Advanced Results on Distance Estimation in Planning

3. Admissible Landmark Heuristics

Cost Partitionings, LM-cut, Hitting Sets

Jörg Hoffmann and Michael Katz

SAARLAND UNIVERSITY
COMPUTER SCIENCE

Summer 2013

Lecture Overview

I assume you're familiar with basic concepts of abstractions (PDBs) and landmarks. The aim of this lecture is to cover recent results.

So what will I talk about?

- Additive admissible composition of heuristics [Katz and Domshlak (2008); Karpas and Domshlak (2009)].
- Generating landmarks and LM-cut procedure [Helmert and Domshlak (2009)].
- From landmarks via hitting sets to h^+. [Bonet and Helmert (2010)].

$*$ Slides for LM-cut and Hitting Sets are partially based on slides by Malte Helmert.

Combining Heuristics

$\rightarrow$ Every h yields good performance only in some domains.

The 4 different methods currently known:

- Critical path heuristics: $\rightarrow$ **See overview**
- Delete relaxation: $\rightarrow$ **Prev. session**
- Abstractions: $\rightarrow$ **See overview**
- Landmarks: $\rightarrow$ **See overview**

$\rightarrow$ Can we exploit their complementary strengths?

Can we exploit their complementary strenghts?

→ Say somebody gives you lower-bounds $h_1, \dots, h_n$. How can you always obtain a lower-bound h that dominates each of them?

Answer: By $h := \max_{i=1\dots n} h_i$.

→ Say somebody gives you lower-bounds $h_1, \dots, h_n$. What would be much better than taking their $\max$?

Answer: Taking their sum!

But how to ensure the sum is still a lower bound?

For Particular Heuristic Families

Reminder: What is a landmark heuristic?

A landmark heuristic h^{LM}_L is a cost of the cheapest action in L if L is a disjunctive action landmark for s, and 0 otherwise.
The disjunctive action landmarks L_i and L_j are independent if $L_i \cap L_j = \emptyset$. Then $h^{\mathsf{LM}}_{L_i} + h^{\mathsf{LM}}_{L_j}$ is admissible.

→ What about all the other possible h? And what about combinations across different methods? Is there something we can do *in general*?

→ The rest of this section points out that the answer is "Yes!!!"

Cost Partitionings

Definition (Cost Partitioning). *Let Π be a planning task with actions A and cost function c. An ensemble of functions $c_1, \dots, c_n : A \mapsto \mathbb{R}_0^+$ is a cost partitioning for Π if, for all $a \in A$, $\sum_{i=1}^n c_i(a) \leq c(a)$.*

$\rightarrow$ A cost partitioning distributes the cost of each action across several separate cost functions.

Terminology: (given a cost partitioning $c_1, \dots, c_n$)

- If h is a heuristic function for Π, then $h[c_i]$ denotes the same heuristic function but computed on the modification of Π where c is replaced by c_i.

  $\rightarrow$ We assume that $h[c_i]$ is *defined*, for any h. (Harmless assumption.)
- If $h_1, \dots, h_n$ is an ensemble of heuristic functions for Π, then the partitioned sum of $h_1, \dots, h_n$ given $c_1, \dots, c_n$ is $\sum_{i=1}^n h_i[c_i]$, for which we use the short-hand $\sum h[c]$.

Partitioned Sums are Admissible

Theorem (Partitioned Sums are Admissible). *Let Π be a planning task, and let $h_1, \dots, h_n$ be heuristic functions for Π. If $c_1, \dots, c_n$ is a cost partitioning for Π, and if $h_i[c_i]$ is consistent and goal-aware for all i, then the partitioned sum $\sum h[c]$ of $h_1, \dots, h_n$ given $c_1, \dots, c_n$ is consistent and goal-aware, and thus also admissible and safe.*

→ Typical case: $h_i[c_i]$ is consistent and goal-aware because h_i is.

Proof.

Goal-awareness: Say s is a goal state. For all i, since $h_i[c_i]$ is goal-aware, $h_i[c_i](s) = 0$. So $\sum h[c](s) = \sum_{i=1}^{n} h_i[c_i](s) = 0$ as desired.

Consistency: We need to show that whenever $(s, a, t) \in T$, $\sum h[c](s) \leq \sum h[c](t) + c(a)$. For all i, $h_i[c_i]$ is consistent. That is, $h_i[c_i](s) \leq h_i[c_i](t) + c_i(a)$ because the cost function underlying $h_i[c_i]$ is c_i (rather than c). But then, $\sum h[c](s) = \sum_{i=1}^{n} h_i[c_i](s) \leq \sum_{i=1}^{n}(h_i[c_i](t) + c_i(a)) = \sum_{i=1}^{n} h_i[c_i](t) + \sum_{i=1}^{n} c_i(a)$. Since $c_1, \dots, c_n$ is a cost partitioning, $\sum_{i=1}^{n} c_i(a) \leq c(a)$ from which the claim follows. □

So What?

→ We can admissibly combine arbitrary heuristic functions.

→ But for the particular methods we have, is that any better than existing additivity criteria?

Yes! (provided we manage to find the right cost partitionings)

For PDBs/landmarks, there always exists a cost partitioning that dominates the canonical heuristic, i.e., the maximum over the sum of additive subsets.

→ In both settings, there are cases where the domination is strict.

What is the Problem?

Given: A collection $h_1, \dots, h_n$ of admissible heuristics, and a state s.

Wanted: A cost partitioning $c_1, \dots, c_n$.

