Landmarks - Bibiliograhie

DCSD/CD

Simon Vernhes

Onera

21 *novembre* 2011

ONERA
THE FRENCH AEROSPACE LAB

Sommaire

ONERA
THE FRENCH AEROSPACE LAB

Définitions

ONERA
THE FRENCH AEROSPACE LAB

Landmark

Definition (Landmark)

Soit (A, I, G) une tâche de planification. Un fluent L est un landmark si $\forall P = \langle a_1, \ \dots \ , a_n \rangle \in A^*, G \subset \mathit{Result}(I, P)$
$(\exists i \in \{1, \ \dots \ , n\}) \quad L \in \mathit{Result}(I, \langle a_1, \dots, a_i \rangle)$

C'est-à-dire

L appartient à au moins un état de tous les plans permettant d'aller de l'état initial I à l'état but G.

ONERA
THE FRENCH AEROSPACE LAB

Necessary Order

Definition ($L \rightarrow_n L'$)

Soient (A, I, G) une tâche de planification et deux landmarks L et L'. On dit qu'il existe un necessary order entre L et L' ($L \rightarrow_n L'$) si et seulement si
$L' \notin I$ et $\forall P = \langle a_1, \ldots, a_n \rangle \in A^*, L' \in \mathit{Result}(I, P)$
$L \in \mathit{Result}(I, \langle a_1, \ldots, a_{n-1} \rangle)$

C'est-à-dire

Pour toutes les séquences d'actions qui amènent à un état contenant L', l'avant dernier état contient nécessairement L.

ONERA
THE FRENCH AEROSPACE LAB

Greedy Necessary

Definition ($L \rightarrow_{gn} L'$)

Soient (A, I, G) une tâche de planification et deux landmarks L et L'. On dit qu'il existe un greedy necessary order entre L et L' $(L \rightarrow_{gn} L')$ si et seulement si
$L' \notin I$ et $\forall P = \langle a_1, \ldots, a_n \rangle \in A^*, L' \in Result(I, P)$ et
$(\forall i \in \{1, \ldots, n-1\}) L' \notin Result(I, \langle a_1, \ldots, a_i \rangle)$
$L \in Result(I, \langle a_1, \ldots, a_{n-1} \rangle)$

C'est-à-dire

Pour toutes les séquences d'actions qui amènent pour la première fois à un état contenant L', l'avant dernier état contient nécessairement L.

ONERA
THE FRENCH AEROSPACE LAB

Reasonable order

Definition ($S_{(L',\neg L)}$)

$S_{(L',\neg L)}$ est l'ensemble de tout les états où L' vient d'être ajouté mais où L n'est jamais encore apparu.

Definition (Aftermath)

L' est un effet/une suite (in the aftermath) de L si depuis tous les $s \in S_{(L',\neg L)}$, L devient vrai dans tous les plans solutions $P \in A^*$ tel que $G \in Result(P, s)$.

Definition ($L \rightarrow_r L'$)

On dit que $L \rightarrow_r L'$ si

$$\begin{cases} L' \text{ est un effet de } L \\ (\forall s \in S_{(L',\neg L)})(\forall P \in A^* \text{ achevant } L) P \text{ supprime nécessairement } L' \end{cases}$$

Reasonable order

Un ordre $L \rightarrow_r L'$ ne veut pas dire qu'il faut absolument obtenir L avant L' mais que ça semble raisonnable.
En effet, on pourrait être obligé de détruire L' pour construire L.

ONERA
THE FRENCH AEROSPACE LAB

Obedient Reasonable Orders

Example

Si on a $L' \rightarrow_n L''$, $L \rightarrow_r L''$ et que les seules actions permettant de produire L détruisent L'.

TODO

Complexité

Décidabilité

- Landmark – PSPACE-complete
- $\rightarrow_n$ – PSPACE-complete
- $\rightarrow_{gn}$ – PSPACE-complete
- $\rightarrow_r$ – PSPACE-complete

ONERA
THE FRENCH AEROSPACE LAB

Algorithmes

ONERA
THE FRENCH AEROSPACE LAB

Landmark

En utilisant un graphe de planification relaxé (RPG), on construit un graphe de génération de landmark. On part des buts, et on remonte.

