

Exercice 1 (Solutionnaire)

1. (5 pts) Quelle est la différence entre une mesure de performance et une fonction d'utilité pour un agent?

Une mesure de performance est une fonction utilisée par un observateur externe pour évaluer la performance d'un agent. Une fonction d'utilité est utilisée par l'agent lui-même pour évaluer à quel point les états sont désirables.

N.B. La fonction d'utilité peut être différente de la mesure de performance. Un agent peut ne pas avoir de fonction d'utilité explicite, mais il y a toujours une mesure de performance.

2. (25 pts) Donner la PAGE et les caractéristiques de l'environnement d'un agent devant jouer au tennis.

Percepts : La balle, le joueur adverse, sa position sur le terrain.

Actions :

- Frapper la balle (coup droit, revers, smash, service, etc.).
- Se déplacer (dans toutes les directions).

But : Le but général est de gagner la partie. Un but intermédiaire serait de frapper la balle pour qu'elle tombe dans la partie de terrain de l'adversaire et que celui-ci ne soit pas capable de la retourner.

Environnement : L'environnement de l'agent est le terrain de tennis sur lequel il joue avec un adversaire. L'environnement est *accessible*, parce que l'agent peut voir tout ce qui se passe sur le terrain. L'environnement est *non-déterminé*, parce que les actions de l'agent n'ont pas un effet garanti et il y a de l'incertain. Lorsque l'agent frappe la balle, il ne sait pas si elle va tomber dans le jeu et il ne sait pas si l'autre joueur va la retourner. L'environnement est *non-épisodique*, car les actions de l'agent dans le passé peut influencer ses actions actuelles. Par exemple, il peut avoir trouvé une faiblesse à son adversaire ou il peut jouer différemment s'il mène la partie. L'environnement est

dynamique, parce que l'environnement peut être modifié pendant que l'agent délibère. L'autre joueur peut se déplacer et la balle se déplace. L'environnement est *continu*, car il y a plusieurs perceptions possibles et plusieurs actions possibles. La force de frappe et la direction de la balle sont des valeurs continues.

- 3. (20 pts) Donner un exemple d'agent pour chacune des quatre architectures d'agent (simple réflexe, réflexe avec état interne, buts, utilité) et expliquer votre choix.**

Tout exemple qui correspond bien aux architectures est accepté.

- 4. (10 pts) Supposons que l'on exécute un algorithme de recherche vorace (greedy search), quelle sorte de recherche l'algorithme de recherche vorace simule-t-il si**

a. $h(n) = -g(n)$?

Profondeur d'abord, parce que les fils auront toujours une valeur plus petite que les parents. Donc, l'algorithme développera le fils, puis le fils du fils, jusqu'à ce qu'il atteigne une feuille, ce qui est le comportement de l'algorithme de profondeur d'abord.

b. $h(n) = g(n)$?

Largeur d'abord, parce que les parents auront toujours une valeur plus petite que leurs fils. Donc, tous les nœuds d'un même niveau seront développés avant de passer au niveau suivant, ce qui est le comportement de l'algorithme de largeur d'abord.

- 5. (25 pts) Dans l'espace de recherche ci-dessous, le nombre placé à côté de chaque lien dénote le coût associé au lien entre les nœuds et h est le coût estimé du nœud jusqu'au but final G.**

- a. **Donner l'ordre dans lequel les nœuds sont visités lorsqu'on utilise A*.**

Donner alors $f(n)$ pour chacun des nœuds n .

$f(B) = 5$, $f(C) = 8$, $f(D) = 20$, $f(E) = 19$, $f(F) = 11$, $f(I) = 31$, $f(G) = 13$, $f(H) = 14$, $f(J) = 34$. Donc, l'ordre de visite est: A-B-C-F-E-D-G

b. Un algorithme « vorace » de type meilleur d'abord peut-il faire mieux? Combien de nœuds, un tel algorithme traverse-t-il avant d'atteindre le but G.

