Planning and Optimization

E2. Landmarks: Cut Landmarks & LM-cut Heuristic

Gabriele Röger and Thomas Keller

Universität Basel

November 12, 2018

Roadmap for this Chapter

- We first introduce a new normal form for delete-free STRIPS tasks that simplifies later definitions.
- We then present a method that computes disjunctive action landmarks for such tasks.
- We conclude with the LM-cut heuristic that builds on this method.

i-g Form

Delete-Free STRIPS Planning Task in i-g Form (1)

In this chapter, we only consider delete-free STRIPS tasks in a special form:

Definition (i-g Form for Delete-free STRIPS)

A delete-free STRIPS planning task $\langle V, I, O, \gamma \rangle$ is in i-g form if

- V contains atoms i and g
- Initially exactly i is true: $I(v) = \mathbf{T}$ iff $v = i$
- g is the only goal atom: $\gamma = g$
- Every action has at least one precondition.

Transformation to i-g Form

Every delete-free STRIPS task $\Pi = \langle V, I, O, \gamma \rangle$ can easily be transformed into an analogous task in i-g form.

- If i or g are in V already, rename them everywhere.
- Add i and g to V.
- Add an operator $\langle i, \bigwedge_{v \in V : I(v) = \mathbf{T}} v, 0 \rangle$.
- Add an operator $\langle \gamma, g, 0 \rangle$.
- Replace all operator preconditions $\top$ with i.
- Replace initial state and goal.

Transformation to i-g Form

Every delete-free STRIPS task $\Pi = \langle V, I, O, \gamma \rangle$ can easily be transformed into an analogous task in i-g form.

- If i or g are in V already, rename them everywhere.
- Add i and g to V.
- Add an operator $\langle i, \bigwedge_{v \in V : I(v) = \mathbf{T}} v, 0 \rangle$.
- Add an operator $\langle \gamma, g, 0 \rangle$.
- Replace all operator preconditions $\top$ with i.
- Replace initial state and goal.

In what sense are the tasks "analogous"?

Delete-Free STRIPS Planning Task in i-g Form (2)

- In the following, we assume tasks in i-g form.
- Providing O suffices to describe the overall task:
  - V are the variables mentioned in the operators in O.
  - always exactly i true in I and $\gamma = g$
- In the following, we only provide O for the description of the task.
- Since we consider delete-free STRIPS tasks, $pre(o)$ and $eff(o)$ are conjunctions of atoms. In the following, we treat them as sets $pre(o)$ and $add(o)$ of atoms.
- We write operator $o = \langle pre(o), add(o), cost(o) \rangle$ as $\langle pre(o) \rightarrow add(o) \rangle_{cost(o)}$, omitting braces for sets.

Example: Delete-Free Planning Task in i-g Form

Example

Operators:

- $o_1 = \langle i \rightarrow x, y \rangle_3$
- $o_2 = \langle i \rightarrow x, z \rangle_4$
- $o_3 = \langle i \rightarrow y, z \rangle_5$
- $o_4 = \langle x, y, z \rightarrow g \rangle_0$

optimal solution?

Example: Delete-Free Planning Task in i-g Form

Example

Operators:

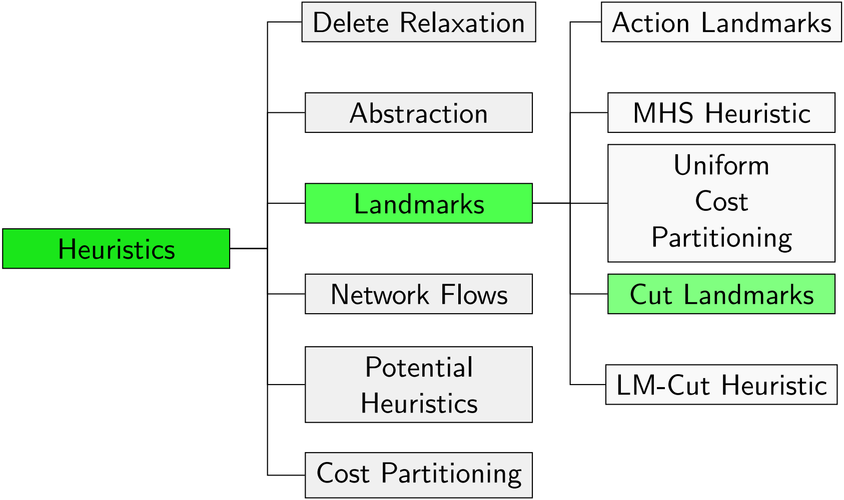
- $o_1 = \langle i \rightarrow x, y \rangle_3$
- $o_2 = \langle i \rightarrow x, z \rangle_4$
- $o_3 = \langle i \rightarrow y, z \rangle_5$
- $o_4 = \langle x, y, z \rightarrow g \rangle_0$

optimal solution to reach g from i:

