New Optimization Functions for Potential Heuristics

Jendrik Seipp Florian Pommerening Malte Helmert

University of Basel

June 11, 2015

Motivation

Setting: Optimal classical planning

Traditional heuristics

- Often hard to understand
- Need lots of code

Potential heuristics (Pommerening et al., AAAI 2015)

- Declaratively specify constraints for heuristic
- Easy to implement
- Optimized solvers exist (e.g. CPLEX)

Overview

Potential Heuristics

Features

Definition (feature)

A feature is a numerical function defined on states: $f : S \to \mathbb{R}$.

Potential Heuristics

Definition (potential heuristic)

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

$$h^{\mathsf{pot}}(s) = \mathsf{P}_1 f_1(s) + \cdots + \mathsf{P}_n f_n(s)$$

with potentials $\mathsf{P}_i \in \mathbb{R}$.

Atomic Potential Heuristics

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

Definition (atomic feature)

Let $X = x$ be an atomic proposition of a planning task.

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

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

- Example for a task with state variables X and Y:

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

How to Set the Potentials?

We want to find good atomic potential heuristics:

- admissible
- consistent
- well-informed

How to achieve this?

How to Set the Potentials?

We want to find good atomic potential heuristics:

- admissible
- consistent
- well-informed

How to achieve this? $\rightarrow$ Linear programming

How to Set the Potentials?

We want to find good atomic potential heuristics:

- admissible $\rightarrow$ Constraints
- consistent $\rightarrow$ Constraints
- well-informed

How to achieve this? $\rightarrow$ Linear programming

How to Set the Potentials?

We want to find good atomic potential heuristics:

- admissible → Constraints
- consistent → Constraints
- well-informed → Optimization function

How to achieve this? → Linear programming

Admissible and Consistent Potential Heuristics

- Admissible and consistent = goal-aware and consistent
- For simplicity, we assume transition normal form

Admissible and Consistent Potential Heuristics

- Admissible and consistent = goal-aware and consistent
- For simplicity, we assume transition normal form

Goal-awareness (i.e., $h^{\mathsf{pot}}(s) \leq 0$ for goal states)

$$\sum_{\text{goal facts } f} \mathsf{P}_f \leq 0$$

Consistency

$$\sum_{f \text{ consumed by } o} \mathsf{P}_f - \sum_{f \text{ produced by } o} \mathsf{P}_f \leq \mathit{cost}(o) \quad \text{for all operators } o$$

Admissible and Consistent Potential Heuristics

- Admissible and consistent = goal-aware and consistent
- For simplicity, we assume transition normal form

Goal-awareness (i.e., $h^{\mathsf{pot}}(s) \leq 0$ for goal states)

$$\sum_{\text{goal facts } f} \mathsf{P}_f \leq 0$$

Consistency

$$\sum_{f \text{ consumed by } o} \mathsf{P}_f - \sum_{f \text{ produced by } o} \mathsf{P}_f \leq \mathit{cost}(o) \quad \text{for all operators } o$$

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

Relation to State Equation Heuristic

Theorem (Pommerening et al., AAAI 2015)

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

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

Relation to State Equation Heuristic

Theorem (Pommerening et al., AAAI 2015)

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

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

- h^{SEQ} upper bound on h^{pot}
- Potential heuristics:
  - only need one LP evaluation
  - trade informedness for evaluation speed
  - can be optimized for any criterion

Optimization Functions

Optimization Functions

Definition (optimization functions)

We can optimize potential heuristics for any linear combination of potentials $\mathsf{P}_1, \ldots, \mathsf{P}_n$:

$$\text{Maximize } w_1\mathsf{P}_1 + \cdots + w_n\mathsf{P}_n$$

Optimization Functions

Definition (optimization functions)

We can optimize potential heuristics for any linear combination of potentials $\mathsf{P}_1, \ldots, \mathsf{P}_n$:

