AI Planning for Autonomy

7. 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*

Nir Lipovetzky

THE UNIVERSITY OF MELBOURNE

Winter Term 2016

Agenda

Motivation

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

→ Every h yields good performance only in some domains! (Search reduction vs. computational overhead, cf. → **lec. 3**.)

We cover 3 out of 4 different methods currently known:

- Critical path heuristics: Done. → **lec. 5**
- Delete relaxation: Done.→ **lec. 6**
- Abstractions:
- Landmarks.→ **Today**

→ LMs heuristics 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!

Motivation for This Particular Chapter Content

- Landmarks were originally introduced as a method for *problem decomposition* [(*Porteous, et al. ECP-01*) Influential paper award ICAPS-13].
- They traditionally come with a colorful variety of concepts defining *orderings* between them.

- Here we only discuss their use for the generation of heuristic functions . . .
- . . . and for simplicity we consider only the two most canonical forms of landmarks, and completely drop all this "orderings" business.

- Landmarks heuristics were a "hot topic", with lots of exciting results up to 2013. Still "hot", but slowly taken protagonism by Flow-based/Counting Heuristics, about to become "the hottest".
- To make the various connections easier to grasp, I summarize the literature "on the spot" (in each section) rather than only at the end of the chapter.

Intuition

"A landmark is something that every plan for the task must satisfy at some point."

Ideas?

Fact Landmarks

$\rightarrow$ *"Something that every plan must satisfy at some point."*

Take 1:

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

$\rightarrow$ A fact landmark is a variable value that must be true at some point along every plan.

$\rightarrow$ We'll often use "LM" for "Landmark".

$\rightarrow$ Any spontaneous ideas for facts that will always be landmarks?

Fact Landmarks in “TSP” in Australia

- Variables: $at : \{Sy, Ad, Br, Pe, Ad\}$;
  $v(x) : \{T, F\}$ for $x \in \{Sy, Ad, Br, Pe, Ad\}$.
- 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 $x \neq Ad$.

→ What are the fact landmarks for the initial state?

→ So which fact(s) are *not* fact landmarks for the initial state?

→ What are the fact landmarks for the state s where all cities have been visited already, and the truck is at Darwin?

Disjunctive Action Landmarks

$\rightarrow$ *"Something that every plan must satisfy at some point."*

Take 2:

Definition (Disjunctive Action Landmark). *Let $\Pi = (F, A, c, I, G)$ be an STRIPS 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 L is minimal if there exists no $L' \subset L$ that is a disjunctive action landmark for s.*

$\rightarrow$ 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 $L(p) := \{a \in A \mid p \in \mathit{eff}_a\}$.

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

Proof.

$\rightarrow$ Is $L(p)$ always minimal?

Disjunctive Action Landmarks in “TSP” in Australia

- Variables: $at : \{Sy, Ad, Br, Pe, Ad\}$; $v(x) : \{T, F\}$ for $x \in \{Sy, Ad, Br, Pe, Ad\}$.
- 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 $x \neq Ad$.

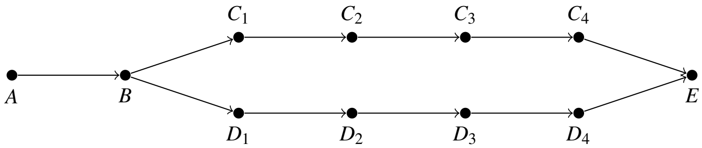
Fact landmarks p for I with $p \notin I$: $v(x) = T$ for $x \neq Ad, Sy$: goal; $at = x$ for all $x \neq Sy$: preconditions (and side effects) of necessary truck moves; and $v(Ad) = T$: side effect of driving to Adelaide.

→ Disjunctive action landmarks $L(p)$ induced by these?

→ Are these *all* disjunctive action landmarks for I?

(Induced) Disjunctive Action Landmarks

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



$\rightarrow$ Fact LMs for I with $p \notin I$?