- plan: o_1, o_2, o_4
- cost: $3 + 4 + 0 = 7$ $\quad (= h^+(I)$ because plan is optimal)

Cut Landmarks

Content of this Course: Heuristics



Justification Graphs

Definition (Precondition Choice Function)

A precondition choice function (pcf) $P : O \to V$ for a delete-free STRIPS task $\Pi = \langle V, I, O, \gamma \rangle$ in i-g form maps each operator to one of its preconditions (i.e. $P(o) \in pre(o)$ for all $o \in O$).

Definition (Justification Graphs)

Let P be a pcf for $\langle V, I, O, \gamma \rangle$ in i-g form. The justification graph for P is the directed, edge-labeled graph $J = \langle V, E \rangle$, where

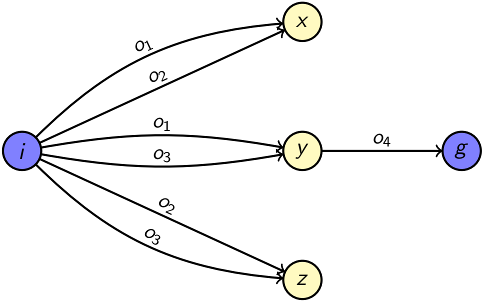
- the vertices are the variables from V, and
- E contains an edge $P(o) \xrightarrow{o} a$ for each $o \in O$, $a \in add(o)$.

Example: Justification Graph

Example

pcf P: $P(o_1) = P(o_2) = P(o_3) = i$, $P(o_4) = y$

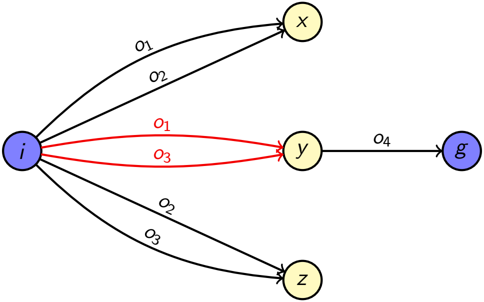
$o_1 = \langle i \rightarrow x, y \rangle_3$
$o_2 = \langle i \rightarrow x, z \rangle_4$
$o_3 = \langle i \rightarrow y, z \rangle_5$
$o_4 = \langle x, y, z \rightarrow g \rangle_0$



Cuts

Definition (Cut)

A cut in a justification graph is a subset C of its edges such that all paths from i to g contain an edge from C.



Cuts are Disjunctive Action Landmarks

Theorem (Cuts are Disjunctive Action Landmarks)

Let P be a pcf for $\langle V, I, O, \gamma\rangle$ (in i-g form) and C be a cut in the justification graph for P.

The set of edge labels from C (formally $\{o \mid \langle v, o, v'\rangle \in C\}$) is a disjunctive action landmark for I.

Proof idea:

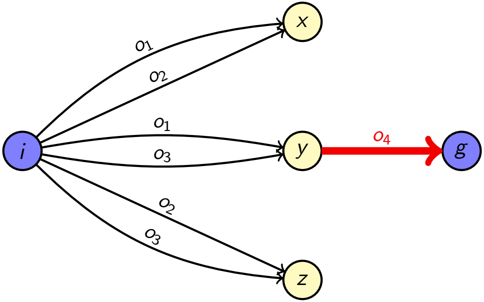
- The justification graph corresponds to a simpler problem where some preconditions (those not picked by the pcf) are ignored.
- Cuts are landmarks for this simplified problem.
- Hence they are also landmarks for the original problem.

