AI Planning

14. Landmark Heuristics

It's a Long Way to the Goal, But How Long Exactly?
Part IV: *Ticking Off the Items On a To-Do List*

Álvaro Torralba, Cosmina Croitoru

SAARLAND UNIVERSITY
COMPUTER SCIENCE

Winter Term 2018/2019

Thanks to Prof. Jörg Hoffmann for slide sources

Agenda

We Need Heuristic Functions!

→ Landmarks (LMs) are a method to relax planning tasks, and thus automatically compute heuristic functions h.

We cover the 4 different methods currently known:

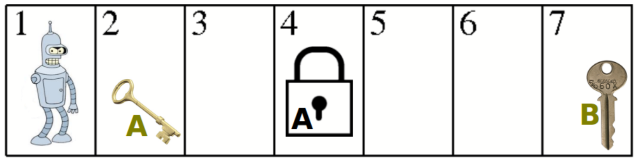
- Critical path heuristics: Done. → **Chapter 8**
- Delete relaxation: Basically done. → **Chapters 9 and 10**
- Abstractions: Done. → **Chapters 11–13**
- Landmarks. → **This Chapter**

→ Each of these have advantages and disadvantages. (We will do a formal comparison in **Chapter 17**.)

→ LM heuristics research yielded lots of exciting results since 2009. They boost the performance of satisficing planning when combined with delete relaxation heuristics, *and* they are among the most successful methods for computing lower-bound estimators.

Landmarks in a Nutshell

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 something that every plan for the task must satisfy at some point.

- Find landmarks in a pre-process to planning.
- Heuristic value(state) := number of yet un-achieved landmarks. ("Number of open items on the to-do list")

Before We Begin

- Landmarks were originally introduced as a method for problem decomposition [Hoffmann *et al.* (2004)].
- They traditionally come with a colorful variety of concepts defining orderings between them.
- Here we only discuss the generation of heuristic functions.
- We consider only the two most canonical forms of landmarks, and we do not cover LM orderings at all.
- Traditionally, LMs are mostly formulated in STRIPS; we'll do FDR (it doesn't really make a difference here). Remember that "facts" p in FDR are variable/value pairs.

Our Agenda for This Chapter

2. **Landmarks:** We start by defining the two forms of landmarks we will consider, and we discuss their connections and differences.
3. **Landmark Heuristics:** We specify how to turn landmarks (assuming they are provided as input) into heuristic functions. We introduce a notion of orthogonality which implies additivity.
4. **Detecting Landmarks:** We state that, in general, detecting landmarks is computationally hard, and we introduce and discuss the most commonly used approximation methods.

Fact Landmarks

"Something that every plan must satisfy at some point." **Take 1:**

Definition (Fact Landmark). *Let $\Pi = (V, A, c, I, G)$ be an FDR planning task, and let s be a state. A fact p is a fact landmark for s if $p \notin s$, and for every plan $\langle a_1, \dots, a_n \rangle$ for s, there exists t so that $p \in s[\![\langle a_1, \dots, a_t \rangle]\!]$.*

→ A fact landmark is a variable value that is currently false, but that must become true at some point along every plan.

→ We'll often use "LM" for "Landmark".

→ Any spontaneous ideas for facts that will always be landmarks? E.g., every goal fact that is not currently true.

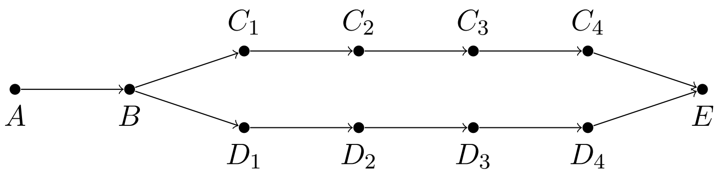
Bartender Fact Landmarks

- V: $Glass_1, Glass_2 : \{Table, Hand\}$; $Empty_1, Empty_2 : \{0, 1\}$; $Wodka, Tomato : \{Bottle, Glass_1, Glass_2, Shaker\}$; $BloodyMary : \{0, 1\}$.
- Initial state I: $Glass_1 = Table$, $Glass_2 = Table$, $Empty_1 = 1$, $Empty_2 = 1$, $Wodka = Bottle$, $Tomato = Bottle$, $BloodyMary = 0$.
- Goal G: $BloodyMary = 1$.
- Actions A: (unit costs)
  - $Take(x)$: pre $Glass_x = Table$; eff $Glass_x = Hand$
  - $Drop(x)$: pre $Glass_x = Hand$; eff $Glass_x = Table$
  - $Fill(x, y)$: pre $Glass_x = Hand$, $Empty_x = 1$, $y = Bottle$; eff $y = Glass_x$
  - $Pour(x, y)$: pre $Glass_x = Hand$, $y = Glass_x$; eff $y = Shaker$, $Empty_x = 1$
  - $Shake()$: pre $Wodka = Shaker$, $Tomato = Shaker$; eff $BloodyMary = 1$

→ What are the fact landmarks for the initial state? $BloodyMary = 1$, $Wodka = Shaker$, $Tomato = Shaker$. Everything else is not a landmark because we can use either glass 1 or glass 2.

Where Fact Landmarks Fail

FindPath example: Actions $move(X,Y)$ pre X eff Y; init A, goal E.



→ Fact LMs for I? B, E.

To the rescue: disjunctive landmarks!

Disjunctive Action Landmarks

"Something that every plan must satisfy at some point." **Take 2:**

Definition (Disjunctive Action Landmark). *Let $\Pi = (V, A, c, I, G)$ be an FDR planning task, and let s be a state. A set $L \subseteq A$ is a disjunctive action landmark for s if every plan for s contains an action $a \in L$. L is minimal if there exists no $L' \subsetneq L$ that is a disjunctive action landmark for s.*

→ A disjunctive action LM is a set of actions at least one of which must occur in every plan. The LM is minimal if it contains no unnecessary actions.

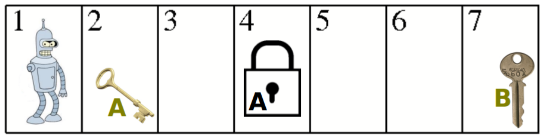
Terminology: The action set induced by a fact p is $L(p) := \{a \in A \mid p \in eff_a\}$.

Proposition (Fact LMs Induce Disjunctive Action LMs). *Let Π be an FDR planning task, let s be a state, and let p be a fact landmark for s. Then $L(p)$ is a disjunctive action landmark for s.*

Proof. Since p must become true at some point, it must be in an action effect.

→ Is $L(p)$ always minimal? No (e.g., 100 actions achieve p, but the preconditions of all but one of these actions are unreachable).

Questionnaire



Fact landmarks p**:** 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.

Actions: MoveXY (*pre* robot-at-X[, lock-open for $Y = 4$]; *eff* robot-at-Y); PickXY (*pre* robot-at-X, key-Y-at-X; *eff* have-key-Y); DropXY (*pre* robot-at-X, have-key-Y; *eff* key-Y-at-X); OpenLockX for $X \in \{3, 5\}$ (*pre* robot-at-X, have-key-A; *eff* lock-open).

Question!

