Landmarks, the Universe, and Everything

Julie Porteous    Laura Sebastia    Jörg Hoffmann

Teesside University, UK
Universidad Politécnica de Valencia, Spain
Saarland University, Germany

June 13, 2013
Imagine there's no Landmarks
It's easy if you try
No benchmarks below us
Above us only Blai
Imagine all the planners
Planning for real
Imagine there’s no Landmarks
   It’s easy if you try
No benchmarks below us
   Above us only Blai
Imagine all the planners
   Planning for real
Imagine there’s no Landmarks
   It’s easy if you try
No benchmarks below us
   Above us only Blai
Imagine all the planners
   Planning for real
Song # 2

Planning, planning, planning,
P-D-D-L scanning,
Keep 'em planners planning, ICAPS!
Uncertain durations,
Truth ramifications,
Wishing FF was by my side!
My soft goals they are kissin'
My landmarks have gone missin'
My stubborn set has turned off the light.
Planning, planning, planning,
Song # 2

Planning, planning, planning,
P-D-D-L scanning,
Planning, planning, planning,
P-D-D-L scanning,
Keep ’em planners planning, ICAPS!
Planning, planning, planning,
P-D-D-L scanning,
Keep 'em planners planning, ICAPS!

Uncertain durations,
Planning, planning, planning,
P-D-D-L scanning,
Keep ’em planners planning, ICAPS!

Uncertain durations,
Truth ramifications,
Song # 2

Planning, planning, planning,
P-D-D-L scanning,
Keep ’em planners planning, ICAPS!

Uncertain durations,
Truth ramifications,
Wishing FF was by my side!
Planning, planning, planning,
  P-D-D-L scanning,
Keep 'em planners planning, ICAPS!

Uncertain durations,
  Truth ramifications,
Wishing FF was by my side!

My soft goals they are kissin’
Planning, planning, planning,
   P-D-D-L scanning,
Keep ’em planners planning, ICAPS!

   Uncertain durations,
   Truth ramifications,
   Wishing FF was by my side!

   My soft goals they are kissin’
   My landmarks have gone missin’
Planning, planning, planning,
    P-D-D-L scanning,
Keep ’em planners planning, ICAPS!

    Uncertain durations,
    Truth ramifications,
Wishing FF was by my side!

    My soft goals they are kissin’
    My landmarks have gone missin’
My stubborn set has turned off the light.
Planning, planning, planning,
P-D-D-L scanning,
Keep 'em planners planning, ICAPS!

Uncertain durations,
Truth ramifications,
Wishing FF was by my side!

My soft goals they are kissin’
My landmarks have gone missin’
My stubborn set has turned off the light.
Agenda
Agenda: Stage 0 (The Dark Ages)
Once Upon a Time, There Was a Landmark . . .

Verbatim from [Porteous et al. (2001)]:

initial state

\[
\begin{array}{ccc}
A & B & C \\
\end{array}
\]

D

goal

\[
\begin{array}{ccc}
C & B \\
A & D \\
\end{array}
\]
What Are Landmarks?

Problem: Bring key B to position 1.

Landmarks:
- robot-at-2
- robot-at-3
- robot-at-4
- robot-at-5
- robot-at-6
- robot-at-7
- Lock-open
- Have-key-A
- Have-key-B

A landmark is a fact that is true at some point on every solution plan. Find landmarks in a pre-process to planning. Can also find landmark orderings $L \leq L'$.
What Are Landmarks?

### Problem
Bring key B to position 1.

#### Landmarks:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="robot-at-2" alt="Robot" /></td>
<td><img src="A" alt="Key A" /></td>
<td></td>
<td><img src="A" alt="Locked Door" /></td>
<td></td>
<td></td>
<td><img src="B" alt="Key B" /></td>
</tr>
</tbody>
</table>

A landmark is a fact that is true at some point on every solution plan. Find landmarks in a pre-process to planning. Can also find landmark orderings $L \leq L'$. 

Julie Porteous Laura Sebastia Jörg Hoffmann

Landmarks, the Universe, and Everything

3/13
What Are Landmarks?

Problem: Bring key B to position 1.

Landmarks:
What Are Landmarks?

Problem: Bring key B to position 1.

Landmarks:
- Lock-open,
What Are Landmarks?

Problem: Bring key B to position 1.

Landmarks:
- Lock-open, Have-key-A,
Problem: Bring key B to position 1.

Landmarks:
- Lock-open, Have-key-A, Have-key-B,
What Are Landmarks?

