Strengthening Landmark Heuristics via Hitting Sets

Blai Bonet[1] Malte Helmert[2]

[1]Universidad Simón Bolívar, Caracas, Venezuela

[2]Albert-Ludwigs-Universität Freiburg, Germany

ECAI 2010 – August 18th, 2010

Our contribution

Area: heuristics for optimal classical planning

Our contribution

- stronger way of exploiting landmarks for heuristic functions
- systematic way of generating landmarks for delete relaxation
- theoretical results relating new ideas to
  - admissible landmark heuristics (Karpas & Domshlak, 2009)
  - landmark-cut heuristic (Helmert & Domshlak, 2009)
  - optimal delete relaxation h^+ (Hoffmann & Nebel, 2001)
  - fixed-parameter tractability of problems of hitting sets
- new poly-time heuristic family that dominates landmark-cut

Relaxed planning

Optimal planning

Optimal planning:

- shortest paths in huge implicit graphs
- no formal definition here

What we need to know:

- state-of-the-art planners: heuristic search
- optimal planners: A* + heuristics
- many use delete relaxation ("relaxed planning tasks")
- want accurate estimates of optimal delete relaxation cost h^+

Relaxed planning tasks

Obtained by removing the deletes of each action

Definition (relaxed planning task)

F: finite set of facts

- initial facts $I \subseteq F$ are given
- goal facts $G \subseteq F$ must be reached
- operators of the form $o[4] : a, b \to c, d$

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

For simplicity (WLOG): assume $I = \{i\}$, $G = \{g\}$, all $pre(o) \neq \emptyset$

Example: relaxed planning task

Example

$o_1[3] : i \to a, b$
$o_2[4] : i \to a, c$
$o_3[5] : i \to b, c$
$o_4[0] : a, b, c \to g$

One way to reach $\{g\}$ from $\{i\}$:

- apply sequence o_1, o_2, o_4 (plan)
- cost: $3 + 4 + 0 = 7$ (optimal)

Optimal relaxed cost

- $h^+(I)$: minimal total cost to reach G from I
- Very good heuristic function for optimal planning
- NP-hard to compute (Bylander, 1994)
  or approximate by constant factor (Betz & Helmert, 2009)

$\rightsquigarrow$ use polynomial-time admissible heuristics

Landmarks

Landmarks

The most accurate current heuristics are based on landmarks.

Definition (landmark)

A (disjunctive action) landmark is a set of operators L such that each plan must contain some element of L.

The cost of a landmark, $cost(L)$, is $\min_{o \in L} cost(o)$.

$\rightsquigarrow$ the cost of any landmark is a (crude) admissible heuristic

Example: landmarks

Example

$o_1[3] : i \to a, b$
$o_2[4] : i \to a, c$
$o_3[5] : i \to b, c$
$o_4[0] : a, b, c \to g$

Some landmarks:

- $W = \{o_4\}$ (cost 0)
- $X = \{o_1, o_2\}$ (cost 3)
- $Y = \{o_1, o_3\}$ (cost 3)
- $Z = \{o_2, o_3\}$ (cost 4)
- but also: $\{o_1, o_2, o_3\}$ (cost 3), $\{o_1, o_2, o_4\}$ (cost 0), ...

Exploiting landmarks

Exploiting landmarks

Assume we are given landmark set $\mathcal{L} = \{W, X, Y, Z\}$
(later: how to find such landmarks)

How do we exploit $\mathcal{L}$ for heuristics?

- sum of costs $0 + 3 + 3 + 4 = 10 \rightsquigarrow$ inadmissible!
- maximum of costs: $\max\{0, 3, 3, 4\} = 4 \rightsquigarrow$ weak
- best previous approach: optimal cost partitioning

Optimal cost partitioning (Karpas & Domshlak (2009))

Example

$cost(o_1) = 3, \quad cost(o_2) = 4, \quad cost(o_3) = 5, \quad cost(o_4) = 0$

$\mathcal{L} = \{W, X, Y, Z\}$
with $W = \{o_4\}, \quad X = \{o_1, o_2\}, \quad Y = \{o_1, o_3\}, \quad Z = \{o_2, o_3\}$

