Higher-Dimensional Potential Heuristics for Optimal Classical Planning

Florian Pommerening\textsuperscript{1} Malte Helmert\textsuperscript{1} Blai Bonet\textsuperscript{2}

\textsuperscript{1}University of Basel, Switzerland
\textsuperscript{2}Universidad Simón Bolívar, Venezuela

February 7, 2017
Higher-Dimensional Potential Heuristics for Optimal Classical Planning
Higher-Dimensional Potential Heuristics
for Optimal Classical Planning

- Find cheapest action sequence to achieve a goal.
- States are variable assignments.
- Operators change variable values.
Higher-Dimensional **Potential Heuristics** for Optimal Classical Planning

\[ h(s) = \sum_{f \in \mathcal{F}} w(f)[s \models f] \]

- Weighted sum of state features
- Two choices
  - Which features to use?
  - How to find good weights?
Features are conjunctions of facts

Size of a feature: number of conjuncts

“Atomic” features (size 1)

\[ w(\text{at-}A) = 10, \ w(\text{at-}B) = 5 \]

“Binary” features (size 2)

\[ w(\text{at-}B \land \text{door-locked}) = 10 \]

...
Why do we care about higher-dimensional features?

Atomic features are often not sufficient for high-quality heuristics.
Goals

- Find good weights automatically
- Ideally:
  - Declare properties of heuristics (admissible, consistent)
  - Constraints characterize heuristics with these properties
  - Select best possible heuristic from the space of solutions
Our Contributions

Describing admissible and consistent potential heuristics

<table>
<thead>
<tr>
<th>Features</th>
<th>Characterization</th>
</tr>
</thead>
<tbody>
<tr>
<td>All atomic features</td>
<td>compact [previous work]</td>
</tr>
<tr>
<td>All binary features</td>
<td>compact [new]</td>
</tr>
<tr>
<td>All ternary features</td>
<td>intractable [new]</td>
</tr>
<tr>
<td>Subset of all features</td>
<td>fixed parameter tractable [new]</td>
</tr>
</tbody>
</table>

Also in the paper

- Potential functions $\simeq$ Transition cost partitioning
Compact Characterizations
Compact Characterization

Characterizing **admissible** and **consistent** heuristics

**Goal awareness**

\[ h(s^*) \leq 0 \]

- Easy: \( h(s^*) \) is a sum of weights

**Consistency**

\[ h(s) - h(s') \leq \text{cost}(o) \quad \forall s \xrightarrow{o} s' \]

- Hard: exponential number of constraints
Consistency

- Consider a single operator
- Three types of features
  - irrelevant: no variables in common with $o$
  - context-independent: all variables in common with $o$
  - context-dependent: some in common with $o$, some not

Heuristic difference caused by operator $o$

$$h(s) - h(s') = \Delta_o^{\text{irr}}(s) + \Delta_o^{\text{ind}}(s) + \Delta_o^{\text{dep}}(s)$$
Heuristic Difference when Applying Operator $o$

Consistency for an operator $o$

$$\Delta_o^{\text{irr}}(s) + \Delta_o^{\text{ind}}(s) + \Delta_o^{\text{dep}}(s) \leq \text{cost}(o) \quad \forall s \xrightarrow{o} s'$$

Irrelevant features

- No variables in common with $o$
- No change in truth value when applying $o$
- Does not cause change in heuristic
Heuristic Difference when Applying Operator $o$

Consistency for an operator $o$

$$0 + \Delta_o^{\text{ind}}(s) + \Delta_o^{\text{dep}}(s) \leq cost(o) \quad \forall s \xrightarrow{o} s'$$

Irrelevant features

- No variables in common with $o$
- No change in truth value when applying $o$
- Does not cause change in heuristic
Heuristic Difference when Applying Operator $o$

Consistency for an operator $o$

$$\Delta^\text{ind}_o(s) + \Delta^\text{dep}_o(s) \leq \text{cost}(o) \quad \forall s \xrightarrow{o} s'$$

Irrelevant features

- No variables in common with $o$
- No change in truth value when applying $o$
- Does not cause change in heuristic
Heuristc Difference when Applying Operator $o$

**Consistency for an operator $o$**

$$\Delta_{o}^{\text{ind}}(s) + \Delta_{o}^{\text{dep}}(s) \leq \text{cost}(o) \quad \forall s \xrightarrow{o} s'$$

**Context-independent features**

- All variables in common with $o$
- Change in truth value fully determined by $o$
- Heuristic change easy to specify and does not depend on state
Heuristic Difference when Applying Operator $o$

**Consistency for an operator $o$**

$$\Delta_o^{\text{ind}}(s) + \Delta_o^{\text{dep}}(s) \leq \text{cost}(o) \quad \forall s \xrightarrow{o} s'$$

**Context-independent features**

- All variables in common with $o$
- Change in truth value fully determined by $o$
- Heuristic change *easy to specify* and *does not depend on state*
Heuristic Difference when Applying Operator $o$

Consistency for an operator $o$

$$\Delta_o^{\text{ind}} + \Delta_o^{\text{dep}}(s) \leq \text{cost}(o) \quad \forall s \xrightarrow{o} s'$$

Context-independent features

- All variables in common with $o$
- Change in truth value fully determined by $o$
- Heuristic change easy to specify and does not depend on state
Heuristic Difference when Applying Operator $o$

**Consistency for an operator $o$**

\[
\Delta_o^{\text{ind}} + \Delta_o^{\text{dep}}(s) \leq \text{cost}(o) \quad \forall s \xrightarrow{o} s'
\]

**Context-dependent features**

- At least one variable in common with $o$
- At least one variable not mentioned by $o$
- Heuristic change depends on state
Context-Dependent Features

- **Atomic features**: no context-dependent features
- **Binary features**: context limited to one variable
  - “Worst value” exists for each variable
  - Worst case: all variables have worst value
  - Constraint for worst state implies all other constraints

**Theorem**

Admissible and consistent potential heuristics over binary features can be characterized by a compact set of linear constraints.
Larger Features
In general

- Change in potential when applying \( o \) depends on more than one variable
- Influence of \( V \) on \( o \) depends on larger context

**Theorem**

Testing whether a given potential function is consistent is coNP-complete. This already holds with only ternary features.

**Proof:**

- Reduction from non-3-colorability
Approach for binary features can be generalized

- Factor out influence of one variable at a time
- Generalization of **Bucket Elimination** algorithm from numerical cost functions to linear expressions

**Theorem**

Computing a set of linear constraints that characterize the admissible and consistent potential heuristics for a set of features is **fixed-parameter tractable**.

Parameter: tree-width of **feature connectivity**.
Take Home Messages
Take Home Messages

Characterization of admissible and consistent potential functions

- Compact for binary features
- coNP-complete for ternary or larger features...
- ... but fixed parameter tractable
  Parameter: tree-width of feature connectivity