Problem: Bring key B to position 1.

Landmarks:
- Lock-open, Have-key-A, Have-key-B, ...

→ A landmark is a fact that is true at some point on every solution plan.

- Find landmarks in a pre-process to planning.
- Can also find landmark orderings $L \leq L'$. 
And Now?

Well, some guy (me, that is) proposed to use this for problem decomposition, but never mind that.

ps. Actually, see [Vernhes et al. (2013)] for an interesting modernized version!
And Now?

Well, some guy (me, that is) proposed to use this for problem decomposition, but never mind that.
And Now?

Well, some guy (me, that is) proposed to use this for problem decomposition, but never mind that.

ps. Actually, see [Vernhes et al. (2013)] for an interesting modernized version!
Agenda: Stage 1 (Preparing for Take-Off)
Agenda: Stage 1 (Preparing for Take-Off)
How To Use Landmarks!

Problem: Bring key B to position 1.

Landmarks set \( \{LM\} \):
- Lock-open, Have-key-A, Have-key-B, ...

\[ h(s) := |\{LM\} \setminus s| . \text{ ("Number of open items on the to-do list")} \]
How To Use Landmarks!

Problem: Bring key B to position 1.

Landmarks set \( \{LM\} \):
- Lock-open, Have-key-A, Have-key-B, ...

\[
\rightarrow h(s) := |\{LM\} \setminus s|. \quad \text{("Number of open items on the to-do list")}
\]

- We can analyze orders and interferences to “put an item back on”. 

Credits to [Zhu and Givan (2003)] for their “forgotten work”...
How To Use Landmarks!

Problem: Bring key B to position 1.

Landmarks set \( \{LM\} \):
- Lock-open, Have-key-A, Have-key-B, …

\[ h(s) := |\{LM\} \setminus s|. \text{ (”Number of open items on the to-do list”)} \]

- We can analyze orders and interferences to “put an item back on”.
- LAMA combines this with relaxed plans, iterated WA*, … [Richter et al. (2008); Richter and Westphal (2010)]
How To Use Landmarks!

Problem: Bring key B to position 1.

Landmarks set \( \{LM\} \):
- Lock-open, Have-key-A, Have-key-B, ...

\[ h(s) := |\{LM\} \setminus s|. \text{ ("Number of open items on the to-do list")} \]

- We can analyze orders and interferences to “put an item back on”.
- LAMA combines this with relaxed plans, iterated WA*, ... [Richter et al. (2008); Richter and Westphal (2010)]
- Credits to [Zhu and Givan (2003)] for their “forgotten work” ...!
The Impact of Stage 1
The Impact of Stage 1

LAMA

R.I.P. FF
Agenda: Stage 2 (Leaving the Atmosphere)
Agenda: Stage 2 (Leaving the Atmosphere)
How To Admissibly Combine Landmarks!

Planning task:
Goals $G = \{A, B\}$, initial state $I = \emptyset$, actions $\text{carA}: \emptyset \rightarrow A$ cost 1, $\text{carB}: \emptyset \rightarrow B$ cost 1, $\text{fancyCar}: \emptyset \rightarrow A \land B$ cost 1.5.

Landmarks set $\{LM\}$: $\{A, B\}$. Thus $h(I) = 2 > h^*(I)$.

Solution: [Karpas and Domshlak (2009)]

Consider disjunctive action landmarks instead: $L_A = \{\text{carA, fancyCar}\}$, $L_B = \{\text{carB, fancyCar}\}$. (= Achievers of each landmark) → Elementary landmark heuristic $h_{LM_L}(s) = \min\{c(a) | a \in L\}$ if $L$ is a disjunctive action landmark for $s$, and $h_{LM_L}(s) = 0$ otherwise.

Partition action costs to make $\sum L h_{LM_L}(s)$ admissible!
Planning task: Goals $G = \{A, B\}$, initial state $I = \emptyset$, actions

$\textit{carA} : \emptyset \rightarrow A \text{ cost } 1$, $\textit{carB} : \emptyset \rightarrow B \text{ cost } 1$, $\textit{fancyCar} : \emptyset \rightarrow A \land B \text{ cost } 1.5$. 
How To Admissibly Combine Landmarks!

Planning task: Goals $G = \{A, B\}$, initial state $I = \emptyset$, actions

$\text{carA} : \emptyset \rightarrow A$ cost 1, $\text{carB} : \emptyset \rightarrow B$ cost 1, $\text{fancyCar} : \emptyset \rightarrow A \land B$ cost 1.5.

