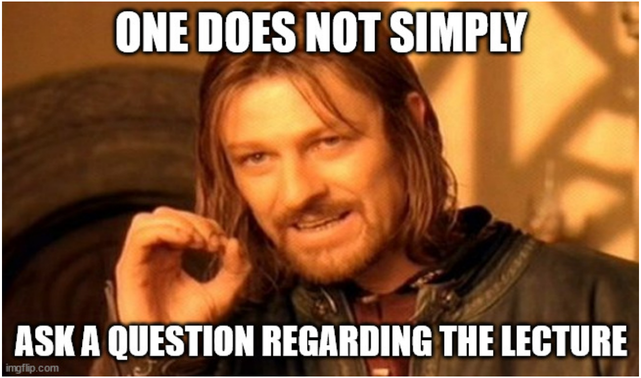
Landmark heuristics

Michaela Urbanovská

PUI Tutorial
Week 4

Lecture check

- Any questions regarding the lecture?



Feedback

- Specific suggestions on how to improve the lectures (I passed that information)
- Overall good scores
- Thank you!

LM-Cut

LM-Cut Heuristic

- Relaxation heuristic
- Uses disjunctive operator landmarks
- Admissible (and actually very successful) heuristic

Things we need to know to compute it

- Disjunctive operator landmark
- Operator supporter
- Justification graph
- s-t cut

LM-Cut

Disjunctive operator landmark

Disjunctive operator landmark $L \subseteq O$ is set of operators such that every plan π contains at least one operator $o \in L$.

Example

Suppose we have problem with 2 existing plans:

- $\pi_1 = (o_1, o_2, o_3, o_4)$
- $\pi_2 = (o_1, o_5, o_2, o_6)$

Which of the following sets are disjunctive operator landmarks?

1. $\{o_1\}$
2. $\{o_1, o_3\}$
3. $\{o_2, o_3\}$

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

$\{o_3, o_4\}$

$\{o_4, o_6\}$

LM-Cut

Disjunctive operator landmark

Disjunctive operator landmark $L \subseteq O$ is set of operators such that every plan π contains at least one operator $o \in L$.

Example

Suppose we have problem with 2 existing plans:

- $\pi_1 = (o_1, o_2, o_3, o_4)$
- $\pi_2 = (o_1, o_5, o_2, o_6)$

Which of the following sets are disjunctive operator landmarks?

- $\{o_1\}$
- $\{o_1, o_3\}$
- $\{o_2, o_3\}$
- $\{o_1, o_2, o_3, o_4\}$
- $\{o_3, o_4\}$
- $\{o_4, o_6\}$

LM-Cut

Delta function

- Function Δ_1 from previous tutorial (h^{max})
- $\Delta_1(s, f) =$
$$\begin{cases} 0 & \text{if } f \in s, \\ \text{inf} & \text{if } \forall o \in O : f \notin add(o), \\ min\{c(o) + \Delta_1(s, o) | o \in O, f \in add(o)\} & otherwise. \end{cases}$$

LM-Cut

Supporter of an operator

Supporter is a **fact**.
Function $supp(o) = argmax_{f \in pre(o)} \Delta_1(s, f)$ maps each operator $o \in O$ to its **supporter**.
(s denotes the state where we compute the heuristic estimate)

LM-Cut

Justification graph

$G = (N, E)$ is a **directed labeled** multigraph.

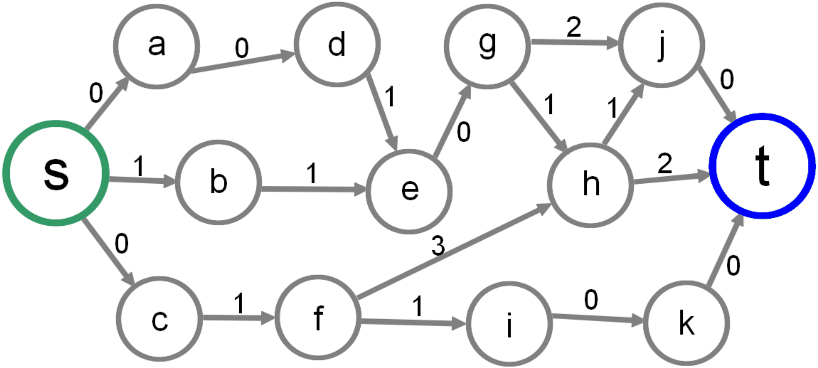
- $N = \{n_f | f \in F\}$ *(set of nodes)*
- $E = \{(n_s, n_t, o) | o \in O, s = supp(o), t \in add(o)\}$ *(set of edges)*
- Edge $e = (a, b, l)$ denotes edge from a to b with label l