$\rightarrow$ How many candidate cost partitionings are there? Infinitely many.

$\rightarrow$ Do all of these yield a good overall lower bound on $h^*(s)$? No! E.g., say $h^{\mathsf{LM}}_{L_1}(s) = 0$ and $h^{\mathsf{LM}}_{L_2}(s) = 100$ in the original task, where L_1 and L_2 are not independent. We *could* choose c_1, c_2 so that, for all a, $c_1(a) = c(a)$ and $c_2(a) = 0$. This would yield $\sum h[c](s) = 0$.

$\rightarrow$ Many (most) cost partitionings are bad. Our challenge is to automatically find good ones.

$\rightarrow$ The challenge is particularly vexing because ideally we want to do this for every search state s! (In particular, if we wish to dominate the canonical heuristics.)

Uniform vs. Optimal Cost Partitionings

→ Uniform cost partitioning: distribute costs evenly.

→ Pros? Cons? Easy to compute, not typically high quality.

Definition (Optimal Cost Partitioning). *Let Π be a planning task, let $h_1, \dots, h_n$ be admissible heuristic functions for Π, and let s be a state. An optimal cost partitioning for s and $h_1, \dots, h_n$ is any cost partitioning $c_1, \dots, c_n$ for which $\sum h[c](s)$ is maximal.*

→ Optimal cost partitionings distribute costs in a way that yields the best possible lower bound, *for a given state*.

→ Pros? Cons? Perfect quality, but can we compute them efficiently? Somewhat surprisingly, in many cases the answer is "yes!"

Optimal Cost Partitionings for h^{LM}

Theorem (Polynomial-Time Optimal Cost Partitionings for h^{LM}). *Let Π be a planning task, let s be a state, and let $L_1, \dots, L_n$ be disjunctive action landmarks for s. Then an optimal cost partitioning for s and $h^{\mathsf{LM}}_{L_1}, \dots, h^{\mathsf{LM}}_{L_n}$ can be computed in time polynomial in $\|\Pi\|$ and n.*

Proof.

The problem of finding an optimal cost partitioning $c_1, \dots, c_n$ can be formulated as a Linear Programming (LP) problem. We use LP variables $c_{i,a}$ encoding the partitioned costs, and variables h_{L_i} encoding the weight the final heuristic will count for the landmark L_i. Simple constraints ensure that $c_{i,a}$ is indeed a cost partitioning, and that the weights h_{L_i} are not larger than allowed. Maximizing $\sum_{i=1}^{n} h_{L_i}$ results in an optimal cost partitioning.

→ Selection of the cheapest action from a landmark L_i can be encoded into LP, giving a weight to each L_i. An optimal cost partitioning corresponds to an LP solution maximizing the summed-up weights.

→ Note: We assume here that the L_i are LMs for s.

Optimal Cost Partitionings for h^{LM}: Proof

Proof.

The detailed LP formulation is as follows.

LP variables:

- For all i and $a \in L_i$: $c_{i,a}$ [value to be assigned to $c_i(a)$].
- For all i: h_{L_i} [weight to be counted for LM L_i].

Maximize: $\sum_{i=1}^{n} h_{L_i}$.

LP constraints:

(i) For all $a \in A$: $\sum_{i:a \in L_i} c_{i,a} \leq c(a)$. [Ensures that the solution can be extended to a cost partitioning $c_1, \dots, c_k$.]

(ii) For all L_i and $a \in L_i$: $h_{L_i} \leq c_{i,a}$. [Ensures that the weight counted for each LM is at most the cost of its cheapest action.]

Due to soundness of (i) and (ii) as explained, we have $\sum_{i=1}^{n} h_{L_i} \leq \sum h[c](s)$. Further, $\sum_{i=1}^{n} h_{L_i} \geq \sum h[c](s)$ because (a) *any reasonable* cost partitioning must satisfy (i), and (b) for any assignment to variables $c_{i,a}$, maximizing over $\sum_{i=1}^{n} h_{L_i}$ together with (ii) results in $h_{L_i} = \min\{c_{i,a} \mid a \in L_i\}$. □

Shorter LP formulation for h^{LM}

$\rightarrow$ For all actions $a \in L_i$, if we demand $h_{L_i} \leq c_{i,a}$, there is no added value in giving to $c_{i,a}$ values strictly higher than h_{L_i}.

$\rightarrow$ We can drop *all* variables $c_{i,a}$ and use only h_{L_i}.

LP variables:

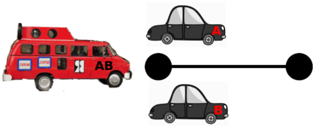
- For all i: h_{L_i} [weight to be counted for LM L_i].

Maximize: $\sum_{i=1}^{n} h_{L_i}$.

LP constraints:

- For all $a \in A$: $\sum_{L_i : a \in L_i} h_{L_i} \leq c(a)$. [Ensures that the solution can be extended to a cost partitioning $c_1, \dots, c_k$ by setting $c_i(a) = h_{L_i}$ if $a \in L_i$ and $c_i(a) = 0$ otherwise.]

Optimal Cost Partitioning LP for h^{LM}: "Films" Example



- V: $carA$, $carB$, $fancy$: $\{L, R\}$; $gotA$, $gotB$: $\{F, T\}$.
- A: $drive(x, L, R)$; $film(x, A)$ where $x \in \{carA, fancy\}$; $film(x, B)$ where $x \in \{carB, fancy\}$.
- c: 1.5 for $drive$ of $x \in \{carA, carB\}$; 2 for $drive$ of $x = fancy$. 0 for all $film$ actions.
- I: $carA = carB = fancy = L$, $gotA = gotB = F$.
- G: $gotA = gotB = T$.