Example: Cuts in Justification Graphs

Example

landmark $A = \{o_4\}$ (cost $= 0$)

$o_1 = \langle i \to x, y \rangle_3$
$o_2 = \langle i \to x, z \rangle_4$
$o_3 = \langle i \to y, z \rangle_5$
$o_4 = \langle x, y, z \to g \rangle_0$

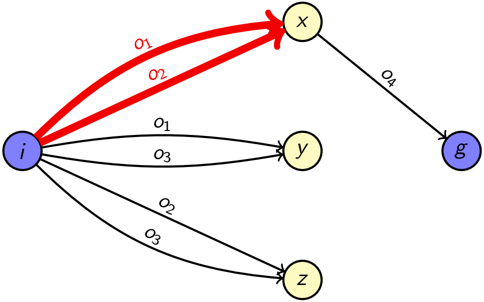


Example: Cuts in Justification Graphs

Example

landmark $B = \{o_1, o_2\}$ (cost $= 3$)

$o_1 = \langle i \rightarrow x, y \rangle_3$
$o_2 = \langle i \rightarrow x, z \rangle_4$
$o_3 = \langle i \rightarrow y, z \rangle_5$
$o_4 = \langle x, y, z \rightarrow g \rangle_0$

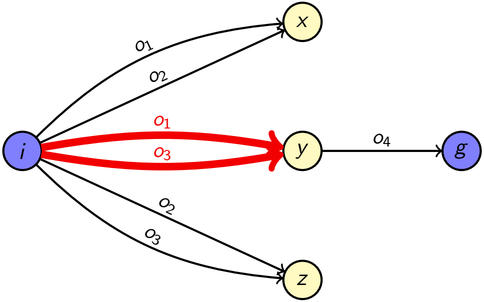


Example: Cuts in Justification Graphs

Example

landmark $C = \{o_1, o_3\}$ (cost $= 3$)

$o_1 = \langle i \to x, y \rangle_3$
$o_2 = \langle i \to x, z \rangle_4$
$o_3 = \langle i \to y, z \rangle_5$
$o_4 = \langle x, y, z \to g \rangle_0$

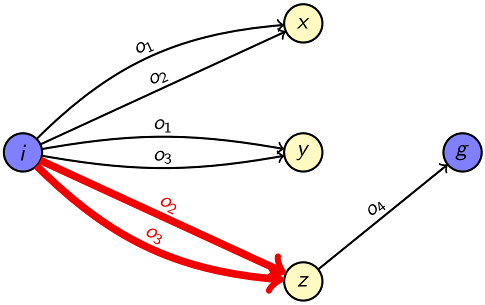


Example: Cuts in Justification Graphs

Example

landmark $D = \{o_2, o_3\}$ (cost $= 4$)

$$o_1 = \langle i \rightarrow x, y \rangle_3$$
$$o_2 = \langle i \rightarrow x, z \rangle_4$$
$$o_3 = \langle i \rightarrow y, z \rangle_5$$
$$o_4 = \langle x, y, z \rightarrow g \rangle_0$$



Power of Cuts in Justification Graphs

- Which landmarks can be computed with the cut method?

Power of Cuts in Justification Graphs

- Which landmarks can be computed with the cut method?
- all interesting ones!

Proposition (perfect hitting set heuristics)

Let $\mathcal{L}$ be the set of all "cut landmarks" of a given planning task with initial state I. Then $h^{MHS}(\mathcal{L}) = h^{+}(I)$.

$\rightsquigarrow$ Hitting set heuristic for $\mathcal{L}$ is perfect.

Power of Cuts in Justification Graphs

- Which landmarks can be computed with the cut method?
- all interesting ones!

Proposition (perfect hitting set heuristics)

Let $\mathcal{L}$ be the set of all "cut landmarks" of a given planning task with initial state I. Then $h^{MHS}(\mathcal{L}) = h^{+}(I)$.

