Foundations of Artificial Intelligence

38. Automated Planning: Landmarks

Thomas Keller and Florian Pommerening

University of Basel

May 10, 2023

Planning Heuristics

We discuss three basic ideas for general heuristics:

- Delete Relaxation
- Abstraction
- Landmarks $\rightsquigarrow$ this and next chapter

Basic Idea: Landmarks

landmark = something (e.g., an action) that must be part of every solution

Estimate solution costs based on unachieved landmarks.

Automated Planning: Overview

Chapter overview: automated planning

- 33. Introduction
- 34. Planning Formalisms
- 35.–36. Planning Heuristics: Delete Relaxation
- 37. Planning Heuristics: Abstraction
- 38.–39. Planning Heuristics: Landmarks
    - 38. Landmarks
    - 39. The LM-cut Heuristic

Delete Relaxation

Landmarks and Delete Relaxation

- In this chapter, we discuss a further technique to compute planning heuristics: landmarks.
- We restrict ourselves to delete-free planning tasks:
  - For a STRIPS task Π, we compute its delete relaxed task Π^+,
  - and then apply landmark heuristics on Π^+.
- Hence the objective of our landmark heuristics is to approximate the optimal delete relaxed heuristic h^+ as accurately as possible.
- More advanced landmark techniques work directly on general planning tasks.

Delete-Free STRIPS planning tasks

reminder:

Definition (delete-free STRIPS planning task)

A delete-free STRIPS planning task is a 4-tuple $\Pi^+ = \langle V, I, G, A \rangle$ with the following components:

- V: finite set of state variables
- $I \subseteq V$: the initial state
- $G \subseteq V$: the set of goals
- A: finite set of actions, where for every $a \in A$, we define
  - $pre(a) \subseteq V$: its preconditions
  - $add(a) \subseteq V$: its add effects
  - $cost(a) \in \mathbb{N}_0$: its cost

denoted as $pre(a) \xrightarrow{cost(a)} add(a)$ (omitting set braces)

Delete-Free STRIPS Planning Task in Normal Form

A delete-free STRIPS planning task $\langle V, I, G, A\rangle$ is in normal form if

- I consists of exactly one element i: $I = \{i\}$
- G consists of exactly one element g: $G = \{g\}$
- Every action has at least one precondition.

Every task can easily be transformed into an equivalent task in normal form. (How?)

- In the following, we assume tasks in normal form.
- Describing A suffices to describe overall task:
  - V are the variables mentioned in A's actions.
  - always $I = \{i\}$ and $G = \{g\}$
- In the following, we only describe A.

Example: Delete-Free Planning Task in Normal Form

Example

actions:

- $a_1 = i \xrightarrow{3} x, y$
- $a_2 = i \xrightarrow{4} x, z$
- $a_3 = i \xrightarrow{5} y, z$
- $a_4 = x, y, z \xrightarrow{0} g$

optimal solution?

Example: Delete-Free Planning Task in Normal Form

Example

actions:

- $a_1 = i \xrightarrow{3} x, y$
- $a_2 = i \xrightarrow{4} x, z$
- $a_3 = i \xrightarrow{5} y, z$
- $a_4 = x, y, z \xrightarrow{0} g$

optimal solution to reach $\{g\}$ from $\{i\}$:

- plan: a_1, a_2, a_4
- cost: $3 + 4 + 0 = 7$ $(= h^+(\{i\})$ because plan is optimal)

Landmarks

Landmarks

Definition (landmark)

A landmark of a planning task Π is a set of actions L such that every plan must contain an action from L.

The cost of a landmark L, $cost(L)$ is defined as $\min_{a \in L} cost(a)$.

$\rightsquigarrow$ landmark cost corresponds to (very simple) admissible heuristic

- Speaking more strictly, landmarks as considered in this course are called disjunctive action landmarks.
- other kinds of landmarks exist (fact landmarks, formula landmarks, ...)

Example: Landmarks

Example

actions:

- $a_1 = i \xrightarrow{3} x, y$
- $a_2 = i \xrightarrow{4} x, z$
- $a_3 = i \xrightarrow{5} y, z$
- $a_4 = x, y, z \xrightarrow{0} g$

landmark examples?

Example: Landmarks

Example

actions:

- $a_1 = i \xrightarrow{3} x, y$
- $a_2 = i \xrightarrow{4} x, z$
- $a_3 = i \xrightarrow{5} y, z$
- $a_4 = x, y, z \xrightarrow{0} g$

some landmarks:

- $A = \{a_4\}$ (cost 0)
- $B = \{a_1, a_2\}$ (cost 3)
- $C = \{a_1, a_3\}$ (cost 3)
- $D = \{a_2, a_3\}$ (cost 4)
- also: $\{a_1, a_2, a_3\}$ (cost 3), $\{a_1, a_2, a_4\}$ (cost 0), ...