How many of the 9 **fact landmarks** p **induce disjunctive action LMs** $L(p)$ **of size** $|L(p)| > 1$**? (And how many of the** $L(p)$ **with** $|L(p)| > 1$ **are minimal?)**

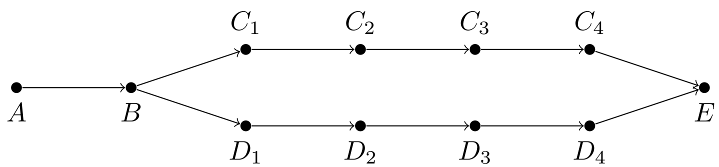
(A): 0 (B): 7

(C): 8 (D): 9

→ All these p, except robot-at-7, have more than one possible achieving action a (where $p \in eff_a$). Thus (C) is correct. None of the $L(p)$ with $|L(p)| > 1$ are minimal.

Induced vs. All Disjunctive Action Landmarks

FindPath example: Actions $move(X,Y)$ pre X eff Y; init A, goal E.



→ Fact LMs for I: B, E.

→ Disjunctive action LMs for I induced by these: $\{move(A,B)\}$, $\{move(C_4,E), move(D_4,E)\}$.

→ Minimal disjunctive action LMs for I **not** induced by these? $\{a_C, a_D\}$ for all $a_C \in \{move(B,C_1), move(C_i,C_{i+1}), move(C_4,E)\}$ and $a_D \in \{move(B,D_1), move(D_i,D_{i+1}), move(D_4,E)\}$, except $\{a_C, a_D\} = \{move(C_4,E), move(D_4,E)\}$.

→ *Some* disjunctive action LMs are induced by fact LMs; most of them aren't.

→ Note the difference in the possible numbers of fact/disjunctive action LMs.

Bartender Disjunctive Action Landmarks

- V: $Glass_1, Glass_2 : \{Table, Hand\}$; $Empty_1, Empty_2 : \{0,1\}$; $Wodka, Tomato : \{Bottle, Glass_1, Glass_2, Shaker\}$; $BloodyMary : \{0,1\}$.
- Initial state I: $Glass_1 = Table$, $Glass_2 = Table$, $Empty_1 = 1$, $Empty_2 = 1$, $Wodka = Bottle$, $Tomato = Bottle$, $BloodyMary = 0$.
- Goal G: $BloodyMary = 1$.
- Actions A: (unit costs)
  $Take(x)$: pre $Glass_x = Table$; eff $Glass_x = Hand$
  $Drop(x)$: pre $Glass_x = Hand$; eff $Glass_x = Table$
  $Fill(x,y)$: pre $Glass_x = Hand$, $Empty_x = 1$, $y = Bottle$; eff $y = Glass_x$
  $Pour(x,y)$: pre $Glass_x = Hand$, $y = Glass_x$; eff $y = Shaker$, $Empty_x = 1$
  $Shake()$: pre $Wodka = Shaker$, $Tomato = Shaker$; eff $BloodyMary = 1$

Fact landmarks p for I: $BloodyMary = 1$, $Wodka = Shaker$, $Tomato = Shaker$.

→ Disjunctive action landmarks $L(p)$ induced by these? $\{Shake()\}$, $\{Pour(1, Wodka), Pour(2, Wodka)\}$, $\{Pour(1, Tomato), Pour(2, Tomato)\}$.

→ Are these *all* disjunctive action landmarks for I? No. For example: $\{Fill(1, Wodka), Fill(2, Wodka)\}$, $\{Fill(1, Tomato), Fill(2, Tomato)\}$, and $\{Take(1), Take(2)\}$.

Elementary Landmark Heuristics

Definition (Elementary Landmark Heuristic). *Let $\Pi = (V, A, c, I, G)$ be an FDR planning task with state space $\Theta_\Pi = (S, A, T, I, G)$, and let $L \subseteq A$. The elementary landmark heuristic h_L^{LM} for Π given L is the function $h_L^{\mathsf{LM}} : S \mapsto \mathbb{R}_0^+$ where $h_L^{\mathsf{LM}}(s) = \min \{c(a) \mid a \in L\}$ if L is a disjunctive action landmark for s, and $h_L^{\mathsf{LM}}(s) = 0$ otherwise.*

→ If L is indeed a landmark, the elementary landmark heuristic given L returns the cost of the cheapest action in L; otherwise, it returns 0.

Remarks:

- h_L^{LM} is just a formal vehicle to elegantly express the goal distance estimates derived from LMs in terms of the heuristic functions framework.
- It has to be "min" over L, not "max" or "sum": intended meaning of L is that the planner may choose which action to use. Neither sum'ing nor max'ing would be admissible.
- If L is induced by a fact landmark p, this just means to "account for the cheapest action that achieves p".

Elementary Landmark Heuristics are Admissible

Theorem (h^{LM} is Admissible). *Let $\Pi = (V, A, c, I, G)$ be an FDR planning task, and let $L \subseteq A$. Then h_L^{LM} is consistent and goal-aware, and thus also admissible and safe.*

Proof. Goal-awareness: If s is a goal state, then L is not a landmark for s (the empty plan does not use any action from L), hence $h_L^{\mathsf{LM}}(s) = 0$.

Consistency: Say $s[\![a]\!] = s'$; we need to prove that $h_L^{\mathsf{LM}}(s) \leq h_L^{\mathsf{LM}}(s') + c(a)$.

If $h_L^{\mathsf{LM}}(s) = 0$, that is trivial. Else, L is a landmark for s and $h_L^{\mathsf{LM}}(s) = \min \{c(a) \mid a \in L\} =: c_{min}$, so we need to show that $c_{min} \leq h_L^{\mathsf{LM}}(s') + c(a)$.

Say $a \notin L$. Then L is a landmark for s' so $h_L^{\mathsf{LM}}(s') = c_{min} = h_L^{\mathsf{LM}}(s)$ and we are done.

Say $a \in L$. Then, by the definition of c_{min}, we have $c_{min} \leq c(a)$. So $c_{min} \leq h_L^{\mathsf{LM}}(s') + c(a)$ holds simply because $0 \leq h_L^{\mathsf{LM}}(s')$.

Questionnaire

Question!

Say s is a dead-end state. What are the (a) fact landmarks and (b) disjunctive action landmarks for s?

→ (a) All facts except those true in s, (b) all action subsets $L \subseteq A$. The reason for both is that, in our definitions, "every plan" quantifies over the empty set.

→ A "to-do list" does not make sense for unsolvable problems.

Question!

Say s is a dead-end state. Can $h_L^{\mathsf{LM}}(s)$ return ∞?

→ Only for $L = \emptyset$, where $\min \{c(a) \mid a \in L\} = \infty$ by convention. For $L \neq \emptyset$, h_L^{LM} returns either 0 or $\min \{c(a) \mid a \in L\}$, both of which are finite.

→ Practical LM heuristics don't find empty LMs. Hence (in difference to all other heuristic functions we consider) they cannot detect dead ends.

