

DIC938K – Planification automatique

Automne 2017 – Examen de synthèse

Éric Beaudry
Département d'informatique
Université du Québec à Montréal

Vendredi 27 octobre 2017 – 9h10 à 12h10 (3 heures) – Local SH-3120

Instructions

- Une ébauche de l'examen a été fournie quelques jours avant l'examen.
- Livre de référence autorisé (max. 25 minutes) : *Automated Planning & Acting* de Ghallab, Nau et Traverso.
- Documentation personnelle autorisée : 4 feuilles (format lettre US ou A4) de notes personnelles recto-verso (incluant les réponses aux exercices proposés dans le livre et aux ébauches de question).
- Les appareils électroniques, incluant les téléphones et les calculatrices, sont interdits.
- Répondez dans le cahier d'examen fourni lors de l'examen.
- À moins d'avis contraire, chaque sous-question vaut 1 point.
- À moins d'avis contraire, chaque réponse doit être justifiée. La justification n'a pas besoin d'être longue, mais elle doit être suffisante pour démontrer votre compréhension.

Identification

Nom : _____

Code permanent : _____

Signature : _____

Résultat

Q1		/ 8
Q2		/ 8
Q3		/ 6
Q4		/ 4
Q5		/ 4
Q6		/ 5
Total		/ 35

1 Résolution de problème par recherche dans un graphe [8 points]

☞ Pour se préparer à cette question, il est recommandé de (re)lire certaines sections du livre *Artificial Intelligence : A Modern Approach* de Russell et Norvig, et les diapositives de la semaine 1.

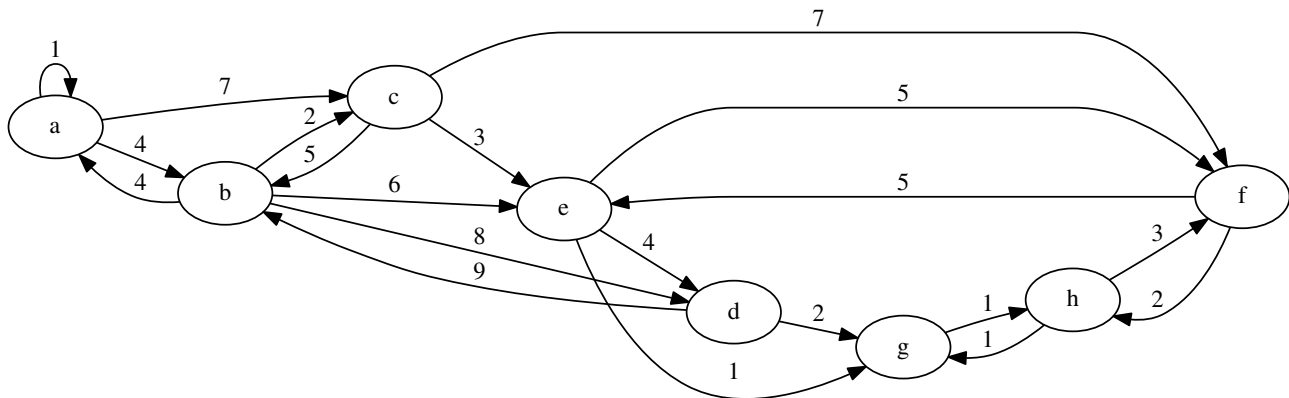
(a) Le chapitre 2 du livre *AI : A Modern Approach* introduit plusieurs types d'agent dont : (1) *simple reflex*; (2) *model-based reflex*; (3) *goal-based reflex*; (4) *utility-based reflex*; (5) *learning agent*. Lesquels peuvent intégrer des techniques de **planification**.

(b) Le livre de référence (*Automated Planning & Acting* de Ghallab, Nau et Traverso) utilise fréquemment le terme «*deliberative*». Définissez dans vos propres mots ce qu'est une **méthode/technique deliberative**.

(c) L'algorithme A* utilise une fonction $f : S \rightarrow \mathbb{R}$ pour déterminer l'ordre de visite des états. Généralement, on utilise la fonction $f(s) = g(s) + h(s)$ où $g(s)$ est le coût pour se rendre de l'état initial s_0 à s , et $h(s)$ est une estimation du coût restant pour se rendre de s jusqu'au but. (i) Quel serait l'intérêt de plutôt utiliser la fonction $f(s) = g(s) + \alpha h(s)$ avec $\alpha > 1$? (ii) Quel serait l'impact sur la qualité de la solution?

(d) Vrai ou faux : Lorsque A* utilise une heuristique inadmissible, l'algorithme peut ne pas trouver de solution, même si une solution existe.

