Compte rendu TP1 IA

Familiarisation avec le problème du Taquin 3x3

1.2)

a)
final_state([[1,2,3,4],[5,6,7,8],[9,10,11,12],[13,14,15,vide]])

b)
1ère : Cherche ou se trouve la pièce d dans Ini, renvoie la ligne L et la position dans la ligne C
2ème : Vérifie dans Fin, si l'élément P est bien à la 2ème place de la 3ème ligne

c)
initial_state(U0), final_state(F), mal_place(c,U0,F).
mal_place et coordonnees ont été initialisé dans le fichier taquin

d)
initial_state(U0), rule(X,1,U0,U1).

e)
initial_state(U0), findall(X,rule(X,1,U0,U1),L).

f)
initial_state(U0), findall([A,S], rule(A,1,U0,S),L).

Développement des 2 heuristiques

2.1)

Choix de la méthode a)
initial_state(U0), final_state(F), findall(X, (mal_place(X,U0,F),X\=vide),L),
length(L, X).

2.2)

```
dm(E,U,K):-
    coordonnees(E,U,[L1,C1]),
```

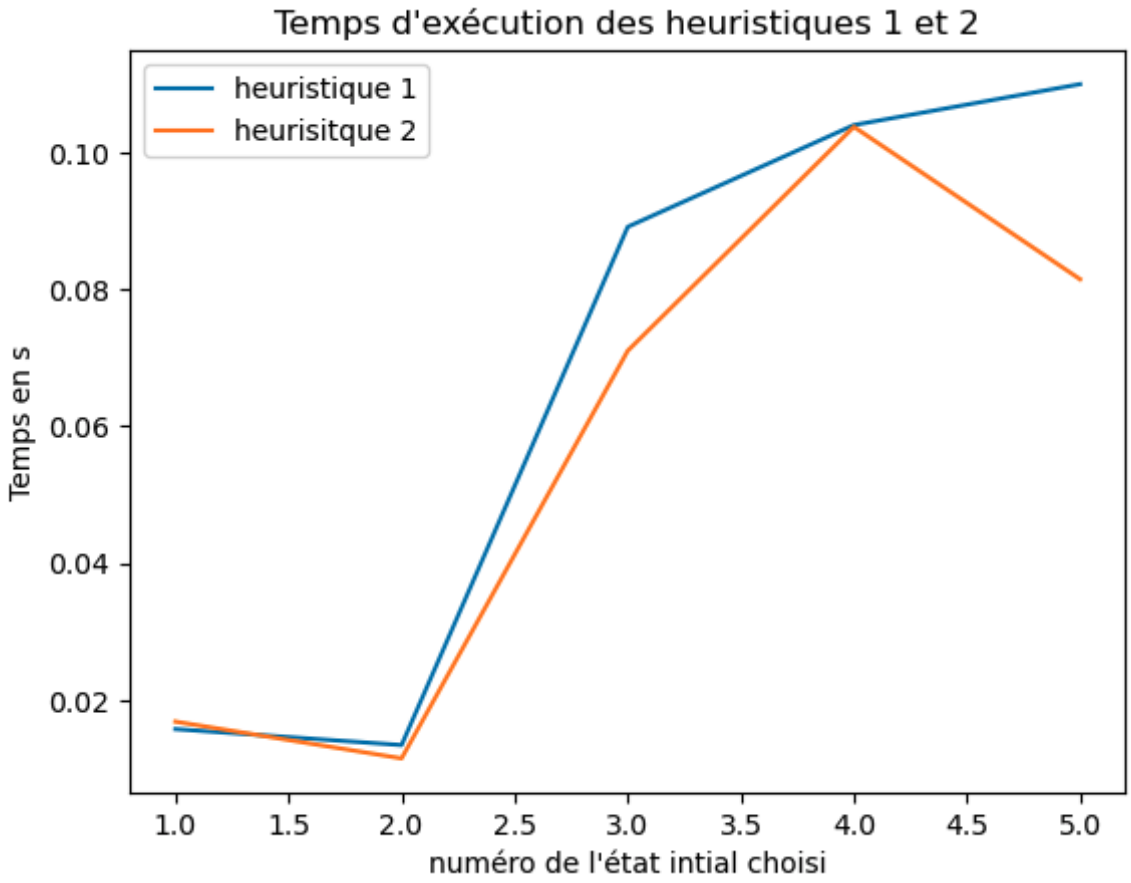
```
    final_state(F),
    coordonnees(E,F,[L2,C2]),
    K is abs(L2-L1)+abs(C2-C1).

heuristique2(U, H) :-
    findall(X,(dm(P,U,X),P\=vide),L),
    sum_list(L,H).
```

Implémentation de A*

3.2)

Avec l'exemple on obtient ce résultat :

Début
up
[b,h,c]
[a,vide,d]
[g,f,e]
up
[b,vide,c]
[a,h,d]
[g,f,e]
left
[vide,b,c]
[a,h,d]
[g,f,e]
down
[a,b,c]
[vide,h,d]
[g,f,e]
right
[a,b,c]
[h,vide,d]
[g,f,e]
9.789606



État 1 : [b, h, c],[a, f, d],[g,vide,e]
État 2 : [a, b, c],[g, h, d], [vide,f, e]
État 3 : [b, c, d], [a,vide,g], [f, h, e]
État 4 : [f, g, a], [h,vide,b], [d, c, e]
État 5 : [e, f, g], [d,vide,h], [c, b, a]
État 6 : [a, b, c], [g,vide,d], [h, f, e] (pas afficher car n'étant pas connexe avec l'état final, il n'est pas résoluble)

Quel longueur de séquence peut-on envisager pour le taquin 4x4 ?

La longueur que l'on peut envisager de résoudre serait similaire à celles que nous venons de résoudre. Cela prendra plus de temps à résoudre et sera plus demandant en espace mémoire mais le méthode de résolution sera identiquement la même.

A* trouve-t-il la solution pour la situation initiale suivante ?
initial_state([[a,b,c],[g,vide,d],[h,f,e]]).

A* ne peut pas résoudre cette situation initiale car elle n'est pas connexe avec la situation finale souhaité.

Quelle représentation de l'état du Rubik's Cube et quel type d'action proposeriez-vous si vous vouliez appliquer A*?

Pour représenter l'état d'un Rubik's Cube, il faut créer une liste de matrice de la même forme que pour le taquin où chaque matrice représente une face du Rubik's Cube mais en remplaçant les lettres par des couleurs avec 6 couleurs différentes.
Il faudrait aussi changer les actions possible. En ayant fait ceci l'algorithme A* devrait pouvoir le résoudre.