Bartender h^{LM}

- V: $Glass_1, Glass_2 : \{Table, Hand\}$; $Empty_1, Empty_2 : \{0, 1\}$; $Wodka, Tomato : \{Bottle, Glass_1, Glass_2, Shaker\}$; $BloodyMary : \{0, 1\}$.
- Initial state I: $Glass_1 = Table$, $Glass_2 = Table$, $Empty_1 = 1$, $Empty_2 = 1$, $Wodka = Bottle$, $Tomato = Bottle$, $BloodyMary = 0$.
- Goal G: $BloodyMary = 1$.
- Actions A: (unit costs)
  - $Take(x)$: pre $Glass_x = Table$; eff $Glass_x = Hand$
  - $Drop(x)$: pre $Glass_x = Hand$; eff $Glass_x = Table$
  - $Fill(x, y)$: pre $Glass_x = Hand$, $Empty_x = 1$, $y = Bottle$; eff $y = Glass_x$
  - $Pour(x, y)$: pre $Glass_x = Hand$, $y = Glass_x$; eff $y = Shaker$, $Empty_x = 1$
  - $Shake()$: pre $Wodka = Shaker$, $Tomato = Shaker$; eff $BloodyMary = 1$

Induced by fact LMs: $\{Shake()\}$, $\{Pour(1, Wodka), Pour(2, Wodka)\}$, $\{Pour(1, Tomato), Pour(2, Tomato)\}$.

→ h_L^{LM} from this: $h = 1$; $h = 1$; $h = 1$.

Additional disjunctive action landmarks: $\{Fill(1, Wodka), Fill(2, Wodka)\}$, $\{Fill(1, Tomato), Fill(2, Tomato)\}$, and $\{Take(1), Take(2)\}$.

→ h_L^{LM} from this: $h = 1$; $h = 1$; $h = 1$.

And Now?

Question!

Is h_L^{LM} a high-quality heuristic function?

(A): Yes. (B): No.

→ Its value is bounded by the cost of the most expensive action in the task! For unit costs: $h_L^{\mathsf{LM}}(s) \in \{0, 1\}$ ... !

→ For h_L^{LM} to be useful, we need to *combine* several of them!

How to admissibly combine $h_{L_1}^{\mathsf{LM}}, \dots, h_{L_k}^{\mathsf{LM}}$?

- max: Works trivially. Also trivially, problem above not solved.
- $\sum$: Can solve above problem, and is a sine-qua-non for LM heuristics (corresponds to "LM counting" in the case of fact LMs).
  → Admissible in general? No, because the same action may be part of more than one disjunctive action LM. See next slide.

When LM Summing (and Counting) is Not Admissble

Planning task:

- Goals G: A and B both true.
- Initial state I: A and B both false.
- Actions: $carA$ effect A cost 1; $carB$ effect B cost 1; $fancyCar$ effect A and B cost 1.5.

- Fact landmarks: A and B.
- Induced disjunctive action landmarks: $\{carA, fancyCar\}$ and $\{carB, fancyCar\}$.
- Summed-up heuristic value: $2 > h^*(I) = 1.5$.

Orthogonal Landmarks

Terminology. $L_1, \dots, L_k \subseteq A$ are orthogonal if $L_i \cap L_j = \emptyset$ for $i \neq j$.

Theorem (The Sum of Orthogonal h^{LM} is Admissible). *Let $\Pi = (V, A, c, I, G)$ be an FDR planning task, and let $L_1, \dots, L_k \subseteq A$ be orthogonal. Then $\sum_{i=1}^{k} h_{L_i}^{\mathsf{LM}}$ is consistent and goal-aware, and thus also admissible and safe.*

Proof. Goal-awareness: Trivial because all component heuristics are goal-aware.

Consistency: Say $s[\![a]\!] = s'$; we need to prove that $\sum_{i=1}^{k} h_{L_i}^{\mathsf{LM}}(s) \leq \sum_{i=1}^{k} h_{L_i}^{\mathsf{LM}}(s') + c(a)$. If any L_i is not a landmark for s, then L_i cannot contribute to disvalidating this inequality, so we can ignore that case.

Say a is not a member of any L_i. Then all L_i still are landmarks for s' so $\sum_{i=1}^{k} h_{L_i}^{\mathsf{LM}}(s) = \sum_{i=1}^{k} h_{L_i}^{\mathsf{LM}}(s')$ and we are done.

Else, a is a member of *exactly one* action set, say of L_j. For all $i \neq j$, L_i still is a landmark in s' so $h_{L_i}^{\mathsf{LM}}(s) = h_{L_i}^{\mathsf{LM}}(s')$.

It remains to show that $h_{L_j}^{\mathsf{LM}}(s) \leq h_{L_j}^{\mathsf{LM}}(s') + c(a)$, which we have by consistency of $h_{L_j}^{\mathsf{LM}}$ (cf. slide 18).

The Canonical Landmark Heuristic

Terminology. The compatibility graph for $\mathcal{C} = \{L_1, \dots, L_n\}$ has vertices L_i and an arc (L_i, L_j) iff $L_i \cap L_j = \emptyset$.

Definition (Canonical Heuristic). *Let Π be an FDR planning task, let $\mathcal{C} = \{L_1, \dots, L_n\}$ be a collection of action subsets, and let $cliques(\mathcal{C})$ be the set of all maximal cliques in the compatibility graph for $\mathcal{C}$. Then the canonical heuristic $h^{\mathcal{C}}$ for $\mathcal{C}$ is defined as $h^{\mathcal{C}}(s) = \max_{\mathcal{D} \in cliques(\mathcal{C})} \sum_{L_i \in \mathcal{D}} h^{\mathsf{LM}}_{L_i}(s)$.*

→ The canonical heuristic maximizes over all largest orthogonal subsets of our landmarks collection.

Remarks:

- To reduce overlaps, minimal disjunctive action LMs are desirable.
- $h^{\mathcal{C}}$ is the best possible admissible heuristic we can derive from $\mathcal{C}$ *using the orthogonality criterion*. Despite this, on slide 22, we get $h^{\mathcal{C}} = 1$.
- Better heuristics can be obtained using cost partitioning or hitting sets (→ Chapter 15).

Bartender Orthogonal Landmarks

- V: $Glass_1, Glass_2 : \{Table, Hand\}$; $Empty_1, Empty_2 : \{0,1\}$; $Wodka, Tomato : \{Bottle, Glass_1, Glass_2, Shaker\}$; $BloodyMary : \{0,1\}$.
- Initial state I: $Glass_1 = Table$, $Glass_2 = Table$, $Empty_1 = 1$, $Empty_2 = 1$, $Wodka = Bottle$, $Tomato = Bottle$, $BloodyMary = 0$.
- Goal G: $BloodyMary = 1$.
- Actions A: (unit costs)
  - $Take(x)$: pre $Glass_x = Table$; eff $Glass_x = Hand$
  - $Drop(x)$: pre $Glass_x = Hand$; eff $Glass_x = Table$
  - $Fill(x,y)$: pre $Glass_x = Hand$, $Empty_x = 1$, $y = Bottle$; eff $y = Glass_x$
  - $Pour(x,y)$: pre $Glass_x = Hand$, $y = Glass_x$; eff $y = Shaker$, $Empty_x = 1$
  - $Shake()$: pre $Wodka = Shaker$, $Tomato = Shaker$; eff $BloodyMary = 1$

Induced by fact LMs: $\{Shake()\}$, $\{Pour(1, Wodka), Pour(2, Wodka)\}$, $\{Pour(1, Tomato), Pour(2, Tomato)\}$.
Additional disjunctive action LMs: $\{Fill(1, Wodka), Fill(2, Wodka)\}$, $\{Fill(1, Tomato), Fill(2, Tomato)\}$, $\{Take(1), Take(2)\}$.