Algorithm 1: Landmark Generation Graph

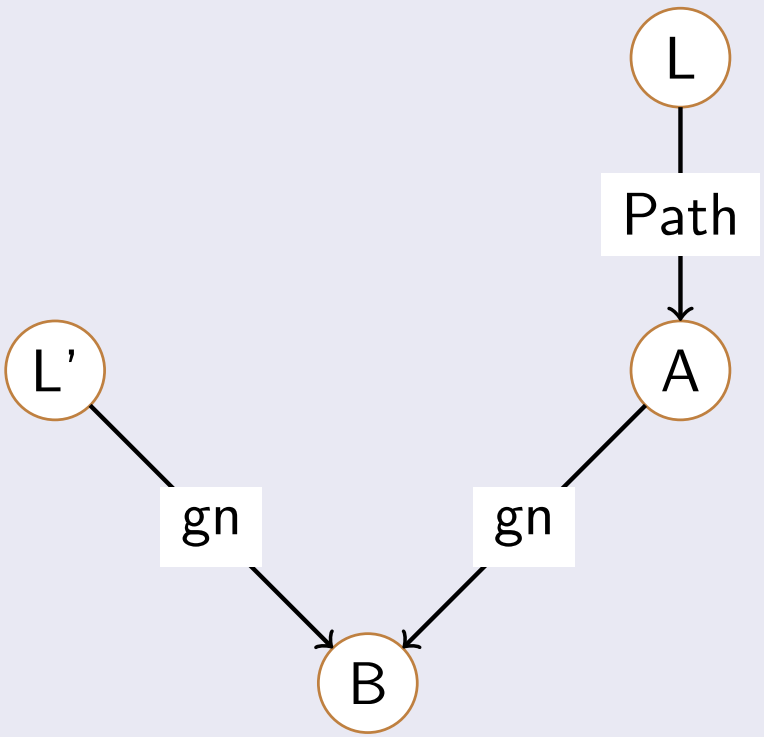
input : (A, O, I, G) a planning task
output: LGG = (N, E) où N est l'ensemble des noeuds, et E les arcs de l'arbre
$LGG = (N, E) \leftarrow (G, \emptyset)$;
$C \leftarrow G$;
while $C \neq \emptyset$ **do**
    $C' \leftarrow \emptyset$;
    foreach $(L' \in C)$ *tel que* $level(L') \neq \emptyset$ **do**
        $A \leftarrow \{(\forall a \in A) \mid L' \in add(a)$ et $level(a) = level(L) - 1\}$;
        foreach L *tel que* $(\forall a \in A)\ L \in pre(a)$ **do**
            if $(L \in O) \notin LGG$ **then**
                $C' \leftarrow C' \cup \{L\}$;
                Insérer L dans LGG;
            if $L \rightarrow_{gn} L' \notin LGG$ **then** Insérer $L \rightarrow_{gn} L'$ dans LGG;
    $C \leftarrow C'$

On obtient pas forcément des landmarks ! (à cause de du test level RPG, ça enlève les actions non présente dans le RPG). En fait, ça permet d'obtenir des pseudo-landmarks (il y en a plus). Pour une recheche heuristique, ça aide (mais est-ce que ça apporte vraiment plus qu'un simple RPG ?).

ONERA
THE FRENCH AEROSPACE LAB

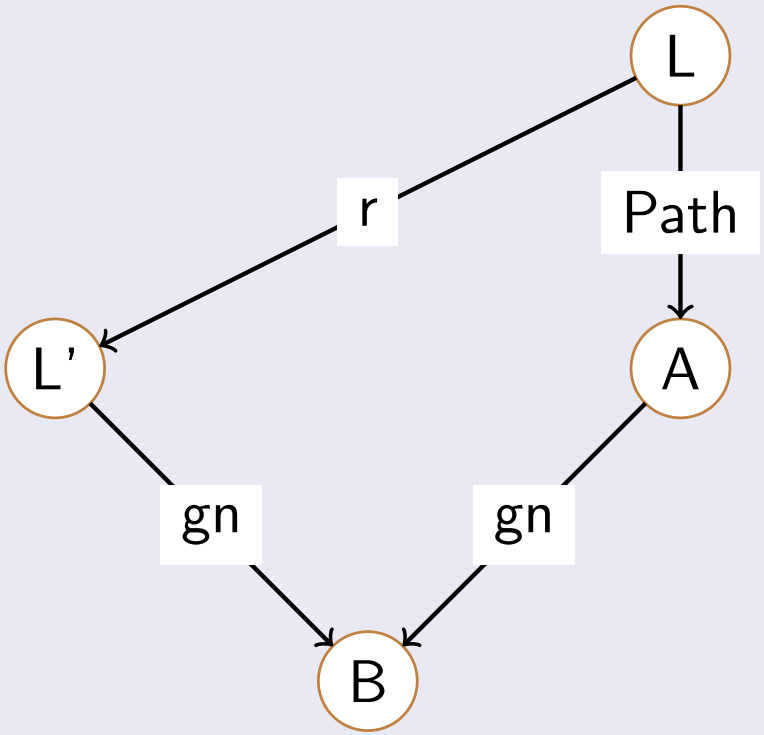
$L \rightarrow_r L'$

Interférence entre 2 landmarks (en gros, des mutexes)



$L \rightarrow_r L'$

Interférence entre 2 landmarks (en gros, des mutexes)



Utilisation

ONERA
THE FRENCH AEROSPACE LAB

Interdiction des actions qui ajoute un landmark qui a des arcs entrants.

Julie Porteous and Laura Sebastia.
Extracting and ordering landmarks for planning.
In *Proceedings UK Planning and Scheduling SIG Workshop*, 2000.

ONERA
THE FRENCH AEROSPACE LAB

Décomposions en sous tâches

Appeler successivement un planification avec comme but : le but du problème ou la disjonction des prochains landmark.
Dès qu'il termine, s'il n'a pas atteint le but, on le fait repartir de l'état final de la même manière.
Et on itère jusqu'à avoir atteint le but.

Jörg Hoffmann, Julie Porteous, and Laura Sebastia.
Ordered landmarks in planning.
Journal of Artificial Intelligence Research, 22 :215–278, 2004.

ONERA
THE FRENCH AEROSPACE LAB