

Planning and Optimization

C17. Critical Path Heuristics: h^m

Malte Helmert and Gabriele Röger

Universität Basel

November 21, 2016

Planning and Optimization

November 21, 2016 — C17. Critical Path Heuristics: h^m

C17.1 Introduction

C17.2 Perfect Regression Heuristic

C17.3 Critical Path Heuristics

C17.4 Computation

C17.5 Summary

C17.1 Introduction

Planning Heuristics: Main Concepts

Major ideas for heuristics in the planning literature:

- ▶ delete relaxation ✓
- ▶ abstraction ✓
- ▶ critical paths ← this and next chapter
- ▶ landmarks
- ▶ network flows

In This (and the Next) Chapter...

- ▶ ... we consider only STRIPS, and ...
- ▶ ... we focus on backward search and regression.

Set Representation of STRIPS Planning Tasks

For a more convenient notation, we will use a different representation of STRIPS task...

Three differences:

- ▶ Represent conjunctions of variables as sets of variables.
- ▶ Represent states as sets of the true variables.
- ▶ Use two sets to represent add and delete effects of operators separately.

STRIPS Operators in Set Representation

- ▶ Previously, every STRIPS operator was of the form

$$\langle v_1 \wedge \dots \wedge v_p, a_1 \wedge \dots \wedge a_q \wedge \neg d_1 \wedge \dots \wedge \neg d_r, c \rangle$$

where v_i, a_j, d_k are state variables and c is the cost.

- ▶ The same operator o in **set representation** is $\langle pre(o), add(o), del(o), cost(o) \rangle$, where
 - ▶ $pre(o) = \{v_1, \dots, v_p\}$ are the **preconditions**,
 - ▶ $add(o) = \{a_1, \dots, a_q\}$ are the **add effects**,
 - ▶ $del(o) = \{d_1, \dots, d_r\}$ are the **delete effects**, and
 - ▶ $cost(o) = c$ is the operator cost.

STRIPS Planning Tasks in Set Representation

A **STRIPS planning task in set representation** is given as a tuple $\langle V, I, O, G \rangle$, where

- ▶ V is a finite set of state variables,
- ▶ $I \subseteq V$ is the initial state,
- ▶ O is a finite set of STRIPS operators in set representation,
- ▶ $G \subseteq V$ is the goal.

The corresponding planning task in the previous notation is $\langle V, I', O', \gamma \rangle$, where

- ▶ $I'(v) = \mathbf{T}$ iff $v \in I$,
- ▶ $O' = \{ \langle \bigwedge_{v \in pre(o)} v, \bigwedge_{v \in add(o)} v \wedge \bigwedge_{v \in del(o)} \neg v, cost(o) \rangle \mid o \in O \}$,
- ▶ $\gamma = \bigwedge_{v \in G} v$.

Reminder: STRIPS Regression

Definition (STRIPS Regression)

Let $\varphi = \varphi_1 \wedge \dots \wedge \varphi_n$ be a conjunction of atoms, and let o be a STRIPS operator which adds the atoms a_1, \dots, a_k and deletes the atoms d_1, \dots, d_l . (W.l.o.g., $a_i \neq d_j$ for all i, j .)

The **STRIPS regression** of φ with respect to o is

$$\text{sregr}_o(\varphi) := \begin{cases} \perp & \text{if } \varphi_i = d_j \text{ for some } i, j \\ \text{pre}(o) \wedge \bigwedge (\{\varphi_1, \dots, \varphi_n\} \setminus \{a_1, \dots, a_k\}) & \text{otherwise} \end{cases}$$

Note: $\text{sregr}_o(\varphi)$ is again a conjunction of atoms, or \perp .

STRIPS Regression in Set Representation

Definition (STRIPS Regression)

Let A be a set of atoms, and let o be a STRIPS operator $o = \langle \text{pre}(o), \text{add}(o), \text{del}(o), \text{cost}(o) \rangle$. (W.l.o.g., $\text{add}(o) \cap \text{del}(o) = \emptyset$.)

The **STRIPS regression** of A with respect to o is

$$\text{sregr}_o(A) := \begin{cases} \perp & \text{if } A \cap \text{del}(o) \neq \emptyset \\ \text{pre}(o) \cup (A \setminus \text{add}(o)) & \text{otherwise} \end{cases}$$

Note: $\text{sregr}_o(A)$ is again a set of atoms, or \perp .