Abbr.: $drA = drive(carA,L,R)$, $drB = drive(carB,L,R)$, $drF = drive(fancy,L,R)$.

Given: $L_A = \{drA, drF\}$ and $L_B = \{drB, drF\}$.

LP variables: h_{L_A}, h_{L_B}.

LP constraints:

$$\begin{array}{lcccl} drA: & h_{L_A} & & & \leq 1.5 \\ drB: & & & h_{L_B} & \leq 1.5 \\ drF: & h_{L_A} & + & h_{L_B} & \leq 2 \end{array}$$

Solution maximizing $h_{L_A} + h_{L_B}$: For example, $h_{L_A} = h_{L_B} = 1$, or $h_{L_A} = 0.5$, $h_{L_B} = 1.5$. In general, any assignment where $0.5 \leq h_{L_A} \leq 1.5$ and $h_{L_B} = 2 - h_{L_A}$.

Optimal Cost Partitionings for method combinations

→ **What about PDBs?** Can we do the same with explicit abstractions? Yes!

→ Cheapest paths in abstract transition systems can be encoded into LP, giving a weight to each abstraction. An optimal cost partitioning corresponds to an LP solution maximizing the summed-up weights.

→ **What about combining landmarks and PDBs?**

Simply put all the LP variables and constraints described previously into a single formulation. Assume for the purpose of notation that $h_1, \dots, h_j$ are LM heuristics, and $h_{j+1}, \dots, h_n$ are abstraction heuristics. Then maximizing $\sum_{i=1}^{j} h_{L_i} + \sum_{i=j+1}^{n} h_{\alpha_i}$ results in an optimal cost partitioning.

(Optimal) Cost Partitionings in Practice

Sometimes, optimal just isn't good enough: LPs can be solved in polynomial time, however may not be fast enough especially if we do it for every state during a search.

Some possible fixes:

- Ditch all this and use our previous additivity criteria (not the worst option, really, at least at the moment). [Haslum *et al.* (2007)]
- Use uniform cost partitionings (not that bad either). [Katz and Domshlak (2010a); Karpas and Domshlak (2009)]
- Just live with it and solve an LP in every search state (useful if you got lots of time and *really* want to solve that challenging problem). [Katz and Domshlak (2010b)]
- Solve an LP just once at the start, getting an optimal cost partitioning for the initial state. Hope that this partitioning is good for other states as well. [Katz and Domshlak (2010b)]
- Solve an LP for some sample states, use a combination/selection of the resulting cost partitionings for each search state. [Karpas *et al.* (2011)]

Relaxed planning tasks

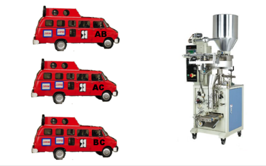
Definition (relaxed planning task)

F: finite set of facts

- initial facts $I \subseteq F$ are given
- goal facts $G \subseteq F$ must be reached
- operators of the form $o[4] : a, b \to c, d$

read: If we already have facts a and b (preconditions $pre(o)$), we can apply o, paying 4 units (cost $c(o)$), to obtain facts c and d (effects $\mathit{eff}(o)$)

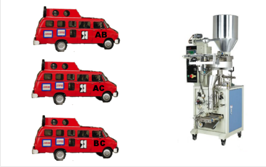
For simplicity: assume $I = \{i\}$, $G = \{g\}$, all $pre(o) \neq \emptyset$



- $F = \{i, A, B, C, g\}$, $I = \{i\}$, $G = \{g\}$.
- A:

$$\begin{aligned} \mathit{fillAB}[3] &: i \to A, B \\ \mathit{fillAC}[4] &: i \to A, C \\ \mathit{fillBC}[5] &: i \to B, C \\ \mathit{devAll}[0] &: A, B, C \to g \end{aligned}$$

Landmarks: Example



- $F = \{i, A, B, C, g\}$, $I = \{i\}$, $G = \{g\}$.
- A: $\begin{array}{ll} \mathit{fillAB}[3] & : i \rightarrow A, B \\ \mathit{fillAC}[4] & : i \rightarrow A, C \\ \mathit{fillBC}[5] & : i \rightarrow B, C \\ \mathit{devAll}[0] & : A, B, C \rightarrow g \end{array}$

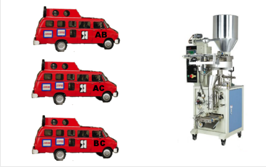
Some landmarks:

- $W = \{\mathit{devAll}\}$, cost: 0
- $X = \{\mathit{fillAB}, \mathit{fillAC}\}$, cost 3
- $Y = \{\mathit{fillAB}, \mathit{fillBC}\}$, cost 3
- $Z = \{\mathit{fillAC}, \mathit{fillBC}\}$, cost 4

But also:

- $\{\mathit{fillAB}, \mathit{fillAC}, \mathit{fillBC}\}$ with cost 3
- $\{\mathit{fillAB}, \mathit{fillBC}, \mathit{devAll}\}$ with cost 0
- $\ldots$

Landmarks: Example



- $F = \{i, A, B, C, g\}$, $I = \{i\}$, $G = \{g\}$.
- A: $$\begin{array}{ll} \mathit{fillAB}[3] & : i \rightarrow A, B \\ \mathit{fillAC}[4] & : i \rightarrow A, C \\ \mathit{fillBC}[5] & : i \rightarrow B, C \\ \mathit{devAll}[0] & : A, B, C \rightarrow g \end{array}$$

$W = \{\mathit{devAll}\}$, $X = \{\mathit{fillAB}, \mathit{fillAC}\}$, $Y = \{\mathit{fillAB}, \mathit{fillBC}\}$, $Z = \{\mathit{fillAC}, \mathit{fillBC}\}$

Given landmark set $\mathcal{L} = \{W, X, Y, Z\}$, how to exploit $\mathcal{L}$? h^{LM}!