$\rightarrow$ Disjunctive action LMs for I induced by these?

$\rightarrow$ Minimal disjunctive action LMs for I **not** induced by these?

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

$\rightarrow$ Note the difference in the possible numbers of fact/disjunctive action LMs!

Remarks

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

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 the plan would become longer", . . .
- 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 landmarks heuristics is by considering the disjunctive action landmarks induced by the fact landmarks.

Elementary Landmark Heuristics

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

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

→ If L is induced by a fact landmark p, this just means to “account for the cheapest way of achieving p”.

Why the $\min$ over L, not the $\max$ or the sum?

Elementary Landmark Heuristics in Practice

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

$\rightarrow$ So will we keep L fixed, and check for every search state s whether or not it's a LM? Could, but don't have to. Here's the **3 options:**

(A) Offline generation, online checking: Generate $L_1, \dots, L_n$ once before planning begins. For each search state s that comes up, check for each L_i whether it is a LM for s.

(B) Offline generation, online update: Generate LMs $L_1, \dots, L_n$ for the initial state once before planning begins. Maintain flags throughout search to determine which still are LMs for any search state s.

(C) Online generation: Generate $L_1, \dots, L_n$ individually for each s.

$\rightarrow$ Currently in use: (B) and (C), see next section. (A) is questionable in practice because checking LMs is expensive.

Elementary Landmark Heuristics are Admissible

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

Proof.

$\rightarrow$ Under what circumstances do we have $h^{\text{LM}}_L(s) = \infty$?

h^{LM} in "TSP" in Australia

- Variables: $at : \{Sy, Ad, Br, Pe, Ad\}$;
  $v(x) : \{T, F\}$ for $x \in \{Sy, Ad, Br, Pe, Ad\}$.
- 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 $x \neq Ad$.

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

$\rightarrow h_L^{\text{LM}}$ from this?

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

$\rightarrow h_L^{\text{LM}}$ from this?

And Now?

Question!

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

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

For elementary landmarks heuristics to be useful, we need to *combine* several of them!

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

- $\max$: Works trivially. Problem above solved?
- $\sum$: Admissible in general?

Independent Elementary Landmark Heuristics

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

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

$\rightarrow$ To reduce overlaps, minimal disjunctive action LMs are desirable.

Proof.

The Canonical Landmarks 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 STRIPS 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 $h^{\mathcal{C}}(s) = \max_{\mathcal{D} \in cliques(\mathcal{C})} \sum_{L_i \in \mathcal{D}} h^{\mathsf{LM}}_{L_i}(s)$.*

$\rightarrow$ The canonical heuristic maximizes over all largest independent subsets of our landmarks collection.

Remarks:

- $h^{\mathcal{C}}$ is *the best possible* admissible heuristic we can derive from $\mathcal{C}$ *using the independence criterion*!
- Even better heuristics can be obtained using cost partitioning or hitting sets (slide 24).

Independent h^{LM} in "TSP" in Australia

- Variables: $at : \{Sy, Ad, Br, Pe, Ad\}$; $v(x) : \{T, F\}$ for $x \in \{Sy, Ad, Br, Pe, Ad\}$.
- 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 $x \neq Ad$.

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

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

Independent h^{LM} in "TSP" in Australia, ctd.

- Variables: $at : \{Sy, Ad, Br, Pe, Ad\}$;
  $v(x) : \{T, F\}$ for $x \in \{Sy, Ad, Br, Pe, Ad\}$.
- 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 $x \neq Ad$.

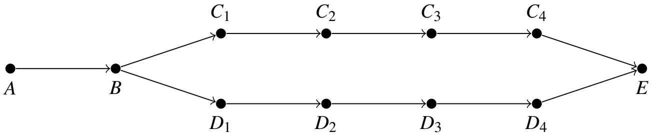
Induced by fact LMs: $\{drive(Ad, Pe)\}$, $\{drive(Ad, Da)\}$, $\{drive(Sy, Br)\}$, $\{drive(Pe, Ad), drive(Da, Ad), drive(Sy, Ad)\}$.
Additional disjunctive action LMs: $\{drive(Ad, Sy), drive(Br, Sy)\}$; $\{drive(Pe, Ad)\}$, $\{drive(Da, Ad)\}$, $\{drive(Sy, Ad)\}$.