LP: maximize $w + x + y + z$ subject to $w, x, y, z \geq 0$ and

$$\begin{array}{ccccccccc} & & x & + & y & & & \leq & 3 & o_1 \\ & & x & + & & & z & \leq & 4 & o_2 \\ & & & & y & + & z & \leq & 5 & o_3 \\ w & & & & & & & \leq & 0 & o_4 \\ W & & X & & Y & & Z & & & \end{array}$$

solution: $w = 0, \; x = 1, \; y = 2, \; z = 3 \quad \rightsquigarrow \quad \boxed{h^L(I) = 6}$

Hitting sets

Definition (hitting set)

Given: finite set A, subset family $\mathcal{F} \subseteq 2^A$, costs $c : A \to \mathbb{R}_0^+$

Hitting set:

- subset $H \subseteq A$ that "hits" all subsets in $\mathcal{F}$: $H \cap S \neq \emptyset$ for all $S \in \mathcal{F}$
- cost of H: $\sum_{a \in H} c(a)$

Minimum hitting set (MHS):

- minimizes cost
- classical NP-complete problem (Karp, 1972)

Landmarks and hitting sets

Can view landmark sets (with operator costs)
as instances of minimum hitting set problem

Example

$A = \{o_1, o_2, o_3, o_4\}$

$\mathcal{F} = \{W, X, Y, Z\}$
with $W = \{o_4\}$, $X = \{o_1, o_2\}$, $Y = \{o_1, o_3\}$, $Z = \{o_2, o_3\}$

$c(o_1) = 3$, $c(o_2) = 4$, $c(o_3) = 5$, $c(o_4) = 0$

Minimum hitting set: $\{o_1, o_2, o_4\}$ with cost $3 + 4 + 0 = 7$

Hitting set heuristics

Let $\mathcal{L}$ be a set of landmarks.

Theorem (hitting set heuristics are admissible)

Let $h^{\mathsf{MHS}}(I)$ be the minimum hitting set cost for $\langle O, \mathcal{L}, \mathit{cost} \rangle$. Then:

1. $h^{\mathsf{MHS}}(I) \geq h^{\mathsf{L}}(I)$ *(hitting sets dominate cost partitioning)*
2. $h^{\mathsf{MHS}}(I) \leq h^{+}(I)$ *(hitting set heuristics are admissible)*

Generating landmarks

Generating landmarks

How do we generate landmarks in the first place?

- most successful previous approach: LM-cut procedure (Helmert & Domshlak, 2009)
- we present a generalization based on:
  - construction of justification graph
  - extraction of landmarks from justification graph

Justification graphs

Definition (precondition choice function)

A precondition choice function (pcf) $D : O \to F$
maps each operator to one of its preconditions.

Definition (justification graph)

The justification graph for pcf D is an arc-labeled digraph with

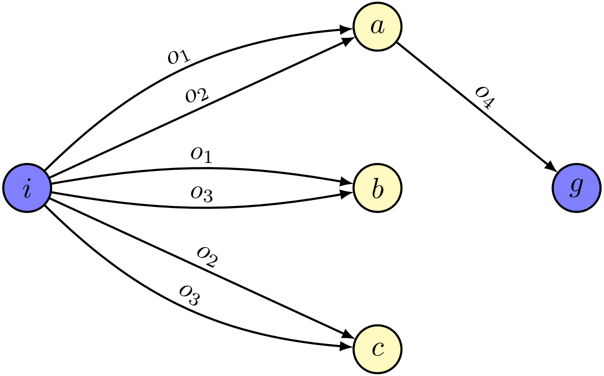
- vertices: the facts F
- arcs: arc $D(o) \xrightarrow{o} e$ for each operator o and effect $e \in \mathit{eff}(o)$

Example: justification graph

Example

pcf D: $D(o_1) = D(o_2) = D(o_3) = i$, $D(o_4) = a$

$o_1[3] : i \to a, b$
$o_2[4] : i \to a, c$
$o_3[5] : i \to b, c$
$o_4[0] : a, b, c \to g$