LP: maximize $w + x + y + z$ subject to $w, x, y, z \geq 0$ and

$$\begin{array}{ccccccccl} w & & & & & & \leq & 0 & \mathit{devAll} \\ & x & + & y & & & \leq & 3 & \mathit{fillAB} \\ & x & + & & & z & \leq & 4 & \mathit{fillAC} \\ & & & y & + & z & \leq & 5 & \mathit{fillBC} \\ W & X & & Y & & Z & & & \end{array}$$

solution: $w = 0$, $x = 1$, $y = 2$, $z = 3$ $\rightsquigarrow$ $h^{\mathsf{LM}}(I) = 6$

But $h^+(I) = 7$! Can we do better? Yes!

Hitting sets

Definition (hitting set)

Given: finite set A, subset family $\mathcal{F} \subseteq 2^A$, costs $c : A \rightarrow \mathbb{R}_0^+$

Hitting set:

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

Minimum hitting set (MHS):

- minimizes cost
- classical NP-complete problem (Karp, 1972)

Example

$A = \{devAll, fillAB, fillAC, fillBC\}$ $\quad \mathcal{F} = \{W, X, Y, Z\}$ with
$W = \{devAll\}, \quad X = \{fillAB, fillAC\}, \quad Y = \{fillAB, fillBC\}, \quad Z = \{fillAC, fillBC\}$

$c(fillAB) = 3, \; c(fillAC) = 4, \; c(fillBC) = 5, \; c(devAll) = 0$

Minimum hitting set: $\{fillAB, fillAC, devAll\}$ with cost $3 + 4 + 0 = 7$

Hitting set heuristics for landmarks

- can view landmark sets (with operator costs) as instances of minimum hitting set problem
- here, we got an admissible estimate that dominated h^{LM}
- coincidence? No!

Let $\mathcal{L}$ be a set of landmarks.

Theorem (hitting set heuristics for landmarks)

Let $h^{\mathit{MHS}}(I)$ be the minimum hitting set cost for $\langle A, \mathcal{L}, c\rangle$. Then:

1. $h^{\mathit{MHS}}(I) \leq h^{+}(I)$ *(hitting set heuristics are admissible)*
2. $h^{\mathit{MHS}}(I) \geq h^{\mathsf{LM}}(I)$ *(hitting sets dominate cost partitioning)*

Hitting set heuristics for LMs are admissible

Proof.

Let π^+ be an optimal relaxed plan for I. Then $\sum_{a \in \pi^+} c(a) = h^+(I)$.

For each landmark $l \in \mathcal{L}$, from the definition of landmarks, there exists an action $a \in \pi^+$ such that $a \in l$.

Thus, the set of all actions in π^+ is a hitting set for $\mathcal{L}$,

and the minimum hitting set cost $h^{\mathsf{MHS}}(I)$ satisfies

$$h^{\mathsf{MHS}}(I) \leq \sum_{a \in \pi^+} c(a) = h^+(I).$$

Hitting set heuristics for LMs dominate cost partitioning

Proof.

ILP formulation for the minimum hitting set problem:

- **Variables:** For each $a \in A$: a binary variable c_a [whether a is in the hitting set]
- **Constraints:** For each landmark $l \in \mathcal{L}$: $\sum_{a \in l} c_a \geq 1$ [the landmark is hit by the set]
- **Objective:** $\min \sum_{a \in A} c_a$

Observations:

(I) An optimal solution to the LP relaxation provides a value smaller or equal to the optimal solution of the ILP.

(II) The strong duality theorem states that if the primal has an optimal solution, then its dual also has an optimal solution, of the same value.

The dual to the LP relaxation is:

- **Variables:** For each landmark $l \in \mathcal{L}$: c_l
- **Constraints:** For each $a \in A$: $\sum_{l \in \mathcal{L}: a \in l} c_l \leq 1$
- **Objective:** $\max \sum_{l \in \mathcal{L}} c_l$

Which is exactly the cost partitioning LP for $\mathcal{L}$.

Generating landmarks: justification graphs

Definition (precondition choice function)

A precondition choice function (pcf) $D : A \to F$ maps each operator to one of its preconditions.