→ Canonical heuristic $h^{\mathcal{C}}(I)$ from these: These L_i are all orthogonal, hence $h^{\mathcal{C}}(I) = 6 = h^*(I)$.

Questionnaire

- Variables: $at : \{Sy, Ad, Br, Pe, Da\}$; $v(x) : \{T, F\}$ for $x \in \{Sy, Ad, Br, Pe, Da\}$.
- Actions: $drive(x,y)$ where x, y have a road.
- Costs: $Sy \leftrightarrow Br : 1$, $Sy \leftrightarrow Ad : 1.5$, $Ad \leftrightarrow Pe : 3.5$, $Ad \leftrightarrow Da : 4$.
- Initial state: $at = Sy, v(Sy) = T, v(x) = F$ for $x \neq Sy$.
- Goal: $at = Sy, v(x) = T$ for all x.

Induced by fact LMs: $\{drive(Ad, Pe)\}$, $\{drive(Ad, Da)\}$, $\{drive(Sy, Br)\}$, $\{drive(Pe, Ad), drive(Da, Ad), drive(Sy, Ad)\}$.

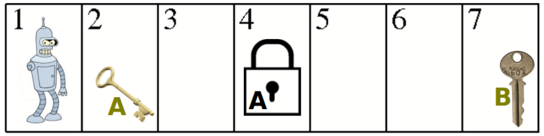
Additional disjunctive action LMs: $\{drive(Ad, Sy), drive(Br, Sy)\}$; $\{drive(Pe, Ad)\}$, $\{drive(Da, Ad)\}$, $\{drive(Sy, Ad)\}$.

Question!

Canonical heuristic $h^{\mathcal{C}}(I)$ from these?

→ The *non-minimal* LM $\{drive(Pe, Ad), drive(Da, Ad), drive(Sy, Ad)\}$ is subsumed by the three orthogonal LMs $\{drive(Pe, Ad)\}$, $\{drive(Da, Ad)\}$, $\{drive(Sy, Ad)\}$. We get $h^{\mathcal{C}}(I) = 3.5[Ad \to Pe] + 4[Ad \to Da] + 1[Sy \to Br] + 1[Br \to Sy] + 3.5[Pe \to Ad] + 4[Da \to Ad] + 1.5[Sy \to Ad] = 18.5$.

Questionnaire, ctd.



Fact landmarks p: 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. (Unit-cost actions)

Actions: MoveXY (*pre* robot-at-X[, lock-open for $Y = 4$]; *eff* robot-at-Y); PickXY (*pre* robot-at-X, key-Y-at-X; *eff* have-key-Y); DropXY (*pre* robot-at-X, have-key-Y; *eff* key-Y-at-X); OpenLockX for $X \in \{3, 5\}$ (*pre* robot-at-X, have-key-A; *eff* lock-open).

Question!

Considering the collection of disjunctive action LMs $L(p)$ induced by these p, what is the value of the canonical heuristic $h^{\mathcal{C}}$?

(A): 6 (B): 7

(C): 8 (D): 9

→ All of these $L(p)$ are orthogonal, so (D) is correct.

Elementary Landmark Heuristics in Practice (Up Next!)

"$h^{\mathsf{LM}}_L(s) = \min\{c(a) \mid a \in L\}$ *if* L *is a disjunctive action landmark for* s, *and* $h^{\mathsf{LM}}_L(s) = 0$ *otherwise.*"

→ So will we keep L fixed, and check for every search state s whether or not it's a LM? No, because checking LMs is expensive. Instead, we design "landmark generation" algorithms, which guarantee to produce *only* LMs, but which do not guarantee to produce *all* LMs.

And then:

- (A) Offline generation, online update: Generate LMs $L_1, \dots, L_n$ for the initial state once before planning begins. Maintain flags throughout search to remember which ones have not been achieved yet.
- (B) Online generation: Generate LMs $L_1, \dots, L_n$ individually for each s.

But How to *Detect* those Landmarks in the First Place?

→ How to obtain a collection of disjunctive action landmarks?

Theorem (Checking Landmarks is Hard). *Let* $\Pi = (V, A, c, I, G)$ *be an FDR planning task, and let* s *be a state. It is* **PSPACE**-*complete to decide whether or not a fact* p *is a fact landmark for* s, *and it is* **PSPACE**-*complete to decide whether or not an action set* $L \subseteq A$ *is a disjunctive action landmark for* s.

Proof. By a reduction from PlanEx. Given the task $\Pi = (V, A, c, I, G)$ for which we need to decide PlanEx, we construct $\Pi' := (V \cup \{x\}, A \cup \{a_1, a_2\}, c', I \cup \{x = 0\}, G)$ by introducing a new variable x with domain $\{0, 1\}$ as well as two new actions a_1, a_2 of which a_1 sets x from 0 to 1, and a_2 has precondition $x = 1$ and effect G. (We obtain c' from c by assigning arbitrary costs to a_1, a_2.) Then $x = 1$ is a fact landmark for I iff Π is unsolvable, and $\{a_1\}$ is a disjunctive action landmark for I iff Π is unsolvable.

→ Something is a landmark if and only if disallowing it renders the task unsolvable. Thus, checking landmarks is as hard as deciding solvability.

Note: This theorem can be proved more easily using the situation where "every plan" quantifies over the empty set, cf. slide 19. I find the present proof more illustrative.

So is all lost?

→ How to obtain a collection of disjunctive action landmarks?

Answer: "It is **PSPACE**-complete to decide whether or not a fact p is a fact landmark for s, and it is **PSPACE**-complete to decide whether or not an action set $L \subseteq A$ is a disjunctive action landmark for s."

Question!

So is all lost?

(A): Yes. (B): No.

→ We approximate ... (business as usual). More precisely, we devise sound but incomplete methods, detecting *a subset of the actual landmarks*.

→ Ideas? Goals and necessary subgoals are landmarks.

Detecting *Some* LMs, Take 1: Necessary Subgoals

Definition (Necessary Subgoals). *Let* Π *be an FDR planning task, and let* s *be a state. A fact* p *is a necessary subgoal in* Π *for* s *if* $p \notin s$ *and either:*

- (i) $p \in G$; *or*
- (ii) *there exists a necessary subgoal* q *in* Π *for* s *so that* $p \in \bigcap_{a \in A, q \in \mathit{eff}_a} \mathit{pre}_a$.

→ Necessary subgoals are top-level goals plus shared preconditions.

("subgoal" here=singleton fact, not fact subset as for critical path heuristics.)

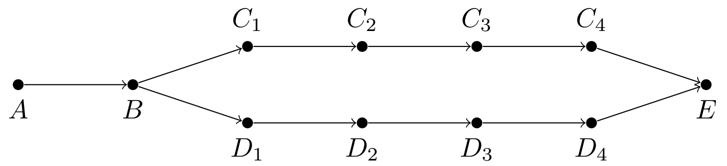
Proposition (Necessary Subgoals are Landmarks). *Let* Π *be an FDR planning task, and let* s *be a state. If* p *is a necessary subgoal in* Π *for* s, *then* p *is a fact landmark for* s.