$f(B) = 4, f(C) = 2, f(D) = 15, f(E) = 16, f(F) = 3, f(I) = 20, f(G) = 0, f(H) = 4$.
 Donc, l'ordre de visite est: A-C-F-B-D-G, ce qui est un nœud de moins que A*, 6 au lieu de 7.

c. Si le nœud I est aussi un but, lequel (ou lesquels) parmi largeur d'abord, profondeur d'abord et profondeur itératif peut (ou peuvent) trouver la solution optimale? Donner pour chaque algorithme la liste des nœuds développés.

Largeur d'abord : A-B-C-D-E-F-G

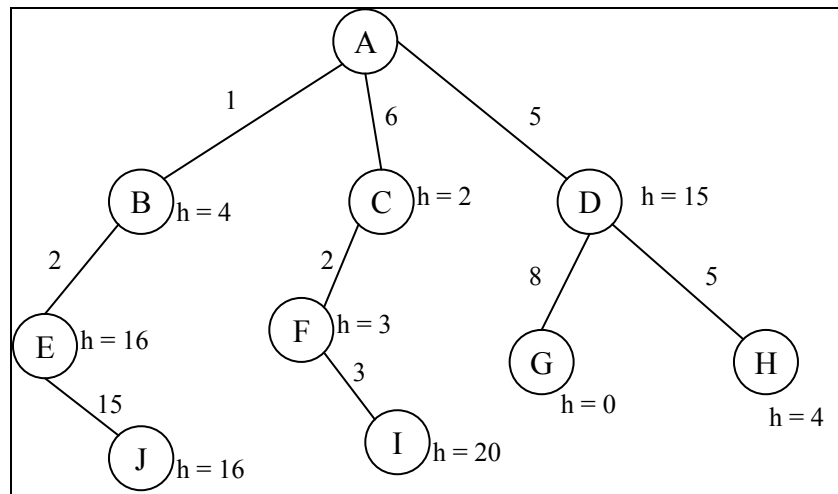
Profondeur d'abord : A-B-E-J-C-F-I

Profondeur itératif : A-B-C-D-A-B-E-C-F-D-G

Seulement la recherche en profondeur d'abord va trouver la solution optimale I.

d. La fonction heuristique h pose un problème dans la mesure où elle surestime trop le coût de D à G. Quelle propriété de A* n'est plus remplie si h a ce problème?

A* n'est plus optimale.



6. (15 pts) Supposons que l'on veut établir les horaires pour un problème de salles de cours. Dans ce problème, on a 5 salles et 10 cours, chaque cours débute à D_i heure et finit à F_i heure, avec D_i et F_i des entiers naturels. Sachant qu'aucun cours ne doit partager avec un autre la même salle en même temps, et que les salles sont seulement disponibles entre 12h00 et 17h00 du lundi au vendredi, formulez ce problème comme un CSP (constraint solving problem), c'est-à-dire en spécifiant les variables, les domaines et les contraintes. Suggérez alors une heuristique pour résoudre ce problème et expliquer pourquoi pensez-vous que votre heuristique peut résoudre un tel problème.

Le problème consiste à choisir une salle pour chaque cours.

Variables : S_1, \dots, S_{10} (les différentes salles pour chacun des cours)

D_1, \dots, D_{10} (les différentes heures de début pour chacun des cours)

F_1, \dots, F_{10} (les différentes heures de fin pour chacun des cours)

Domaine : $S_i : \{1 \text{ à } 5\}$

$D_i : \{12 \text{ à } 16\text{h}30\}$ (Si on suppose des cours d'une demi heure)

$F_i : \{12\text{h}30 \text{ à } 17\}$ (Si on suppose des cours d'une demi heure)

Contraintes : Pour tous les i, j , si $(F_i \geq F_j > D_i \text{ OU } D_i \leq D_j < F_i)$ alors $(S_i \neq S_j)$

$D_i < F_i, D_i \geq 12$ (pour 12h00), $F_i \leq 17$ (pour 17h00)

Vous pouvez utiliser n'importe quelle heuristique vue en classe, comme par exemple l'heuristique des conflits minimums (min-conflicts).