Definition (justification graph)

The justification graph $G(D)$ for pcf D is an arc-labeled digraph with

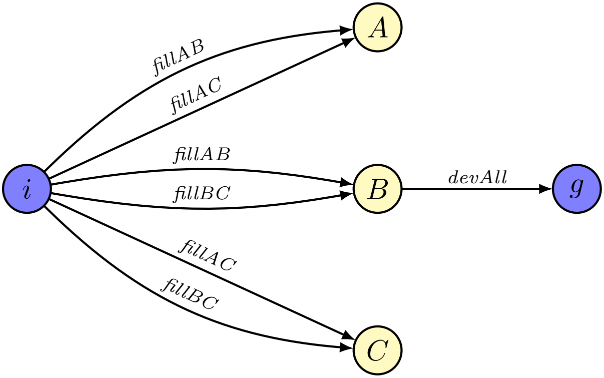
- vertices: the facts F
- arcs: arc $D(a) \xrightarrow{a} e$ for each operator a and effect $e \in \mathit{eff}(a)$

Example: justification graph

Example

pcf D: $D(\mathit{fillAB}) = D(\mathit{fillAC}) = D(\mathit{fillBC}) = i,\ D(\mathit{devAll}) = B$

$\mathit{fillAB}[3] : i \rightarrow A, B$
$\mathit{fillAC}[4] : i \rightarrow A, C$
$\mathit{fillBC}[5] : i \rightarrow B, C$
$\mathit{devAll}[0] : A, B, C \rightarrow g$



Cuts

Definition (cut)

A cut of a justification graph is a subset of its arcs C such that all paths from i to g use some arc in C.

Theorem (cuts are landmarks)

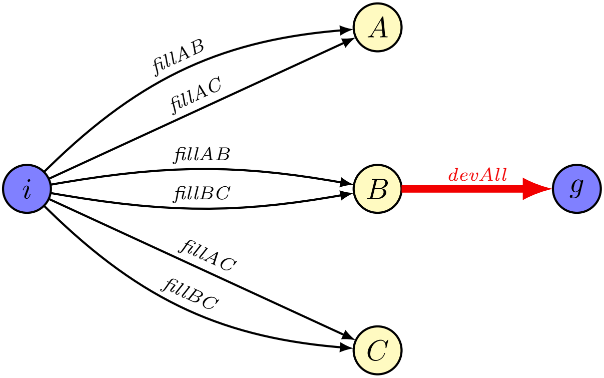
Let C be any cut of the justification graph for any pcf. Then the labels of C form a landmark.

Example: cuts of a justification graph

Example

Landmark $W = \{devAll\}$ (cost 0)

$fillAB[3] : i \rightarrow A, B$
$fillAC[4] : i \rightarrow A, C$
$fillBC[5] : i \rightarrow B, C$
$devAll[0] : A, B, C \rightarrow g$

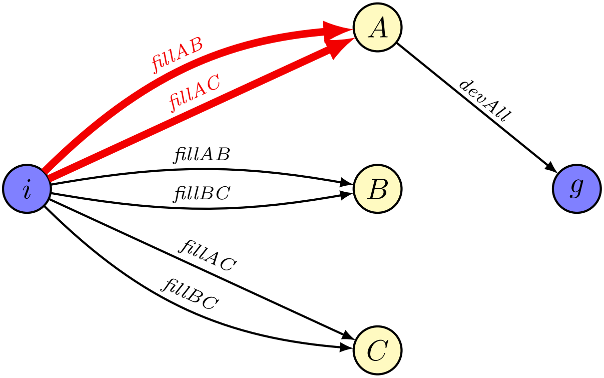


Example: cuts of a justification graph

Example

Landmark $X = \{\mathit{fillAB}, \mathit{fillAC}\}$ (cost 3)

$\mathit{fillAB}[3] : i \rightarrow A, B$
$\mathit{fillAC}[4] : i \rightarrow A, C$
$\mathit{fillBC}[5] : i \rightarrow B, C$
$\mathit{devAll}[0] : A, B, C \rightarrow g$

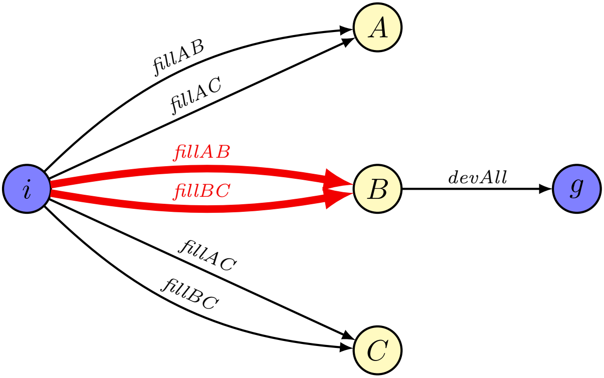


Example: cuts of a justification graph

Example

Landmark $Y = \{\mathit{fillAB}, \mathit{fillBC}\}$ (cost 3)

$\mathit{fillAB}[3] : i \to A, B$
$\mathit{fillAC}[4] : i \to A, C$
$\mathit{fillBC}[5] : i \to B, C$
$\mathit{devAll}[0] : A, B, C \to g$

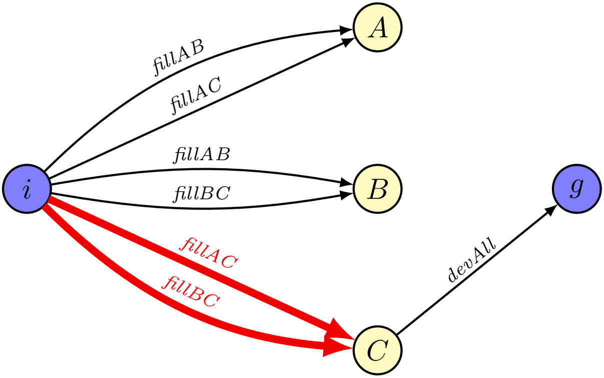


Example: cuts of a justification graph

Example

Landmark $Z = \{fillAC, fillBC\}$ (cost 4)

