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

Song # 1

Song # 1

```
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 # 1

```
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

Song # 2

`Planning, planning, planning,`

Song # 2

```
Planning, planning, planning,
      P-D-D-L scanning,
```

Song # 2

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

Song # 2

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

Uncertain durations,
```

Song # 2

```
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!

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’

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'

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.

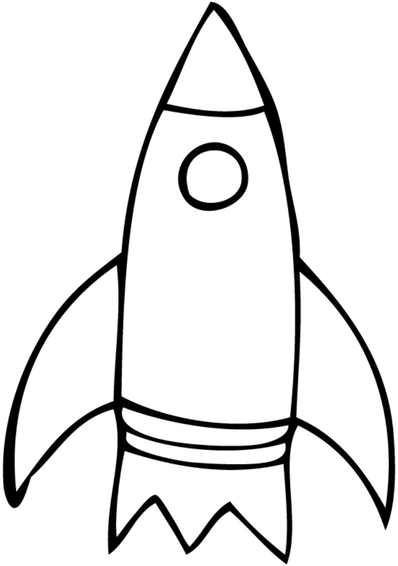
Song # 2

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

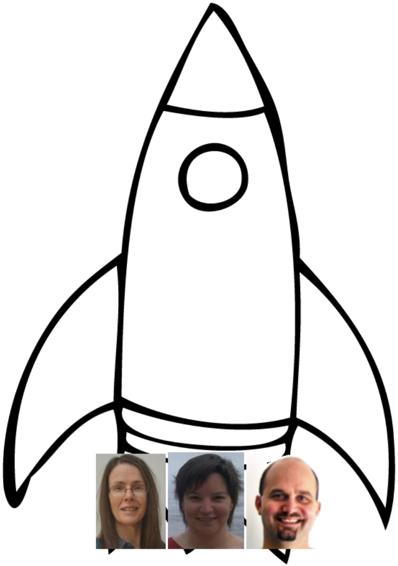
Uncertain durations,
Truth ramifications,
Wishing FF was by my side!

My soft goals they are kissin'
My landmarks have gone missin'
My stubborn set has turned off the light.

Agenda

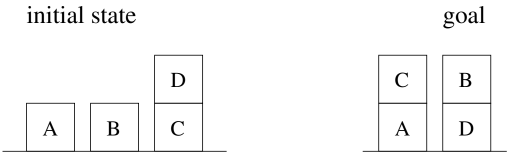


Agenda: Stage 0 (The Dark Ages)



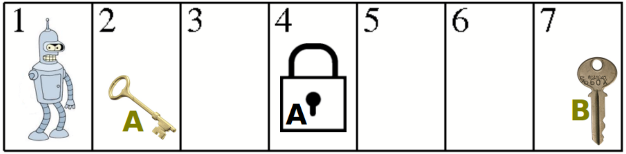
Once Upon a Time, There Was a Landmark . . .

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



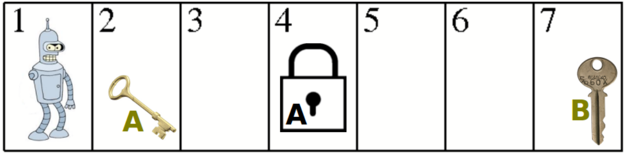
What Are Landmarks?

Problem: Bring key B to position 1.



What Are Landmarks?

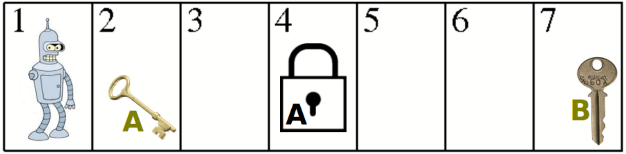
Problem: Bring key B to position 1.



Landmarks:

What Are Landmarks?

Problem: Bring key B to position 1.

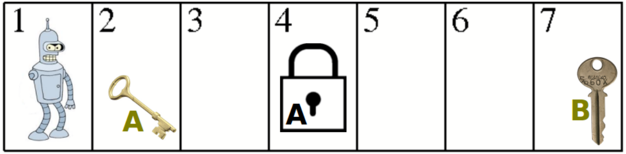


Landmarks:

- robot-at-2, robot-at-3, robot-at-4, robot-at-5, robot-at-6, robot-at-7.

What Are Landmarks?

Problem: Bring key B to position 1.

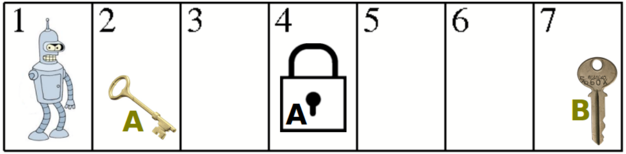


Landmarks:

- robot-at-2, robot-at-3, robot-at-4, robot-at-5, robot-at-6, robot-at-7.
- Lock-open,

What Are Landmarks?

Problem: Bring key B to position 1.

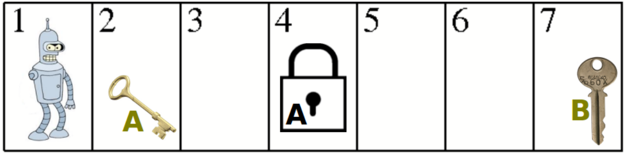


Landmarks:

- robot-at-2, robot-at-3, robot-at-4, robot-at-5, robot-at-6, robot-at-7.
- Lock-open, Have-key-A,

What Are Landmarks?

Problem: Bring key B to position 1.

