Intelligence Artificielle

Projet 1 : Sliding Block Puzzles

Description du problème : On se propose dans ce projet de résoudre des *sliding block puzzles* ou puzzles de blocs coulissants. Chaque puzzle de cette classe contient des éléments qu'il faut faire coulisser jusqu'à l'obtention d'une configuration désirée.

On se concentrera sur des puzzles simples : un tel puzzle peut être vu comme une matrice de taille $n \times n$. Dans cette matrice apparait chaque nombre de l'ensemble $\{1, \ldots, n^2-1\}$, ainsi qu'un symbole particulier, le symbole vide, nommé *blank*. Une action du puzzle consiste à déplacer le symbole *blank* en lui faisant échanger sa position avec un symbole d'une case adjacente. La configuration désirée consiste à avoir *blank* dans la case $(0,0)$ (représentation (abscisse, ordonnée)), puis chaque numéro i dans la case $(i\%3, \lfloor i/3 \rfloor)$. Pour des raisons pédagogiques, on étudie une version du problème dans laquelle déplacer une case avec un numéro pair a un coût de 1 et une case avec un numéro impair a un coût de 2.

Voici ci-dessous, dans les Figures 1 et 2, des exemples de 8-puzzles ($n = 3$) et 15-puzzles ($n = 4$).

3	1	2
4	7	b
6	8	5

b	1	2
3	4	5
6	7	8

Figure 1: Un 8-puzzle ($n = 3$) mélangé à gauche et résolu à droite. Le symbole *blank* est représenté par la lettre b.

Travail demandé : Le travail s'effectue en binôme. Il s'agit de réaliser un projet informatique pour tester les méthodes de recherche du Cours 2 pour résoudre ces puzzles : *Breadth-first Search, Uniform-cost Search, Depth-first Search, Depth-limited Search, Iterative Deepening Search, Greedy Best-first Search, A^* Search.*

Vous pouvez également tester les méthodes suivantes : *Iterative-lengthening search, Bidirectional Search, Iterative-Deepening A^*.*

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

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

Figure 2: Un 15-puzzle ($n = 4$) mélangé à gauche et résolu à droite. Le symbole *blank* est représenté par la lettre b.

Enfin, vous devrez proposer de nouvelles heuristiques pour guider vos méthodes de recherche informée.

Quelques remarques :

- On ne demande pas la réalisation d'une interface graphique. Cependant, on doit pouvoir tester votre programme facilement et de manière intuitive.
- Vous pouvez utiliser les outils que vous voulez et le langage de programmation que vous désirez. Cependant, il est préférable de demander l'autorisation si vous souhaitez utiliser un langage "exotique".
- Posez-vous la question suivante : "est-ce que chaque disposition initiale du puzzle consititue une instance réalisable du problème ?"

Vous devez proposer un protocole expérimental pour tester ces méthodes, mettre en place ces expérimentations, puis réaliser un rapport détaillant votre travail (e.g., structure du code, définition des heuristiques, résultats expérimentaux).

Vous devrez y détailler la mise-en-place et les résultats de vos expérimentations numériques : génération des instances, nombre de noeuds générés par chaque méthode, nombre maximum de noeuds stockés en mémoire, valeur de n que peut atteindre chaque méthode ... et comparer ces résultats aux compléxités théoriques.

Si vous utilisez des ressources (e.g., mémoires, papiers de recherche, autres cours, code), vous devez y faire référence dans votre rapport.