$fillAB[3] : i \rightarrow A, B$
$fillAC[4] : i \rightarrow A, C$
$fillBC[5] : i \rightarrow B, C$
$devAll[0] : A, B, C \rightarrow g$



LM-cut heuristic

$h^{\text{LM-cut}}$: Helmert & Domshlak (2009)

Initialize $h^{\text{LM-cut}}(I) := 0$. Then loop:

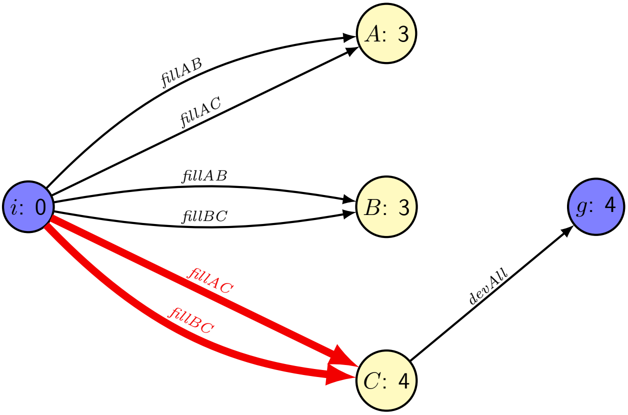
1. Compute $h^{\max}$ costs of facts. Stop if $h^{\max}(g) = 0$.
2. Let D be a pcf that picks preconditions maximizing $h^{\max}$.
3. Compute the justification graph for D.
4. Compute a cut using a particular procedure that guarantees that $c(L) > 0$ for the induced landmark L.
5. Increase $h^{\text{LM-cut}}(I)$ by $c(L)$.
6. Decrease $c(o)$ by $c(L)$ for all $o \in L$.

Example: LM-cut computation

Example

round 1: $D(g) = C \rightsquigarrow L = \{\mathit{fillAC}, \mathit{fillBC}\}$ [4]

$\mathit{fillAB}[3] : i \to A, B$
$\mathit{fillAC}[4] : i \to A, C$
$\mathit{fillBC}[5] : i \to B, C$
$\mathit{devAll}[0] : A, B, C \to g$

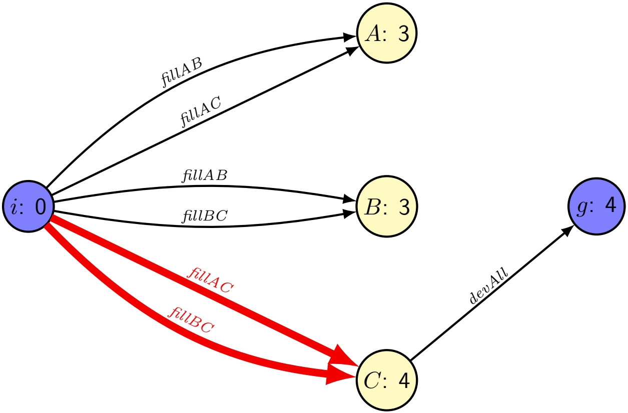


Example: LM-cut computation

Example

round 1: $D(g) = C \rightsquigarrow L = \{\mathit{fillAC}, \mathit{fillBC}\}$ [4] $\rightsquigarrow h^{\text{LM-cut}}(I) := 4$

$\mathit{fillAB}[3] : i \rightarrow A, B$
$\mathit{fillAC}[0] : i \rightarrow A, C$
$\mathit{fillBC}[1] : i \rightarrow B, C$
$\mathit{devAll}[0] : A, B, C \rightarrow g$



Example: LM-cut computation

Example

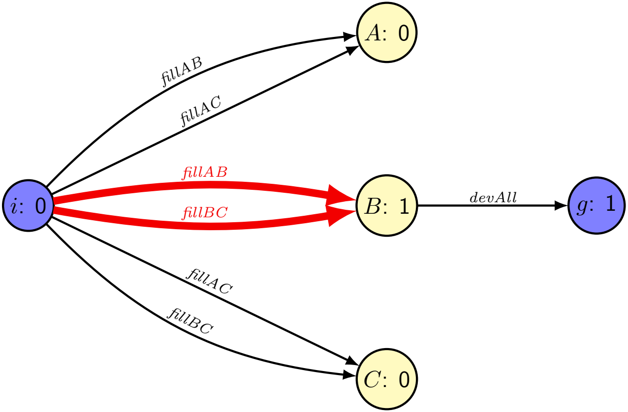
round 2: $D(g) = B \rightsquigarrow L = \{\mathit{fillAB}, \mathit{fillBC}\}$ [1]

$\mathit{fillAB}[3] : i \to A, B$

$\mathit{fillAC}[0] : i \to A, C$

$\mathit{fillBC}[1] : i \to B, C$

$\mathit{devAll}[0] : A, B, C \to g$

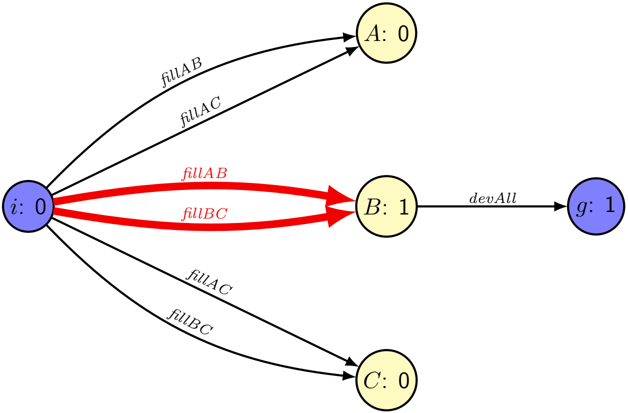


Example: LM-cut computation

Example

round 2: $D(g) = B \rightsquigarrow L = \{\mathit{fillAB}, \mathit{fillBC}\}$ [1] $\rightsquigarrow h^{\text{LM-cut}}(I) := 4 + 1 = 5$