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

Note: Every plan here must use every drive action at least once, so if we find *all* minimal disjunctive action landmarks, we get $h^{\mathcal{C}}(I) = 20 = h^*(I)$.

(Induced) Independent h^{LM}

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



Induced by fact LMs: $\{get(A, B)\}$, $\{get(C_4, E), get(D_4, E)\}$. $\rightarrow h^{\mathcal{C}}(I) =$

Additional disjunctive action LMs: e.g., $\{get(B, C_1), get(B, D_1)\}$ and $\{get(C_i, C_{i+1}), get(D_i, D_{i+1})\}$ for $1 \leq i < 4$. $\rightarrow h^{\mathcal{C}}(I) =$

$\rightarrow$ Can we *always* get a set of LM so that $h = h^*$??

Remarks

Technical: (Say we're given a collection $\{L_i\}$ of disjunctive action landmarks.)

- Cost partitioning: Alternative combination of $h^{\text{LM}}_{L_i}$. The optimal cost partitioning dominates the canonical heuristic.
- Even stronger alternative: cost $c(H)$ of a minimum-cost hitting set H of $\{L_i\}$. This is admissible and dominates the optimal cost partitioning. (E.g. for independent $\{L_i\}$, the hitting set contains one distinct action per L_i.)
  $\rightarrow$ On bottom example from previous slide, $c(H) =$
- However, we still do not get h^* in general: We may have to apply the same action *more than once*.
- In delete-relaxed planning, on the other hand, every action must be applied at most once. And indeed, there always exists $\{L_i\}$ so that $c(H) = h^+$! [*bonet:helmert:ecai-10*]

Remarks, ctd.

Historical:

- The idea to generate heuristics based on landmarks was first conceived by [*zhu:givan:icaps-dc-03*], never properly published and forgotten all about.
- The (basic) idea was re-discovered by the authors of LAMA [*richter:etal:aaai-08,richter:westphal:jair-10*]. 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 landmarks heuristics induced by fact landmarks, without ensuring independence).
- The first admissible landmarks heuristic used cost partitioning [*karpas:domshlak:ijcai-09*].
- Cost partitioning also underlies LM-cut [*helmert:domshlak:icaps-09*], the currently best-performing admissible landmarks heuristic in practice. $\rightarrow$ We don't cover LM-cut here

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

$\rightarrow$ How to obtain the *"collection $\{L_i\}$ of disjunctive action landmarks"*??

Theorem (Checking Landmarks is Hard). *Let $\Pi = (F, A, c, I, G)$ be an STRIPS 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.*

> $\rightarrow$ Something is a landmark if and only if disallowing it renders the task unsolvable, and thus checking landmarks is hard.

So is all lost?

$\rightarrow$ How to obtain the *"collection $\{L_i\}$ 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.

Detecting *Some* LMs, Take 1: Necessary Sub-Goals

Definition (Necessary Sub-Goals). *Let Π be an STRIPS planning task, and let s be a state. A fact p is a necessary sub-goal in Π for s if either:*

(i) $p \in G$; *or*

(ii) *there exists a necessary sub-goal q in Π for s so that $q \notin s$ and $p \in \bigcap_{a \in A, q \in eff_a} pre_a$.*

→ Necessary sub-goals are top-level goals plus shared preconditions. (Note: "sub-goal" here=singleton fact, not fact subset as for critical path heuristics.)