LM-Cut

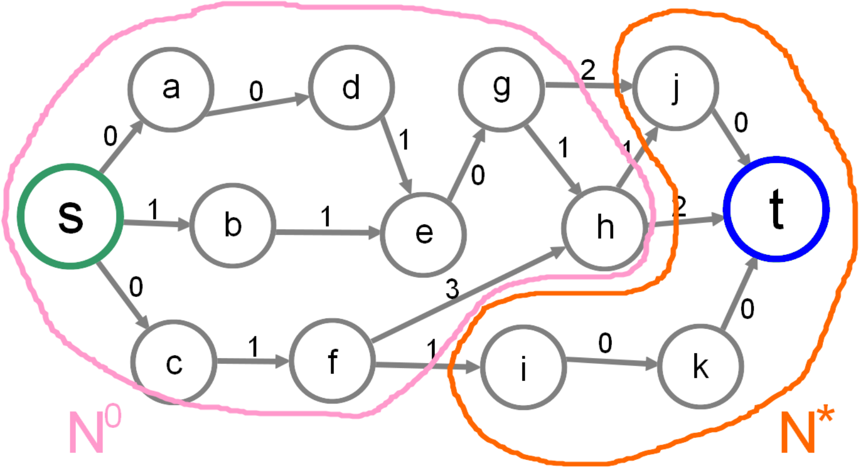
s-t cut

- **s-t cut** $C(G, s, t) = (N^0, N^* \cup N^b)$
- partitioning of nodes from the **justification graph** $G = (N, E)$
- N^* contains nodes from which t can be reached with zero-cost path
- N^0 contains nodes which can be reached from s without passing any node from N^*
- $N^b = N \setminus (N^0 \cup N^*)$ *(all the other nodes)*

LM-Cut



LM-Cut



Example

Compute $h^{lm-cut}(s_{init})$ for problem $\Pi = \langle F, O, s_{init}, s_{goal}, c \rangle$:

$F = \{s, q_1, q_2, q_3, t\}$

$s_{init} = \{s\}$

$s_{goal} = \{t\}$

$O =$

	pre	add	del	c
o_1	s	q_1, q_2	$\emptyset$	1
o_2	s	q_1, q_3	$\emptyset$	1
o_3	s	q_2, q_3	$\emptyset$	1
fin	q_1, q_2, q_3	t	$\emptyset$	0

LM-Cut

Algorithm 2: Algorithm for computing $h^{\text{lm-cut}}(s)$.

Input: $\Pi = \langle \mathcal{F}, \mathcal{O}, s_{init}, s_{goal}, c \rangle$, state s
Output: $h^{\text{lm-cut}}(s)$

if $h^{\max}(\Pi, s_{init}) = \infty$ **then**
    $h^{\text{lm-cut}}(s) \leftarrow \infty$ and terminate;
end
$h^{\text{lm-cut}}(s) \leftarrow 0$;
$\Pi_1 = \langle \mathcal{F}' = \mathcal{F} \cup \{I, G\}, \mathcal{O}' = \mathcal{O} \cup \{o_{init}, o_{goal}\}, s'_{init} = \{I\}, s'_{goal} = \{G\}, c_1 \rangle$, where $\text{pre}(o_{init}) = \{I\}$, $\text{add}(o_{init}) = s$, $\text{del}(o_{init}) = \emptyset$, $\text{pre}(o_{goal}) = s_{goal}$, $\text{add}(o_{goal}) = \{G\}$, $\text{del}(o_{goal}) = \emptyset$, $c_1(o_{init}) = 0$, $c_1(o_{goal}) = 0$, and $c_1(o) = c(o)$ for all $o \in \mathcal{O}$;
$i \leftarrow 1$;
while $h^{\max}(\Pi_i, s'_{init}) \neq 0$ **do**
    Construct a justification graph G_i from Π_i;
    Construct an s-t-cut $\mathcal{C}_i(G_i, n_I, n_G) = (N_i^0, N_i^\star \cup N_i^b)$;
    Create a landmark L_i as a set of labels of edges that cross the cut $\mathcal{C}_i$, i.e., they lead from N_i^0 to $N_i^\star$;
    $m_i \leftarrow \min_{o \in L_i} c_i(o)$;
    $h^{\text{lm-cut}}(s) \leftarrow h^{\text{lm-cut}}(s) + m_i$;
    Set $\Pi_{i+1} = \langle \mathcal{F}', \mathcal{O}', s'_{init}, s'_{goal}, c_{i+1} \rangle$, where $c_{i+1}(o) = c_i(o) - m_i$ if $o \in L_i$, and $c_{i+1}(o) = c_i(o)$ otherwise;
    $i \leftarrow i + 1$;
end