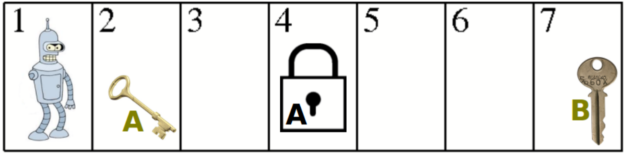


Landmarks:

- 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,

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 $L \leq L'$.

And Now?

And Now?

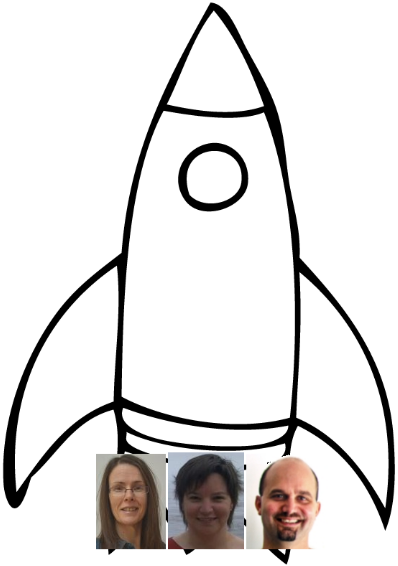
Well, some guy (me, that is) proposed to use this for problem decomposition, but never mind that.

And Now?

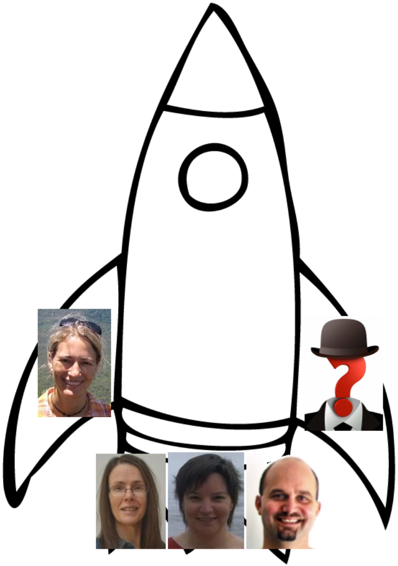
Well, some guy (me, that is) proposed to use this for problem decomposition, but never mind that.

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

Agenda: Stage 1 (Preparing for Take-Off)

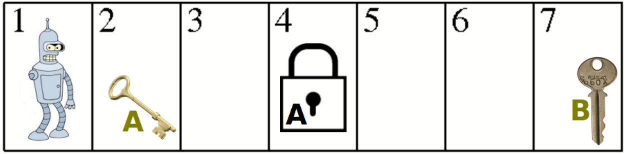


Agenda: Stage 1 (Preparing for Take-Off)



How To Use Landmarks!

Problem: Bring key B to position 1.



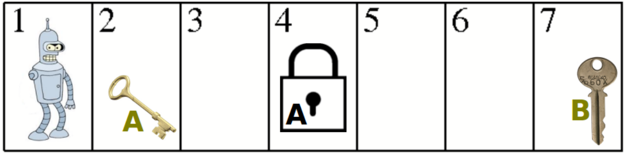
Landmarks set $\{LM\}$**:**

- 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, . . .

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

How To Use Landmarks!

Problem: Bring key B to position 1.



Landmarks set $\{LM\}$:

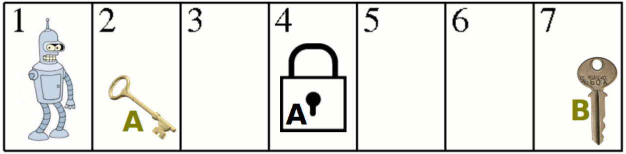
- robot-at-2, robot-at-3, robot-at-4, robot-at-5, robot-at-6, robot-at-7.
- Lock-open, Have-key-A, Have-key-B, ...

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

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

How To Use Landmarks!

Problem: Bring key B to position 1.



Landmarks set $\{LM\}$:

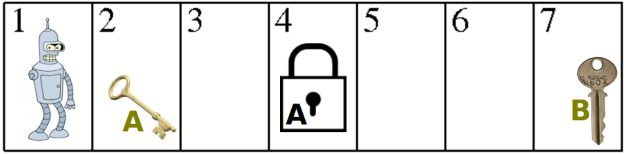
- robot-at-2, robot-at-3, robot-at-4, robot-at-5, robot-at-6, robot-at-7.
- Lock-open, Have-key-A, Have-key-B, . . .

$\rightarrow h(s) := |\{LM\} \setminus s|$. ("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 $\{LM\}$**:**

- 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, . . .

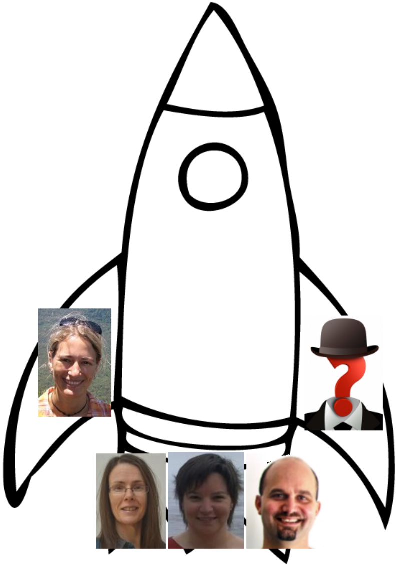
→ $h(s) := |\{LM\} \setminus s|$. ("Number of open items on the to-do list")

- 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" . . . !