$\mathit{fillAB}[2] : i \rightarrow A, B$
$\mathit{fillAC}[0] : i \rightarrow A, C$
$\mathit{fillBC}[0] : i \rightarrow B, C$
$\mathit{devAll}[0] : A, B, C \rightarrow g$

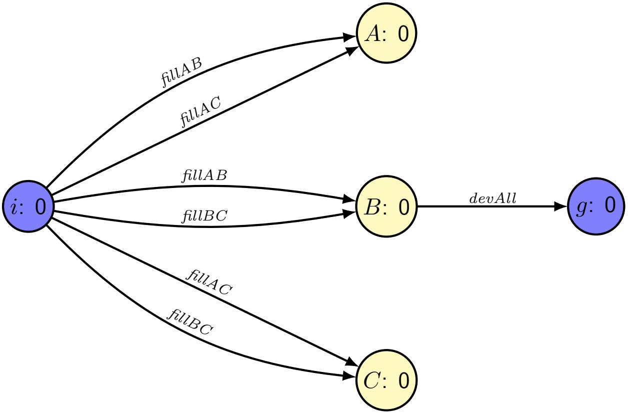


Example: LM-cut computation

Example

round 3: $h^{\max}(g) = 0 \rightsquigarrow$ done! $\rightsquigarrow h^{\text{LM-cut}}(I) = 5$

$fillAB[2] : i \rightarrow A, B$
$fillAC[0] : i \rightarrow A, C$
$fillBC[0] : i \rightarrow B, C$
$devAll[0] : A, B, C \rightarrow g$



LM-cut heuristic dominating $h^{\max}$

Theorem (LM-cut dominates $h^{\max}$)

Let $h^{\text{LM-cut}}(I)$ be computed as above. Then $h^{\text{LM-cut}}(I) \geq h^{\max}(g)$.

Proof.

If $h^{\max}(g) = 0$, there is nothing to prove. Otherwise, steps (2)–(6) allow defining a cost partitioning $c = c_1 + c_2$ with two heuristics h_1 and h_2 such that

- h_1 is an elementary landmark heuristic with cost function c_1,
- h_2 is the $h^{\max}$ heuristic with cost function c_2, and
- $h^{\max}(g) \leq h_1(I) + h_2(g)$.

→ c_1 assigns $c(L)$ to all operators in L and 0 to all other operators; $c_2 = c - c_1$.

LM-cut heuristic dominating $h^{\max}$ (ctd.)

Proof (ctd.)

To see that $h^{\max}(g) \leq h_1(I) + h_2(g)$, note that

(a) $h^{\max}(g) =$ cost of the cheapest path from i to g in $G(D)$, and

(b) the cut L can be chosen s.t. each i-g-path has exactly one edge in L.

$\rightarrow$ The cost of the cheapest path cannot be reduced by more than $c(L)$.

The reduction is then applied recursively to h_2 if $h_2(g) > 0$.

The set of zero cost operators is strictly larger for c_2 than for c, so the process terminates in polynomially many (at most $|O|$) steps. □

Perfect Hitting Set heuristics

- Which landmarks can be generated with the cut method?
- All interesting ones!

Theorem (perfect hitting set heuristics)

Let $\mathcal{L}$ be the set of all "cut landmarks". Then $h^{\mathsf{MHS}}(I) = h^+(I)$.

$\rightsquigarrow$ hitting set heuristic over $\mathcal{L}$ is perfect

Proof.

Let H be some hitting set for $\mathcal{L}$. If we show that H can be turned into a (relaxed) plan, we are done, since (a) we already know that $h^{\mathsf{MHS}}(I) \leq h^+(I)$, and (b) in particular, a minimal hitting set can be turned into a plan.

Perfect Hitting Set heuristics (ctd.)

Proof (ctd.)

Assume that H cannot be turned into a plan. Let R be the set of all facts that can be reached by only using operators in H. Then $g \notin R$.

We construct a pcf D such that $G(D)$ contains a cut not hit by H, thus reaching a contradiction.

We classify operators into two types and define D:

(T1) $pre(a) \subseteq R$, then set $D(a)$ arbitrarily to some $p \in pre(a)$.
(T2) $pre(a) \not\subseteq R$, then set $D(a)$ to some $p \in pre(a) \setminus R$.

Now consider the cut of $G(D)$, from R to all facts not in R. It is a cut since $i \in R$ and $g \notin R$.

Assume now that there exist an operator $a \in H$ that labels an edge of $G(D)$ going from some fact in R to some fact not in R. What is its type?
Not T1, since $pre(a) \subseteq R$ and $a \in H$ imply $\mathit{eff}(a) \subseteq R$.
Not T2, since edges labeled by type T2 operators do not start in R.

Hence, no such operator exists and the cut is not hit by H! □

Conclusion

Summary:

- Hitting sets for landmarks are more informative than optimal cost partitioning.
- Cuts in justification graphs offer a principled way of generating landmarks.
- Hitting sets over all cut landmarks are perfect heuristics for delete relaxations.
- These concepts can be exploited in practical heuristics.

