A Complete Algorithm for Generating Landmarks

Blai Bonet    Julio Castillo

Universidad Simón Bolívar, Caracas, Venezuela

ICAPS 2011 – Freiburg, June 2011
Introduction

- multiple uses of landmarks in planning
- most powerful admissible heuristics are based on landmarks
- we know . . .
  - a lot about exploiting landmarks
  - little about generation of landmarks
- this work is about generation of landmarks
Our contribution

- principled algorithm for generating landmarks
- landmarks can be used for different purposes
- a general framework for heuristics based on landmarks:
  - admissible for optimal planning
  - non-admissible for satisfacing planning
- polytime admissible heuristic
Relaxed Planning
Obtained by removing the deletes of each action

Relaxed task characterized by:

- finite set $F$ of facts
- initial facts $I \subseteq F$
- goal facts $G \subseteq F$ that must be reached
- operators of the form $o[4] : a, b \rightarrow c, d$

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

**Assume WLOG:** $I = \{i\}$, $G = \{g\}$, all $\text{pre}(o) \neq \emptyset$
One way to reach goal $G = \{g\}$ from $I = \{i\}$:

- apply sequence $o_1, o_2, o_4, o_5$ (plan)

- cost: $3 + 4 + 1 + 1 = 9$ (optimal)
Optimal Relaxed Cost

- $h^+$: minimal total cost to reach $G$ from $I$
- **Very good heuristic** function for optimal planning
- **NP-hard** to compute or approximate by constant factor
Landmarks
Most accurate admissible heuristics are based on landmarks

**Def:** a (disjunctive action) **landmark** is a set of operators $L$ such that each plan must contain some action in $L$
Example

\( o_1[3] : i \rightarrow a, b \)
\( o_2[4] : i \rightarrow a, c \)
\( o_3[5] : i \rightarrow b, c \)
\( o_4[1] : a, b \rightarrow d \)
\( o_5[1] : a, d, c \rightarrow g \)

Some landmarks:

- need \( g \): \( W = \{o_5\} \) (hence \( h^+ \geq 1 \))
- need \( a \): \( X = \{o_1, o_2\} \) (hence \( h^+ \geq 3 \))
- need \( c \): \( Y = \{o_2, o_3\} \) (hence \( h^+ \geq 4 \))
- need \( d \): \( Z = \{o_4\} \) (hence \( h^+ \geq 1 \))
- ...
Exploiting Landmarks: Hitting Sets

Given:

- finite set $A$
- collection $\mathcal{F}$ of subsets from $A$
- non-negative costs $c : A \rightarrow \mathbb{R}_0^+$

Hitting set:

- subset $H \subseteq A$ that hits every $S \in \mathcal{F}$ (i.e. $S \cap H \neq \emptyset$)
- cost of $H = \sum_{a \in H} c(a)$

Minimum-cost Hitting Set (MHS):

- minimizes cost
- classical NP-complete problem
Can view collection of landmarks as instance of MHS problem

**Example (Landmarks)**

\[ A = \{o_1, o_2, o_3, o_4, o_5\} \]

\[ \mathcal{F} = \{\{o_5\}, \{o_1, o_2\}, \{o_2, o_3\}, \{o_4\}\} \]

\[ W \quad X \quad Y \quad Z \]

**Costs:** \[ c(o_1) = 3, \quad c(o_2) = 4, \quad c(o_3) = 5, \quad c(o_4) = 1, \quad c(o_5) = 1 \]

**Minimum hitting set:** \[ \{o_2, o_4, o_5\} \] with cost \[ 4 + 1 + 1 = 6 \]
Precondition choice function (pcf): function $D$ that maps operators to preconditions

Justification graph for pcf $D$: arc-labeled digraph with:

- vertices: the facts $F$
- arcs: $D(o) \xrightarrow{o} e$ for each operator $o$ and effect $e \in \text{eff}(o)$
pcf $D$: \[
\begin{array}{cccccc}
 & o & o_1 & o_2 & o_3 & o_4 & o_5 \\
 D(o) & i & i & i & a & a \\
\end{array}
\]