Proof. By structural induction. The claim holds trivially for necessary subgoals of kind (i). For (ii), if q is a fact landmark for s, then q must be achieved at some point which by construction involves achieving p first.

Strategy: Given state s, detect necessary subgoals p_i for s by simple backchaining from the goal: start at $p \in G \setminus s$, then iteratively apply (ii) until no more new necessary subgoals are found.

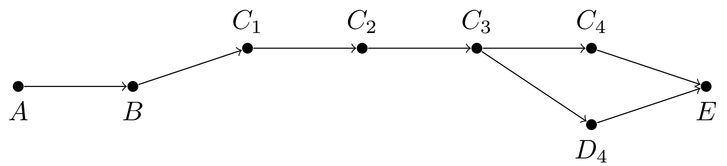
Necessary Subgoals vs. Fact Landmarks

FindPath example: Actions $move(X,Y)$ pre X eff Y; init A, goal E.



→ Fact landmarks for I? B, E.

→ Necessary subgoals for I? E.



→ Fact landmarks for I? B, C_1, C_2, C_3, E.

→ Necessary subgoals for I? E.

Bartender Necessary Subgoals

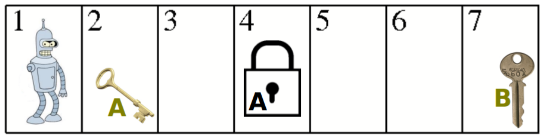
- V: $HandClean : \{0,1\}$, $Glass_1, Glass_2 : \{Table, Hand\}$; $Empty_1, Empty_2 : \{0,1\}$; $Wodka, Tomato : \{Bottle, Glass_1, Glass_2, Shaker\}$; $BloodyMary : \{0,1\}$.
- Initial state I: $HandClean = 0$, $Glass_1 = Table$, $Glass_2 = Table$, $Empty_1 = 1$, $Empty_2 = 1$, $Wodka = Bottle$, $Tomato = Bottle$, $BloodyMary = 0$.
- Goal G: $BloodyMary = 1$.
- Actions A: (unit costs)
  - $CleanHand()$: pre empty; eff $HandClean = 1$
  - $Take(x)$: pre $HandClean = 1$, $Glass_x = Table$; eff $Glass_x = Hand$
  - $Drop(x)$: pre $Glass_x = Hand$; eff $Glass_x = Table$
  - $Fill(x,y)$: pre $Glass_x = Hand$, $Empty_x = 1$, $y = Bottle$; eff $y = Glass_x$
  - $Pour(x,y)$: pre $Glass_x = Hand$, $y = Glass_x$; eff $y = Shaker$, $Empty_x = 1$
  - $Shake()$: pre $Wodka = Shaker$, $Tomato = Shaker$; eff $BloodyMary = 1$

→ Fact landmarks p for I: $BloodyMary = 1$, $Wodka = Shaker$, $Tomato = Shaker$, $HandClean = 1$.

→ Necessary subgoals for I? $BloodyMary = 1$: goal; $Wodka = Shaker$ and $Tomato = Shaker$: precondition for $Shake()$. The landmark $HandClean = 1$ is not detected as the backchaining stops at $Wodka = Shaker$ and $Tomato = Shaker$ which do not yield shared preconditions.

Questionnaire

Problem: Bring key B to position 1.



Actions: MoveXY (*pre* robot-at-X[, lock-open for $Y = 4$]; *eff* robot-at-Y); PickXY (*pre* robot-at-X, key-Y-at-X; *eff* have-key-Y); DropXY (*pre* robot-at-X, have-key-Y; *eff* key-Y-at-X); OpenLockX for $X \in \{3,5\}$ (*pre* robot-at-X, have-key-A; *eff* lock-open).

Question!

What are the necessary subgoals for I in this planning task?

→ Just key-B-at-1 and have-key-B: The only action that achieves the goal key-B-at-1 is to drop key-B at 1, which has preconditions robot-at-1 and have-key-B. The former is true in I so is not a necessary subgoal. The actions achieving have-key-B (all PickXB actions) do not share any precondition.

Detecting *Some* LMs, Take 2: Delete Relaxation LMs

Definition (Delete Relaxation LM). *Let $\Pi = (V, A, c, I, G)$ be an FDR planning task, and let s be a state. A fact p [respectively an action set $L \subseteq A$] is a delete relaxation landmark for s if $p \notin s$, and for every relaxed plan $\langle a_1^+, \dots, a_n^+ \rangle$ for s, there exists t so that $p \in s[\![\langle a_1^+, \dots, a_t^+ \rangle]\!]$ [respectively so that $a_t \in L$].*

Proposition (Checking Delete Relaxation LMs is Easy). *Let $\Pi = (V, A, c, I, G)$ be an FDR planning task, and let s be a state. It can be decided in polynomial time whether or not a fact p, respectively an action set L, is a delete relaxation landmark for s.*

Proof. To decide whether L is a delete relaxation landmark, simply test whether $(V, A \setminus L, c, I, G)$ does not have a relaxed plan.

To decide whether p is a fact landmark, simply test whether $p \notin s$ and Π does not have a relaxed plan when excluding all actions whose effect contains p.

→ Something is a landmark if and only if disallowing it renders the task unsolvable. For the delete relaxation, this can be checked in polynomial time.

Detecting Delete Relaxation LMs

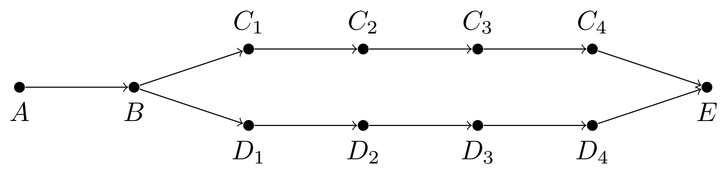
How to detect delete relaxation fact LMs?

- How to find all? For every fact, run test on previous slide.
- Not such a good idea in practice: Relaxed planning is polynomial time but not dirt-cheap, and there may be 100s–1000s of facts.
- A direct method computes all "causal" delete relaxation fact landmarks by a fixed point computation [Keyder *et al.* (2010)].

How to detect delete relaxation disjunctive action LMs?

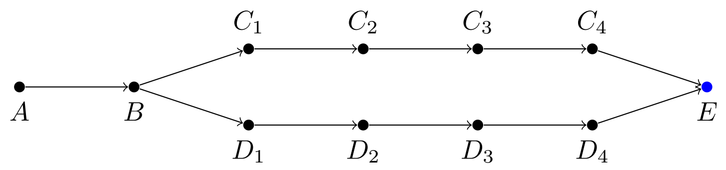
- How to find all? For every $L \subseteq A$, run test on previous slide.
- Completely useless idea in practice: Exponentially many L.
- **Vanilla solution:** Use $L(p)$ induced by delete relaxation fact LM p.
- **Advanced solution LM-cut:** [Helmert and Domshlak (2009)]
  Get L as a *cut* between the initial state and the "0-cost goal zone"; reduce the cost of each action in L by $\min_{a \in L} c(a)$; iterate.
  We'll give details in **Chapter 17**; illustration see next slide.