Summary

- Action cost partitioning has proved very powerful tool for additively combining different heuristic functions.
- Different heuristic families can be additively combined.
- Informative admissible estimates h^{LM} and $h^{\mathsf{LM\text{-}cut}}$ are due to action cost partitioning.
- Hitting sets can be used to obtain informative estimate from a given set of landmarks that dominate h^{LM}.
- MHS is NP-complete, but a width parameter k can be introduced such that MHS is fixed-parameter tractable w.r.t. k.
- $h^{\mathsf{LM\text{-}cut}}$ is an effective procedure that generates landmarks.
- It can be significantly improved by performing it several times, with different tie-breakings for pcf.
- Once a collection of landmarks is found (w.r.t. the width k), MHS can be used to obtain informative estimates in polynomial time.

Reading

- *Optimal Additive Composition of Abstraction-Based Admissible Heuristics* [Katz and Domshlak (2008)].

  Available at:

  `http://fai.cs.uni-saarland.de/katz/papers/icaps08b.pdf`

  Content: Original paper proposing cost partitionings, and showing that, for certain classes of heuristics, optimal cost partitionings can be computed in polynomial time using Linear Programming. Specifically, the paper established this for abstractions as handled here, and for implicit abstractions represented through planning task fragments identified based on the causal graph.

  Journal version: Optimal Admissible Composition of Abstraction Heuristics [Katz and Domshlak (2010b)], for the most persistent amongst you.

Reading, ctd.

- *Cost-Optimal Planning with Landmarks* [Karpas and Domshlak (2009)].

  Available at:

  http://iew3.technion.ac.il/~dcarmel/Papers/Sources/ijcai09a.pdf

  Content: The "alarm clock" waking LMs up to the modern age of cost-optimal planning. Introduces cost partitionings for elementary landmarks heuristics, and the computation of optimal cost partitionings for such heuristics using Linear Programming. Introduces uniform cost partitionings, which are used in the experiments due to being more runtime-effective.

Reading, ctd.

- *Landmarks, Critical Paths and Abstractions: What's the Difference Anyway?* [Helmert and Domshlak (2009)].

  Available at:

  `http://ai.cs.unibas.ch/papers/helmert-domshlak-icaps2009.pdf`

  Content: The paper investigates the dominance relations between the four heuristic families known to us today. As a result of such an investigation, the LM-cut procedure for obtaining a set of landmarks is presented. The procedure can alternatively be viewed as a landmark heuristic, a cost partitioning scheme for additive $h^{\max}$, or an approximation to the (intractable) optimal relaxation heuristic h^{+}.

Reading, ctd.

- *Strengthening Landmark Heuristics via Hitting Sets* [Bonet and Helmert (2010)].

  Available at:

  http://ai.cs.unibas.ch/papers/bonet-helmert-ecai2010.pdf

  Content: The paper shows that $h^{\textsf{LM-cut}}$ can be understood as a simple relaxation of a hitting set problem, and that stronger heuristics can be obtained by considering stronger relaxations.

References I

Blai Bonet and Malte Helmert. Strengthening landmark heuristics via hitting sets. In Helder Coelho, Rudi Studer, and Michael Wooldridge, editors, *Proceedings of the 19th European Conference on Artificial Intelligence (ECAI'10)*, pages 329–334, Lisbon, Portugal, August 2010. IOS Press.

Patrik Haslum, Adi Botea, Malte Helmert, Blai Bonet, and Sven Koenig. Domain-independent construction of pattern database heuristics for cost-optimal planning. In Adele Howe and Robert C. Holte, editors, *Proceedings of the 22nd National Conference of the American Association for Artificial Intelligence (AAAI-07)*, pages 1007–1012, Vancouver, BC, Canada, July 2007. AAAI Press.

Malte Helmert and Carmel Domshlak. Landmarks, critical paths and abstractions: What's the difference anyway? In Alfonso Gerevini, Adele Howe, Amedeo Cesta, and Ioannis Refanidis, editors, *Proceedings of the 19th International Conference on Automated Planning and Scheduling (ICAPS 2009)*, pages 162–169. AAAI Press, 2009.

References II

Erez Karpas and Carmel Domshlak. Cost-optimal planning with landmarks. In C. Boutilier, editor, *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI 2009)*, pages 1728–1733, Pasadena, California, USA, July 2009. Morgan Kaufmann.

Erez Karpas, Michael Katz, and Shaul Markovitch. When optimal is just not good enough: Learning fast informative action cost-partitionings. In Fahiem Bacchus, Carmel Domshlak, Stefan Edelkamp, and Malte Helmert, editors, *Proceedings of the 21st International Conference on Automated Planning and Scheduling (ICAPS 2011)*, pages 122–129. AAAI Press, 2011.

Michael Katz and Carmel Domshlak. Optimal additive composition of abstraction-based admissible heuristics. In Jussi Rintanen, Bernhard Nebel, J. Christopher Beck, and Eric Hansen, editors, *Proceedings of the 18th International Conference on Automated Planning and Scheduling (ICAPS 2008)*, pages 174–181. AAAI Press, 2008.

Michael Katz and Carmel Domshlak. Implicit abstraction heuristics. *Journal of Artificial Intelligence Research*, 39:51–126, 2010.

Michael Katz and Carmel Domshlak. Optimal admissible composition of abstraction heuristics. 174(12–13):767–798, 2010.