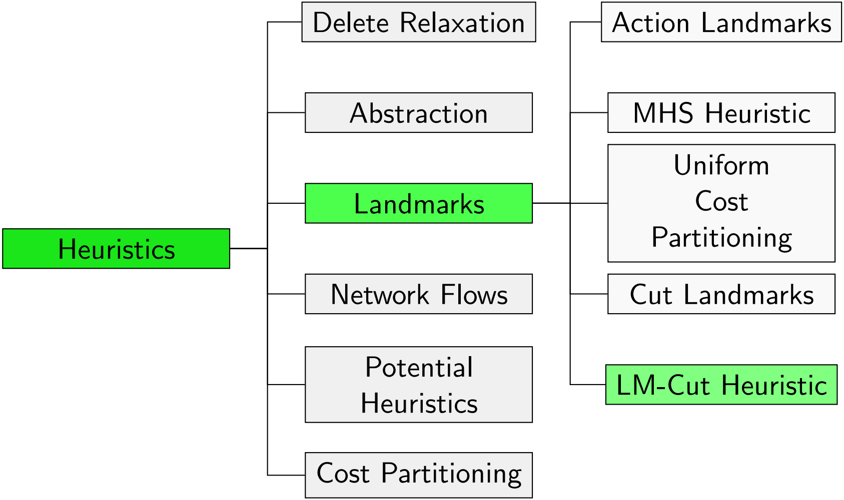
$\rightsquigarrow$ Hitting set heuristic for $\mathcal{L}$ is perfect.

Proof idea:

- Show 1:1 correspondence of hitting sets H for $\mathcal{L}$ and plans, i.e., each hitting set for $\mathcal{L}$ corresponds to a plan, and vice versa.

The LM-Cut Heuristic

Content of this Course: Heuristics



LM-Cut Heuristic: Motivation

- In general, there are exponentially many pcfs, hence computing all relevant landmarks is not tractable.
- The LM-cut heuristic is a method that chooses pcfs and computes cuts in a goal-oriented way.
- As a side effect, it computes a (non-uniform) cost partitioning.

$\rightsquigarrow$ currently one of the best admissible planning heuristic

LM-Cut Heuristic (1)

$h^{\text{LM-cut}}$: Helmert & Domshlak (2009)

Initialize $h^{\text{LM-cut}}(I) := 0$. Then iterate:

1. Compute $h^{\max}$ values of the variables. Stop if $h^{\max}(g) = 0$.
2. Let P be a pcf that chooses preconditions with maximal $h^{\max}$ value.
3. Compute the justification graph for P.
4. Compute a cut which guarantees $cost(L) > 0$ for the corresponding landmark L (next slide).
5. Increase $h^{\text{LM-cut}}(I)$ by $cost(L)$.
6. Decrease $cost(o)$ by $cost(L)$ for all $o \in L$.

LM-Cut Heuristic (2)

$h^{\text{LM-cut}}$: Helmert & Domshlak (2009)

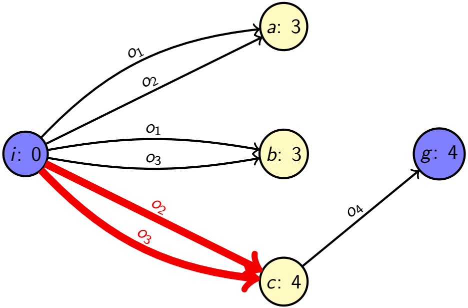
4. Compute a cut which guarantees $cost(L) > 0$ for the corresponding landmark L as follows:
   - The goal zone V_g of the justification graph consists of all nodes that have a path to g where all edges are labelled with zero-cost operators.
   - The cut contains all edges $\langle v, o, v' \rangle$ such that $v \notin V_g$, $v' \in V_g$ and v can be reached from i without traversing a node in V_g.

Example: Computation of LM-Cut

Example

round 1: $P(o_4) = c \rightsquigarrow L = \{o_2, o_3\}$ [4]

$o_1 = \langle i \rightarrow a, b \rangle_3$
$o_2 = \langle i \rightarrow a, c \rangle_4$
$o_3 = \langle i \rightarrow b, c \rangle_5$
$o_4 = \langle a, b, c \rightarrow g \rangle_0$

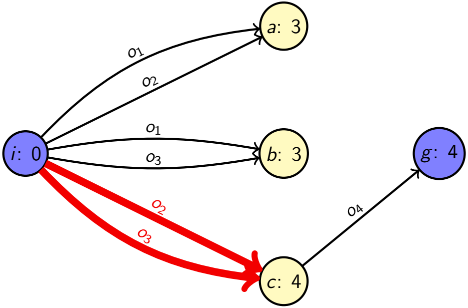


Example: Computation of LM-Cut

Example

round 1: $P(o_4) = c \rightsquigarrow L = \{o_2, o_3\}$ [4] $\rightsquigarrow h^{\text{LM-cut}}(I) := 4$