$$\text{Maximize } w_1\mathsf{P}_1 + \cdots + w_n\mathsf{P}_n$$

- Initial state
- All syntactic states
- Samples

Initial State

Optimization function

$$\text{Maximize} \sum_{f \in s_{\text{init}}} \mathrm{P}_f$$

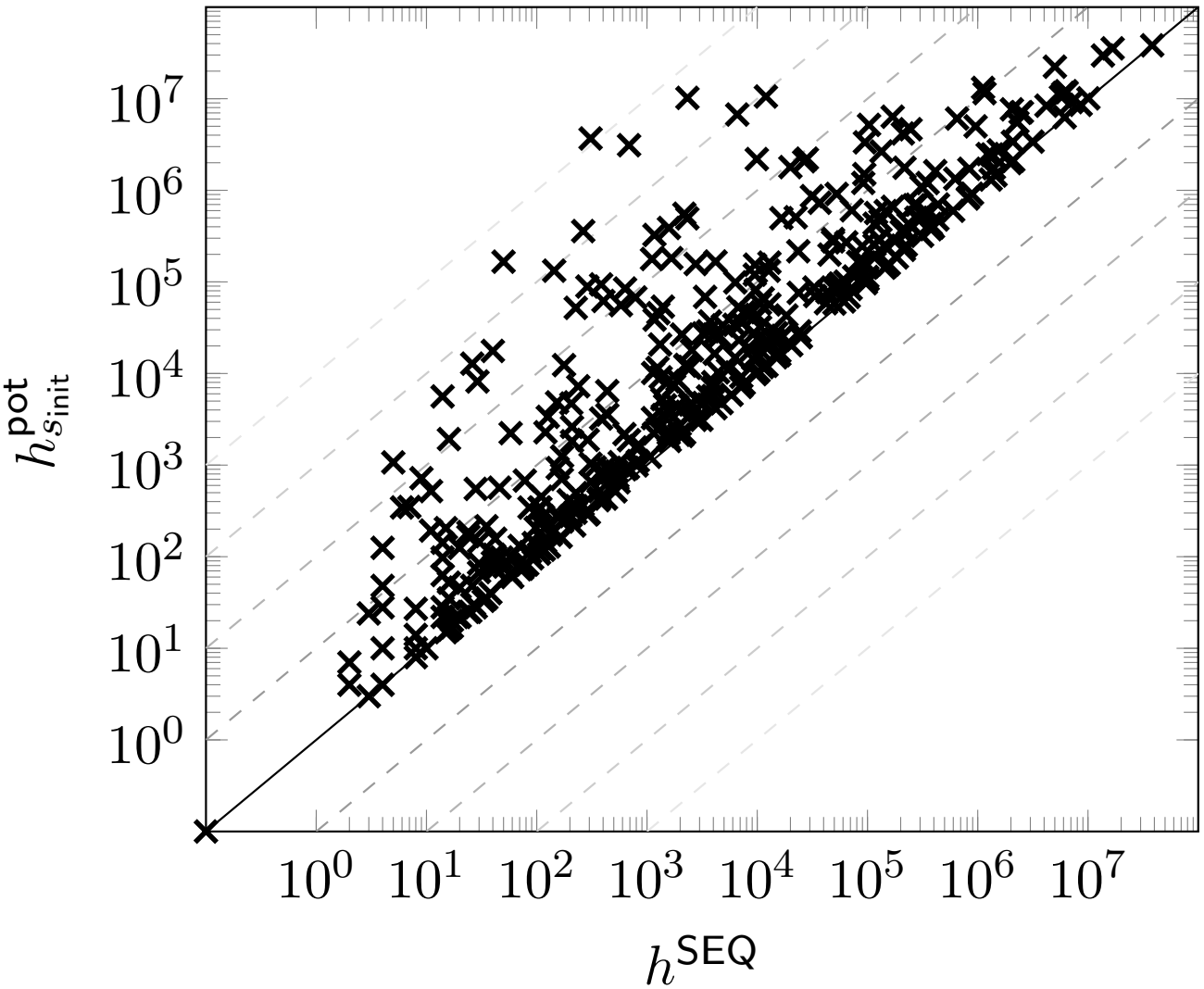
- Maximize potentials of the initial state's facts
- Used by Pommerening et al., 2015

Initial State — Expansions

How close to h^{SEQ} are we?

Initial State — Expansions

How close to h^{SEQ} are we?



Comparison of optimization functions

Solved Tasks	h^{SEQ}	$h^{\mathsf{pot}}_{s_{\mathsf{init}}}$
Sum (1396)	631	611

Initial State

- Problem: Only considers facts from the initial state
  No conscious attempt to achieve high heuristic values for other states

All Syntactic States

Optimization function

$$\text{Maximize} \sum_{V=v \in \mathcal{F}} \frac{1}{|\text{dom}(V)|} \text{P}_{V=v}$$

- Maximize heuristic for all syntactic states

All Syntactic States

Optimization function

$$\text{Maximize} \sum_{V=v \in \mathcal{F}} \frac{1}{|\text{dom}(V)|} \text{P}_{V=v}$$

- Maximize heuristic for all syntactic states
- Problem: LP may be unbounded for tasks with dead ends
- Solution: bound potentials

$$\text{P}_f \leq M \text{ for all facts } f$$

- "Prunes" dead ends

Comparison of optimization functions

Solved Tasks	h^{SEQ}	$h^{\text{pot}}_{s_{\text{init}}}$	$h^{\text{pot}}_{\text{all-states}}$
Sum (1396)	631	611	660

All Syntactic States $\rightarrow$ Samples

- Problem: optimizing for all states includes unreachable ones
- Solution: Maximize heuristic for (1000) sampled states