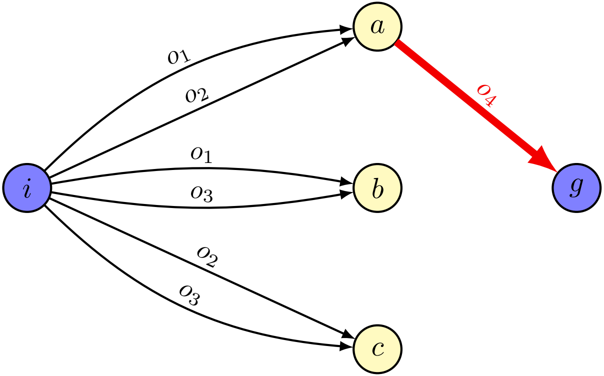


Example: cuts of a justification graph

Example

Landmark $W = \{o_4\}$ (cost 0)

$o_1[3] : i \rightarrow a, b$
$o_2[4] : i \rightarrow a, c$
$o_3[5] : i \rightarrow b, c$
$o_4[0] : a, b, c \rightarrow g$



Example: cuts of a justification graph

Example

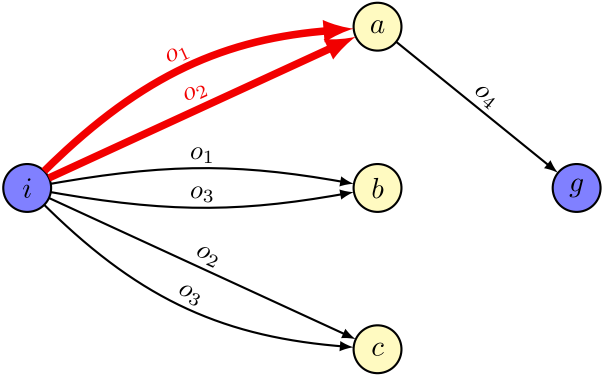
Landmark $X = \{o_1, o_2\}$ (cost 3)

$o_1[3] : i \rightarrow a, b$

$o_2[4] : i \rightarrow a, c$

$o_3[5] : i \rightarrow b, c$

$o_4[0] : a, b, c \rightarrow g$

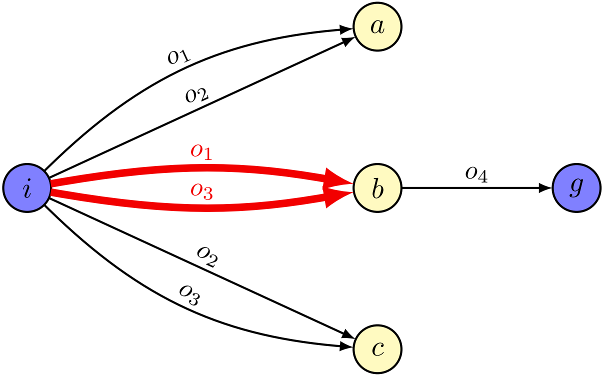


Example: cuts of a justification graph

Example

Landmark $Y = \{o_1, o_3\}$ (cost 3)

$o_1[3] : i \rightarrow a, b$
$o_2[4] : i \rightarrow a, c$
$o_3[5] : i \rightarrow b, c$
$o_4[0] : a, b, c \rightarrow g$



Example: cuts of a justification graph

Example

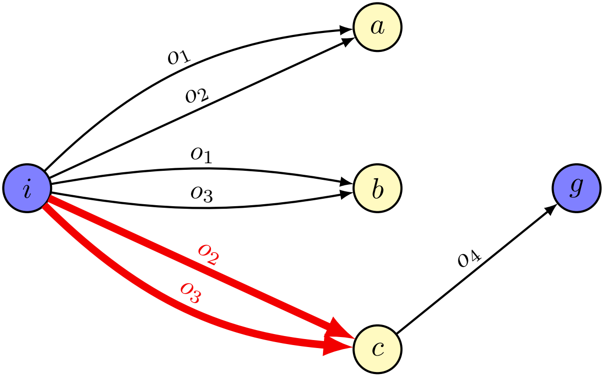
Landmark $Z = \{o_2, o_3\}$ (cost 4)

$o_1[3] : i \rightarrow a, b$

$o_2[4] : i \rightarrow a, c$

$o_3[5] : i \rightarrow b, c$

$o_4[0] : a, b, c \rightarrow g$



Power of justification graph cuts

- Which landmarks can be generated with the cut method?
- All interesting ones!