$o_1 = \langle i \rightarrow a, b\rangle_3$
$o_2 = \langle i \rightarrow a, c\rangle_0$
$o_3 = \langle i \rightarrow b, c\rangle_1$
$o_4 = \langle a, b, c \rightarrow g\rangle_0$

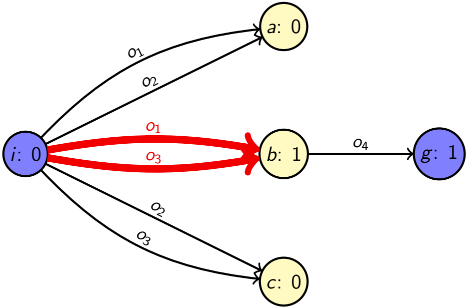


Example: Computation of LM-Cut

Example

round 2: $P(o_4) = b \rightsquigarrow L = \{o_1, o_3\}$ [1]

$o_1 = \langle i \to a, b \rangle_3$
$o_2 = \langle i \to a, c \rangle_0$
$o_3 = \langle i \to b, c \rangle_1$
$o_4 = \langle a, b, c \to g \rangle_0$

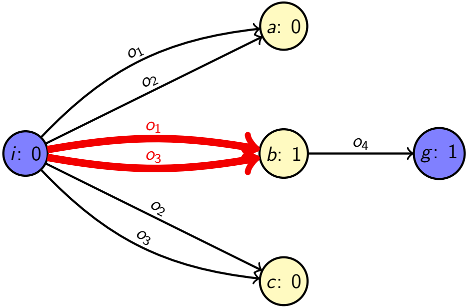


Example: Computation of LM-Cut

Example

round 2: $P(o_4) = b \rightsquigarrow L = \{o_1, o_3\}$ [1] $\rightsquigarrow h^{\text{LM-cut}}(I) := 4 + 1 = 5$

$o_1 = \langle i \rightarrow a, b \rangle_2$
$o_2 = \langle i \rightarrow a, c \rangle_0$
$o_3 = \langle i \rightarrow b, c \rangle_0$
$o_4 = \langle a, b, c \rightarrow g \rangle_0$

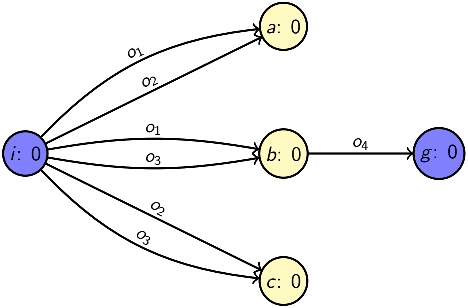


Example: Computation of LM-Cut

Example

round 3: $h^{\max}(g) = 0 \rightsquigarrow$ done! $\rightsquigarrow h^{\text{LM-cut}}(I) = 5$

$o_1 = \langle i \to a, b\rangle_2$
$o_2 = \langle i \to a, c\rangle_0$
$o_3 = \langle i \to b, c\rangle_0$
$o_4 = \langle a, b, c \to g\rangle_0$



Properties of LM-Cut Heuristic

Theorem

Let $\langle V, I, O, G\rangle$ be a delete-free STRIPS task in i-g normal form. The LM-cut heuristic is admissible: $h^{\text{LM-cut}}(I) \leq h^*(I)$.

(Proof omitted.)

If Π is not delete-free, we can compute $h^{\text{LM-cut}}$ on Π^+.
Then $h^{\text{LM-cut}}$ is bound by h^+.

Summary & Outlook

Summary

- Cuts in justification graphs are a general method to find disjunctive action landmarks.
- Hitting sets over all cut landmarks yield a perfect heuristic for delete-free planning tasks.
- The LM-cut heuristic is an admissible heuristic based on these ideas.

Outlook

- We have only considered (disjunctive) action landmarks, not atom or formula landmarks.
- There are other landmark generation methods, e.g. based on a version of relaxed task graphs.
- The LM-cut heuristic extracts the landmarks for each state.
- Other methods extract landmarks once, propagating them over the course of the search.
- Such methods are usually enhanced with orderings (e.g. stating that some landmark must be achieved before some other landmark).
- The (inadmissible) LM-Count heuristic counts the number of formula landmarks that still need to be achieved.