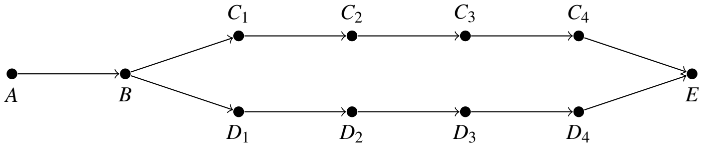
Proposition (Necessary Sub-Goals are Landmarks). *Let Π be an STRIPS planning task, and let s be a state. If p is a necessary sub-goal in Π for s, then p is a fact landmark for s.*

Proof. The claim holds trivially for necessary sub-goals of kind (i). As for case (ii), if q is a fact landmark for s where $q \notin s$, then

→ Given state s, detect necessary sub-goals $\{p_i\}$ for s by simple backchaining, then consider disjunctive action LM L_i induced by $\{p_i \mid p_i \notin s\}$.

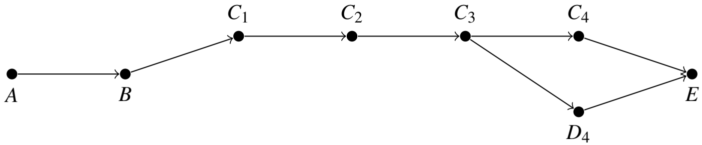
Necessary Sub-Goals vs. Fact Landmarks

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



→ Fact landmarks for I?

→ Necessary sub-goals for I?



→ Fact landmarks for I?

→ Necessary sub-goals for I?

Necessary Sub-Goals in “TSP” in Australia

- Variables: $at : \{Sy, Ad, Br, Pe, Ad\}$; $v(x) : \{T, F\}$ for $x \in \{Sy, Ad, Br, Pe, Ad\}$.
- 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 $x \neq Ad$.

Fact LMs for I: $v(x) = T$ for $x \neq Ad$, and $at = Sy$: goal; $v(x) = F$ for all $x \neq Sy$: true initially; $at = x$ for all $x \neq Sy$: preconditions (and side effects) of necessary truck moves; and $v(Ad) = T$: side effect of driving to Adelaide.

$\rightarrow$ Necessary sub-goals for I?

$\rightarrow$ Fact LMs that are *not* necessary sub-goals?

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

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

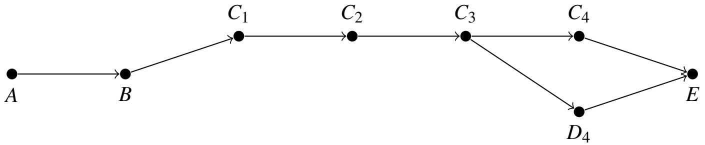
Proposition (Delete Relaxation LM Properties). *Let Π be an STRIPS planning task, and let s be a state. Then all of the following hold:*

(i) *If p is a necessary sub-goal for s, then p is a delete relaxation LM for s.*
(ii) *If p respectively L is a delete relaxation LM for s, then it is a LM for s.*
(iii) *If L is a delete relaxation LM for s, then $h_L^{\mathrm{LM}}(s) \leq h^+(s)$.*

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

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

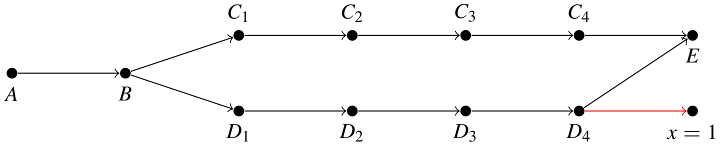
Necessary Sub-Goals vs. Delete Relaxation LMs vs. LMs



Fact LMs for I: A, B, C_1, C_2, C_3, E. **Necessary sub-goals for I:** E.

$\rightarrow$ Delete relaxation fact LMs for I?

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



$\rightarrow$ Fact LMs for I?

$\rightarrow$ Delete relaxation fact LMs for I?

$\rightarrow$ Necessary sub-goals for I?