Landmarks set $\{LM\}$:
Planning task: Goals \( G = \{A, B\} \), initial state \( I = \emptyset \), actions

\[ \text{car}A : \emptyset \rightarrow A \text{ cost 1}, \text{car}B : \emptyset \rightarrow B \text{ cost 1}, \text{fancyCar} : \emptyset \rightarrow A \land B \text{ cost 1.5}. \]

Landmarks set \( \{LM\} : \{A, B\} \). Thus \( h(I) = \)
**Planning task:** Goals $G = \{A, B\}$, initial state $I = \emptyset$, actions

$\text{car}_A : \emptyset \rightarrow A$ cost 1, $\text{car}_B : \emptyset \rightarrow B$ cost 1, $\text{fancyCar} : \emptyset \rightarrow A \land B$ cost 1.5.

**Landmarks set** $\{LM\}: \{A, B\}$. Thus $h(I) = 2 > h^*(I)$. 

---

**References**

1. Karpas and Domshlak (2009)
How To Admissibly Combine Landmarks!

Planning task: Goals $G = \{A, B\}$, initial state $I = \emptyset$, actions $carA : \emptyset \rightarrow A$ cost 1, $carB : \emptyset \rightarrow B$ cost 1, $fancyCar : \emptyset \rightarrow A \land B$ cost 1.5.

Landmarks set $\{LM\}: \{A, B\}$. Thus $h(I) = 2 > h^*(I)$.

Solution: [Karpas and Domshlak (2009)]

1. Consider disjunctive action landmarks instead: $L_A = \{carA, fancyCar\}$, $L_B = \{carB, fancyCar\}$. (= Achievers of each landmark)

→ Elementary landmark heuristic $h_{LM}^L(s) = \min \{c(a) \mid a \in L\}$ if $L$ is a disjunctive action landmark for $s$, and $h_{LM}^L(s) = 0$ otherwise.
How To Admissibly Combine Landmarks!

Planning task: Goals $G = \{A, B\}$, initial state $I = \emptyset$, actions

$\text{carA} : \emptyset \rightarrow A \text{ cost 1}$, $\text{carB} : \emptyset \rightarrow B \text{ cost 1}$, $\text{fancyCar} : \emptyset \rightarrow A \land B \text{ cost } 1.5$.

Landmarks set $\{LM\}: \{A, B\}$. Thus $h(I) = 2 > h^*(I)$.

Solution: [Karpas and Domshlak (2009)]

1. Consider disjunctive action landmarks instead: $L_A = \{\text{carA, fancyCar}\}$, $L_B = \{\text{carB, fancyCar}\}$. (= Achievers of each landmark)
   
   $\rightarrow$ Elementary landmark heuristic $h_{L_M}^{LM}(s) = \min \{c(a) \mid a \in L\}$ if $L$ is a disjunctive action landmark for $s$, and $h_{L_M}^{LM}(s) = 0$ otherwise.

2. Partition action costs to make $\sum_L h_{L_M}^{LM}(s)$ admissible!
Cost Partitionings

**Cost Partitioning:** Ensemble of functions \( c_1, \ldots, c_n : A \mapsto \mathbb{R}^+_0 \) s.t. for all \( a \in A \),
\[ \sum_{i=1}^n c_i(a) \leq \text{cost}(a). \]
Cost Partitionings

Cost Partitioning: Ensemble of functions $c_1, \ldots, c_n : A \mapsto \mathbb{R}_0^+$ s.t. for all $a \in A$, $
abla_{i=1}^n c_i(a) \leq \text{cost}(a)$.

Admissible Sum: For heuristics $h_1, \ldots, h_n$, $
abla_{i=1}^n h_i[c_i] \leq h^*$. 
Cost Partitionings

Cost Partitioning: Ensemble of functions $c_1, \ldots, c_n : A \mapsto \mathbb{R}^+_0$ s.t. for all $a \in A$, $
abla_{i=1}^n c_i(a) \leq \text{cost}(a)$.

Admissible Sum: For heuristics $h_1, \ldots, h_n$, $\nabla_{i=1}^n h_i[c_i] \leq h^*$.

