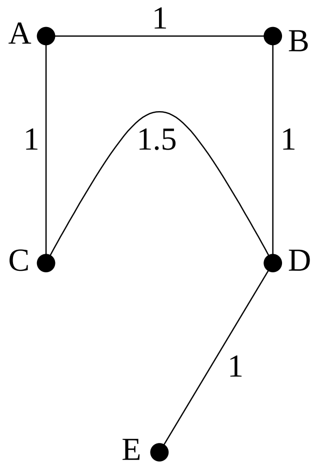
Master d'Ingéniérie Informatique de Paris Diderot - Paris 7
M1: Introduction à l'Intelligence Artificielle

Examen partiel du 10 novembre 2011 - Durée: 1h45

Documents autorisés; le barème est donné à titre indicatif.

Exercice 1 (**6 points**) On considère le problème de trouver un chemin de A à E dans le graphe pondéré suivant:



1. Définir deux fonctions heuristiques admissibles h_1 et h_2 telles que l'algorithme de recherche glouton soit optimal relativement à h_1 et non optimal relativement à h_2.

2. Définir une fonction heuristique h_3 relativement à laquelle A^* à parcours d'arbre ne soit pas optimal.

3. Nous savons que dans cet exemple, l'algorithme A^* à parcours d'arbre est optimal relativement à toute heuristique admissible[1]. Il est toutefois possible que, avant de trouver la solution optimale, un même sommet soit visité (c.à.d. sélectionné de la liste des sommets ouverts) plusieurs fois par l'algorithme A^* à parcours d'arbre. Donner un exemple de fonction heuristique admissible h_4 pour laquelle cette situation se vérifie.

4. Montrer que, relativement à h_4, l'algorithme A^* *à parcours de graphe*[2] n'est pas optimal. Énoncer une condition suffisante portant sur une heuristique admissible h, pour que A^* à parcours de graphe soit optimal relativement à h.

Exercice 2 (**8 points**)

Nous allons considérer des problèmes de recherche locale sur l'ensemble des séquences binaires de longueur 20: $E = \{0,1\}^{20}$. Pour $s \in E$, $1 \leq i \leq 20$, s_i désigne le i-ème élément de s, $|s|_0$ (resp. $|s|_1$) désigne les nombre d'occurrences de 0 (resp. de 1) dans s. Les successeurs de s sont les séquences t obtenues en changeant un bit de s. Formellement, $t \in Succ(s)$ si $\exists 1 \leq j \leq 20$ tel que

$$t_k = \begin{cases} s_k & \text{si } k \neq j \\ \overline{s_k} & \text{si } k = j \end{cases}$$

[1] En effet, l'exemple satisfait les hypothèses de branchement fini et de coûts non nuls.

[2] A^* à parcours de graphe maintient une liste de "sommets fermées", vide au début, dans laquelle sont insérés au fur et à mesure les sommets visités. Lorsque on ajoute à la liste des sommets ouverts les successeurs du sommet qui vient d'être visité, les sommets déjà présent dans la liste des fermées sont ignorés.

Pour les problèmes de la liste suivante, donner la valeur de l'optimum, une séquence qui réalise cet optimum, puis montrer soit que l'algorithme de recherche locale *hill climbing* est optimale, soit qu'il existe des valeur localement optimales qui ne sont pas optimum.

1. $f_1(s) = ||s|_0 - |s|_1|$, où pour un entier p, $|p|$ désigne sa valeur absolue; on recherche un minimum de f_1.

2. $f_2(s) = \#\{i \mid 1 \leq i \leq 19 \text{ et } s_i \neq s_{i+1}\}$, où pour un ensemble A, $\#A$ désigne le nombre d'éléments (la *cardinalité*) de A; on recherche un maximum de f_2.

3. $f_3(s) = \#\{i \mid 1 \leq i \leq 10 \text{ et } s_i = s_{21-i}\}$; on recherche un maximum de f_3.

On envisage l'application d'algorithmes génétiques sur E avec fonction de fitness f_1, f_2 et f_3 respectivement. Nous allons considérer une version restreinte de l'algorithme génétique de base, dans laquelle le croisement de deux individus se fait toujours au rang 10. C'est à dire que les séquences s et t peuvent engendrer uniquement les individus u et v définis par:

$$u_k = \begin{cases} s_k & \text{si } k \leq 10 \\ t_k & \text{si } k > 10 \end{cases} \quad \text{et} \quad v_k = \begin{cases} t_k & \text{si } k \leq 10 \\ s_k & \text{si } k > 10 \end{cases}$$

- Montrer que dans les trois cas, les fils de couple d'individus bien adapté peuvent être très mal adapté.

- Dans quel cas ce phénomène est-il plus sensible?

- Tenter de donner une définition générale d'*hérédité* telle que les performances de l'algorithme génétique restreint présenté ici s'améliorent si la fonction de fitness est héréditaire. Testez votre définition sur les trois exemples donnés, et éventuellement sur d'autres.

Exercice 3 (**6 points**) On considère un agent réactif A disposant de n percepteurs booléens, et devant choisir à chaque instant une action parmi m actions possibles. L'éventuelle mémoire de l'agent sera implémentée par l variables booléennes.

1. Déterminez la valeur de l, au-dessous de laquelle A ne peut pas fonctionner correctement.

2. On dira que A est *k-réactif* si l'action choisie à l'instant t dépend uniquement des valeurs des percepteurs aux instants $t-1, \ldots, t-k$. Donc, un agent 1-réactif choisit son action uniquement en fonction des valeurs des percepteurs à l'instant précèdent: il est *purement réactif*. Supposons que A soit *k-réactif*. Déterminez, en fonction de k et de n, la valeur maximale de m et une valeur de l suffisante à l'implémentions de A.

3. Donner un exemple d'agent 1-réactif et un exemple d'agent 2-réactif[3].

4. Que peut-on envisager comme agent k-réactif, quand k croît?

[3] La concision des exemples sera appréciée. Il n'est pas nécessaire de spécifier les règle des agents; une description en quelques phrases sera suffisante.