Planning and Optimization

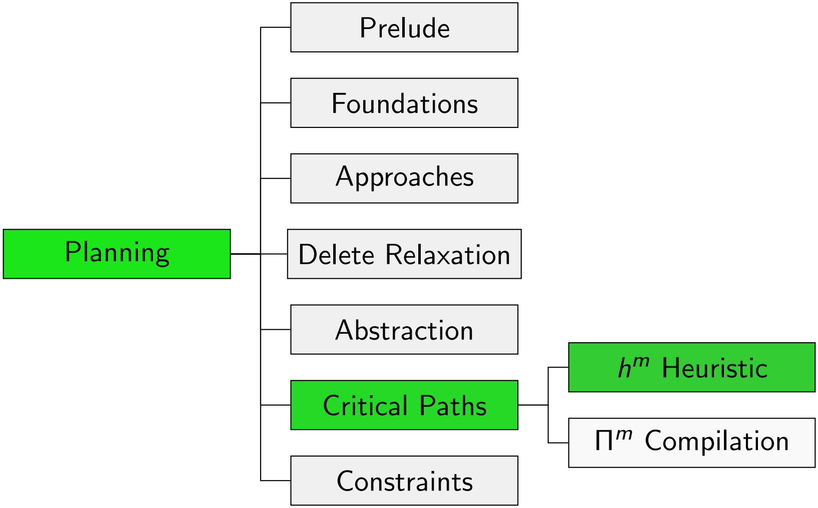
F2. Critical Path Heuristics: Properties and Π^m Compilation

Malte Helmert and Gabriele Röger

Universität Basel

November 27, 2023

Content of this Course



Heuristic Properties

Heuristic for Forward or Backward Search? (1)

Any heuristic can be used for both, forward and backward search:

- Let h_f be a forward search heuristic (as in earlier chapters). We can use it to get estimate for state S in backward search on task (V, I, O, G), computing $h_f(I)$ on task (V, I, O, S).

Heuristic for Forward or Backward Search? (1)

Any heuristic can be used for both, forward and backward search:

- Let h_f be a forward search heuristic (as in earlier chapters). We can use it to get estimate for state S in backward search on task (V, I, O, G), computing $h_f(I)$ on task (V, I, O, S).
- We also can use a backward search heuristic h_b in forward search on task (V, I, O, G), determining estimate for state s as $h_b(G)$ on task (V, s, O, G).

Heuristic for Forward or Backward Search? (2)

We defined h^m so that it can directly be used for both directions on task (V, I, O, G) as

- $h^m_f(s) := h^m(s, G)$ for forward search, or
- $h^m_b(S) := h^m(I, S)$ for backward search.

Precomputation determines $h^m(s, B)$ for all $B \subseteq V$ with $|B| \leq m$.

- For h^m_f, we can only use these values for a single heuristic evaluation, because the state s changes.
- For h^m_b, we can re-use these values and all subsequent heuristic evaluations are quite cheap.

$\rightarrow$ h^m better suited for backward search

$\rightarrow$ We examine it in the following in this context.

Heuristic Properties (1)

Theorem

Let $\Pi = \langle V, I, O, G \rangle$ be a STRIPS planning tasks and $S \subseteq V$ be a backward search state. Then $h^m_b(S) := h^m(I, S)$ is a safe, goal-aware, consistent, and admissible heuristic for Π.

Proof.

We prove goal-awareness and consistency, the other properties follow from these two.

Goal-awareness: S is a goal state iff $S \subseteq I$. Then $h^m_b(S) = h^m(I, S) = 0$. ...

Heuristic Properties (2)

Proof (continued).

Consistency: Assume h_b^m is not consistent, i.e., there is a state S and an operator o, where $R := sregr(S, o) \neq \perp$ such that $h_b^m(S) > cost(o) + h_b^m(R)$.

Then $h_b^m(S) = h^m(I, S)$ and there is $S' \subseteq S$ with $|S'| \leq m$ and $h^m(I, S') = h^m(I, S)$: if $|S| \leq m$, choose $S' = S$, otherwise choose any maximizing subset from the last h^m equation.