Landmark (cut): \{5\}:
- $\rightarrow a, b$
- $\rightarrow a, c$
- $\rightarrow b, c$
- $\rightarrow d$
- $\rightarrow a, c, d \rightarrow g$
\[
\begin{array}{c|cccccc}
\text{pcf } D: & o & o_1 & o_2 & o_3 & o_4 & o_5 \\
\hline
D(o) & i & i & i & a & a \\
\end{array}
\]

Landmark (cut): \( W = \{o_5\} \)

\( o_1[3]: i \rightarrow a, b \)
\( o_2[4]: i \rightarrow a, c \)
\( o_3[5]: i \rightarrow b, c \)
\( o_4[1]: a, b \rightarrow d \)
\( o_5[1]: a, c, d \rightarrow g \)
pcf $D$: \[
\begin{array}{c|cccccc}
D(o) & o & o_1 & o_2 & o_3 & o_4 & o_5 \\
\hline
D(o) & i & i & i & a & a
\end{array}
\]

Landmark (cut): $X = \{o_1, o_2\}$

- $o_1[3]: i \rightarrow a, b$
- $o_2[4]: i \rightarrow a, c$
- $o_3[5]: i \rightarrow b, c$
- $o_4[1]: a, b \rightarrow d$
- $o_5[1]: a, c, d \rightarrow g$
**pcf** $D$: \[
\begin{array}{c|cccc}
  o & o_1 & o_2 & o_3 & o_4 & o_5 \\
  i & i & i & i & a & d \\
\end{array}
\]

(new pcf)

**Landmark (cut):** $W = \{o_5\}$

\[o_1[3] : i \rightarrow a, b\]
\[o_2[4] : i \rightarrow a, c\]
\[o_3[5] : i \rightarrow b, c\]
\[o_4[1] : a, b \rightarrow d\]
\[o_5[1] : a, c, d \rightarrow g\]
pcf $D: \begin{array}{cccccc}
  o & o_1 & o_2 & o_3 & o_4 & o_5 \\
  D(o) & i & i & i & a & d 
\end{array}$

Landmark (cut): $Z = \{o_4\}$

- $o_1[3]: i \rightarrow a, b$
- $o_2[4]: i \rightarrow a, c$
- $o_3[5]: i \rightarrow b, c$
- $o_4[1]: a, b \rightarrow d$
- $o_5[1]: a, c, d \rightarrow g$
\[ \text{pcf } D: \begin{array}{c|cccccc} o & o_1 & o_2 & o_3 & o_4 & o_5 \\ \hline D(o) & i & i & i & a & d \end{array} \]

Landmark (cut): \( X = \{ o_1, o_2 \} \)

\[ o_1[3] : i \rightarrow a, b \]
\[ o_2[4] : i \rightarrow a, c \]
\[ o_3[5] : i \rightarrow b, c \]
\[ o_4[1] : a, b \rightarrow d \]
\[ o_5[1] : a, c, d \rightarrow g \]
Thm (B. & Helmert, 2010): Let $\mathcal{L}$ be all “cut landmarks”. Then, $h^+ = \text{cost of MHS for } \mathcal{L}$. 
Thm (B. & Helmert, 2010): Let $\mathcal{L}$ be all “cut landmarks”. Then, $h^+ = \text{cost of MHS for } \mathcal{L}$.

Impractical to generate all landmarks!

Do we need all of them to get $h^+$ or a good approximation?
Principled Generation of Landmarks
\( H = \) subset of operators

\( R = \) fluents reachable from \( I \) using only operators in \( H \)
\( H = \text{subset of operators} \)