C17.2 Perfect Regression Heuristic

Perfect Regression Heuristic

Definition (Perfect Regression Heuristic)

For a STRIPS planning task $\langle V, I, O, G \rangle$ the **perfect regression heuristic** r^* for state s and variable set $A \subseteq V$ is defined as the (point-wise) greatest fixed-point solution of the equations:

$$r^*(s, A) = \begin{cases} 0 & \text{if } A \subseteq s \\ \min_{(B, o) \in R_o(A)} [\text{cost}(o) + r^*(s, B)] & \text{otherwise} \end{cases}$$

$$R_o(A) = \{(B, o) \mid o \in O, B = \text{sregr}_o(A) \neq \perp\}$$

Perfect Regression Heuristic r^* vs. Perfect Heuristic h^*

Theorem

For a STRIPS planning task $\langle V, I, O, G \rangle$ it holds for each state s that $h^*(s) = r^*(s, G)$.

Intuition: We can extract a path from the operators in the minimizing pairs (B, o) , starting from the goal.

$\rightsquigarrow r^*$ cannot be computed efficiently.

C17.3 Critical Path Heuristics

Running Example

We will use the following running example throughout this chapter:

$\Pi = \langle V, I, \{o_1, o_2, o_3\}, G \rangle$ with

$$V = \{a, b, c\}$$

$$I = \{a\}$$

$$o_1 = \langle \{a, b\}, \{c\}, \{b\}, 1 \rangle$$

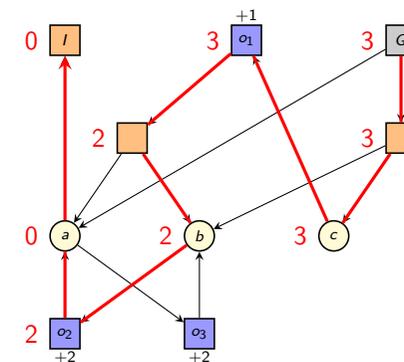
$$o_2 = \langle \{a\}, \{b\}, \{a\}, 2 \rangle$$

$$o_3 = \langle \{b\}, \{a\}, \emptyset, 2 \rangle$$

$$G = \{a, b, c\}$$

Optimal plan o_2, o_3, o_1, o_2, o_3 has cost 9.

Back to RTG and h^{\max}



The critical path justifies the heuristic estimate $h^{\max}(I) = 3$

h^{\max} as Critical Path Heuristic

Definition (h^{\max} Heuristic)

For a STRIPS planning task $\langle V, I, O, G \rangle$ the heuristic h^{\max} for state s and variable set $A \subseteq V$ is defined as the (point-wise) greatest fixed-point solution of $h^{\max}(s, A) =$

$$\begin{cases} 0 & \text{if } A \subseteq s \\ \min_{(B,o) \in R_O(A)} [\text{cost}(o) + h^{\max}(s, B)] & \text{if } |A| \leq 1 \text{ and } A \not\subseteq s \\ \max_{v \in A} h^m(s, \{v\}) & \text{otherwise} \end{cases}$$

$$R_O(A) = \{ \langle B, o \rangle \mid o \in O, B = \text{sregr}_o(A) \neq \perp \}$$

Estimate $r^*(s, A)$ as cost of most expensive $v \in A$.

This definition specifies the same heuristic h^{\max} as in the chapter on relaxation heuristics.

Critical Path Heuristics

Definition (h^m Heuristics)

For a STRIPS planning task $\langle V, I, O, G \rangle$ and $m \in \mathbb{N}$ the heuristic h^m for state s and variable set $A \subseteq V$ is defined as the (point-wise) greatest fixed-point solution of $h^m(s, A) =$

$$\begin{cases} 0 & \text{if } A \subseteq s \\ \min_{(B,o) \in R_O(A)} [\text{cost}(o) + h^m(s, B)] & \text{if } |A| \leq m \text{ and } A \not\subseteq s \\ \max_{B \subseteq A, 1 \leq |B| \leq m} h^m(s, B) & \text{otherwise} \end{cases}$$

$$R_O(A) = \{ \langle B, o \rangle \mid o \in O, B = \text{sregr}_o(A) \neq \perp \}$$

Estimate $r^*(s, A)$ as cost of most expensive $B \subseteq A$ with $|B| \leq m$.

C17.4 Computation

Critical Path Heuristics: Computation

Definition (h^m Heuristics)

For a STRIPS planning task $\langle V, I, O, G \rangle$ and $m \in \mathbb{N}$ the heuristic h^m for state s and variable set $A \subseteq V$ is defined as the (point-wise) greatest fixed-point solution of $h^m(s, A) =$

$$\begin{cases} 0 & \text{if } A \subseteq s \\ \min_{(B,o) \in R_O(A)} [\text{cost}(o) + h^m(s, B)] & \text{if } |A| \leq m \text{ and } A \not\subseteq s \\ \max_{B \subseteq A, 1 \leq |B| \leq m} h^m(s, B) & \text{otherwise} \end{cases}$$

$$R_O(A) = \{ \langle B, o \rangle \mid o \in O, B = \text{sregr}_o(A) \neq \perp \}$$

Cheap to evaluate given $h^m(s, B)$ for all $B \subseteq V$ with $1 \leq |B| \leq m$.
We precompute these values.

h^m Precomputation (1)

For value m and state s of task with variables V and operators O

Computing h^m Values for Variable Sets up to Size m

$S := \{A \subseteq V \mid |A| \leq m\}$

Associate a *cost* attribute with each set $A \in S$.

for all sets $A \in S$:

if $A \subseteq s$ then $A.cost := 0$

else $A.cost := \infty$

while no fixed point is reached:

Choose a variable set A from S .

