Annexe : Preuve des propriétés de A*

Rappel sur A* : On insère les noeuds dans OPEN selon une fonction $f(n) = g(n) + h(n)$, avec

- $g(n)$ le *coût du chemin optimal* du noeud initial au neoud n.
- $h(n)$ l'*estimation* du coût additionel pour atteindre le but à partir de n.
- $h(n) \leq h^*(n)$, $\forall n$, avec $h^*(n)$ le *vrai coût*.

Remarque : $h^*(n) \geq 0$, et $c(n_1, n_2) > 0$, avec $c(n_1, n_2)$ le coût d'un arc. Donc, $g(n) \geq 0$.

Théorème 1
L'algorithme *recherche* termine toujours pour des graphes finis (indépendamment de la stratégie, profondeur, largeur, A^*).

Preuve :

L'algorithme termine quand il n'y a plus de noeuds dans $OPEN$. Ceci arrivera tôt ou tard, puisque à chaque étape on enlève un noeud de $OPEN$, et on ne le remet jamais dans $OPEN$.

Lemme 1

Si une solution existe, alors à tout instant avant la terminaison de A*, il existe dans $OPEN$ un noeud n' se trouvant sur un chemin optimal de s au but, avec $f(n') \leq f^*(s)$.

Preuve :

Soit $n_0, n_1, \ldots, n_k$, avec $n_0 = s$ et n_k un noeud objectif, une séquence optimale.
A tout instant précédent la terminaison de A*, soit n' le premier noeud de cette séquence se trouvant dans $OPEN$.
Par définition de f :

$$f(n') = g(n') + h(n').$$

Puisque la sous-séquence jusqu'à n' a été explorée, et vu qu'elle est optimale

$$g(n') = g^*(n').$$

Donc

$$f(n') = g^*(n') + h(n').$$

Puisque $h(n') \leq h^*(n')$, par définition de A*, on en déduit

$$f(n') \leq g'(n) + h^*(n') = f^*(n').$$

Or, $f^*(n) = f^*(s)$ pour tout noeud n sur un chemin optimal.
Donc, $f(n') \leq f^*(s)$.
On en déduit $f(n') \leq f^*(s)$.

Theorem 2
S'il existe un chemin de s au but, A* termine tôt ou tard.

Preuve :

Soit $d^*(n)$ le nombre d'arcs du chemin le plus court de s à n, pour un noeud donné n dans l'espace d'état.
Soit ϵ le plus petit de coûts des arcs.
Puisque $\epsilon > 0$, nous avons $g^*(n) \geq d^*(n) \times \epsilon$.
Puisque $g(n) \geq g^*(n)$ et $h(n) \geq 0$ (donc $f(n) \geq g(n)$), on a $f(n) \geq d^*(n) \times \epsilon$.
Donc si A* ne termine pas, tous les noeuds dans OPEN auront tôt ou tard une valeur $d^*(n)$ arbitrairement grande, donc une valeur $f(n)$ arbitrairement grande.
Ceci est en contradiction avec lemme 1.

Théorème 3
A* est *admissible*, c-à-d., trouve un chemin optimal si une solution existe.

Preuve :

Si une solution existe, A* termine sur un noeud objectif.
En effet, A* termine quand OPEN est vide ou quand un noeud objectif est choisi pour expansion. Le premièr cas ne peut pas arriver d'après lemme 1, donc seul le deuxième cas est possible.

Si une solution existe, A* termine toujours avec une solution optimale.

Supposons que A* termine sur un noeud n qui n'est pas optimal.

Donc, $f(n) = g(n) > f^*(s)$.

Or, lemme 1.6.2, dit qu'il doit exister un noeud n' tel que $f(n') \leq f^*(s) < f(n)$.

Donc, à cet étape, A* aurait dû choisir n' pour l'expansion, plutôt que n.

Ceci contradit le fait que A* ait terminé.

Donc A* est admissible.

Théorème 4

Pour chaque noeud n, qui est choisi pour être expansé dans A^*, on a toujours $f(n) \leq f^*(s)$.

Preuve :

Si n est un noeud objectif, on a $f(n) = f^*(s)$ d'après le théorème précédent.
Sinon, si n est choisi pour l'expansion, d'après l'algorithme, cela veut dire que n a la plus petite valeur de f.
D'après lemme 1, il y a un noeud n' dans OPEN tel que $f(n') \leq f^*(s)$, toujours avant terminaison.
Donc, $f(n) \leq f(n') \leq f^*(s)$.

Heuristique Monotone

h est dit *monotone* $\forall n_i, n_j$, tel que n_j est successeur de n_i :

$$h(n_i) - h(n_j) \leq c(n_i, n_j).$$

Avec cette condition :

- A* devient plus efficace et plus simple.
- A* devient incrémental et peut donc être utilisé dans des systèmes temps-réel.

Théorème 5
Si h est monotone, alors chaque fois que A* choisit un noeud pour expanser, cela signifie qu'il a déjà trouvé un chemin optimal vers ce noeud.