Comparison of optimization functions

Solved Tasks	h^{SEQ}	$h^{\text{pot}}_{s_{\text{init}}}$	$h^{\text{pot}}_{\text{all-states}}$	$h^{\text{pot}}_{\text{samples}}$
Sum (1396)	631	611	660	678

Multiple Potential Heuristics

Multiple Potential Heuristics

- Multiple sample-based heuristics
- Automatic diversification

Multiple Sample-based Heuristics

	Heuristics	Samples
Baseline	1	1000
	1–1000	1
	1–1000	1000

Multiple Sample-based Heuristics

	Heuristics	Samples
Baseline	1	1000
	1–1000	1
	1–1000	1000

- No big improvement over baseline
- Explanation: heuristics too similar
- Problem: need to set number of heuristics manually

Automatic Diversification

Automatic diversification

- Sample n states
- Until all samples are covered:
  - Find potential heuristic for remaining samples
  - If it covers no new sample, find heuristic for random sample
  - Remove all covered samples
- Return found heuristics

Comparison of optimization functions

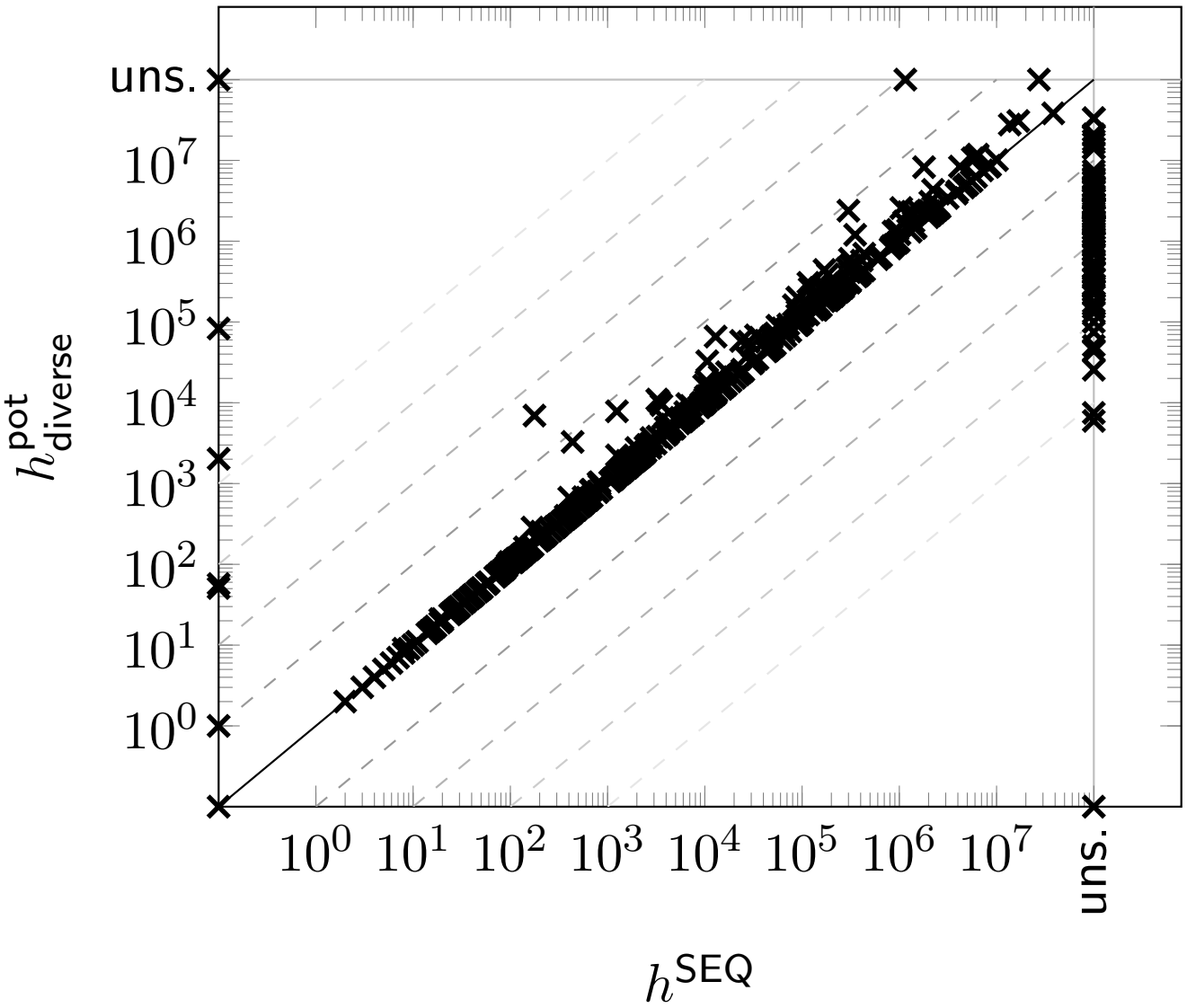
Solved Tasks	h^{SEQ}	$h^{\text{pot}}_{s_{\text{init}}}$	$h^{\text{pot}}_{\text{all-states}}$	$h^{\text{pot}}_{\text{samples}}$	$h^{\text{pot}}_{\text{diverse}}$
Sum (1396)	631	611	660	678	**698**