$\rightarrow c_1, \ldots, c_n$ optimal for $h_1, \ldots, h_n$ and $s$ if $\nabla_{i=1}^n h_i[c_i](s)$ is maximal.
Cost Partitionings

**Cost Partitioning:** Ensemble of functions $c_1, \ldots, c_n : A \mapsto \mathbb{R}^+_0$ s.t. for all $a \in A$, 

$$\sum_{i=1}^{n} c_i(a) \leq \text{cost}(a).$$

**Admissible Sum:** For heuristics $h_1, \ldots, h_n$, 

$$\sum_{i=1}^{n} h_i[c_i] \leq h^*.$$

$\rightarrow$ $c_1, \ldots, c_n$ optimal for $h_1, \ldots, h_n$ and $s$ if 

$$\sum_{i=1}^{n} h_i[c_i](s) \text{ is maximal.}$$

**Theorem.** Let $s$ be a state, and let $L_1, \ldots, L_n$ be disjunctive action landmarks for $s$. Then an optimal cost partitioning for $s$ and $h_{L_1}^{\text{LM}}, \ldots, h_{L_n}^{\text{LM}}$ can be computed in polynomial time.

**Proof.** We can encode this optimization problem into Linear Programming.
## Cost Partitionings

**Cost Partitioning:** Ensemble of functions $c_1, \ldots, c_n : A \mapsto \mathbb{R}^+_0$ s.t. for all $a \in A$, 
$$\sum_{i=1}^{n} c_i(a) \leq \text{cost}(a).$$

**Admissible Sum:** For heuristics $h_1, \ldots, h_n$, 
$$\sum_{i=1}^{n} h_i[c_i] \leq h^*.$$  
$\rightarrow c_1, \ldots, c_n$ optimal for $h_1, \ldots, h_n$ and $s$ if 
$$\sum_{i=1}^{n} h_i[c_i](s)$$ is maximal.

**Theorem.** Let $s$ be a state, and let $L_1, \ldots, L_n$ be disjunctive action landmarks for $s$. Then an optimal cost partitioning for $s$ and $h_{LM}^{L_1}, \ldots, h_{LM}^{L_n}$ can be computed in polynomial time.

**Proof.** We can encode this optimization problem into Linear Programming.

**Example:** $L_A = \{\text{carA, fancyCar}\}$, $L_B = \{\text{carB, fancyCar}\}$.

- $\text{carA : } h_{LA} \leq 1$
- $\text{carB : } h_{LB} \leq 1$
- $\text{fancyCar : } h_{LA} + h_{LB} \leq 1.5$

$\rightarrow$ Maximizing $h_{LA} + h_{LB}$ yields $h(I) = 1.5$. 

Note: First done for abstraction heuristics [Katz and Domshlak (2008)].
Cost Partitionings

**Cost Partitioning:** Ensemble of functions \( c_1, \ldots, c_n : A \mapsto \mathbb{R}^+ \) s.t. for all \( a \in A \),
\[
\sum_{i=1}^{n} c_i(a) \leq \text{cost}(a).
\]

**Admissible Sum:** For heuristics \( h_1, \ldots, h_n \), \[
\sum_{i=1}^{n} h_i[c_i] \leq h^*.
\]

\( \rightarrow \) \( c_1, \ldots, c_n \) optimal for \( h_1, \ldots, h_n \) and \( s \) if \( \sum_{i=1}^{n} h_i[c_i](s) \) is maximal.

**Theorem.** Let \( s \) be a state, and let \( L_1, \ldots, L_n \) be disjunctive action landmarks for \( s \). Then an optimal cost partitioning for \( s \) and \( h_{LM}^{L_1}, \ldots, h_{LM}^{L_n} \) can be computed in polynomial time.

**Proof.** We can encode this optimization problem into Linear Programming.

**Example:** \( L_A = \{\text{carA}, \text{fancyCar}\} \), \( L_B = \{\text{carB}, \text{fancyCar}\} \).

- \( \text{carA} : \quad h_{L_A} \quad \leq \quad 1 \)
- \( \text{carB} : \quad h_{L_B} \quad \leq \quad 1 \)
- \( \text{fancyCar} : \quad h_{L_A} + h_{L_B} \quad \leq \quad 1.5 \)

\( \rightarrow \) Maximizing \( h_{L_A} + h_{L_B} \) yields \( h(I) = 1.5 \).

**Note:** First done for abstraction heuristics [Katz and Domshlak (2008)].
The Impact of Stage 2

For those of you who don't remember that scene: It didn't happen.