Preuve :
Soit $P = (n_0, n_1, \ldots, n_k)$, avec $n_0 = s$ et $n_k = n$, un chemin optimal de s à n.
Donc, $g^*(n) = \sum_{i=o}^{n-1} c(n_i, n_{i+1})$
Soit n_{i-1} le dernier dans la séquence qui est dans CLOSED lorsque A* selectionne n pour l'expansion.
Donc, n_i est dans OPEN quand S* selectionne n.
Donc, $f(n) \leq f(n_i)$
D'où, $g(n) + h(n) \leq g(n_i) + h(n_i)$,
c-à-d., $g(n) \leq g(n_i) + h(n_i) - h(n)$.

Vu que $g(n_i) = g^*(n_i)$, on en déduit $g(n) \leq g^*(n_i) + h(n_i) - h(n)$.
Or,

$$\begin{gathered} h(n_i) - h(n_{i+1}) \leq c(n_i, n_{i+1}) \\ h(n_{i+1}) - h(n_{i+2}) \leq c(n_{i+1}, n_{i+2}) \\ \vdots \\ h(n_{k-1}) - h(n) \leq c(n_{k-1}, n), \end{gathered}$$

d'où on obtient

$$h(n_i) - h(n) \leq \sum_{j=i}^{k-1} c(n_j, n_{j+1}).$$

Donc, $g(n) \leq g^*(n_i) + \sum_{j=i}^{k-1} c(n_j, n_{j+1})$,
c-à-d., $g(n) \leq g^*(n)$.

Théorème 6
Si h est monotonne, alors les valeurs f pour la séquence des noeuds expansés par A* sont non-décroissant.

Preuve :
Supposons que n_2 est expansé juste après n_1.
Si n_2 était sur $OPEN$ lorsque n_1 fut expansé, alors on a $f(n_1) \leq f(n_2)$.
Sinon, cela veut dire que n_2 est un successeur immédiat de n_1.
Puisque n_2 est choisi pour l'expansion, on a :

$$\begin{aligned}
f(n_2) \quad &= g(n_2) + h(n_2) \\
&= g^*(n_2) + h(n_2) \text{ Thm. 5} \\
&= g^*(n_1) + c(n_1, n_2) + h(n_2) \\
&= g(n_1) + c(n_1, n_2) + h(n_2).
\end{aligned}$$

Or $c(n_1, n_2) + h(n_2) \geq h(n_1)$.
D'où nous obtenons $f(n_2) \geq g(n_1) + h(n_1) = f(n_1)$.

Exploration en temps réel

Si h est monotonne, chaque expansion réduit l'erreur dans l'estimation de l'optimalité du noeud. On a donc une recherche dont le résultat est amélioré incrémentalement.

Preuve :

Soit n_1 le noeud qu'on décide d'expanser.
L'erreur dans notre estimation de n_2 est

$$\begin{aligned} f^*(n_1) - f(n_1) &= g^*(n_1) + h^*(n_1) - g(n_1) - h(n_1) \\ &= h^*(n_1) - h(n_1) \ \text{ d'après théorème 5} \end{aligned}$$

Supposons qu'on a encore du temps pour expanser n.
Soit n_2 le successeur de n_1 ayant la plus petite valuer f
L'estimation du coût pour atteindre le but via n_1 devient

$$f(n_1) = g(n_1) + c(n_1, n_2) + h(n_2),$$

et l'erreur devient

$$\begin{aligned} & f^*(n_1) - (g(n_1) + c(n_1, n_2) + h(n_2)) \\ & = h^*(n_1) - (c(n_1, n_2) + h(n_2)) \\ & \leq h^*(n_1) - h(n_1) \;\; \text{d'après la monotonie} \end{aligned}$$

C-à-d., l'erreur dans une exploration vers l'avant décroit avec l'avancement.

La méthode dite *exploration bornée* (*bounded look-ahead*) est basée sur une telle notion de monotonie. L'idée est d'expanser les noeuds seulement jusqu'à une certaine distance du but.

Heuristique non-admissibles

Soit la restriction

$$h(n) \leq h^*(n) + \epsilon,$$

avec ϵ une constante.

Si une solution existe, A* termine toujours avec un but n, tel que

$$f^*(n) \leq f^*(s) + \epsilon.$$

Preuve :

Supposons que A* termine avec n qui n'est pas sur un chemin optimal.
A tout instant avant que A* termine, il existe dans OPEN, un

noeud n' sur un chemin optimal, donc avec $g(n') = g^*(n')$.

Puisque n a été expansé avant n', on doit avoir $f(n) \leq f(n')$.

C-à-d., $g(n) + h(n) \leq g(n') + h(n') \quad (1)$.

Or, on sait que toujours $g^*(n) \leq g(n)$.

Donc, on a $g^*(n) + h(n) \leq g(n) + h(n)$,

Ce qui, combiné avec (1) et le fait que $h(n) \geq 0$, donne

$g^*(n) \leq g(n') + h(n')$.

Puisque $h^*(n) = 0$ (vu que n est un but), on a $f^*(n) = g^*(n)$.

Donc, $f^*(n) \leq g(n') + h(n')$.

Or $g(n') = g^*(n')$, et $h(n) \leq h^*(n) + \epsilon$.

Donc, $f^*(n) \leq g^*(n') + h^*(n') + \epsilon$.

C-à-d., $f^*(n) \leq f^*(s) + \epsilon$, puisque n' est sur un chemin optimal.