

TP : Intelligence Artificielle : Jeu du taquin (Partie 2)

Jeudi 13 novembre 2014

Rappels des objectifs

L'objectif final de ce travail pratique est de résoudre pleinement le problème du taquin 4x4 en utilisant l'algorithme IDA* avec des heuristiques avancées. Vous utiliserez le langage C. Bien sûr, il n'y a pas de restrictions sur les langages pour les interfaces (pour saisir les données ou afficher le déroulement du jeu).

Une petite adaptation

Pour commencer, adapter votre algorithme A*, réalisé lors du TP précédent, pour résoudre des taquins 4x4. Vous allez vite vous rendre compte que certains taquins sont impossibles à résoudre avec les deux heuristiques évoquées dans le TP précédent, comme celui donné dans la table 1 :

Table 1: Taquin 4x4 difficile à résoudre

15	11	8	12
14	10	9	13
2	6	1	4
3	7	5	

Pour y aller plus loin, il faut de nouvelles heuristiques. Je rappelle que plus l'heuristique est proche du coût réel d'une solution (tout en restant admissible) plus sera court le temps nécessaire à IDA* (ou à A*) de trouver une solution. Plusieurs nouvelles heuristiques ont été proposées dans la littérature. L'inconvénient majeur rencontré est que le temps gagné par ces nouvelles heuristiques est souvent balancé par le temps nécessaire au calcul des heuristiques elles-mêmes. L'idée est alors de faire un pré-traitement (une sorte de compilation) où les valeurs des heuristiques des taquins sont calculées au préalable. Cependant, le nombre de taquins 4x4 possibles est trop large pour être tous répertoriés dans un tableau (ou une base de données). Il suffit alors d'enregistrer un ensemble de taquins partiellement spécifiés. On parle alors de base de données (ou une compilation) de motifs "pattern databases" (ou de signatures) de taquins. Les heuristiques sont calculées au préalable et enregistrées dans une base de données.

Avant de résoudre entièrement le problème du taquin 4x4, dans ce TP on ne s'intéresse qu'à la résolution des taquins partiellement spécifiés.

Taquins partiellement spécifiés ...

Le travail demandé dans ce TP est de programmer A* et IDA* sur des taquins 4x4 partiellement spécifiés. Un exemple de taquin partiellement spécifié, appelé Frange (The "Fringe"), est donnée dans la table 2. Il s'agit d'un taquin partiellement spécifiée car certaines cases ne contiennent pas encore de valeurs : seuls la position des nombres {3, 7, 11, 12, 13, 14, 15} ainsi que la position de la case "blanc" sont précisées. Le reste des nombres n'est pas encore placé. Un taquin de type Frange va être vu comme un taquin but mais partiellement spécifié.

Le travail demandé est le calcul du coût réel nécessaire, en utilisant A* et IDA*,

Table 2: Taquin partiellement spécifié appelé Frange

	x	x	3
x	x	x	7
x	x	x	11
12	13	14	15

- pour passer d'une permutation quelconque d'un taquin Frange (un taquin 4x4 ordinaire) vers le taquin Frange donné dans 2.
- pour passer d'une permutation quelconque d'un taquin Frange (un taquin 4x4 ordinaire) vers un autre taquin Frange donné dans 3:

Table 3: Exemple d'un autre taquin but partiellement spécifié

	1	2	x
4	5	6	x
8	9	10	x
x	x	x	x

On utilisera les trois heuristiques suivantes :

- h_0 : La valeur heuristique associée à chaque état est égal à 0 (on retrouve alors l'algorithme de recherche en coût uniforme ou encore l'algorithme de recherche en largeur car les coûts sont tous identiques).
- h_1 : Nombre de cases qui ne sont pas dans leurs positions initiales.
- h_2 : La distance de Manhattan.

Pour chaque solution obtenue, il est demandé de fournir le nombre de déplacements obtenus dans le cas de taquins non résolus. Nous utiliserons l'exemple donné dans la table 4 pour illustrer vos algorithmes.

Table 4: Exemple de taquin à tester

5	10	14	7
8	3	6	1
15		12	9
2	11	4	13

Remarque : La différence importante entre la résolution d'un taquin 4x4 et celui partiellement spécifié est la définition de l'état but : dans un taquin partiellement spécifié seules les cases du "Fringe" doivent être à leur place définitive.