Overview: Landmarks

in the following:

- exploiting landmarks:
  How can we compute an accurate heuristic
  for a given set of landmarks?
  $\rightsquigarrow$ this chapter
- finding landmarks:
  How can we find landmarks?
  $\rightsquigarrow$ next chapter
- LM-cut heuristic:
  an algorithm to find landmarks and exploit them as a heuristic
  $\rightsquigarrow$ next chapter

Exploiting Landmarks

Exploiting Landmarks

Assume the set of landmarks $\mathcal{L} = \{A, B, C, D\}$.

How to use $\mathcal{L}$ for computing heuristics?

- sum the costs: $0 + 3 + 3 + 4 = 10$
  $\rightsquigarrow$ not admissible!
- maximize the costs: $\max\{0, 3, 3, 4\} = 4$
  $\rightsquigarrow$ usually yields a weak heuristic
- better: hitting sets or cost partitioning

Hitting Sets

Definition (hitting set)

given: finite support set X, family of subsets $\mathcal{F} \subseteq 2^X$,
cost $c : X \to \mathbb{R}_0^+$

hitting set:

- subset $H \subseteq X$ that "hits" all subsets in $\mathcal{F}$:
  $H \cap S \neq \emptyset$ for all $S \in \mathcal{F}$
- cost of H: $\sum_{x \in H} c(x)$

minimum hitting set (MHS):

- hitting set with minimal cost
- "classical" NP-complete problem (Karp, 1972)

Example: Hitting Sets

Example

$X = \{a_1, a_2, a_3, a_4\}$

$\mathcal{F} = \{A, B, C, D\}$
with $A = \{a_4\}$, $B = \{a_1, a_2\}$, $C = \{a_1, a_3\}$, $D = \{a_2, a_3\}$

$c(a_1) = 3$, $c(a_2) = 4$, $c(a_3) = 5$, $c(a_4) = 0$

minimum hitting set:

Example: Hitting Sets

Example

$X = \{a_1, a_2, a_3, a_4\}$

$\mathcal{F} = \{A, B, C, D\}$
with $A = \{a_4\}, \;\; B = \{a_1, a_2\}, \;\; C = \{a_1, a_3\}, \;\; D = \{a_2, a_3\}$

$c(a_1) = 3, \;\; c(a_2) = 4, \;\; c(a_3) = 5, \;\; c(a_4) = 0$

minimum hitting set: $\{a_1, a_2, a_4\}$ with cost $3 + 4 + 0 = 7$

Hitting Sets for Landmarks

idea: landmarks are interpreted as instance of minimum hitting set

Definition (hitting set heuristic)

Let $\mathcal{L}$ be a set of landmarks for a delete-free planning task in normal form with actions A, action costs *cost* and initial state I.

The hitting set heuristic $h^{\mathrm{MHS}}(I)$ is defined as the minimal solution cost for the minimum hitting set instance with support set A, family of subsets $\mathcal{L}$ and costs *cost*.

Proposition (Hitting Set Heuristic is Admissible)

The minimum hitting set heuristic h^{MHS} is admissible.

Why?

Computing Hitting Sets with Integer Programs

Minimal hitting sets can be computed with Integer Programs:

- one binary variable u_x for every element $x \in X$
  $\rightsquigarrow$ value 1 iff x is used as part of the hitting set H
- one constraint for each set $S \in \mathcal{F}$
  $\rightsquigarrow$ encodes that at least one element from S has to be used
- objective is to minimize total cost of used items

Definition (hitting set IP)

Minimize $\sum_{x \in X} u_x \cdot c(x)$ subject to

$$\sum_{x \in S} u_x \geq 1 \qquad \text{for all } S \in \mathcal{F}$$

$$u_x \in \{0, 1\} \qquad \text{for all } x \in X$$

Computing h^{MHS} with Integer Programs

h^{MHS} can be computed with Integer Programs:

- one binary variable u_a for every action $a \in A$
  $\rightsquigarrow$ value 1 iff a is used as part of the hitting set H
- one constraint for each landmark $L \in \mathcal{L}$
  $\rightsquigarrow$ encodes that at least one action from L has to be used
- objective is to minimize total cost of used actions

Definition (h^{MHS} IP)

Minimize $\sum_{a \in A} u_a \cdot cost(a)$ subject to

$$\sum_{a \in L} u_a \geq 1 \qquad \text{for all } L \in \mathcal{L}$$

$$u_a \in \{0, 1\} \qquad \text{for all } a \in A$$

Approximation of h^{MHS}

- As computing minimal hitting sets is NP-hard, we want to approximate h^{MHS} in polynomial time.
- Solving the LP-relaxation of the IP is possible in polynomial time and gives a lower bound.