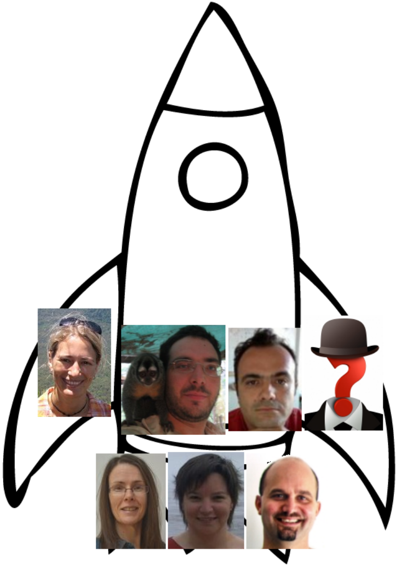
The Impact of Stage 1

The Impact of Stage 1

Agenda: Stage 2 (Leaving the Atmosphere)



Agenda: Stage 2 (Leaving the Atmosphere)



How To *Admissibly Combine* Landmarks!

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, $fancyCar : \emptyset \rightarrow A \wedge B$ cost 1.5.

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, $fancyCar : \emptyset \rightarrow A \wedge B$ cost 1.5.

Landmarks set $\{LM\}$:

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, $fancyCar : \emptyset \rightarrow A \wedge B$ cost 1.5.

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

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, $fancyCar : \emptyset \rightarrow A \wedge B$ cost 1.5.

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

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, $fancyCar : \emptyset \rightarrow A \wedge B$ 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)
   $\rightarrow$ Elementary landmark heuristic $h^{\mathsf{LM}}_L(s) = \min\{c(a) \mid a \in L\}$ if L is a disjunctive action landmark for s, and $h^{\mathsf{LM}}_L(s) = 0$ otherwise.

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, $fancyCar : \emptyset \rightarrow A \wedge B$ 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)
   $\rightarrow$ Elementary landmark heuristic $h_L^{\text{LM}}(s) = \min\{c(a) \mid a \in L\}$ if L is a disjunctive action landmark for s, and $h_L^{\text{LM}}(s) = 0$ otherwise.
2. Partition action costs to make $\sum_L h_L^{\text{LM}}(s)$ admissible!

Cost Partitionings

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

Cost Partitionings

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

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

Cost Partitionings

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

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

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

Cost Partitionings

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

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

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

Theorem. *Let s be a state, and let $L_1, \dots, L_n$ 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 polynomial time.*

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

Cost Partitionings

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

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

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

Theorem. *Let s be a state, and let $L_1, \dots, L_n$ 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 polynomial time.*

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

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

$$
\begin{array}{rcccl}
carA: & h_{L_A} & & & \leq 1 \\
carB: & & & h_{L_B} & \leq 1 \\
fancyCar: & h_{L_A} & + & h_{L_B} & \leq 1.5
\end{array}
$$

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

Cost Partitionings

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

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

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

Theorem. *Let s be a state, and let $L_1, \dots, L_n$ 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 polynomial time.*

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

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

$$\begin{array}{rcccll} carA: & h_{L_A} & & & \leq & 1 \\ carB: & & & h_{L_B} & \leq & 1 \\ fancyCar: & h_{L_A} & + & h_{L_B} & \leq & 1.5 \end{array}$$

$\rightarrow$ 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

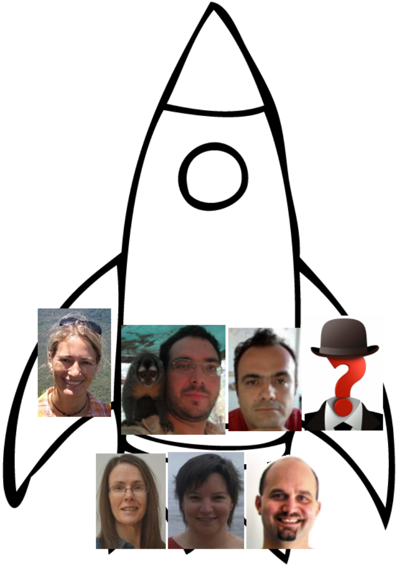
The Impact of Stage 2

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

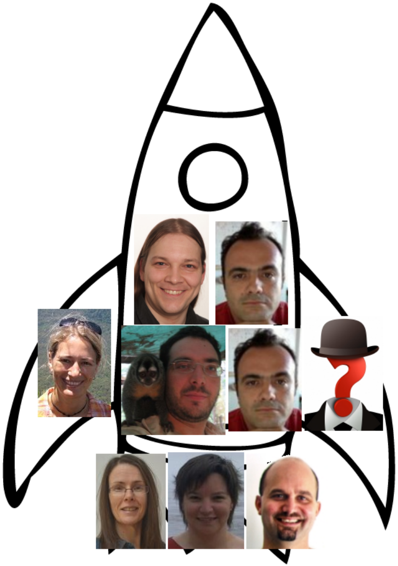
The Impact of Stage 2

→ 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?)

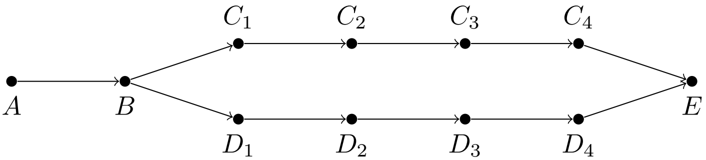


Agenda: Stage 3 (Wasn't That Mars We Just Passed?)



Many Disjunctive Action Landmarks!

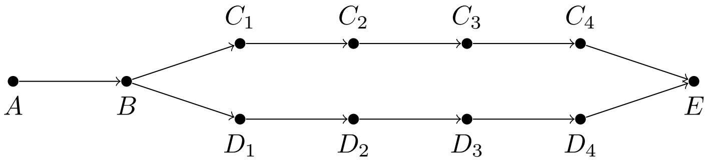
Pre-Eff Structure: Actions $get(X, Y)$; init A, goal E.