As $S' \subseteq S$ and $sregr(S, o) \neq \perp$, also $R' := sregr(S', o) \neq \perp$ and $(R', o) \in R(S', O)$. This gives $h^m(I, S') \leq cost(o) + h^m(I, R')$.

As $S' \subseteq S$, it holds that $R' \subseteq R$ and $h^m(I, R') \leq h^m(I, R)$.

Overall, we get $h_b^m(S) = h^m(I, S) = h^m(I, S') \leq cost(o) + h^m(I, R') \leq cost(o) + h^m(I, R) = cost(o) + h_b^m(R)$. ↯ □

Heuristic Properties (3)

Theorem

For $m, m' \in \mathbb{N}_1$ with $m < m'$ it holds that $h^m \leq h^{m'}$.

(Proof omitted.)

Heuristic Properties (4)

Theorem

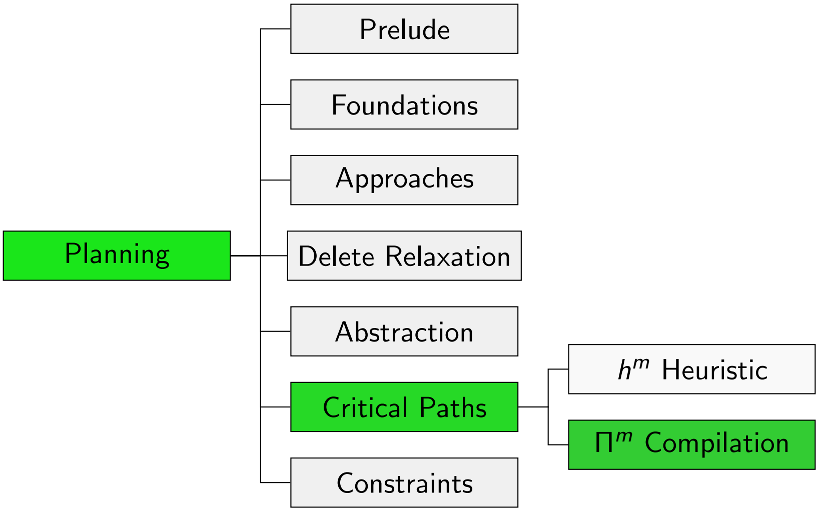
Let $\Pi = \langle V, I, O, G\rangle$ be a STRIPS planning task.
For a sufficiently large m, it holds that $h^m = r^*$ on Π.

Proof Sketch.

It is easy to check that for $m = |V|$ the heuristic definition of h^m can be simplified so that it becomes the definition of r^*.

Π^m Compilation

Content of this Course



Π^m Compilation: Motivation

- We have seen that $h^1 = h^{\max}$ and that $h^{\max}$ corresponds to the cost of a critical path in the relaxed task graph.
- What about $m > 1$?
- Π^m compilation derives for a given m a task Π^m from the original task Π.
- h^m corresponds to cost of critical path in the relaxed task graph of Π^m.

$\rightarrow$ Better understanding of h^m

$\rightarrow$ Also interesting in the context of landmark heuristics

Idea of Π^m Compilation

- $h^{\max}$ only considers variables individually.
- For example, it cannot detect that a goal $\{a, b\}$ is unreachable from the empty set if every action that adds a deletes b and vice versa.
- Idea: Use meta-variable $v_{\{a,b\}}$ to capture such interactions.
- Intuitively $v_{\{a,b\}}$ is reachable in Π^m if a state where a and b are both true would be reachable in Π when only capturing interactions of at most m variables.

Some Notation

- For a set X of variables and $m \in \mathbb{N}_1$ we define $X^m := \{v_Y \mid Y \subseteq X, |Y| \leq m\}$.
- Example: $\{a, b, c\}^2 = \{v_\emptyset, v_{\{a\}}, v_{\{b\}}, v_{\{c\}}, v_{\{a,b\}}, v_{\{a,c\}}, v_{\{b,c\}}\}$

Π^m Compilation

Definition (Π^m)

Let $\Pi = \langle V, I, O, G \rangle$ be a STRIPS planning task.
For $m \in \mathbb{N}_1$, the task Π^m is the STRIPS planning task $\langle V^m, I^m, O^m, G^m \rangle$, where
$O^m = \{a_{o,S} \mid o \in O, S \subseteq V, |S| < m, S \cap (add(o) \cup del(o)) = \emptyset\}$
with