Delete Relaxation *Fact* LMs in "TSP" in Australia

- Variables: $at : \{Sy, Ad, Br, Pe, Ad\}$; $v(x) : \{T, F\}$ for $x \in \{Sy, Ad, Br, Pe, Ad\}$.
- 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 $x \neq Ad$.

Fact LMs for I: $v(x) = T$ for $x \neq Ad$, and $at = Sy$: goal; $v(x) = F$ for all $x \neq Sy$: true initially; $at = x$ for all $x \neq Sy$: preconditions (and side effects) of necessary truck moves; and $v(Ad) = T$: side effect of driving to Adelaide.

$\rightarrow$ Delete relaxation fact LMs for I?

$\rightarrow$ The delete relaxation happens to not lose information with respect to fact landmarks here; it *does* lose information with respect to disjunctive action landmarks, see next slide.

Delete Relaxation *Action* LMs in "TSP" in Australia

- Variables: $at : \{Sy, Ad, Br, Pe, Ad\}$;
  $v(x) : \{T, F\}$ for $x \in \{Sy, Ad, Br, Pe, Ad\}$.
- 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 $x \neq Ad$.

$\rightarrow$ Minimal disjunctive action LMs for I?

$\rightarrow$ Minimal delete relaxation disjunctive action LMs for I?

Note: The canonical heuristic here equals $h^+(I)$. This is not entirely coincidental: This heuristic will always be $\leq h^+$ (simply insert $\Pi := \Pi^+$ in the theorem on slide 20). The heuristic can be equal to h^+ but doesn't have to be (counter-example on the bottom of slide 24).

Ok, But How to *Detect* Delete Relaxation LMs?

Proposition (Checking Delete Relaxation LMs is Easy). *Let* $\Pi = (F, A, c, I, G)$ *be an STRIPS 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.

$\rightarrow$ 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.

I Said *"Detect"*, not *"Check"*, Delete Relaxation LMs!

→ How can we detect all delete relaxation fact landmarks for s?

- Not such a good idea in practice: Relaxed planning is polynomial time but not dirt-cheap, and there may be 100s–1000s of facts.
- There is a more direct method computing all "causal" delete relaxation fact landmarks by a simple fixed point computation [*keyder:etal:ecai-10*].

→ How can we detect all delete relaxation disjunctive action landmarks for s?

- Completely useless idea in practice: Exponentially many L.
- Vanilla solution: Use $L(p)$ induced by delete relaxation fact LM p.
- Advanced solutions detecing L *not* induced by fact LMs: Construct L iteratively based on *cuts* between reached and unreached parts of planning task [*helmert:domshlak:icaps-09,haslum:etal:icaps-12*]; see also slide 40.

Detect, and Re-Detect, and Re-Detect All Over Again?

→ So, given any search state s, we can detect some disjunctive action landmarks L for s and compute a heuristic. Are we happy now?

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

→ 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.)

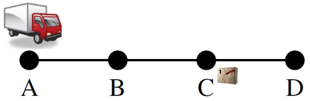
→ =option (B) on slide 16. Re-computation for s =option (C) on slide 16.

Notes (see also next slide)**:** Option (B) is a *path-dependent* heuristic; for optimal planning, need to modify A^* a bit. Option (C) actually sometimes works very well.

Historical Remarks

- The original LMs detection method was restricted to delete relaxation fact LMs, mostly the necessary sub-goals [*hoffmann:etal:jair-04*].
- LAMA does that, plus additional methods based on DTGs; it propagates LMs for I to avoid having to re-detect [*richter:westphal:jair-10*].
- The first admissible LMs heuristic uses the disjunctive action LMs induced by LAMA's (propagated) fact LMs [*karpas:domshlak:ijcai-09*]. This paper also introduced *LM-*A^*, which is optimal while making the best of this path-dependent heuristic.
- The first technique using disjunctive action LMs *not* induced by fact LMs was LM-cut [*helmert:domshlak:icaps-09*], which introduced the idea to iteratively "cut" between reached and unreached parts of the planning task. 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.
- The latest technique alternatingly generates a hitting set and adds a new landmark based on the complement thereof [*haslum:etal:icaps-12*].