Fact landmarks:

Many Disjunctive Action Landmarks!

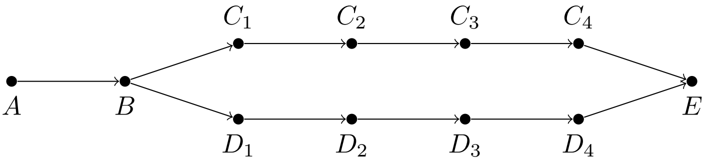
Pre-Eff Structure: Actions $get(X, Y)$; init A, goal E.



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

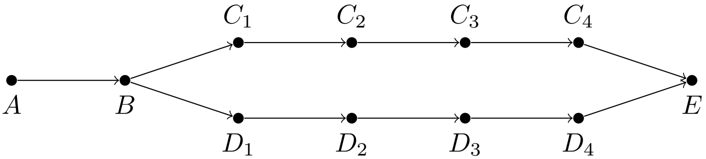
Many Disjunctive Action Landmarks!

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



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

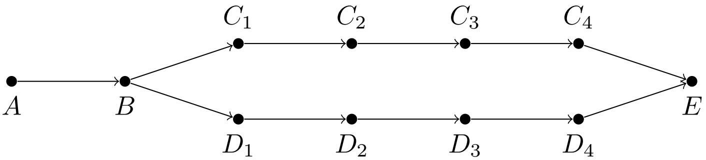
And now, let's pass Mars: LM-cut! [Helmert and Domshlak (2009)]



$\rightarrow h(I) = 0$

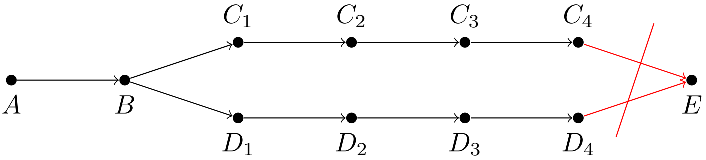
Many Disjunctive Action Landmarks!

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



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

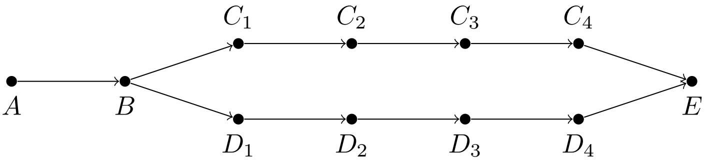
And now, let's pass Mars: LM-cut! [Helmert and Domshlak (2009)]



$\rightarrow h(I) = 1$

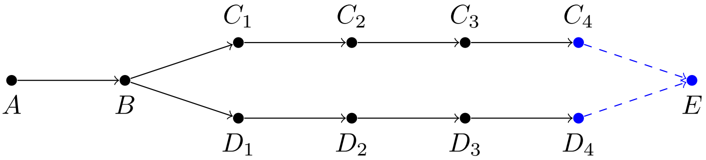
Many Disjunctive Action Landmarks!

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



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

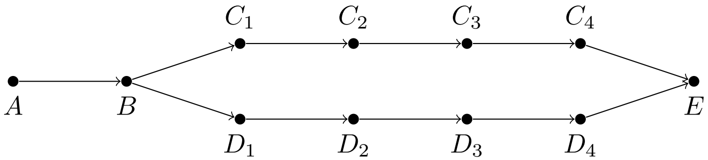
And now, let's pass Mars: LM-cut! [Helmert and Domshlak (2009)]



$\rightarrow h(I) = 1$

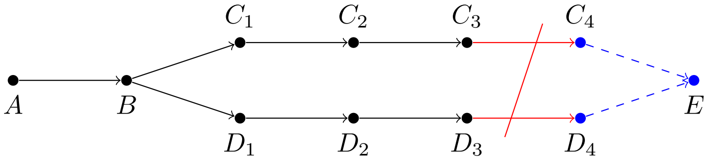
Many Disjunctive Action Landmarks!

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



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

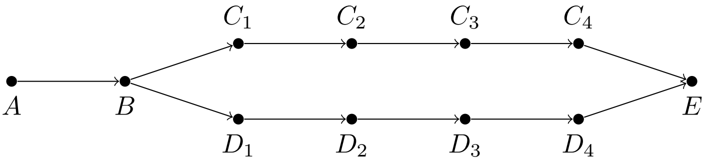
And now, let's pass Mars: LM-cut! [Helmert and Domshlak (2009)]



$\rightarrow h(I) = 2$

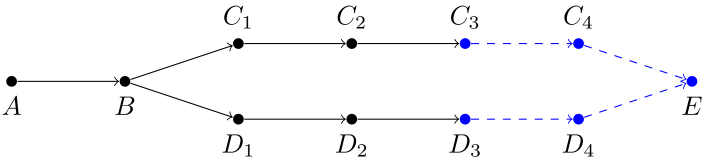
Many Disjunctive Action Landmarks!

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



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

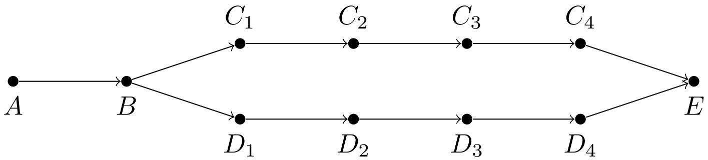
And now, let's pass Mars: LM-cut! [Helmert and Domshlak (2009)]



$\rightarrow h(I) = 2$

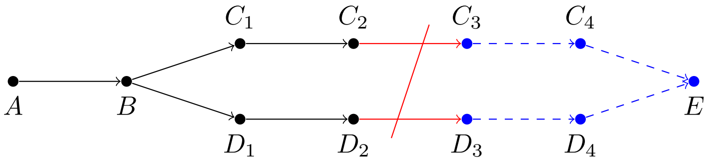
Many Disjunctive Action Landmarks!

