

Rapport_{Graphe}

ghettaszineddine

June 2017

0.1 Question 1

0.1.1 Question 1.a

Plusieurs solutions s'offrent à nous. Il est possible d'utiliser une structure de données (un tableau par exemple) qui représenterai le plateau à chaque état, ainsi deep blue (et l'utilisateur du robot) pourrait visualiser à chaque noeud l'échiquier. Une solution plus vraisemblable, étant donné que nous avons la fonction S , serait d'évaluer $S(e)$ pour chaque état parcouru du graphe (ou $S(e)S(e')$, e' étant l'état suivant e) et évaluer le parcours idéal, c'est à dire le parcours qui conduit à une valeur de $S(e)$ maximale. On pourrait également utiliser un enregistrement contenant l'échiquier et sa valeur associée $S(e)$. Si on associe à chaque état la valeur $dif = S(e) - S(e')$, alors on peut calculer aisément la valeur $S(e)$ puisqu'on connaît la différence et la valeur de $S(e')$. Si on utilise un tableau, il faudra annuler le dernier coup de l'adversaire (à creuser). Une fois le dag rempli, il faut essayer d'atteindre l'état idéal, c'est à dire l'état pour lequel $S(e)$ sera maximal pour Deep Blue et minimal pour l'adversaire. Le calcul doit être réitéré à chaque état selon les coups de l'adversaire, car le chemin calculé initialement (le chemin idéal) ne sera probablement pas atteignable si l'adversaire n'est pas mauvais.

0.1.2 Question 1.b

5, ce chiffre est monstrueux.

0.1.3 Question 1.c

La méthode prioristique consiste à calculer un chemin à partir d'un état choisi aléatoirement (ou par intuition). Si au 12ème demi coup, cet état est satisfaisant, alors on le choisit. S'il n'est pas satisfaisant, alors on ne va pas revenir à la racine mais plutôt parcourir l'arbre en profondeur afin de chercher des états voisins plus intéressants. Si au bout du calcul d'un certain nombre d'états, on se rend compte que la variation de $S(e)$ vire à la catastrophe, alors on arrête de parcourir ce chemin afin d'évaluer des chemins plus rentables.

0.2 Question 2

0.2.1 Question 2.a

On cherche à calculer $P(M)$, c'est à dire qu'à partir des causes de l'évènement M , on veut déterminer la probabilité des effets (Herbe mouillée). On sait que les évènements $(A \text{ et } P)$, $(A \text{ et non}(P))$, $(\text{non}(A) \text{ et } P)$, $(\text{non}(A) \text{ et non}(P))$ sont dis-joints. On peut donc appliquer le théorème des probabilités totales et obtenir la formule suivante : $P(M) = P(M/A.P) * P(A.P) + P(M/n(A).P) * P(n(A).P) + P(M/A.n(P)) * P(A.n(P)) + P(M/n(P)) * P(n(A).n(P))$

Les valeurs de $P(M/A.P)$, $P(M/n(A).P)$, $P(M/n(A).n(P))$ et $P(M/A.n(P))$ sont données, il reste donc à calculer le reste: $P(A.P) = P(P) * P(A/P) = 0.5 * 0.1 = 0.05$
 $P(A.n(P)) = P(n(P)) * P(A/n(P)) = 0.5 * 0.5 = 0.25$ Pour $P(n(A).P)$ et $P(n(A).n(P))$ on va devoir calculer $P(A)$, ce qui se fait aussi grâce au théorème des probabilités totales : $P(A) = P(A/P) * P(P) + P(A/n(P)) * n(P) = 0.1 * 0.5 + 0.5 * 0.5 = 0.3$ Ce qui donne $P(A) = 0.3$ et par la suite : $P(n(A).P) = P(P) * P(n(A)/P) = P(P) * P(P/nA) * P(nA)/P(P) = P(P) * (P(P) - P(P/A) * P(A)) / P(P) = P(P) * (P(P) - P(P.A)) / P(P) = (0.5 - 0.05) = 0.45$
De même on calcule $P(n(A).n(P))$: $P(n(A).n(P)) = P(n(P)) * P(n(A)/n(P)) = P(n(P)) * P(nP/nA) * P(nA)/P(nP) = P(nP) * (P(nP) - P(nP/A) * P(A)) / P(nP) = P(nP) * (P(nP) - P(nP.A)) / P(nP) = 0.5 - 0.25 = 0.25$ Finalement on obtient : $P(M) = 0.95 * 0.05 + 0.7 * 0.45 + 0.01 * 0.25 + 0.8 * 0.25 = 0.565$ $P(M) = 0.565$

0.2.2 Question 2.b

BG va dépendre des deux choix du candidat C1 et C2. BR va dépendre de BG et de C1, étant donné que le présentateur ne révélera ni la boîte gagnante, ni le premier choix du candidat. C2 va dépendre du premier choix du candidat C1 ainsi que de la boîte révélée BR C1 ne dépend d'aucune des variables, étant donné que le premier choix du candidat est complètement aléatoire.

0.2.3 Question 2.c

0.3 Question 3

0.3.1 Question 3.a

Exemples : Dans le cas où les propriétés de sûreté et de vivacité ne sont pas vérifiées pour un processus, le coût est alors inconnu. Jeu d'échec : le temps de réflexion du joueur ne peut être évalué.

0.3.2 Question 3.b

Sachant que nous avons 2 ressources, et que chaque tâche a un coût égal à 1, on peut alors calculer le coût optimal : D'abord, les deux ressources s'occupent des tâches $(a1...aN)$ et $(x1...xN)$, ce qui donne un coût de N . Puis la ressource qui s'occupe de $(a1...aN)$ passe à $(b1...bN)$ tandis que la deuxième ressource est utilisée pour les tâches $(xN+1...xNx)$. Finalement, le coût optimal vaut $2N$.

0.3.3 Question 3.c

Les deux ressources s'occupent des tâches ($a_1 \dots a_N$) et ($b_1 \dots b_N$), ce qui donne un temps d'exécution égal à N . Puis les tâches ($x_1 \dots x_N$) utilisent les deux ressources pour leur exécution, ce qui divise ce temps par 2. Le coût optimal obtenu vaut alors $1.5N$.

1 Partie 4 : Extension aux tâches et ressources hétérogènes

1.1 Question 11

Si un ordonnancement est calculé sans prendre en compte l'hétérogénéité, alors on se retrouve avec un temps d'exécution t . Cela signifie qu'on possède soit une ressource qui exécute une tâche en un temps t , soit 2 ressources r_1 et r_2 exécutant deux tâches t_1 et t_2 et ce en un temps t . Dans le deuxième cas, qui est le cas proposé par le sujet, on a donc r_1 qui exécute t_1 et r_2 qui exécute t_2 . Le pire cas est celui où r_1 exécute t_2 et r_2 exécute t_1 . Sachant que r_1 exécute t_2 , le temps d'exécution de cette tâche sera donc de t/α_2 (α_2 étant le paramètre d'hétérogénéité pour l'exécution de t_2 par r_1). De même le temps d'exécution de t_1 par r_2 est de t/α_1 . Cela donne un temps d'exécution de t/α^* , $\alpha^* = \min(\alpha_1, \alpha_2)$.