Karpas and Domshlak (2009)'s heuristic was part of Fast Downward Stone Soup and Selective Max in IPC'11.
The Impact of Stage 2

→ For those of you who don’t remember that scene: It didn’t happen.
The Impact of Stage 2

→ For those of you who don’t remember that scene: It didn’t happen. Karpas and Domshlak (2009)’s heuristic was part of Fast Downward Stone Soup and Selective Max in IPC’11.
Agenda: Stage 3 (Wasn’t That Mars We Just Passed?)
Agenda: Stage 3 (Wasn’t That Mars We Just Passed?)
Many Disjunctive Action Landmarks!

Pre-Eff Structure: Actions $\text{get}(X, Y)$; init $A$, goal $E$.

Fact landmarks:
Many Disjunctive Action Landmarks!

**Pre-Eff Structure:** Actions $get(X, Y)$; init $A$, goal $E$.

**Fact landmarks:** $\{B, E\}$, yielding $h(I) = 2$. 
Many Disjunctive Action Landmarks!

**Pre-Eff Structure:** Actions $\text{get}(X, Y)$; init $A$, goal $E$.

Fact landmarks: $\{B, E\}$, yielding $h(I) = 2$.

And now, let’s pass Mars: LM-cut! [Helmert and Domshlak (2009)]

$\rightarrow h(I) = 0$
Many Disjunctive Action Landmarks!

Pre-Eff Structure: Actions \( \text{get}(X, Y) \); init \( A \), goal \( E \).

Fact landmarks: \( \{B, E\} \), yielding \( h(I) = 2 \).

And now, let’s pass Mars: LM-cut! [Helmert and Domshlak (2009)]

\[ \rightarrow h(I) = 1 \]
Many Disjunctive Action Landmarks!

Pre-Eff Structure: Actions $get(X, Y)$; init $A$, goal $E$.

Fact landmarks: $\{B, E\}$, yielding $h(I) = 2$.

And now, let’s pass Mars: LM-cut! [Helmert and Domshlak (2009)]

$\rightarrow h(I) = 1$
Many Disjunctive Action Landmarks!

Pre-Eff Structure: Actions $get(X, Y)$; init $A$, goal $E$.

Fact landmarks: $\{B, E\}$, yielding $h(I) = 2$.

And now, let’s pass Mars: LM-cut! [Helmert and Domshlak (2009)]

$\rightarrow h(I) = 2$
Many Disjunctive Action Landmarks!

Pre-Eff Structure: Actions $\text{get}(X, Y)$; init $A$, goal $E$.

Fact landmarks: $\{B, E\}$, yielding $h(I) = 2$.

And now, let’s pass Mars: LM-cut! [Helmert and Domshlak (2009)]

→ $h(I) = 2$
Many Disjunctive Action Landmarks!

Pre-Eff Structure: Actions $\text{get}(X, Y)$; init $A$, goal $E$.

Fact landmarks: $\{B, E\}$, yielding $h(I) = 2$.

And now, let’s pass Mars: LM-cut! [Helmert and Domshlak (2009)]

$\rightarrow h(I) = 3$
Many Disjunctive Action Landmarks!

Pre-Eff Structure: Actions $get(X, Y)$; init $A$, goal $E$.

Fact landmarks: $\{B, E\}$, yielding $h(I) = 2$.

And now, let’s pass Mars: LM-cut! [Helmert and Domshlak (2009)]

$\rightarrow h(I) = 3$
Many Disjunctive Action Landmarks!

**Pre-Eff Structure:** Actions $get(X, Y)$; init $A$, goal $E$.  

![Diagram](image)

**Fact landmarks:** $\{B, E\}$, yielding $h(I) = 2$.

**And now, let’s pass Mars:** LM-cut! [Helmert and Domshlak (2009)]

![Diagram](image)

$\rightarrow h(I) = 4$
Many Disjunctive Action Landmarks!

**Pre-Eff Structure:** Actions $\text{get}(X, Y)$; init $A$, goal $E$.

![Graph of pre-eff structure](image)

**Fact landmarks:** $\{B, E\}$, yielding $h(I) = 2$.

**And now, let’s pass Mars:** LM-cut! [Helmert and Domshlak (2009)]

![Graph with LM-cut](image)

$\rightarrow h(I) = 4$
Many Disjunctive Action Landmarks!

Pre-Eff Structure: Actions $get(X, Y)$; init $A$, goal $E$.

Fact landmarks: $\{B, E\}$, yielding $h(I) = 2$.

And now, let’s pass Mars: LM-cut! [Helmert and Domshlak (2009)]

$\rightarrow h(I) = 5$
Many Disjunctive Action Landmarks!