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



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

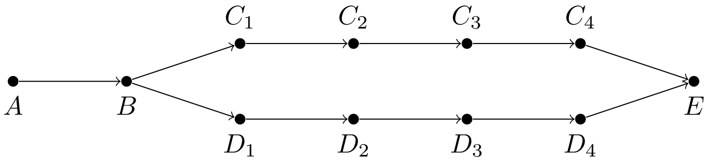
And now, let's pass Mars: LM-cut! [Helmert and Domshlak (2009)]



$\rightarrow h(I) = 3$

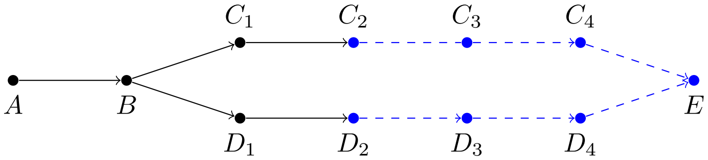
Many Disjunctive Action Landmarks!

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



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

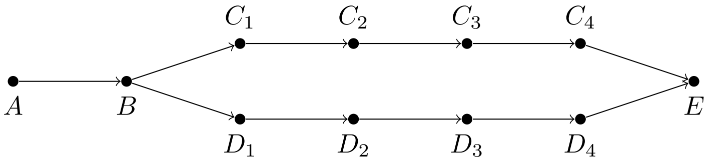
And now, let's pass Mars: LM-cut! [Helmert and Domshlak (2009)]



$\rightarrow h(I) = 3$

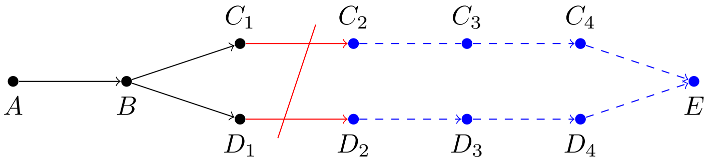
Many Disjunctive Action Landmarks!

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



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

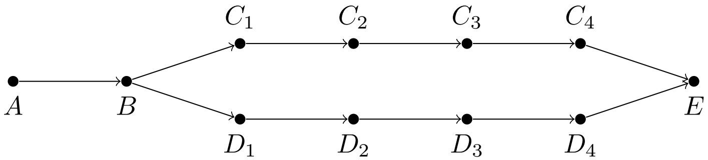
And now, let's pass Mars: LM-cut! [Helmert and Domshlak (2009)]



$\rightarrow h(I) = 4$

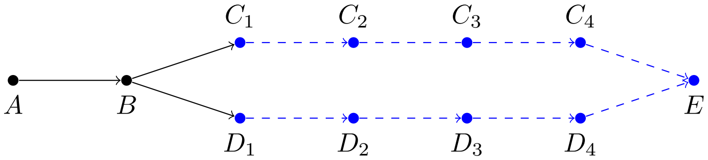
Many Disjunctive Action Landmarks!

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



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

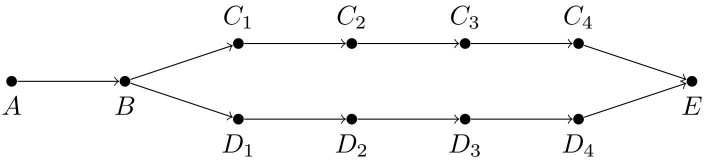
And now, let's pass Mars: LM-cut! [Helmert and Domshlak (2009)]



$\rightarrow h(I) = 4$

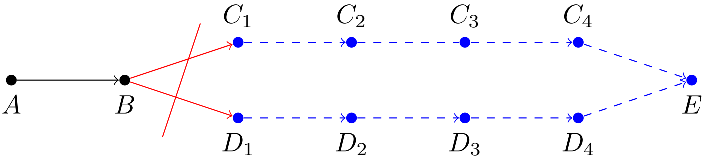
Many Disjunctive Action Landmarks!

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



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

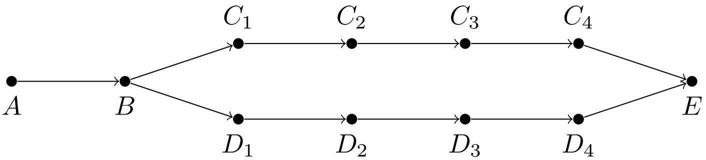
And now, let's pass Mars: LM-cut! [Helmert and Domshlak (2009)]



$\rightarrow h(I) = 5$

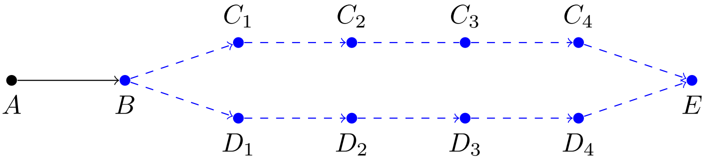
Many Disjunctive Action Landmarks!