- $pre(a_{o,S}) = (pre(o) \cup S)^m$
- $add(a_{o,S}) = \{v_Y \mid Y \subseteq add(o) \cup S, |Y| \leq m, Y \cap add(o) \neq \emptyset\}$
- $del(a_{o,S}) = \emptyset$
- $cost(a_{o,S}) = cost(o)$

Π^m for Running Example with $m = 2$

For running example Π we get $\Pi^2 = \langle V', I', O', G' \rangle$, where

Π^m for Running Example with $m = 2$

For running example Π we get $\Pi^2 = \langle V', I', O', G' \rangle$, where

$$V' = \{v_\emptyset, v_{\{a\}}, v_{\{b\}}, v_{\{c\}}, v_{\{a,b\}}, v_{\{a,c\}}, v_{\{b,c\}}\}$$

$V = \{a, b, c\}$
$V' = V^2 = \{v_Y \mid Y \subseteq V, |Y| \leq 2\}$

Π^m for Running Example with $m = 2$

For running example Π we get $\Pi^2 = \langle V', I', O', G' \rangle$, where

$$V' = \{v_\emptyset, v_{\{a\}}, v_{\{b\}}, v_{\{c\}}, v_{\{a,b\}}, v_{\{a,c\}}, v_{\{b,c\}}\}$$
$$I' = \{v_\emptyset, v_{\{a\}}\}$$

$I = \{a\}$
$I' = I^2 = \{v_Y \mid Y \subseteq I, |Y| \leq 2\}$

Π^m for Running Example with $m = 2$

For running example Π we get $\Pi^2 = \langle V', I', O', G' \rangle$, where

$$\begin{aligned} V' &= \{v_\emptyset, v_{\{a\}}, v_{\{b\}}, v_{\{c\}}, v_{\{a,b\}}, v_{\{a,c\}}, v_{\{b,c\}}\} \\ I' &= \{v_\emptyset, v_{\{a\}}\} \\ G' &= \{v_\emptyset, v_{\{a\}}, v_{\{b\}}, v_{\{c\}}, v_{\{a,b\}}, v_{\{a,c\}}, v_{\{b,c\}}\} \end{aligned}$$

$G = \{a, b, c\}$
$G' = G^2 = \{v_Y \mid Y \subseteq G, |Y| \leq 2\}$

Π^m for Running Example with $m = 2$

For running example Π we get $\Pi^2 = \langle V', I', O', G' \rangle$, where

$$
\begin{aligned}
V' &= \{v_\emptyset, v_{\{a\}}, v_{\{b\}}, v_{\{c\}}, v_{\{a,b\}}, v_{\{a,c\}}, v_{\{b,c\}}\} \\
I' &= \{v_\emptyset, v_{\{a\}}\} \\
G' &= \{v_\emptyset, v_{\{a\}}, v_{\{b\}}, v_{\{c\}}, v_{\{a,b\}}, v_{\{a,c\}}, v_{\{b,c\}}\} \\
O' &= \{a_{o_1,\emptyset}, a_{o_1,\{a\}}, a_{o_2,\emptyset}, a_{o_2,\{c\}}, a_{o_3,\emptyset}, a_{o_3,\{b\}}, a_{o_3,\{c\}}\}
\end{aligned}
$$

$o_1 = \langle \{a, b\}, \{c\}, \{b\}, 1 \rangle$
$o_2 = \langle \{a\}, \{b\}, \{a\}, 2 \rangle$
$o_3 = \langle \{b\}, \{a\}, \emptyset, 2 \rangle$
$O' = \{a_{o,S} \mid o \in O, S \subseteq V, |S| < m, S \cap (add(o) \cup del(o)) = \emptyset\}$