Theorem (perfect hitting set heuristics)

Let $\mathcal{L}$ be the set of all "cut landmarks".
Then $h^{\mathit{MHS}}(I) = h^{+}(I)$.

$\rightsquigarrow$ hitting set heuristic over $\mathcal{L}$ is perfect

Improving the LM-cut heuristic

Polynomial hitting set heuristics

How practical are our results?

- minimum hitting set is NP-hard
- number of cut landmarks is exponential

We show how to apply our results to derive

- polynomial heuristics which
- dominate the LM-cut heuristic

LM-cut heuristic

- Computes a collection of landmarks by using pcfs that choose preconditions maximizing h^{max}
- Derived landmarks are pairwise disjoint
- Thus, costs can be combined (admissibly) with addition

Improved LM-cut

Improve the LM-cut heuristic by

1. Generating more landmarks:
   - Perform the LM-cut computation p times (parameter)
   - Use random tie-breaking to make runs different
   - Collect all generated landmarks in a set $\mathcal{L}$.
2. Exploiting them in a smarter way:
   - Introduce a width parameter k for hitting set instances such that MHS is fixed-parameter tractable w.r.t. k
   - Remove some landmarks from $\mathcal{L}$ to bound the width
   - Solve resulting MHS problem in polynomial time

Preliminary experiments

#	LM-cut	$h_{p,k}^{\text{LM-cut}}$ with $k=5$ $p=3$	$p=4$	$p=5$	$h_{p,k}^{\text{LM-cut}}$ with $k=10$ $p=3$	$p=4$	$p=5$	$h_{p,k}^{\text{LM-cut}}$ with $k=15$ $p=3$	$p=4$	$p=5$
Pipesworld-NoTankage (rel. error of LM-cut wrt $h^+ = 19.45\%$)										
06	107	45.8	54.2	67.3	49.5	54.2	68.2	49.5	54.2	68.2
07	3	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
08	84	47.6	57.1	81.0	58.3	75.0	76.2	58.3	75.0	76.2
10	137,092	30.2	40.1	46.9	32.9	43.9	50.0	33.7	47.0	55.1
Pipesworld-Tankage (rel. error of LM-cut wrt $h^+ = 18.42\%$)										
05	74	58.1	70.3	70.3	58.1	67.6	70.3	58.1	67.6	70.3
06	223	41.7	52.0	60.5	43.0	55.6	70.0	43.0	55.6	70.0
07	323	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
08	36,203	77.3	84.9	87.6	77.5	85.0	88.2	77.9	85.8	89.2
Openstacks (rel. error of LM-cut wrt $h^+ = 18.09\%$)										
04	1,195	53.4	57.8	59.0	58.5	63.9	66.7	63.7	66.8	71.5
05	1,195	52.6	57.4	59.7	58.8	65.0	66.6	61.5	65.6	69.8
06	211,175	64.6	64.9	65.2	69.0	70.7	71.7	69.8	71.2	72.0
07	266,865	60.7	61.3	61.8	65.1	66.4	67.2	65.4	66.8	67.3
Freecell (rel. error of LM-cut wrt $h^+ = 13.92\%$)										
pf4	36,603	70.7	75.2	78.4	70.3	76.3	79.6	72.3	77.3	79.8
pf5	53,670	73.6	76.0	77.9	74.4	77.1	78.8	75.0	77.6	79.3
2-5	277	72.9	73.3	74.0	72.9	73.3	74.0	72.9	73.3	74.0
3-4	17,763	44.6	62.8	73.1	44.7	62.8	72.1	44.7	62.6	72.1

Conclusion

Conclusion

Summary:

- Hitting sets for landmarks are more informative than optimal cost partitioning
- Cuts in justification graphs offer a principled and complete method for generating landmarks
- Hitting sets over all cut landmarks are perfect heuristics for delete relaxations
- These concepts can be exploited in practical heuristics