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



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

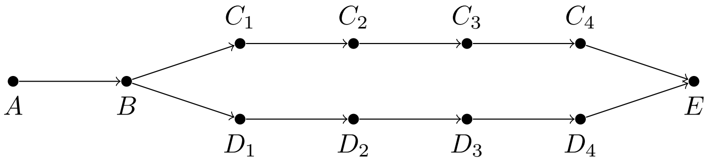
And now, let's pass Mars: LM-cut! [Helmert and Domshlak (2009)]



$\rightarrow h(I) = 5$

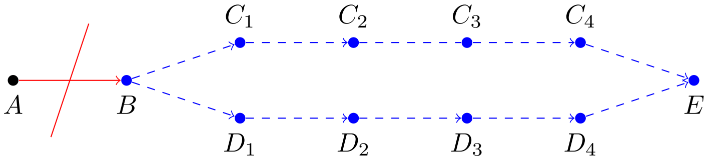
Many Disjunctive Action Landmarks!

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



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

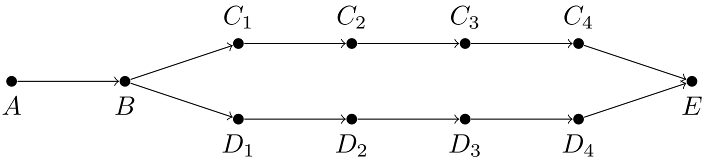
And now, let's pass Mars: LM-cut! [Helmert and Domshlak (2009)]



$\rightarrow h(I) = 6$

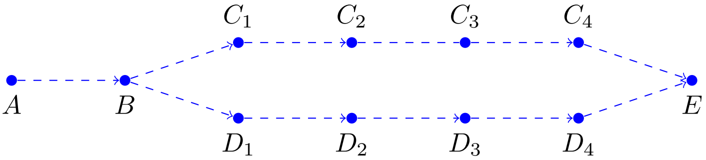
Many Disjunctive Action Landmarks!

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



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

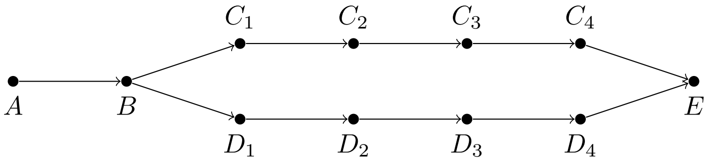
And now, let's pass Mars: LM-cut! [Helmert and Domshlak (2009)]



$\rightarrow h(I) = 6$

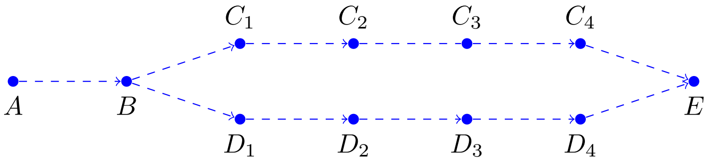
Many Disjunctive Action Landmarks!

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



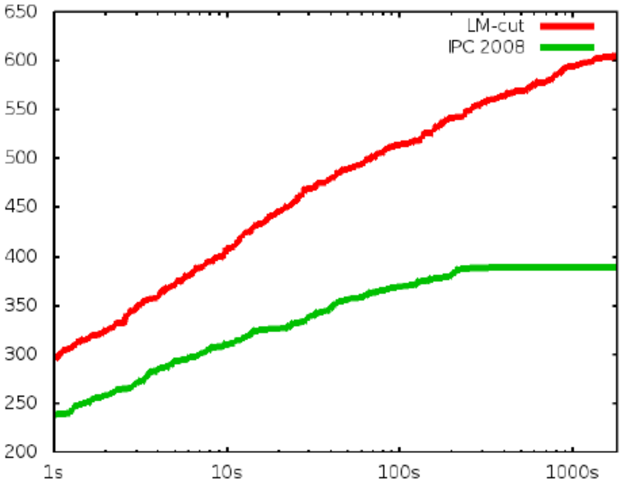
Fact landmarks: $\{B,E\}$, yielding $h(I)=2$.

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



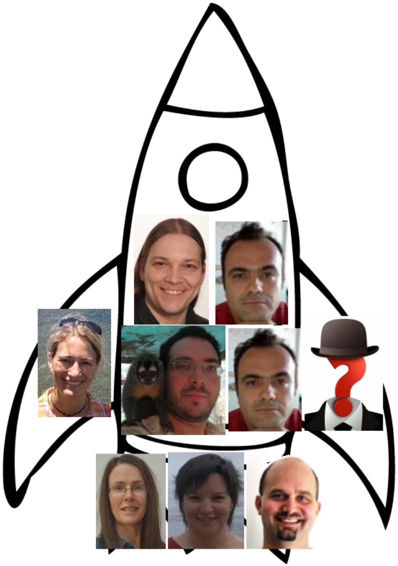
$\rightarrow h(I) = 6 = h^*(I)$.

The Impact of Stage 3

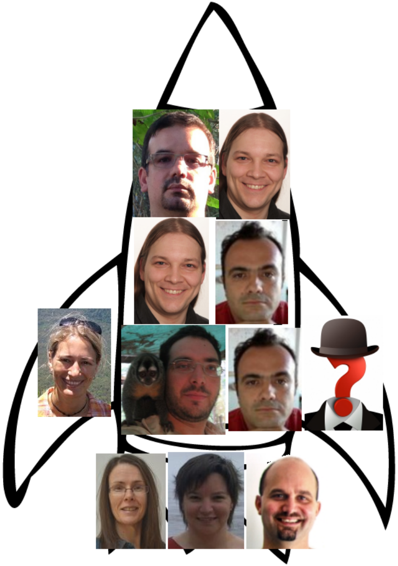


IPC 2008: Best optimal planner in the competition.