Pre-Eff Structure: Actions $\text{get}(X, Y)$; init $A$, goal $E$.

Fact landmarks: $\{B, E\}$, yielding $h(I) = 2$.

And now, let’s pass Mars: LM-cut! [Helmert and Domshlak (2009)]

$\rightarrow h(I) = 5$
Many Disjunctive Action Landmarks!

Pre-Eff Structure: Actions $\text{get}(X, Y)$; init $A$, goal $E$.

Fact landmarks: $\{B, E\}$, yielding $h(I) = 2$.

And now, let’s pass Mars: LM-cut! [Helmert and Domshlak (2009)]

$\rightarrow h(I) = 6$
Many Disjunctive Action Landmarks!

Pre-Eff Structure: Actions $\mathit{get}(X, Y)$; init $A$, goal $E$.

Fact landmarks: $\{B, E\}$, yielding $h(I) = 2$.

And now, let’s pass Mars: LM-cut! [Helmert and Domshlak (2009)]

$\rightarrow h(I) = 6$
Many Disjunctive Action Landmarks!

Pre-Eff Structure: Actions $\text{get}(X, Y)$; init $A$, goal $E$.

Fact landmarks: $\{B, E\}$, yielding $h(I) = 2$.

And now, let’s pass Mars: LM-cut! [Helmert and Domshlak (2009)]

$\rightarrow h(I) = 6 = h^*(I)$. 
The Impact of Stage 3

IPC 2008: Best optimal planner in the competition.
Agenda: Stage 4 (Off to the Milky Way!!)
Agenda: Stage 4 (Off to the Milky Way!!)
Agenda: Stage 4 (Off to the Milky Way!!)
Hitting Sets Over Landmarks!

Stage 0
Stage 1
Stage 2
Stage 3
Stage 4
References

Hitting Sets Over Landmarks!

A

B

C

I

\{carAB, carBC\}

\{carAB, carAC\}

\{carAC, carBC\}

Precondition-Choice Functions

Landmarks:

Optimal cost partitioning:

\( h(I) = 1.5 < h^*(I) \):

\( \text{Set } h_L^A = h_L^B = h_L^C = 0.5 \).

Minimum cost hitting set:

\( h(I) = 2 = h^*(I) \): E.g.,

\( H := \{\text{carAB, carAC}\} \).

Hitting sets are admissible:

Let \( L_1, \ldots, L_n \) be disjunctive action landmarks for \( s \). Let \( H \) be a minimum-cost hitting set. Then

\[ \sum_{a \in H} \text{cost}(a) \leq h^*(s) . \]

(Simply because by definition every plan must hit every \( L_i \).)
Hitting Sets Over Landmarks!

Landmarks:

Precondition-Choice Functions

\[
\text{Landmarks: } A \quad B \quad C
\]

\[
\{\text{carAB, carBC}\}
\]

\[
\{\text{carAB, carAC}\}
\]

\[
\{\text{carAC, carBC}\}
\]

\[
\text{Optimal cost partitioning: } h(I) = 1.5 < h^*(I)
\]

\[
\text{Set } h_L = h_L = h_L = 0.5.
\]

\[
\text{Minimum cost hitting set: } h(I) = 2 = h^*(I)
\]

\[
H = \{\text{carAB, carAC}\}
\]

\[
\text{Hitting sets are admissible: Let } L_1, \ldots, L_n \text{ be disjunctive action landmarks for s. Let } H \text{ be a minimum-cost hitting set. Then } \sum_{a \in H} \text{cost}(a) \leq h^*(s).
\]

(Simply because by definition every plan must hit every \(L_i\).)
Hitting Sets Over Landmarks!

Landmarks: \{carAB, carAC\}

Precondition-Choice Functions

Optimal cost partitioning: \( h(I) = 1.5 \), \( h^*(I) \):

- Set \( h_L = h_L' = h_L'' = 0.5 \).

Minimum cost hitting set: \( h(I) = 2 = h^*(I) \): E.g., \( H = \{carAB, carAC\} \).

Hitting sets are admissible: Let \( L_1, \ldots, L_n \) be disjunctive action landmarks for \( s \). Let \( H \) be a minimum-cost hitting set. Then \( \sum_{a \in H} \text{cost}(a) \leq h^*(s) \).

(Simply because by definition every plan must hit every \( L_i \).)
Hitting Sets Over Landmarks!