(e) Simulez l'algorithme A* sur le problème suivant : l'espace d'états $S = \{a, b, c, d, e, f, g, h\}$, l'état initial est a , l'ensemble des états satisfaisant le but sont $S_g = \{g, h\}$, le graphe de transitions et la fonction h ci-bas.



État :	a	b	c	d	e	f	g	h
$h(\cdot)$	10	7	3	1	5	2	0	0

(f) En vous référant au problème en (e) ci-haut : (i) L'heuristique h est-elle admissible? (ii) L'heuristique h est-elle consistante (monotone)?

(g) Vrai ou faux : une heuristique consistante (monotone) est forcément admissible.

(h) Vrai ou faux : une heuristique inconsistante (non monotone) est forcément inadmissible.

2 Chapitre 2 – Modèles déterministes (2.1 à 2.3) [8 points]

(a) Donnez un problème de planification à variables d'état P_1 et une solution π_1 pour P_1 tel que π_1 est **minimal** mais pas **la plus courte** (*shortest*). Donnez un problème P_2 et une solution π_2 pour P_2 tel que π_2 est acyclique mais non minimal. [Exercice 2.3 du livre]

(b) Soit le problème de planification spécifié à l'Annexe A. Estimez la taille de l'espace d'états ($|S|$) pour ce problème.

(c) Considérez l'Annexe A. Estimez la taille de l'espace d'états **accessibles** depuis s_0 .

(d) [2 points] Calculez l'état $s_1 = \gamma(s_0, a)$ où $a = \text{deverrouiller}(r_1, e_1)$.

(e) Évaluez les fonctions heuristiques h^{add} et h^{max} pour l'état initial s_0 et l'état s_1 obtenu en (d).

(f) Comment faudrait-il modifier le problème pour qu'un camion ne puisse que charger un seul colis à la fois? Une modification peut être l'ajout ou le retrait de variables d'états, de préconditions, d'effets, etc.

(g) La section 2.3.2 aborde les heuristiques *delete-relaxation*. À quoi fait référence le terme *delete*?

3 Chapitre 2 – Modèles déterministes (2.4 à 2.7) [6 points]

- (a) Considérez le problème de planification à l'Annexe A. Dans une approche PSP (plan-space planning) que serait le plan (partiel) initial π_0 .
- (b) Dans le plan initial π_0 en (a), quels sont les défauts (*flaws*) ?
- (c) Choisissez un défaut (*flaws*) identifié en (b) et calculez une étape de raffinement pour obtenir le plan (partiel) π_0 .
- (d) Expliquez dans vos propres mots ce qu'est un lien causal (*causal link*).
- (e) Considérez le problème de planification à l'annexe A. À quoi pourrait ressembler un plan partiellement ordonné produit par l'algorithme PSP ? Écrivez uniquement l'ensemble des actions et les ordres partiels.
- (f) Combien de séquences d'exécution sont possible à partir du plan obtenu en (e) ?

4 Chapitre 3 – Méthodes de raffinement + Modèles temporels [4 points]

- (a) [2 points] Identifiez et expliquez deux différences et similitudes fondamentales qui distinguent les approches du chapitre 3 par rapport à celles du chapitre 2.
- (b) Relativement à la terminologie utilisée dans le chapitre 3, quelle est la différence entre une tâche et une méthode ?
- (c) Dans la section 4.2 (*Temporal Representation*), qu'est-ce qu'une *persistence* ? Quel est son rôle ?

5 Chapitre 5 – Modèles non déterministes [4 points]

Pour les sous-questions suivantes, supposez que l'action *deverrouiller* à l'Annexe soit incertaine, qu'elle puisse parfois échouer pour rendre $ver(e) = faux$.

- (a) Existe-t-il encore une **solution** (définition 5.7) ?
- (b) Existe-t-il une **safe solution** (définition 5.8) ?
- (c) Existe-t-il une **acyclic safe solution** (définition 5.11) ?
- (d) Quelle est la taille minimale de la «meilleure» politique possible.

6 Chapitre 6 – Modèles probabilistes [5 points]

- (a) Qu'est-ce qu'une politique ?
- (b) L'équation (6.3) définit la valeur $V^\pi(s)$. Ensuite, l'équation (6.4) définit le principe d'optimalité dans lequel on retrouve $V^*(s) = \min_{a \in A} \{cost(s, a) + \sum_{s' \in \gamma(s, a)} Pr(s', s, a) V^*(s')\}$ lorsque $s \notin S_g$. Qu'arrive-t-il à la valeur $V^*(s)$ lorsqu'aucune action n'est applicable dans l'état s ?
- (c) Dans un problème de type *stochastic shortest path*, peut-on générer une politique *unsafe* ?
- (d) Donnez une politique pouvant être calculée par l'algorithme d'itération par valeurs (VI, *value iteration*) pour le graphe d'états présenté à l'Annexe B. Les étapes intermédiaires ne sont pas demandées. Écrivez seulement le résultat final.
- (e) Simulez deux (2) itérations de l'algorithme 6.17 LRTDP sur le graphe d'états présenté à l'Annexe B. Lorsqu'il faut tirer un état au hasard, choisissez le résultat le plus probable. Montrez clairement l'ordre de parcours des états et les valeurs $V(s)$ suite aux «Bellman-Update».