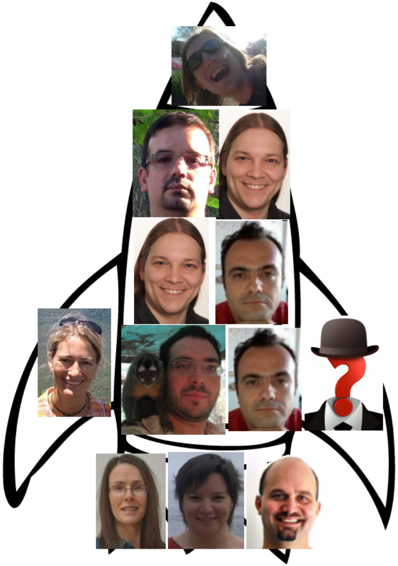
Agenda: Stage 4 (Off to the Milky Way!!)



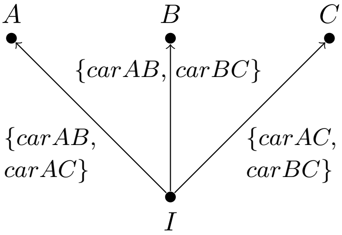
Agenda: Stage 4 (Off to the Milky Way!!)



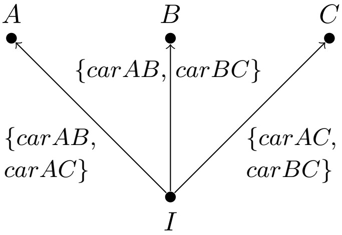
Agenda: Stage 4 (Off to the Milky Way!!)



Hitting Sets Over Landmarks!



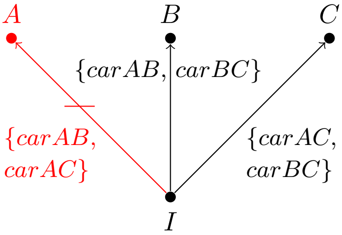
Hitting Sets Over Landmarks!



Precondition-Choice Functions

Landmarks:

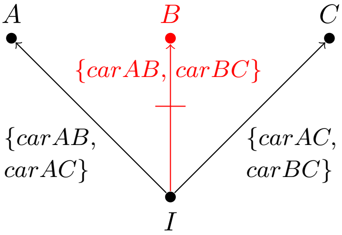
Hitting Sets Over Landmarks!



Precondition-Choice Functions

Landmarks: $\{carAB, carAC\}$

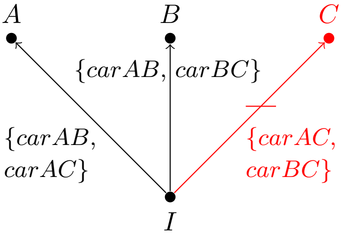
Hitting Sets Over Landmarks!



Precondition-Choice Functions

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

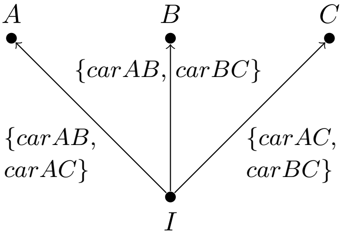
Hitting Sets Over Landmarks!



Precondition-Choice Functions

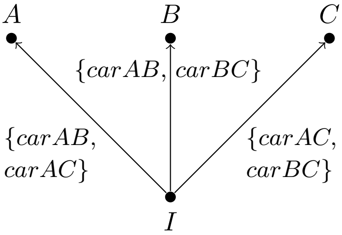
Landmarks: $\{carAB, carAC\}$, $\{carAB, carBC\}$, $\{carAC, carBC\}$

Hitting Sets Over Landmarks!



Landmarks: $\{carAB, carAC\}$, $\{carAB, carBC\}$, $\{carAC, carBC\}$. (Action costs: Uniform 1.)

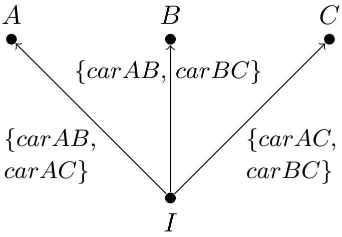
Hitting Sets Over Landmarks!



Landmarks: $\{carAB, carAC\}$, $\{carAB, carBC\}$, $\{carAC, carBC\}$. (Action costs: Uniform 1.)

Optimal cost partitioning: $h(I) = 1.5 < h^*(I)$: Set $h_{L_A} = h_{L_B} = h_{L_C} = 0.5$.

Hitting Sets Over Landmarks!

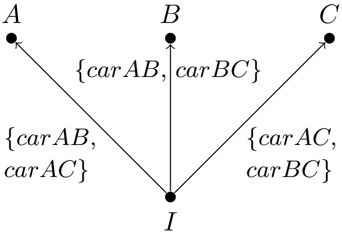


Landmarks: $\{carAB, carAC\}$, $\{carAB, carBC\}$, $\{carAC, carBC\}$. (Action costs: Uniform 1.)

Optimal cost partitioning: $h(I) = 1.5 < h^*(I)$: Set $h_{L_A} = h_{L_B} = h_{L_C} = 0.5$.

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

Hitting Sets Over Landmarks!



Landmarks: $\{carAB, carAC\}$, $\{carAB, carBC\}$, $\{carAC, carBC\}$. (Action costs: Uniform 1.)

Optimal cost partitioning: $h(I) = 1.5 < h^*(I)$: Set $h_{L_A} = h_{L_B} = h_{L_C} = 0.5$.

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

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

Proof. Any relaxed plan must hit $L_1, \dots, L_n$ so $\sum_{a \in H} cost(a) \leq h^+(s)$.

