (potential heuristic)

A potential heuristic for a set of features $\mathcal{F} = \{f_1, \dots, f_n\}$ 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:

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

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

$$\sum_{\text{goal atoms } a} w_a = 0$$

Consistency

$$\sum_{\substack{a \text{ consumed} \\ \text{by } o}} w_a - \sum_{\substack{a \text{ produced} \\ \text{by } o}} w_a \leq \mathit{cost}(o) \quad \text{for all operators } o$$

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?

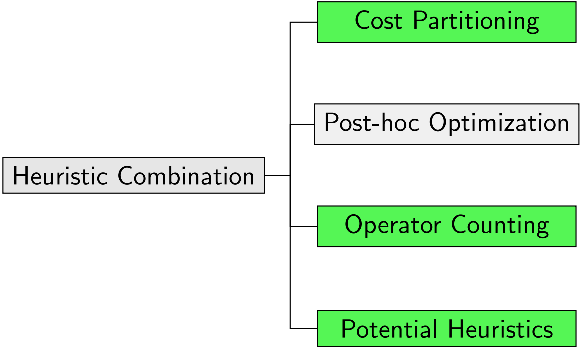
$\rightsquigarrow$ encode quality metric in the objective function and use LP solver to find a heuristic maximizing it

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

Connections

Content of this Course: Heuristic Combination



Potential and Flow Heuristic

Theorem

For state s, let $h^{\text{maxpot}}(s)$ denote the maximal heuristic value of all admissible and consistent atomic potential heuristics in s.

Then $h^{\text{maxpot}}(s) = h^{flow}(s)$.

Proof idea: compare dual of $h^{\text{flow}}(s)$ LP to potential heuristic constraints optimized for state s.

If we optimize the potentials for a given state then for this state it equals the flow heuristic.

Operator Counting and General Cost Partitioning

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.

Use the Theorem to Combine Heuristics

- Easy way to compute cost partitioning of heuristics
  - LP can be more compact (variable elimination)
  - No need for one variable per operator and subproblem
- Even better combination of heuristics with IP heuristic
  - Considers that operator cannot be used 1.5 times
  - But computation is no longer polynomial

Use the Theorem to Analyze Heuristics

Analyze operator counting heuristics

1. Group linear constraints into sets of operator-counting constraints
2. Figure out what heuristic is computed with just one such set
3. Your original operator-counting heuristic computes the optimal general cost partition of those component heuristics

Use the Theorem to Analyze Heuristics

Analyze operator counting heuristics

Example: flow heuristic

1. Group linear constraints into sets of operator-counting constraints
2. Figure out what heuristic is computed with just one such set
3. Your original operator-counting heuristic computes the optimal general cost partition of those component heuristics

Use the Theorem to Analyze Heuristics

Analyze operator counting heuristics

Example: flow heuristic

1. Group linear constraints into sets of operator-counting constraints
   - One group of flow constraints per variable
2. Figure out what heuristic is computed with just one such set
3. Your original operator-counting heuristic computes the optimal general cost partition of those component heuristics

Use the Theorem to Analyze Heuristics

Analyze operator counting heuristics

Example: flow heuristic

1. Group linear constraints into sets of operator-counting constraints
   - One group of flow constraints per variable
2. Figure out what heuristic is computed with just one such set
   - Minimizing total cost while respecting flow in projection to one variable
   - Shortest path in projection
3. Your original operator-counting heuristic computes the optimal general cost partition of those component heuristics

Use the Theorem to Analyze Heuristics

Analyze operator counting heuristics

Example: flow heuristic

1. Group linear constraints into sets of operator-counting constraints
   - One group of flow constraints per variable
2. Figure out what heuristic is computed with just one such set
   - Minimizing total cost while respecting flow in projection to one variable
   - Shortest path in projection
3. Your original operator-counting heuristic computes the optimal general cost partition of those component heuristics
   - Flow heuristic = gOCP(atomic projection heuristics)

Other Examples

What about the rest of our examples?

- Landmark constraints
  - gOCP(individual landmark heuristics)
- Post-hoc optimization heuristic
  - gOCP(heuristics that spend a minimum cost on relevant ops)
  - Also: cost partitioning over atomic projection heuristics
    - Operator costs not independent
    - Scale with one factor per projection

Summary

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.
- Potential heuristics can be used as fast admissible approximations of h^{flow}.
- Generalization beyond h^{flow}: use non-atomic features
- If features are cheap to compute, the heuristic evaluation for every state is extremely fast.

Literature

Florian Pommerening, Malte Helmert, Gabriele Röger and Jendrik Seipp.
From Non-Negative to General Operator Cost Partitioning.
Proc. AAAI 2015, pp. 3335–3341, 2015.
Introduces potential heuristics and shows relation between general cost partitioning and operator counting.

Jendrik Seipp, Florian Pommerening and Malte Helmert.
New Optimization Functions for Potential Heuristics.
Proc. ICAPS 2015, pp. 193–201, 2015.
Studies effect of different optimization functions.