Π^m for Running Example with $m = 2$

For running example Π we get $\Pi^2 = \langle V', I', O', G' \rangle$, where

$$\begin{aligned}
V' &= \{v_\emptyset, v_{\{a\}}, v_{\{b\}}, v_{\{c\}}, v_{\{a,b\}}, v_{\{a,c\}}, v_{\{b,c\}}\}\\
I' &= \{v_\emptyset, v_{\{a\}}\}\\
G' &= \{v_\emptyset, v_{\{a\}}, v_{\{b\}}, v_{\{c\}}, v_{\{a,b\}}, v_{\{a,c\}}, v_{\{b,c\}}\}\\
O' &= \{a_{o_1,\emptyset}, a_{o_1,\{a\}}, a_{o_2,\emptyset}, a_{o_2,\{c\}}, a_{o_3,\emptyset}, a_{o_3,\{b\}}, a_{o_3,\{c\}}\}
\end{aligned}$$

with (for example)

$$a_{o_3,\{c\}} = \langle\{v_\emptyset, v_{\{b\}}, v_{\{c\}}, v_{\{b,c\}}\}, \ldots, \ldots, \ldots\rangle$$

$o_3 = \langle\{b\}, \{a\}, \emptyset, 2\rangle$
$pre(a_{o,S}) = (pre(o) \cup S)^2$

Π^m for Running Example with $m = 2$

For running example Π we get $\Pi^2 = \langle V', I', O', G' \rangle$, where

$$
\begin{aligned}
V' &= \{v_\emptyset, v_{\{a\}}, v_{\{b\}}, v_{\{c\}}, v_{\{a,b\}}, v_{\{a,c\}}, v_{\{b,c\}}\} \\
I' &= \{v_\emptyset, v_{\{a\}}\} \\
G' &= \{v_\emptyset, v_{\{a\}}, v_{\{b\}}, v_{\{c\}}, v_{\{a,b\}}, v_{\{a,c\}}, v_{\{b,c\}}\} \\
O' &= \{a_{o_1,\emptyset}, a_{o_1,\{a\}}, a_{o_2,\emptyset}, a_{o_2,\{c\}}, a_{o_3,\emptyset}, a_{o_3,\{b\}}, a_{o_3,\{c\}}\}
\end{aligned}
$$

with (for example)

$$a_{o_3,\{c\}} = \langle \{v_\emptyset, v_{\{b\}}, v_{\{c\}}, v_{\{b,c\}}\}, \{v_{\{a\}}, v_{\{a,c\}}\}, \dots, \dots \rangle$$

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

$add(a_{o,S}) = \{v_Y \mid Y \subseteq add(o) \cup S, |Y| \leq m, Y \cap add(o) \neq \emptyset\}$

Π^m for Running Example with $m = 2$

For running example Π we get $\Pi^2 = \langle V', I', O', G' \rangle$, where

$$\begin{aligned}
V' &= \{v_\emptyset, v_{\{a\}}, v_{\{b\}}, v_{\{c\}}, v_{\{a,b\}}, v_{\{a,c\}}, v_{\{b,c\}}\} \\
I' &= \{v_\emptyset, v_{\{a\}}\} \\
G' &= \{v_\emptyset, v_{\{a\}}, v_{\{b\}}, v_{\{c\}}, v_{\{a,b\}}, v_{\{a,c\}}, v_{\{b,c\}}\} \\
O' &= \{a_{o_1,\emptyset}, a_{o_1,\{a\}}, a_{o_2,\emptyset}, a_{o_2,\{c\}}, a_{o_3,\emptyset}, a_{o_3,\{b\}}, a_{o_3,\{c\}}\}
\end{aligned}$$

with (for example)

$$a_{o_3,\{c\}} = \langle \{v_\emptyset, v_{\{b\}}, v_{\{c\}}, v_{\{b,c\}}\}, \{v_{\{a\}}, v_{\{a,c\}}\}, \emptyset, \ldots \rangle$$

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

$del(a_{o,s}) = \emptyset$

Π^m for Running Example with $m = 2$

For running example Π we get $\Pi^2 = \langle V', I', O', G' \rangle$, where

$$\begin{aligned}
V' &= \{v_\emptyset, v_{\{a\}}, v_{\{b\}}, v_{\{c\}}, v_{\{a,b\}}, v_{\{a,c\}}, v_{\{b,c\}}\} \\
I' &= \{v_\emptyset, v_{\{a\}}\} \\
G' &= \{v_\emptyset, v_{\{a\}}, v_{\{b\}}, v_{\{c\}}, v_{\{a,b\}}, v_{\{a,c\}}, v_{\{b,c\}}\} \\
O' &= \{a_{o_1,\emptyset}, a_{o_1,\{a\}}, a_{o_2,\emptyset}, a_{o_2,\{c\}}, a_{o_3,\emptyset}, a_{o_3,\{b\}}, a_{o_3,\{c\}}\}
\end{aligned}$$

with (for example)

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

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

$cost(a_{o,S}) = cost(o)$

Π^m: Properties

Theorem ($h^m_\Pi = h^{\max}_{\Pi^m}$)

Let Π be a STRIPS planning task and $m \in \mathbb{N}_1$.