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

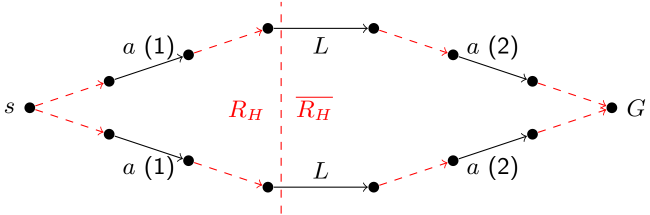
Theorem. *Let s be a state, and let $L_1, \dots, L_n$ be the collection of disjunctive action landmarks for s resulting from all precondition-choice functions and cuts. Let H be a minimum-cost hitting set. Then $\sum_{a \in H} cost(a) = h^+(s)$.*

Proof. Any relaxed plan must hit $L_1, \dots, L_n$ so $\sum_{a \in H} cost(a) \leq h^+(s)$. 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 $L_1, \dots, L_n$ 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 $\sum_{a \in H} cost(a) = h^+(s)$.*

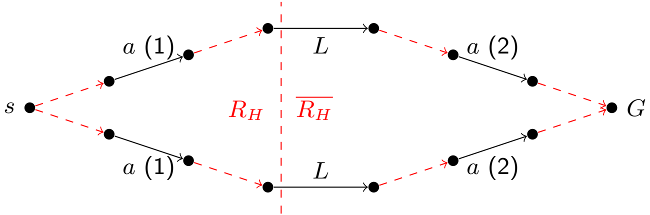
Proof. Any relaxed plan must hit $L_1, \dots, L_n$ so $\sum_{a \in H} cost(a) \leq h^+(s)$. We now prove that any hitting set H contains a relaxed plan. With $R_H := \{p \mid p$ *can be reached in delete-relaxation using only* $H\}$, assume to the contrary that $G \not\subseteq R_H$. Consider the cut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



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

Theorem. *Let s be a state, and let $L_1, \dots, L_n$ 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 $\sum_{a \in H} cost(a) = h^+(s)$.*

Proof. Any relaxed plan must hit $L_1, \dots, L_n$ so $\sum_{a \in H} cost(a) \leq h^+(s)$. 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 $R_H, \overline{R_H}$:

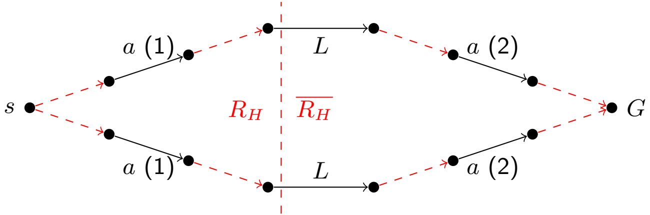


Case (1): If $pre_a \subseteq R_H$ then $add_a \subseteq R_H$ so $a \notin L$.

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

Theorem. *Let s be a state, and let $L_1, \dots, L_n$ 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 $\sum_{a \in H} cost(a) = h^+(s)$.*

Proof. Any relaxed plan must hit $L_1, \dots, L_n$ so $\sum_{a \in H} cost(a) \leq h^+(s)$. 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 $R_H, \overline{R_H}$:



Case (1): If $pre_a \subseteq R_H$ then $add_a \subseteq R_H$ so $a \notin L$.
Case (2): If $pre_a \not\subseteq R_H$ then our precondition-choice function can select $p \in pre_a \setminus R_H$ so, again, $a \notin L$. 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 Π^C, which converges to h^* [Haslum *et al.* (2012)].

Last Slide

And now: No questions. Off to dinner!

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

References I

Blai Bonet and Julio Castillo. A complete algorithm for generating landmarks. In Fahiem Bacchus, Carmel Domshlak, Stefan Edelkamp, and Malte Helmert, editors, *Proceedings of the 21st International Conference on Automated Planning and Scheduling (ICAPS 2011)*. AAAI Press, 2011.

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, John Slaney, and Sylvie Thiébaux. Minimal landmarks for optimal delete-free planning. In Blai Bonet, Lee McCluskey, José Reinaldo Silva, and Brian Williams, editors, *Proceedings of the 22nd International Conference on Automated Planning and Scheduling (ICAPS 2012)*, pages 353–357. AAAI Press, 2012.

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.

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.

Julie Porteous, Laura Sebastia, and Jörg Hoffmann. On the extraction, ordering, and usage of landmarks in planning. In A. Cesta and D. Borrajo, editors, *Recent Advances in AI Planning. 6th European Conference on Planning (ECP-01)*, Lecture Notes in Artificial Intelligence, pages 37–48, Toledo, Spain, September 2001. Springer-Verlag.

Silvia Richter and Matthias Westphal. The LAMA planner: Guiding cost-based anytime planning with landmarks. *Journal of Artificial Intelligence Research*, 39:127–177, 2010.

Silvia Richter, Malte Helmert, and Matthias Westphal. Landmarks revisited. In Dieter Fox and Carla Gomes, editors, *Proceedings of the 23rd National Conference of the American Association for Artificial Intelligence (AAAI-08)*, pages 975–982, Chicago, Illinois, USA, July 2008. AAAI Press.

References III

Simon Vernhes, Guillaume Infantes, and Vincent Vidal. Problem splitting using heuristic search in landmark orderings. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI-2013)*, Beijing, China, August 2013. AAAI Press.

Lin Zhu and Robert Givan. Landmark extraction via planning graph propagation. In *ICAPS 2003 Doctoral Consortium*, pages 156–160, 2003.