

Planning and Optimization

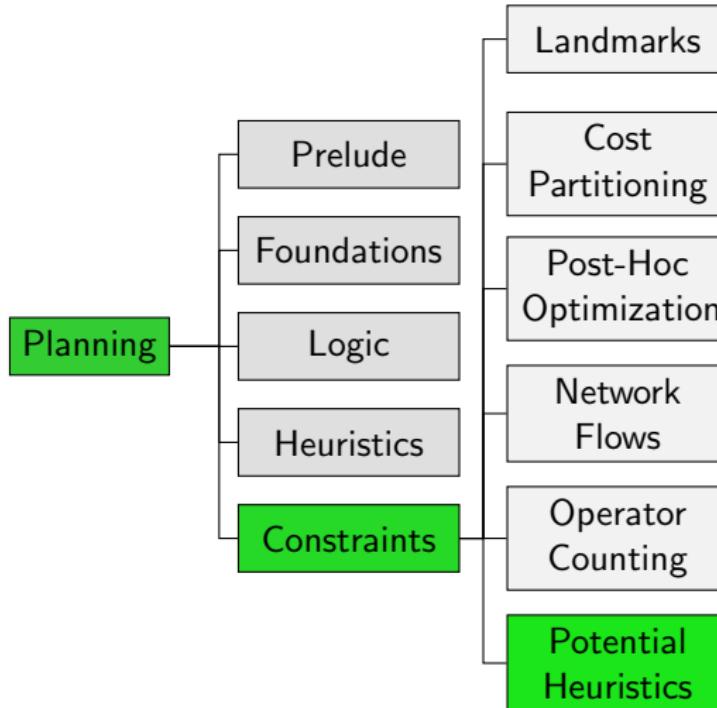
G12. Potential Heuristics

Malte Helmert and Gabriele Röger

Universität Basel

December 14, 2022

Content of this Course



Introduction

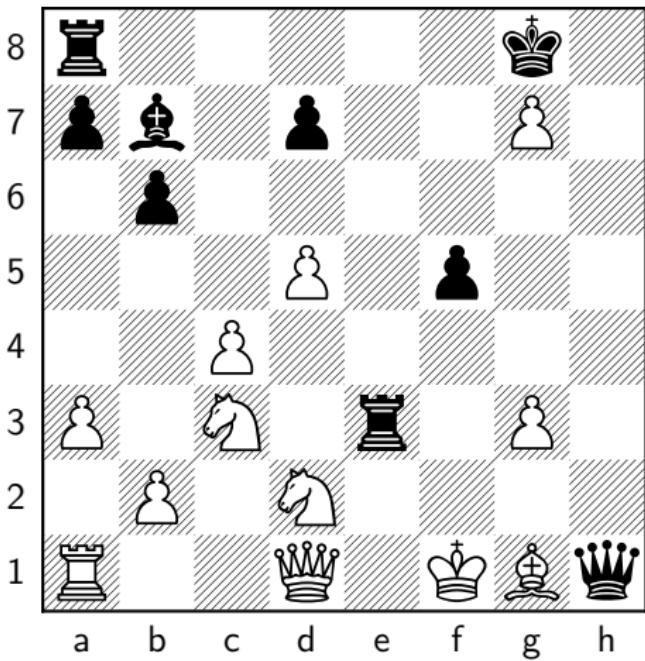
Reminder: Transition Normal Form

In this chapter, we consider SAS⁺ tasks in transition normal form.

- A TNF operator mentions the **same variables** in the precondition and in the effect.
- A TNF goal specifies a value for **every** variable.

Material Value of a Chess Position

Material value for white:



+ 1 · 6 (white pawns)
- 1 · 4 (black pawns)
+ 3 · 2 (white knights)
- 3 · 0 (black knights)
+ 3 · 1 (white bishops)
- 3 · 1 (black bishops)
+ 5 · 1 (white rooks)
- 5 · 2 (black rooks)
+ 9 · 1 (white queen)
- 9 · 1 (black queen)
= 3

Idea

- 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**) $w_i \in \mathbb{R}$

- heuristic **very fast to compute** if feature values are

Potential Heuristics

Definition

Definition (Feature)

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

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:

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

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

Definition

Definition (Feature)

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

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:

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

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

Many possibilities \Rightarrow need some restrictions

Features for SAS⁺ Planning Tasks

Which features are good for planning?