Definition (h^{MHS} IP)

Minimize $\sum_{a \in A} u_a \cdot cost(a)$ subject to

$$\sum_{a \in L} u_a \geq 1 \qquad \text{for all } L \in \mathcal{L}$$

$$u_a \in \{0, 1\} \qquad \text{for all } a \in A$$

Approximation of h^{MHS}

- As computing minimal hitting sets is NP-hard, we want to approximate h^{MHS} in polynomial time.
- Solving the LP-relaxation of the IP is possible in polynomial time and gives a lower bound.

Definition (h^{MHS} LP-relaxation)

Minimize $\sum_{a \in A} u_a \cdot cost(a)$ subject to

$$\sum_{a \in L} u_a \geq 1 \qquad \text{for all } L \in \mathcal{L}$$

$$u_a \in \mathbb{R}^+ \qquad \text{for all } a \in A$$

Approximation of h^{MHS}

- As computing minimal hitting sets is NP-hard, we want to approximate h^{MHS} in polynomial time.
- Solving the LP-relaxation of the IP is possible in polynomial time and gives a lower bound.

Definition ($h^{\mathrm{MHS\text{-}LP}}$)

$$\text{Minimize } \sum_{a \in A} u_a \cdot cost(a) \text{ subject to}$$

$$\begin{aligned} \sum_{a \in L} u_a &\geq 1 && \text{for all } L \in \mathcal{L} \\ u_a &\in \mathbb{R}^+ && \text{for all } a \in A \end{aligned}$$

Originally expressed in a different form as optimal cost partitioning (Karpas & Domshlak, 2009).

Example: $h^{\text{MHS-LP}}$

Example

$cost(a_1) = 3, \quad cost(a_2) = 4, \quad cost(a_3) = 5, \quad cost(a_4) = 0$

$\mathcal{L} = \{A, B, C, D\}$
with $A = \{a_4\}, \quad B = \{a_1, a_2\}, \quad C = \{a_1, a_3\}, \quad D = \{a_2, a_3\}$

LP:

Minimize $3u_{a_1} + 4u_{a_2} + 5u_{a_3} + 0u_{a_4}$ subject to

$$
\begin{aligned}
u_{a_4} &\geq 1 && (\rightsquigarrow A) \\
u_{a_1} + u_{a_2} &\geq 1 && (\rightsquigarrow B) \\
u_{a_1} + u_{a_3} &\geq 1 && (\rightsquigarrow C) \\
u_{a_2} + u_{a_3} &\geq 1 && (\rightsquigarrow D) \\
u_{a_i} &\in \mathbb{R}^+ && \text{for } i \in \{1, 2, 3, 4\}
\end{aligned}
$$

optimal solution: ?

Example: $h^{\text{MHS-LP}}$

Example

$cost(a_1) = 3, \quad cost(a_2) = 4, \quad cost(a_3) = 5, \quad cost(a_4) = 0$

$\mathcal{L} = \{A, B, C, D\}$
with $A = \{a_4\}, \quad B = \{a_1, a_2\}, \quad C = \{a_1, a_3\}, \quad D = \{a_2, a_3\}$

LP:

Minimize $3u_{a_1} + 4u_{a_2} + 5u_{a_3} + 0u_{a_4}$ subject to

$$
\begin{aligned}
u_{a_4} &\geq 1 && (\rightsquigarrow A)\\
u_{a_1} + u_{a_2} &\geq 1 && (\rightsquigarrow B)\\
u_{a_1} + u_{a_3} &\geq 1 && (\rightsquigarrow C)\\
u_{a_2} + u_{a_3} &\geq 1 && (\rightsquigarrow D)\\
u_{a_i} &\in \mathbb{R}^+ && \text{for } i \in \{1, 2, 3, 4\}
\end{aligned}
$$

optimal solution:

$u_{a_1} = 0.5, \ u_{a_2} = 0.5, \ u_{a_3} = 0.5, \ u_{a_4} = 1 \quad \rightsquigarrow \quad h^{\text{MHS-LP}}(I) = 6$

Relationship of Heuristics

Proposition ($h^{\text{MHS-LP}}$ vs. h^{MHS})

Let $\mathcal{L}$ be a set of landmarks for a planning task with initial state I.

Then $h^{\text{MHS-LP}}(I) \leq h^{\text{MHS}}(I) \leq h^{+}(I)$

Summary

Summary

- Landmarks are action sets such that every plan must contain at least one of the actions.
- Hitting sets yield the most accurate heuristic for a given set of landmarks, but the computation is NP-hard.
- With LP-relaxation we get a polynomial approach for the computation of informative landmark heuristics.