Automatic Diversification — Expansions

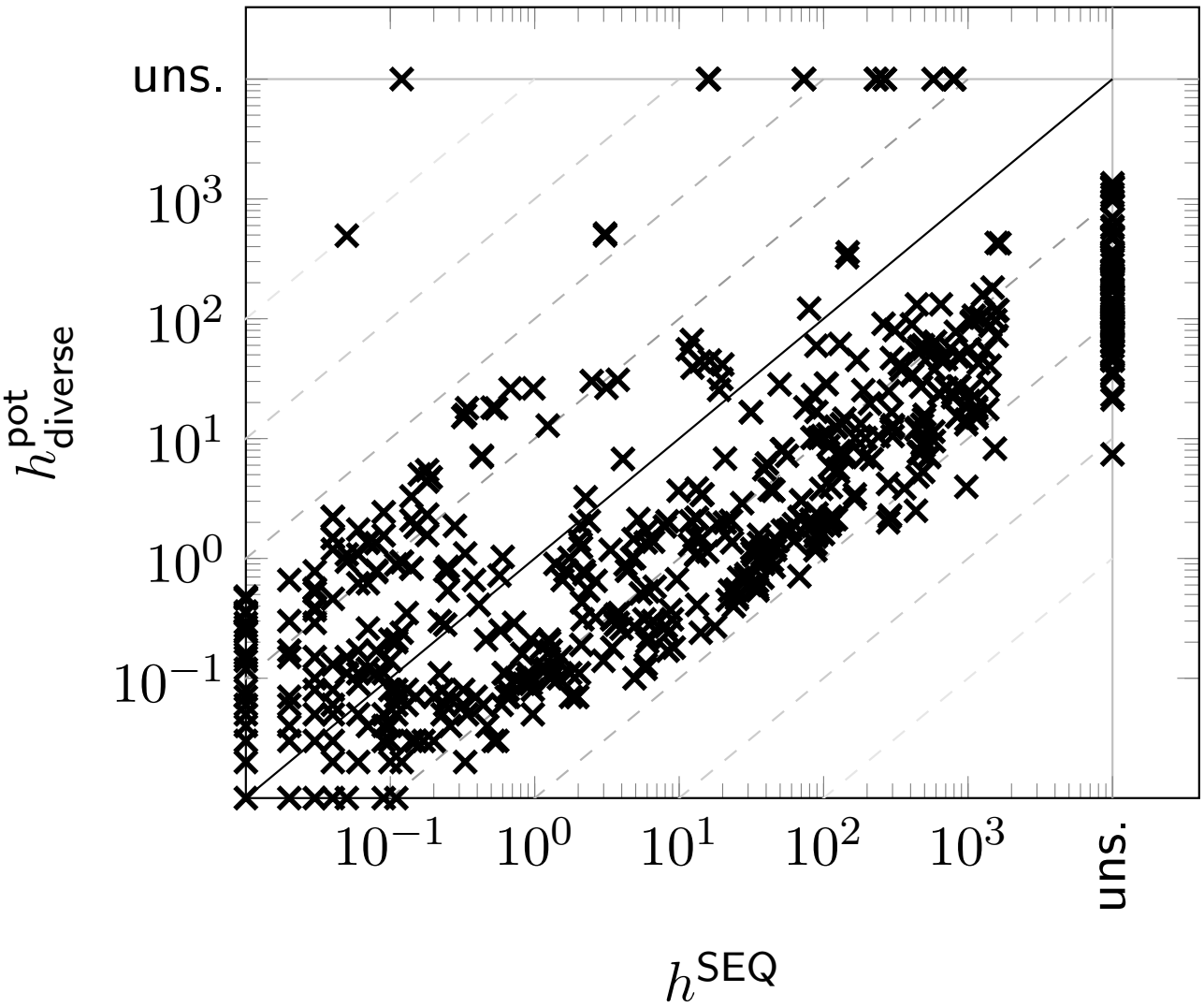
How close to h^{SEQ} are we?

Automatic Diversification — Expansions

How close to h^{SEQ} are we?



Automatic Diversification — Runtimes



Future Work

- Not much sense in improving atomic potential heuristics
- Look at other features instead

Summary

Potential heuristics:

- Declarative formulation of admissible heuristics
- Work for any optimization function
- Approximate state equation heuristic very well
- Much faster to compute $\rightarrow$ more tasks solved