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

Definition (Atomic Feature)

Let $v = d$ be an atom of a FDR planning task.

The atomic feature $f_{v=d}$ is defined as:

$$f_{v=d}(s) = [(v = d) \in s] = \begin{cases} 1 & \text{if variable } v \text{ has value } d \text{ in state } s \\ 0 & \text{otherwise} \end{cases}$$

Offer good tradeoff between computation time and guidance

Example: Atomic Features

Example

Consider a planning task Π with state variables v_1 and v_2 and $\text{dom}(v_1) = \text{dom}(v_2) = \{d_1, d_2, d_3\}$. The set

$$\mathcal{F} = \{f_{v_i=d_j} \mid i \in \{1, 2\}, j \in \{1, 2, 3\}\}$$

is the **set of atomic features** of Π and the function

$$h(s) = 3f_{v_1=d_1} + 0.5f_{v_1=d_2} - 2f_{v_1=d_3} + 2.5f_{v_2=d_1}$$

is a **potential heuristic** for \mathcal{F} .

The heuristic estimate for a state $s = \{v_1 \mapsto d_2, v_2 \mapsto d_1\}$ is

$$h(s) = 3 \cdot 0 + 0.5 \cdot 1 - 2 \cdot 0 + 2.5 \cdot 1 = 3.$$

Potentials for Optimal Planning

Which potentials are good for optimal planning
and how can we compute them?

- We seek potentials for which h is admissible and well-informed
⇒ **declarative approach** to heuristic design
- We derive potentials by solving an **optimization problem**

How to achieve this? **Linear programming to the rescue!**

Admissible and Consistent Potential Heuristics

We achieve admissibility through goal-awareness and consistency

Goal-awareness

$$\sum_{a \in \gamma} w_a = 0$$

Admissible and Consistent Potential Heuristics

We achieve admissibility through goal-awareness and consistency

Goal-awareness

$$\sum_{a \in \gamma} w_a = 0$$

Consistency

$$\sum_{a \in s} w_a - \sum_{a \in s'} w_a \leq cost(o) \quad \text{for all transitions } s \xrightarrow{o} s'$$

One constraint transition per transition.

Can we do this more compactly?

Admissible and Consistent Potential Heuristics

Consistency for a transition $s \xrightarrow{o} s'$

$$\begin{aligned} \text{cost}(o) &\geq \sum_{a \in s} w_a - \sum_{a \in s'} w_a \\ &= \sum_a w_a[a \in s] - \sum_a w_a[a \in s'] \\ &= \sum_a w_a([a \in s] - [a \in s']) \\ &= \sum_a w_a[a \in s \text{ but } a \notin s'] - \sum_a w_a[a \notin s \text{ but } a \in s'] \\ &= \sum_{\substack{a \text{ consumed} \\ \text{by } o}} w_a - \sum_{\substack{a \text{ produced} \\ \text{by } o}} w_a \end{aligned}$$

Admissible and Consistent Potential Heuristics

Goal-awareness and Consistency independent of s

Goal-awareness

$$\sum_{a \in \gamma} w_a = 0$$

Consistency

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

Potential Heuristics

- All potential heuristics that satisfy these constraints are admissible and consistent
- Furthermore, all admissible and consistent potential heuristics satisfy these constraints

Constraints are a compact **characterization** of all admissible and consistent potential heuristics.

LP can be used to find **the best** admissible and consistent potential heuristics by encoding a **quality metric** in the **objective function**

Well-Informed Potential Heuristics

What do we mean by **the best** potential heuristic?

Different possibilities, e.g., the potential heuristic that

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

Potential and Flow Heuristic

Theorem

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

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

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.

Summary

Summary

- Potential heuristics are computed as a **weighted sum of state features**
- Admissibility and consistency can be **encoded compactly** in constraints
- With linear programming, we can efficiently compute the **best potential heuristic** wrt some objective
- Potential heuristics can be used as **fast admissible approximations** of h^{flow} .