Detecting Action LMs: Fact-Induced vs. LM-cut



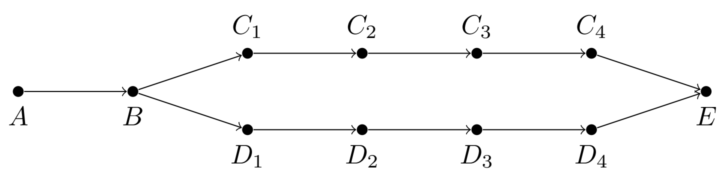
Fact-induced LMs: Fact LMs B and E yield $\{move(A, B)\}$ and $\{move(C_4, E), move(D_4, E)\}$. Thus $h^{\mathcal{C}}(I) = 2$.

LM-cut: (blue edges = cost reduced to 0, blue nodes = "0-cost goal zone")



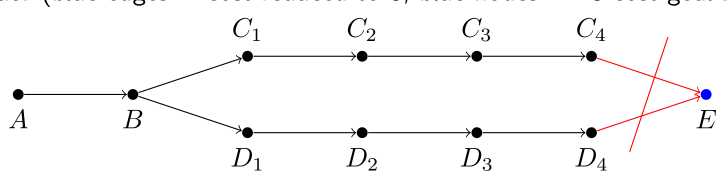
LM-cut LMs:

Detecting Action LMs: Fact-Induced vs. LM-cut



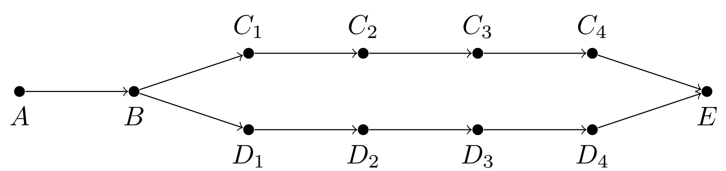
Fact-induced LMs: Fact LMs B and E yield $\{move(A, B)\}$ and $\{move(C_4, E), move(D_4, E)\}$. Thus $h^{\mathcal{C}}(I) = 2$.

LM-cut: (blue edges = cost reduced to 0, blue nodes = "0-cost goal zone")



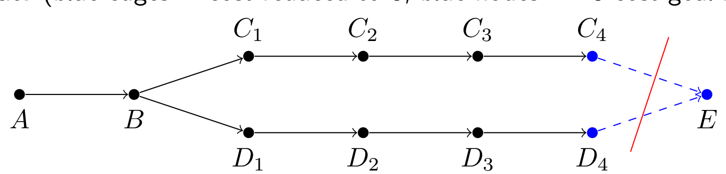
LM-cut LMs: $\{move(C_4, E), move(D_4, E)\}$

Detecting Action LMs: Fact-Induced vs. LM-cut



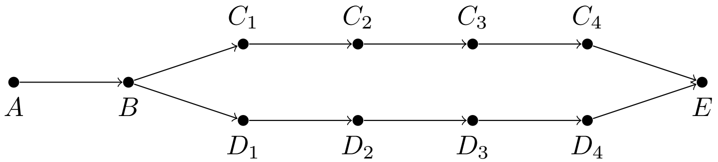
Fact-induced LMs: Fact LMs B and E yield $\{move(A, B)\}$ and $\{move(C_4, E), move(D_4, E)\}$. Thus $h^{\mathcal{C}}(I) = 2$.

LM-cut: (blue edges = cost reduced to 0, blue nodes = "0-cost goal zone")



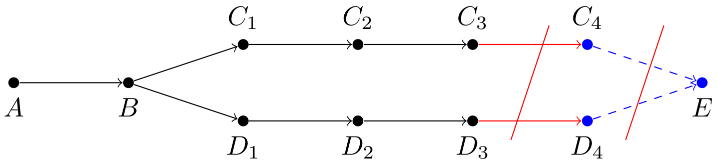
LM-cut LMs: $\{move(C_4, E), move(D_4, E)\}$

Detecting Action LMs: Fact-Induced vs. LM-cut



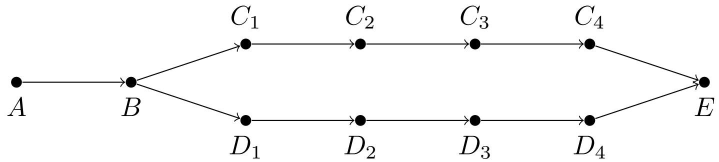
Fact-induced LMs: Fact LMs B and E yield $\{move(A,B)\}$ and $\{move(C_4,E), move(D_4,E)\}$. Thus $h^{\mathcal{C}}(I) = 2$.

LM-cut: (blue edges = cost reduced to 0, blue nodes = "0-cost goal zone")



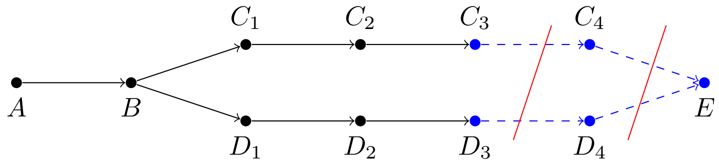
LM-cut LMs: $\{move(C_4,E), move(D_4,E)\}$, $\{move(C_3,C_4), move(D_3,D_4)\}$

Detecting Action LMs: Fact-Induced vs. LM-cut



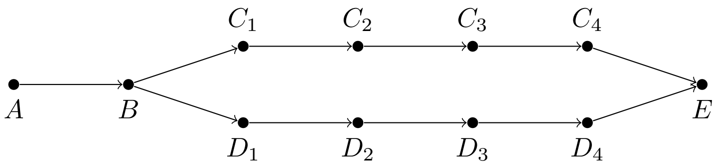
Fact-induced LMs: Fact LMs B and E yield $\{move(A,B)\}$ and $\{move(C_4,E), move(D_4,E)\}$. Thus $h^{\mathcal{C}}(I) = 2$.

LM-cut: (blue edges = cost reduced to 0, blue nodes = "0-cost goal zone")



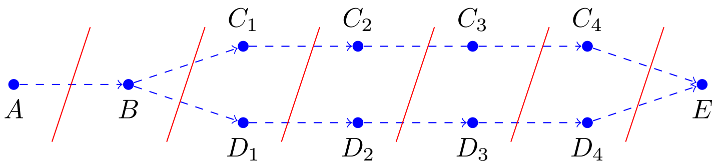
LM-cut LMs: $\{move(C_4,E), move(D_4,E)\}$, $\{move(C_3,C_4), move(D_3,D_4)\}$

Detecting Action LMs: Fact-Induced vs. LM-cut



Fact-induced LMs: Fact LMs B and E yield $\{move(A,B)\}$ and $\{move(C_4,E), move(D_4,E)\}$. Thus $h^{\mathcal{C}}(I) = 2$.

LM-cut: (blue edges = cost reduced to 0, blue nodes = "0-cost goal zone")



LM-cut LMs: $\{move(C_4,E), move(D_4,E)\}$, $\{move(C_3,C_4), move(D_3,D_4)\}$, $\{move(C_2,C_3), move(D_2,D_3)\}$, $\{move(C_1,C_2), move(D_1,D_2)\}$, $\{move(B,C_1), move(B,D_1)\}$, $\{move(A,B)\}$. Thus $h^{\mathcal{C}}(I) = 6 = h^*(I)$.