Example 2

Compute $h^{lm-cut}(s_{init})$ for problem $\Pi = \langle F, O, s_{init}, s_{goal}, c \rangle$:

$F = \{a, b, c, d, e, i, g\}$

$s_{init} = \{i\}$

$s_{goal} = \{g\}$

$O =$

	pre	add	del	c
o_1	i	a,b	$\emptyset$	2
o_2	i	b,c	$\emptyset$	3
o_3	a,c	d	c	1
o_4	b,d	e	b	3
o_5	a,c,e	g	c,d	1
o_6	a	e	a,c	5

LM-Cut

Algorithm 2: Algorithm for computing $\mathrm{h}^{\text{lm-cut}}(s)$.

Input: $\Pi = \langle \mathcal{F}, \mathcal{O}, s_{init}, s_{goal}, \mathrm{c} \rangle$, state s
Output: $\mathrm{h}^{\text{lm-cut}}(s)$

if $\mathrm{h}^{\max}(\Pi, s_{init}) = \infty$ **then**
  $\mathrm{h}^{\text{lm-cut}}(s) \leftarrow \infty$ and terminate;
end
$\mathrm{h}^{\text{lm-cut}}(s) \leftarrow 0$;
$\Pi_1 = \langle \mathcal{F}' = \mathcal{F} \cup \{I, G\}, \mathcal{O}' = \mathcal{O} \cup \{o_{init}, o_{goal}\}, s'_{init} = \{I\}, s'_{goal} = \{G\}, \mathrm{c}_1 \rangle$, where $\mathrm{pre}(o_{init}) = \{I\}$, $\mathrm{add}(o_{init}) = s$, $\mathrm{del}(o_{init}) = \emptyset$, $\mathrm{pre}(o_{goal}) = s_{goal}$, $\mathrm{add}(o_{goal}) = \{G\}$, $\mathrm{del}(o_{goal}) = \emptyset$, $\mathrm{c}_1(o_{init}) = 0$, $\mathrm{c}_1(o_{goal}) = 0$, and $\mathrm{c}_1(o) = \mathrm{c}(o)$ for all $o \in \mathcal{O}$;
$i \leftarrow 1$;
while $\mathrm{h}^{\max}(\Pi_i, s'_{init}) \neq 0$ **do**
  Construct a justification graph G_i from Π_i;
  Construct an s-t-cut $\mathcal{C}_i(G_i, n_I, n_G) = (N_i^0, N_i^\star \cup N_i^b)$;
  Create a landmark L_i as a set of labels of edges that cross the cut $\mathcal{C}_i$, i.e., they lead from N_i^0 to $N_i^\star$;
  $m_i \leftarrow \min_{o \in L_i} \mathrm{c}_i(o)$;
  $\mathrm{h}^{\text{lm-cut}}(s) \leftarrow \mathrm{h}^{\text{lm-cut}}(s) + m_i$;
  Set $\Pi_{i+1} = \langle \mathcal{F}', \mathcal{O}', s'_{init}, s'_{goal}, \mathrm{c}_{i+1} \rangle$, where $\mathrm{c}_{i+1}(o) = \mathrm{c}_i(o) - m_i$ if $o \in L_i$, and $\mathrm{c}_{i+1}(o) = \mathrm{c}_i(o)$ otherwise;
  $i \leftarrow i + 1$;
end

Recap

- You should be able to compute and implement LM-Cut
- Everything you need to do the assignment 1!

The End

Feedback form link