Then for each state s of Π it holds that $h^m_\Pi(s) = h^{max}_{\Pi^m}(s^m)$, where the subscript denotes on which task the heuristic is computed.

(Proof omitted.)

Can we in general compute an admissible heuristic on Π^m and get admissible estimates for Π? $\rightsquigarrow$ No!

Theorem

There are STRIPS planning tasks Π, $m \in \mathbb{N}_1$ and admissible heuristics h such that $h^*_\Pi(s) < h^*_{\Pi^m}(s^m)$ for some state s of Π.

(Proof omitted.)

Intuition: we may need separate copies of the same action to achieve different meta-fluents

Π^C Compilation

Outlook: Π^C and Π^C_{ce} Compilation

- Π^m (and h^m) must consider all subsets up to size m.
- $h^*_{\Pi^m}$ is in general not admissible for Π.

Outlook: Π^C and Π^C_{ce} Compilation

- Π^m (and h^m) must consider all subsets up to size m.
- $h^*_{\Pi^m}$ is in general not admissible for Π.
- The compilation Π^C is defined for a set C of atom sets.
  - C can contain arbitrary subsets of arbitrary size.
  - Task Π^C is again delete-free.
  - $h^+_{\Pi^C} = h^*_{\Pi^C}$ is admissible for Π.
  - The task representation is exponential in $|C|$ (one action copy for every set of meta-variables the action can make true).

Outlook: Π^C and Π^C_{ce} Compilation

- Π^m (and h^m) must consider all subsets up to size m.
- $h^*_{\Pi^m}$ is in general not admissible for Π.
- The compilation Π^C is defined for a set C of atom sets.
  - C can contain arbitrary subsets of arbitrary size.
  - Task Π^C is again delete-free.
  - $h^+_{\Pi^C} = h^*_{\Pi^C}$ is admissible for Π.
  - The task representation is exponential in $|C|$ (one action copy for every set of meta-variables the action can make true).
- Π^C_{ce} is an alternative to Π^C using conditional effects
  - Π^C_{ce} can be exponentially smaller (in $|C|$) than Π^C.
  - $h^+_{\Pi^C}$ dominates $h^+_{\Pi^C_{\mathrm{ce}}}$ for set C of non-unit sets.

Summary

Summary

- h^m heuristics are best suited for backward search.
- h^m heuristics are safe, goal aware, consistent and admissible.
- The Π^m compilation explicitly represents sets ($\hat{=}$ conjunctions) of variables as meta-variables.
- $h^m_\Pi(s) = h^{\max}_{\Pi^m}(s^m)$
- The ideas underlying the Π^m compilation have been generalized to the Π^C and Π^C_{ce} compilation.

Literature

Literature (1)

References on critical path heuristics:

Patrik Haslum and Hector Geffner.
Admissible Heuristics for Optimal Planning.
Proc. AIPS 2000, pp. 140–149, 2000.
Introduces h^m heuristics.

Patrik Haslum.
$h^m(P) = h^1(P^m)$: Alternative Characterisations of the Generalisation From $h^{\max}$ to h^m.
Proc. ICAPS 2009, pp. 354–357, 2009.
Introduces Π^m compilation.

Literature (2)

Patrik Haslum.
Incremental Lower Bounds for Additive Cost Planning Problems.
Proc. ICAPS 2012, pp. 74–82, 2012.
Introduces Π^C compilation.

Emil Keyder, Jörg Hoffmann and Patrik Haslum.
Improving Delete Relaxation Heuristics Through Explicitly Represented Conjunctions.
JAIR 50, pp. 487–533, 2014.
Introduces Π^C_{ce} compilation.