Propagating Landmarks

Remember? *"Heuristic value(state) := number of yet un-achieved landmarks (number of open items on the to-do list)."*

→ Here's how to "maintain the to-do list":

Proposition (Propagating Landmarks). *Let Π be an FDR planning task, let L be a disjunctive action LM for I, and let s be a state. If $s = I[\![\vec{a}]\!]$ where $\vec{a}$ does not use any action from L, then L is a disjunctive action LM for s.*

Strategy: Before search, detect disjunctive action landmarks for I. During forward search, maintain a flag for each L saying whether or not it was used yet. (For fact LMs p, the flag says whether p has already been true at some point.)

→ This is option (A) on slide 28. Re-computation for each s is option (B) on slide 28.

Delete Relaxation LMs: Properties

→ Necessary subgoals ⊆ delete relaxation landmarks ⊆ real landmarks.
→ Delete relaxation landmarks lower-bound h^+.

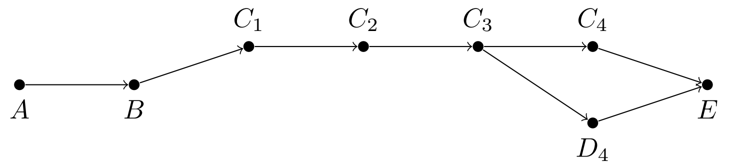
Precisely:

Proposition (Delete Relaxation LM Properties). *Let Π be an FDR planning task, and let s be a state. Then all of the following hold:*

1. *If p is a necessary subgoal for s, then p is a delete relaxation LM for s.*
2. *If p respectively L is a delete relaxation LM for s, then it is a LM for s.*
3. *If L is a delete relaxation LM for s, then $h_L^{\text{LM}}(s) \leq h^+(s)$.*

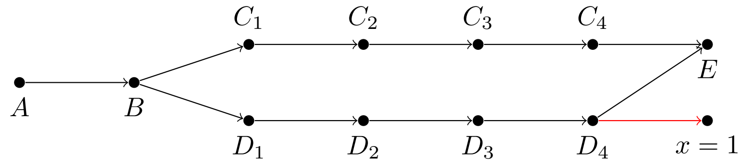
Proof. (i): Same argument as in the proof that p is a LM. (ii): Every real plan for s is also a relaxed plan for s, so must use p respectively L. (iii): Trivial.

Necessary Subgoals vs. Delete Relaxation LMs vs. LMs



Fact LMs for I: B, C_1, C_2, C_3, E. **Necessary subgoals for I:** E.
→ Delete relaxation fact LMs for I? B, C_1, C_2, C_3, E.

And now: Say init $A, (x = 1)$; goal $E, (x = 1)$; $move(D_4, E)$ sets $x := 0$.



→ Fact LMs for I? B, C_1, C_2, C_3, C_4, E.
→ Delete relaxation fact LMs for I? B, E.

Questionnaire

- $P = \{alive, haveTiger, tamedTiger, haveJump\}$.
  Short: $P = \{A, hT, tT, J\}$.
- Initial state I: $alive$. Goal G: $alive, haveJump$.
- Actions A:
  $getTiger$: pre $alive$; add $haveTiger$
  $tameTiger$: pre $alive$, $haveTiger$; add $tamedTiger$
  $jumpTamedTiger$: pre $alive$, $tamedTiger$; add $haveJump$
  $jumpTiger$: pre $alive$, $haveTiger$; add $haveJump$; del $alive$

Question!

What are, for the initial state in this example, the delete-relaxation fact landmarks? Are all fact landmarks delete-relaxation fact landmarks?

→ The fact landmarks for the inital state are $haveTiger$, $tamedTiger$, and $haveJump$: all except $alive$, because while this is a goal, it is already true in I.

→ The delete-relaxation fact landmarks are only $haveTiger$ and $haveJump$: $tamedTiger$ is not a delete-relaxation fact landmark because there exists a delete-relaxed (but not real) plan which does not tame the Tiger.

→ There are more delete-relaxed plans than real plans. Hence there are less things (landmarks) shared across all of them.

Bartender Delete Relaxation Landmarks

- V: $Glass_1, Glass_2 : \{Table, Hand\}$; $Empty_1, Empty_2 : \{0, 1\}$; $Wodka, Tomato : \{Bottle, Glass_1, Glass_2, Shaker\}$; $BloodyMary : \{0, 1\}$; $CustomerHappy : \{0, 1\}$.
- Initial state I: $Glass_1 = Table$, $Glass_2 = Table$, $Empty_1 = 1$, $Empty_2 = 1$, $Wodka = Bottle$, $Tomato = Bottle$, $BloodyMary = 0$, $CustomerHappy = 1$.
- Goal G: $BloodyMary = 1$, $CustomerHappy = 1$.
- Actions A: (unit costs)
  $Take(x)$: pre $Glass_x = Table$; eff $Glass_x = Hand$
  $Drop(x)$: pre $Glass_x = Hand$; eff $Glass_x = Table$
  $Fill(x, y)$: pre $Glass_x = Hand$, $Empty_x = 1$, $y = Bottle$; eff $y = Glass_x$
  $Pour(x, y)$: pre $Glass_x = Hand$, $y = Glass_x$; eff $y = Shaker$, $Empty_x = 1$
  $Shake()$: pre $Wodka = Shaker$, $Tomato = Shaker$; eff $BloodyMary = 1$
  $TakePreMix()$: pre empty; eff $BloodyMary = 1$, $CustomerHappy = 0$

→ Fact landmarks p for I: $BloodyMary = 1$, $Wodka = Shaker$, $Tomato = Shaker$.

→ Delete relaxation fact LMs for I? Only $BloodyMary = 1$. While the alternative plan $TakePreMix()$ does not work and hence does not affect the real fact landmarks, it does constitute an alternative relaxed plan and hence no more facts are shared across plans.

Questionnaire

- Actions: $drive(x, y)$ where x, y have a road.
- Costs: $Sy \leftrightarrow Br : 1$, $Sy \leftrightarrow Ad : 1.5$, $Ad \leftrightarrow Pe : 3.5$, $Ad \leftrightarrow Da : 4$.
- Initial state: $at = Sy, v(Sy) = T, v(x) = F$ for $x \neq Sy$.
- Goal: $at = Sy, v(x) = T$ for all x.

Question!

Minimal disjunctive action LMs for I, and $h^{\mathcal{C}}(I)$?

$\rightarrow \{drive(x, y)\}$ for *every* road (x, y) on the map, so $h^{\mathcal{C}}(I) = 20 = h^*(I)$.

Question!

Minimal delete relaxation disjunctive action LMs for I, and $h^{\mathcal{C}}(I)$?

$\rightarrow \{drive(x, y)\}$ for $(x, y) \in \{(Sy, Br), (Sy, Ad), (Ad, Pe), (Ad, Da)\}$; we do *not* get any of the "driving-back" actions. So $h^{\mathcal{C}}(I) = 10 = h^+(I)$.