\( R = \text{fluents reachable from } I \text{ using only operators in } H \)

\[ g \in R \implies H \text{ “contains” a relaxed plan} \]

\[ g \notin R \implies (R, R^c) \text{ is cut of some justification graph } G(D) \]

and \( H \text{ does not hit cutset}(R, R^c) \)
$H =$ subset of operators

$R =$ fluents reachable from $I$ using only operators in $H$

$g \in R \implies H$ “contains” a relaxed plan

$g \notin R \implies (R, R^c)$ is cut of some justification graph $G(D)$

and $H$ does not hit cutset$(R, R^c)$

Indeed, it’s enough to define pcf $D$ as $D(o) = p$ where

$$
\begin{cases}
    p \in pre(o) & \text{if } pre(o) \subseteq R \\
    p \in pre(o) \setminus R & \text{if } pre(o) \not\subseteq R
\end{cases}
$$
For such pcf $D$,

$$L = \text{cutset}(R, R^c) = \{ o : D(o) \in R \text{ and } \text{eff}(o) \notin R^c \}$$

is landmark not hit by $H$!
For such pcf $D$,

$$L = \text{cutset}(R, R^c) = \{ o : D(o) \in R \text{ and } \text{eff}(o) \notin R^c \}$$

is landmark not hit by $H$!

$L$ improved by removing from $G(D)$ facts irrelevant to $g$
Algorithm $A$

**Input:** subset $H$ of actions

**Output:** YES if $H$ contains plan, or landmark not hit by $H$

**Method:**

1. $R :=$ set of reachable fluents using actions in $H$
2. if $g \in H$ then return YES
3. compute pcf $D$ and justification graph $G(D)$
4. simplify graph $G(D)$
5. return cutset of $(R, R^c)$ in simplified graph

**Time:** linear with correct data structures!
Landmarks $= \emptyset$
Landmarks = $\emptyset$

\[ H = \emptyset \ ; \ R = \{i\} \ ; \ R^c = \{a, b, c, d, g\} \ ; \ L = \{o_1, o_2\} \]
Landmarks = \{\{o_1, o_2\}\}

H = \{o_1\} ; R = \{i, a, b\} ; R^c = \{c, d, g\} ; L = \{o_4\}
Landmarks = \{ \{o_1, o_2\}, \{o_4\} \}

H = \{o_1, o_4\}; R = \{i, a, b, d\}; R^c = \{c, g\}; L = \{o_2, o_3\}
Landmarks = \{ \{ o_1, o_2 \}, \{ o_4 \}, \{ o_2, o_3 \} \} \\
X \quad Z \quad Y

H = \{ o_2, o_4 \} ;
R = \{ i, a, c \} ;
R^c = \{ b, g \} ;
L = \{ o_1, o_3 \}
Landmarks = \{\{o_1, o_2\}, \{o_4\}, \{o_2, o_3\}, \{o_1, o_3\}\} \\
\begin{align*}
\text{H} &= \{o_1, o_2, o_4\} ; \\
\text{R} &= \{i, a, b, c, d\} ; \\
\text{R}^c &= \{g\} ; \\
\text{L} &= \{o_5\}
\end{align*}
Landmarks = \{\{o_1, o_2\}, \{o_4\}, \{o_2, o_3\}, \{o_1, o_3\}, \{o_5\}\}\] \text{complete!}

H = \{o_1, o_2, o_4, o_5\}; \ R = \{i, a, b, c, d, g\}; \ R^c = \emptyset
Algorithm \(C1\)

**Input:** initial collection \(\mathcal{L}\) (maybe empty)

**Output:** a complete collection and \(h^+(I)\)

**Method:**

1. \(H := \text{min-cost hitting set for } \mathcal{L}\)
2. \(L := A(H)\)
3. if \(L = \text{YES}\) then return \(\mathcal{L}\) and cost of \(H\)
4. \(\mathcal{L} := \mathcal{L} \cup \{L\}\)
5. goto 2

Algorithm \(C1\) does not run in polytime because:

- computing min-cost hitting sets is \textbf{NP-hard}
- number of iterations may be \textbf{exponential}
Flaws can be **overcomed** to get a polytime approximation by:

- controlling number of iterations
- controlling difficulty of solving MHS problem

**See paper for:**

- details about algorithm $C1$ and variants $C2$ and $C3$
- how to use $A$ to get heuristics for satisficing planning
- novel polytime admissible heuristics that dominate best-known heuristics (**in number of expanded nodes**)  
  **slower than state-of-the-art heuristics (i.e. LM-Cut)**
Thanks!