**Landmarks:** \{carAB, carAC\}, \{carAB, carBC\}

**Precondition-Choice Functions**

\[ h(I) = 1.5 < h^*(I) : \text{Set } h_LA = h_LB = h_LC = 0.5. \]

**Minimum cost hitting set:** \[ h(I) = 2 = h^*(I) : \text{E.g., } H = \{carAB, carAC\}. \]

**Hitting sets are admissible:** Let \( L_1, \ldots, L_n \) be disjunctive action landmarks for \( s \). Let \( H \) be a minimum-cost hitting set. Then \[ \sum_{a \in H} \text{cost}(a) \leq h^*(s). \] (Simply because by definition every plan must hit every \( L_i \).)
Hitting Sets Over Landmarks!

Landmarks: \{\text{car AB, car AC}\}, \{\text{car AB, car BC}\}, \{\text{car AC, car BC}\}

Precondition-Choice Functions

\[ h(I) = 1.5 < h^*(I) \]  
\[ \sum_{a \in H} \text{cost}(a) \leq h^*(s) \]  

(Simply because by definition every plan must hit every \( L_i \).)
Hitting Sets Over Landmarks!

**Landmarks:** \{car_{AB}, car_{AC}\}, \{car_{AB}, car_{BC}\}, \{car_{AC}, car_{BC}\}. (Action costs: Uniform 1.)
Hitting Sets Over Landmarks!

**Landmarks:** \{car_{AB}, car_{AC}\}, \{car_{AB}, car_{BC}\}, \{car_{AC}, car_{BC}\}. (Action costs: Uniform 1.)

**Optimal cost partitioning:** \(h(I) = 1.5 < h^*(I)\): Set \(h_{LA} = h_{LB} = h_{LC} = 0.5\).
Hitting Sets Over Landmarks!

**Landmarks:** \{car\textsubscript{\textit{AB}}, car\textsubscript{\textit{AC}}\}, \{car\textsubscript{\textit{AB}}, car\textsubscript{\textit{BC}}\}, \{car\textsubscript{\textit{AC}}, car\textsubscript{\textit{BC}}\}. (Action costs: Uniform 1.)

**Optimal cost partitioning:** \(h(I) = 1.5 < h^*(I)\): Set \(h_{L\textsubscript{A}} = h_{L\textsubscript{B}} = h_{L\textsubscript{C}} = 0.5\).

**Minimum cost hitting set:** \(h(I) = 2 = h^*(I)\): E.g., \(H := \{car\textsubscript{\textit{AB}}, car\textsubscript{\textit{AC}}\}\).
Hitting Sets Over Landmarks!

Landmarks: \{\text{car}AB, \text{car}AC\}, \{\text{car}AB, \text{car}BC\}, \{\text{car}AC, \text{car}BC\}. (Action costs: Uniform 1.)

Optimal cost partitioning: \(h(I) = 1.5 < h^*(I)\): Set \(h_{LA} = h_{LB} = h_{LC} = 0.5\).

Minimum cost hitting set: \(h(I) = 2 = h^*(I)\): E.g., \(H := \{\text{car}AB, \text{car}AC\}\).

Hitting sets are admissible: Let \(L_1, \ldots, L_n\) be disjunctive action landmarks for \(s\). Let \(H\) be a minimum-cost hitting set. Then \(\sum_{a \in H} \text{cost}(a) \leq h^*(s)\). (Simply because by definition every plan must hit every \(L_i\).)
From Landmarks to $h^+$! [Bonet and Helmert (2010)]

**Theorem.** Let $s$ be a state, and let $L_1, \ldots, L_n$ be the collection of disjunctive action landmarks for $s$ resulting from all precondition-choice functions and cuts. Let $H$ be a minimum-cost hitting set. Then 

$$\sum_{a \in H} \text{cost}(a) = h^+(s).$$

**Proof.** Any relaxed plan must hit $L_1, \ldots, L_n$ so

$$\sum_{a \in H} \text{cost}(a) \leq h^+(s).$$
From Landmarks to $h^+$! [Bonet and Helmert (2010)]

**Theorem.** Let $s$ be a state, and let $L_1, \ldots, L_n$ be the collection of disjunctive action landmarks for $s$ resulting from all precondition-choice functions and cuts. Let $H$ be a minimum-cost hitting set. Then $\sum_{a \in H} \text{cost}(a) = h^+(s)$.

