

TP 5 : Introduction à l'Intelligence Artificielle

2020/2021

Objectifs : L'objectif de ce travail pratique (noté) est de résoudre le problème du taquin 4x4 partiel en utilisant l'algorithme IDA*.

Langages de programmation : Il est préférable d'utiliser le langage C (le langage Python est accepté).

Rappels des définitions des taquins complets/partiels

- Un taquin **but complet** (ou simplement taquin but) est un taquin où toutes les cases sont à leurs positions finales. Pour ce TP, nous utiliserons le taquin but suivant :

■	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

Taquin but complet

- Un taquin **complet initial** ou de départ est une permutation du taquin but complet. Il s'agit d'un taquin que l'on cherche à résoudre. Par exemple, :

15	11	8	12
14	10	9	13
2	6	1	4
3	7	5	■

Taquin initial complet

- Un taquin **but partiel** est un taquin où seules certaines cases (dans la suite + la case "blanc") doivent-être à leur place. Voici deux exemples de taquins buts partiels qui seront utilisés dans ce TP :

■	x	x	3
x	x	x	7
x	x	x	11
12	13	14	15

Taquin but partiel bp1

■	1	2	x
4	5	6	x
8	9	10	x
x	x	x	x

Taquin but partiel bp2

L'heuristique h à utiliser : Pour l'heuristique admissible h on peut simplement utiliser la somme des distances de Manhattan associées aux cases spécifiées (sauf la case vide) dans un taquin but partiel. Pour rappel, un taquin but complet est un taquin but partiel où toutes les cases sont spécifiées.

Les fonctions coûts g à utiliser :

Pour la fonction coût, nous utiliserons les deux possibilités suivantes :

- La première possibilité consiste à prendre en compte tous les déplacements de la case "blanc". C'est-à-dire si e' est un successeur de e (suite à un déplacement de la case "blanc") alors :

$$g_1(e') = g_1(e) + 1$$

- La deuxième possibilité ne tient compte que des déplacements qui impliqueraient les cases spécifiées dans le taquin but partiel. C'est-à-dire si e' est un successeur de e (suite à un déplacement de la case "blanc") alors :

$$g_2(e') = \begin{cases} g_2(e) + 1 & \text{si } e' \text{ est obtenu en déplaçant une case spécifiée par le taquin but partiel} \\ g_2(e) & \text{sinon} \end{cases}$$

Travail demandé

D'abord, il est important d'implémenter les fonctions de pré-traitements (une adaptation des fonctions déjà données dans les TPs précédents) :

1. Ecrire une fonction qui vérifie si un taquin 4x4 est solvable ou non (un cours sur ce sujet est disponible sous Moodle).
2. Ecrire une fonction qui vérifie si un taquin 4x4 initial e est un taquin but partiel bp1 (resp. bp2) ou non. Je rappelle que l'on se focalise uniquement sur certaines cases spécifiées par les taquins buts partiels (bp1 & bp2).
3. Ecrire une fonction qui prend un taquin initial 4x4 en paramètre et calcule ses taquins successeurs. Un taquin successeur est obtenu depuis le taquin initial en déplaçant (quand cela est possible) la case vide vers la gauche, le haut, la droite, le bas respectivement.
4. Ecrire une fonction, qui prend en paramètres un taquin 4x4 et un taquin but partiel, et calcule l'heuristique basée sur la distance de Manhattan associée aux cases spécifiées dans le taquin but partiel.

Ensuite, on demande de réaliser les tâches suivantes :

1. Ecrire un programme qui implémente l'algorithme de recherche IDA* pour la résolution d'un taquin initial complet e vers un taquin but partiel. L'algorithme IDA* retourne le coût optimal (la plus petite valeur de la variable "seuil") de la résolution du taquin e .
2. On s'intéresse maintenant à l'évaluation votre programme qui sera testé i) d'une part avec les deux exemples de taquins buts partiels bp1 et bp2 décrits ci-dessus et ii) d'autre part

avec les deux fonctions coût g_1 et g_2 définis ci-dessus. Pour le résultat, on donne d'une part le coût optimal obtenu par l'algorithme IDA* (-1 si le taquin n'est pas solvable) ainsi qu'une séquence de résolution¹. Plus précisément :

- On demande d'abord de générer un échantillon de taquins 4x4 complets.
- Le résultat attendu de votre TP est un fichier résultat (resultat.txt) où chaque taquin initial complet e (généré à la première étape) lui est associé une ligne avec les dix champs suivants :
 - (a) Le premier champs contient le taquin initial complet e (les 16 chiffres du taquins, ligne par ligne. Le vide sera codé par le chiffre 0).
 - (b) Le deuxième champs contient la somme des distances de Manhattan associées à **toutes** les cases.
 - (c) Le troisième champs contient le coût optimal obtenu par l'algorithme IDA* pour la résolution du taquin e vers le taquin but partiel bp1 avec la fonction g_1
 - (d) Le quatrième champs contient une séquence de résolution par l'algorithme IDA* du taquin e vers le taquin but partiel bp1 avec la fonction g_1 .
 - (e) Le cinquième champs contient le coût optimal obtenu par l'algorithme IDA* pour la résolution du taquin e vers le taquin but partiel bp1 avec la fonction g_2 .
 - (f) Le sixième champs contient une séquence de résolution par l'algorithme IDA* du taquin e vers le taquin but partiel bp1 avec la fonction g_2 .
 - (g) Le septième champs contient le coût optimal obtenu par l'algorithme IDA* pour la résolution du taquin e vers le taquin but partiel bp2 avec la fonction g_1 .
 - (h) Le huitième champs contient une séquence de résolution par l'algorithme IDA* du taquin e vers le taquin but partiel bp2 avec la fonction g_1 .
 - (i) Le neuvième champs contient le coût optimal obtenu par l'algorithme IDA* pour la résolution du taquin e vers le taquin but partiel bp2 avec la fonction g_2 .
 - (j) Le dixième champs contient une séquence de résolution par l'algorithme IDA* du taquin e vers le taquin but partiel bp2 avec la fonction g_2 .

3. Les TPs/rapports sont à rendre pour le vendredi 04 décembre 12h (sous Moodle).

- Un seul rapport, de 2 à 4 pages, pour les deux TPs est demandé.
- Dans le TP4, on demande principalement de résumer (très rapidement) les fonctions codées, le nombre de taquins testés et un petit paragraphe sur l'utilisation de votre programme.
- Dans le TP5, on demande également une brève étude expérimentale pour comparer les différents résultats obtenus. Plus précisément, on vous demande de comparer les valeurs obtenues du champs 2 [item b] avec le maximum des champs 3 et 7 [items c & g], et la somme des champs 5 et 9 [items e & i] de votre fichier sortie.

¹c'est-à-dire une chaîne de caractères, composée de l'un des quatre caractères : "G" (Gauche), "D" (Droite), "H" (Haut) et "B" (Bas). Cette chaîne de caractères représente la séquence des déplacements de la case vide de l'état initial vers l'état but. Cette séquence permet de vérifier que votre solution est correcte (voir TPs précédents). On mettra une chaîne vide pour les taquins non-solvables.

Remarques :

- Il est important de réfléchir à une bonne structure de données et à l'algorithmique avant de se lancer dans la programmation.
- Les travaux pratiques doivent-être réalisés par vous-même (il n'est pas autorisé de reprendre un code existant; peu importe sa provenance).
- Les TPs notés peuvent-être faits en binôme.