Questionnaire



- Initial state I: $t(A), p(C)$.
- Goal G: $t(A), p(D)$.
- Actions A: $dr(X, Y), lo(X), ul(X)$.

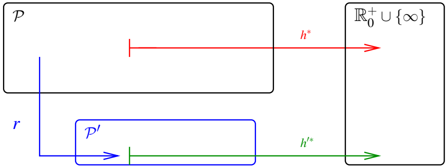
Question!

What is the canonical heuristic $h^{\mathcal{C}}(I)$ for $\mathcal{C} = \{L_i\}$ induced by fact LMs?

Question!

What is the canonical heuristic $h^{\mathcal{C}}(I)$ for $\mathcal{C} =$ all disjunctive action LMs?

Landmarks Heuristics as Relaxations



where, for all $\Pi \in \mathcal{P}$, $h'^*(r(\Pi)) \leq h^*(\Pi)$.

For landmarks heuristics:

- Problem $\mathcal{P}$: All STRIPS planning tasks Π.
- Simpler problem $\mathcal{P}'$: Admissibly combining $h^{\mathsf{LM}}_{L_i}$ for a given collection $\{L_i\}$ of disjunctive action landmarks.
- Perfect heuristic h'^* for $\mathcal{P}'$: The best possible admissible combination.
- Transformation r: Detect disjunctive action landmarks $\{L_i\}$.

This is not native. $\rightarrow$ Efficiently constructible? No: Detecting LMs is hard.

$\rightarrow$ Efficiently computable? Yes! The optimal cost partitioning can be found in polynomial time, Linear approximation. (If we want the stronger hitting set, then it's not efficiently computable.)

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.
- Independent elementary LM heuristics are summed admissibly in the canonical heuristic. Stronger methods are cost partitioning and (even stronger) hitting sets.
- Checking LMs is hard. Practical methods are sound but incomplete, detecting some LMs, namely necessary sub-goals or delete relaxation LMs.
- Vanilla method: Detect (some) fact LMs and use the induced disjunctive action LMs. There are methods to directly detect more general disjunctive action LMs (e.g. LM-cut, cf. slide 48).

Remarks

Planning tools and performance using landmarks:

- Original use for probem decomposition gave reasonable speed-ups for FF and another satisficing heuristic search planner [*hoffmann:etal:jair-04*].
- LAMA [*richter:westphal:jair-10*] introduced the idea to use both, a delete relaxation heuristic and a LMs heuristic, in Fast Downward's dual-queue greedy best-first search framework. The LMs heuristic improves performance significantly in some domains. LAMA won the 1st prizes for satisficing planners at IPC'08 and IPC'11.
- BJOLP [*karpas:domshlak:ijcai-09,domshlak:etal:ipc-11*] 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:domshlak:icaps-09*] also uses admissible combination of disjunctive action LMs, but of more general such LMs not induced by fact LMs. 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:etal:jair-04*].

  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 lecture, but useful as a background read.

- *Sound and Complete Landmarks for And/Or Graphs* [*keyder:etal:ecai-10*].

  Available at:

  `http://ai.cs.unibas.ch/papers/keyder-etal-ecai2010.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:domshlak:ijcai-09*].

  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, 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:domshlak:icaps-09*]. Best Paper Award at ICAPS'09.

  Available at:

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

  Content: The LM-cut paper (cf. slide 46). As if LM-cut wasn't enough, it also introduces the comparison framework for admissible heuristics.

  Wonderful paper, one of the best ever in planning!

Reading, ctd.

- *Strengthening Landmark Heuristics via Hitting Sets* [*bonet:helmert:ecai-10*].
  Best Paper Award at ECAI'10.

  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 landmarks heuristics. Shows that the minimum-cost hitting set over sufficiently large collections of delete relaxation disjunctive action landmarks is equal to h^+.