**Proof.** Any relaxed plan must hit $L_1, \ldots, L_n$ so $\sum_{a \in H} \text{cost}(a) \leq h^+(s)$. We now prove that any hitting set $H$ contains a relaxed plan.
From Landmarks to $h^+$! [Bonet and Helmert (2010)]

**Theorem.** Let $s$ be a state, and let $L_1, \ldots, L_n$ be the collection of disjunctive action landmarks for $s$ resulting from all precondition-choice functions and cuts. Let $H$ be a minimum-cost hitting set. Then $\sum_{a \in H} \text{cost}(a) = h^+(s)$.

**Proof.** Any relaxed plan must hit $L_1, \ldots, L_n$ so $\sum_{a \in H} \text{cost}(a) \leq h^+(s)$. We now prove that any hitting set $H$ contains a relaxed plan. With $R_H := \{p \mid p \text{ can be reached in delete-relaxation using only } H\}$, assume to the contrary that $G \not\subseteq R_H$. Consider the cut $L$ defined by $R_H, \overline{R_H}$:

```
  s  \rightarrow a (1) \rightarrow L \rightarrow a (2) \rightarrow G
  \hspace{1cm}
  R_H \hspace{1cm} \overline{R_H}
  \hspace{1cm}
  a (1) \rightarrow L \rightarrow a (2)
```
**Theorem.** Let $s$ be a state, and let $L_1, \ldots, L_n$ be the collection of disjunctive action landmarks for $s$ resulting from all precondition-choice functions and cuts. Let $H$ be a minimum-cost hitting set. Then $\sum_{a \in H} \text{cost}(a) = h^+(s)$.

**Proof.** Any relaxed plan must hit $L_1, \ldots, L_n$ so $\sum_{a \in H} \text{cost}(a) \leq h^+(s)$. We now prove that any hitting set $H$ contains a relaxed plan. With $R_H := \{p \mid p$ can be reached in delete-relaxation using only $H\}$, assume to the contrary that $G \not\subseteq R_H$. Consider the cut $L$ defined by $R_H, \overline{R_H}$:

![Diagram](image)

Case (1): If $\text{pre}_a \subseteq R_H$ then $\text{add}_a \subseteq R_H$ so $a \not\in L$. 

From Landmarks to $h^+$! [Bonet and Helmert (2010)]
From Landmarks to $h^+$! [Bonet and Helmert (2010)]

**Theorem.** Let $s$ be a state, and let $L_1, \ldots, L_n$ be the collection of disjunctive action landmarks for $s$ resulting from all precondition-choice functions and cuts. Let $H$ be a minimum-cost hitting set. Then $\sum_{a \in H} \text{cost}(a) = h^+(s)$.

**Proof.** Any relaxed plan must hit $L_1, \ldots, L_n$ so $\sum_{a \in H} \text{cost}(a) \leq h^+(s)$. We now prove that any hitting set $H$ contains a relaxed plan. With $R_H := \{p \mid p \text{ can be reached in delete-relaxation using only } H\}$, assume to the contrary that $G \not\subseteq R_H$. Consider the cut $L$ defined by $R_H, \overline{R_H}$:

Case (1): If $\text{pre}_a \subseteq R_H$ then $\text{add}_a \subseteq R_H$ so $a \not\in L$.
Case (2): If $\text{pre}_a \not\subseteq R_H$ then our precondition-choice function can select $p \in \text{pre}_a \setminus R_H$ so, again, $a \not\in L$. So $H$ does not hit $L$, in contradiction.
The Impact of Stage 4

Well, isn’t it just beautiful?
The Impact of Stage 4

Well, isn’t it just beautiful?

More concretely:

- Improved LM-cut, runtime-effective in cases with large search space reduction [Bonet and Helmert (2010); Bonet and Castillo (2011)].
The Impact of Stage 4

Well, isn’t it just beautiful?

More concretely:

- Improved LM-cut, runtime-effective in cases with large search space reduction [Bonet and Helmert (2010); Bonet and Castillo (2011)].

- State of the art method for computing $h^+$ [Haslum et al. (2012)].
The Impact of Stage 4

Well, isn’t it just beautiful?

More concretely:

- Improved LM-cut, runtime-effective in cases with large search space reduction [Bonet and Helmert (2010); Bonet and Castillo (2011)].

- State of the art method for computing $h^+$ [Haslum et al. (2012)].

- State of the art method for computing $h^{++}$, i.e., $h^+$ computed in compilation $\Pi^C$, which converges to $h^*$ [Haslum et al. (2012)].
And now: No questions. Off to dinner!

p.s.: Apologies and thanks to everybody who worked on landmarks but is not mentioned here!