$newcost := \min_{\langle B, o \rangle \in R_O(A)} [cost(o) + currentcost(B, S)]$

if $newcost < A.cost$ then $A.cost := newcost$

currentcost(B, S)

if $|B| \leq m$ then return $B.cost$ else return $\max_{A \in S, A \subseteq B} A.cost$

 h^m Precomputation (2)

- ▶ Fixed point reached $\Rightarrow A.cost = h^m(s, A)$ for all $A \in S$.
- ▶ Intuition:
 - ▶ cost values satisfy h^m equations, and
 - ▶ no larger values can satisfy the equations: initialized to ∞ and values are only reduced if it is otherwise impossible to satisfy an equation.
- ▶ With suitable data structures, we can choose A in each iteration so that it directly gets assigned its final value (Generalized Dijkstra's algorithm).
- ▶ With such a strategy, the runtime is **polynomial for fixed m** .
- ▶ Runtime is **exponential in m** $\rightsquigarrow h^m$ typically used with $m \leq 3$

Example with $m = 1$ to Initial State

$$R_{\{o_1, o_2, o_3\}}(\{a\}) = \{(\{a, b, c\}, o_1), (\{b\}, o_3)\}$$

$$R_{\{o_1, o_2, o_3\}}(\{b\}) = \{(\{a\}, o_2), (\{b\}, o_3)\}$$

$$R_{\{o_1, o_2, o_3\}}(\{c\}) = \{(\{a, b\}, o_1), (\{a, c\}, o_2), (\{b, c\}, o_3)\}$$

	$\{a\}$	$\{b\}$	$\{c\}$
cost	0	2	3

$$\{b\}: \min\{2 + \{a\}.cost, 2 + \{b\}.cost\} = 2$$

$$\{c\}: \min\{1 + \max\{\{a\}.cost, \{b\}.cost\}, \\ 2 + \max\{\{a\}.cost, \{c\}.cost\}, \\ 2 + \max\{\{b\}.cost, \{c\}.cost\}\} = 3$$

Example with $m = 1$ to Initial State

	$\{a\}$	$\{b\}$	$\{c\}$
cost	0	2	3

$$\{b\}: \min\{2 + \{a\}.cost, 2 + \{b\}.cost\} = 2$$

$$\{c\}: \min\{1 + \max\{\{a\}.cost, \{b\}.cost\}, \\ 2 + \max\{\{a\}.cost, \{c\}.cost\}, \\ 2 + \max\{\{b\}.cost, \{c\}.cost\}\} = 3$$

Fixed point reached

$$h^1(I, \{a, b, c\}) = \max\{h^1(I, \{a\}), h^1(I, \{b\}), h^1(I, \{c\})\} \\ = \max\{0, 2, 3\} = 3$$

Example with $m = 2$ to Initial State

	$\{a\}$	$\{b\}$	$\{c\}$	$\{a, b\}$	$\{a, c\}$	$\{b, c\}$
<i>cost</i>	0	2	5	4	5	7

$$\begin{aligned} \{b\}: & \min\{2 + \{a\}.cost, 2 + \{b\}.cost\} = 2 \\ \{a, b\}: & \min\{2 + \{b\}.cost\} = 4 \\ \{c\}: & \min\{1 + \{a, b\}.cost, 2 + \{a, c\}.cost, 2 + \{b, c\}.cost\} = 5 \\ \{a, c\}: & \min\{1 + \{a, b\}.cost, 2 + \{b, c\}.cost\} = 5 \\ \{b, c\}: & \min\{2 + \{a, c\}.cost, 2 + \{b, c\}.cost\} = 7 \end{aligned}$$

$$\begin{aligned} h^2(I, \{a, b, c\}) &= \max\{h^2(I, \{a\}), h^2(I, \{b\}), h^2(I, \{c\})\} \\ &\quad h^2(I, \{a, b\}), h^2(I, \{a, c\}), h^2(I, \{b, c\})\} \\ &= \max\{0, 2, 5, 4, 5, 7\} = 7 \end{aligned}$$

C17.5 Summary

Summary

- ▶ **Critical path heuristic h^m** estimates the cost of reaching a set ($\hat{=}$ conjunction) of variables as the **cost of reaching the most expensive subset of size at most m** .
- ▶ h^m computation is **polynomial for fixed m** .
- ▶ h^m computation is **exponential in m** .
- ▶ In practice, we use $m \in \{1, 2, 3\}$.