Note: In both cases here, $h^{\mathcal{C}}(I)$ is "perfect". This is *not* in general so, even if we detect all disjunctive action LMs. E.g., on slide 22, the disjunctive action LMs are $\{carA, fancyCar\}$, $\{carB, fancyCar\}$, $\{carA, carB, fancyCar\}$; and $h^{\mathcal{C}}(I) = 1 < 1.5 = h^+(I) = h^*(I)$. We get back to this in the Next Chapter.

Summary

- A landmark (LM) is something that every plan must satisfy. A fact LM must hold at some point on every plan, a disjunctive action LM is a set of actions one of which must be used by every plan.
- Fact LMs induce disjunctive action LMs; however, most disjunctive action LMs are not induced in this way.
- The elementary LM heuristic returns the cost of the cheapest action in a disjunctive action LM.
- Disjunctive action LMs are orthogonal if they are disjoint. Orthogonal elementary LM heuristics are summed admissibly in the canonical heuristic.

  $\rightarrow$ Stronger methods are cost partitioning and (even stronger) hitting sets, to be considered in the Next Chapter.
- Checking LMs is hard. Practical methods are sound but incomplete, detecting some LMs, namely necessary subgoals or delete relaxation LMs.
- Vanilla method: Detect (some) fact LMs and use the induced disjunctive action LMs. Much stronger method LM-cut: Iteratively cut between the initial state and the "0-cost goal zone".

Remarks: LM Definitions

Historical:

- Landmarks were originally just fact landmarks, and were introduced as a means to *decompose* the task: Find LMs for I in a pre-process, feed them one-by-one to the planner [Hoffmann *et al.* (2004)].

Technical:

- Various kinds of *orderings* between landmarks are in use: "A must be achieved (directly) before B", "A should be achieved before B or else we would need to delete B and re-achieve it after A", ...
- Instead of just facts, we can use arbitrary propositional formulas ϕ over the facts (or even quantification over PDDL objects).
- If ϕ is a disjunction of facts, then that corresponds very closely to disjunctive action landmarks.
- I've chosen the two particular notions as presented because the "vanilla method" to compute landmark heuristics is by considering the disjunctive action landmarks induced by the fact landmarks.

Remarks: LM Heuristics

Historical:

- The idea to generate heuristics based on landmarks was first conceived by [Zhu and Givan (2003)], never properly published and forgotten all about.
- The (basic) idea was re-discovered by the authors of LAMA [Richter *et al.* (2008); Richter and Westphal (2010)]. Which subsequently won two IPCs.
- Both the initial attempt and LAMA use non-admissible landmarks heuristics, basically counting the number of non-achieved fact landmarks (= summing up elementary landmark heuristics induced by fact landmarks, without ensuring independence).

Technical: (We will consider this in detail in the Next Chapter)

- The best admissible landmark heuristics in practice use cost partitioning [Karpas and Domshlak (2009); Helmert and Domshlak (2009)].
- One can use hitting sets over landmarks to obtain even better heuristics, but these tend to be too costly computationally [Bonet and Helmert (2010)].

Remarks: Detecting LMs

- The original LMs detection method found delete relaxation fact LMs, mostly the necessary subgoals [Hoffmann *et al.* (2004)].
- LAMA does that, plus additional methods based on domain transition graphs (cf. **Chapter 5**); it propagates LMs for I to avoid having to re-detect [Richter and Westphal (2010)].
- The first admissible LM heuristic uses the disjunctive action LMs induced by LAMA's fact LMs [Karpas and Domshlak (2009)].
- The first technique using disjunctive action LMs *not* induced by fact LMs was LM-cut [Helmert and Domshlak (2009)]. The iterated cut algorithm is done anew for every search state. Despite this, LM-cut is the most successful admissible LM heuristic in practice, to date.

Remarks: Planning Tools and Performance Using LMs

- Original use for probem decomposition gave reasonable speed-ups for FF and another satisficing heuristic search planner [Hoffmann *et al.* (2004)].
- LAMA [Richter and Westphal (2010)] introduced the idea to use both, a delete relaxation heuristic and a LM heuristic, in Fast Downward's dual-queue greedy best-first search framework. The LM heuristic improves performance significantly in some domains. LAMA won the 1st prizes for satisficing planners at IPC'08 and IPC'11.
- BJOLP [Karpas and Domshlak (2009); Domshlak *et al.* (2011)] uses admissible combination of disjunctive action LMs induced by fact LMs. It was part of the 1st-prize winning portfolio in the optimal track of IPC'11.
- LM-cut [Helmert and Domshlak (2009)] also uses admissible combination of disjunctive action LMs, but of more general such LMs not induced by fact LMs (cf. slide 38). It was part of the 1st-prize winning portfolio, and of the 2nd-prize winning portfolio, in the optimal track of IPC'11. It was the strongest single-heuristic optimal planner in IPC'11.

Reading

- *Ordered Landmarks in Planning* [Hoffmann *et al.* (2004)].

  Available at:

  `http://fai.cs.uni-saarland.de/hoffmann/papers/jair04.pdf`

  Content: The first paper on landmarks. Focusses mostly on ordering relations and problem decomposition; not directly relevant to the content of this chapter, but useful as a background read.

- *Sound and Complete Landmarks for And/Or Graphs* [Keyder *et al.* (2010)].

  Available at:

  `http://www.dtic.upf.edu/~ekeyder/ECAI10_Landmarks.pdf`

  Content: A nice and clean "modern" paper on landmarks. Contains, among other things, the fixed point algorithm computing all (causal) delete relaxation fact landmarks.

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! Admissible combination by going from fact LMs to disjunctive action LMs, optimal cost partitioning by compilation to linear programming **(→ Chapters 15–16)**, LM-A^* to handle this path-dependent heuristic.

  Recapitulates LAMA's heuristic along the way so may be used to get an idea of that one as well.

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 LM-cut paper. As if LM-cut wasn't enough, it also introduces the comparison framework for admissible heuristics **(→ Chapter 17)**.

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: Introduces the idea to use minimum-cost hitting sets over disjunctive action landmarks, instead of combining elementary landmark heuristics. Shows that the minimum-cost hitting set over sufficiently large collections of delete relaxation disjunctive action landmarks is equal to h^+.

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.

Carmel Domshlak, Malte Helmert, Erez Karpas, Emil Keyder, Silvia Richter, Gabriele Röger, Jendrik Seipp, and Matthias Westphal. BJOLP: The big joint optimal landmarks planner. In *IPC 2011 planner abstracts*, pages 91–95, 2011.

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'09)*, pages 162–169. AAAI Press, 2009.

Jörg Hoffmann, Julie Porteous, and Laura Sebastia. Ordered landmarks in planning. *Journal of Artificial Intelligence Research*, 22:215–278, 2004.

References II

Erez Karpas and Carmel Domshlak. Cost-optimal planning with landmarks. In Craig Boutilier, editor, *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI'09)*, pages 1728–1733, Pasadena, California, USA, July 2009. Morgan Kaufmann.

Emil Keyder, Silvia Richter, and Malte Helmert. Sound and complete landmarks for and/or graphs. In Helder Coelho, Rudi Studer, and Michael Wooldridge, editors, *Proceedings of the 19th European Conference on Artificial Intelligence (ECAI'10)*, pages 335–340, Lisbon, Portugal, August 2010. IOS Press.

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.

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