Annexe A

Soit les objets $B = Emplacements \cup Camions \cup Colis$ où

$Emplacements = \{e_1, \dots, e_6\}$, $Camions = \{r_1, r_2\}$ et $Colis = \{c_1, c_2\}$.

Soit une relation rigide $adjacent = \{(e_1, e_2), (e_2, e_1), (e_2, e_3), (e_2, e_4), (e_3, e_2), (e_4, e_2), (e_4, e_5), (e_5, e_4)\}$ et une fonction partielle de distances : $d(e_1, e_2) = 2, d(e_2, e_1) = 2, d(e_2, e_3) = 3, d(e_2, e_4) = 4, d(e_3, e_2) = 3, d(e_4, e_2) = 4, d(e_4, e_5) = 1, d(e_5, e_4) = 1$.

Soit les variables d'états $X = \{pos(o), ver(l) \mid o \in Camions \cup Colis, l \in Emplacements\}$, où $Range(pos(o)) \in Emplacements$ et $Range(ver(l)) \in \{faux, vrai\}$.

Soit les actions A comprenant :

move(r, e, e')

pre : $adjacent(e, e'), ver(e) = vrai, pos(r) = e$

eff : $pos(r) \leftarrow e'$

cost : $d(e, e')$

charger(r, e, c)

pre : $ver(e) = faux, pos(r) = e, pos(c) = e$

eff : $pos(c) \leftarrow r$

cost : 1

décharger(r, e, c)

pre : $ver(e) = faux, pos(r) = e, pos(c) = r$

eff : $pos(c) \leftarrow e$

cost : 1

verrouiller(r, e)

pre : $ver(e) = faux, pos(r) = e$

eff : $ver(e) \leftarrow vrai$

cost : 1

déverrouiller(r, e)

pre : $ver(e) = vrai, pos(r) = e$

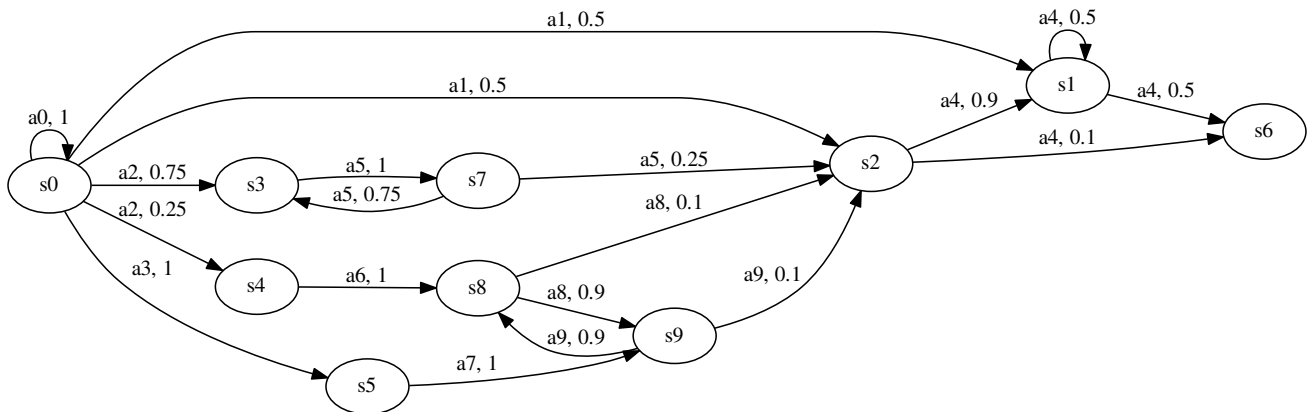
eff : $ver(e) \leftarrow faux$

cost : 1

Soit l'état initiale $s_0 = \{pos(r_1) = e_1, pos(r_2) = e_5, pos(c_1) = e_4, pos(c_2) = e_6, ver(e) = vrai | e \in Emplacements\}$.

Et le but : $g = \{pos(c_1) = e_5, pos(c_2) = e_1\}$.

Annexe B



Les actions ont un coût unitaires et uniforme :

Action :	a0	a1	a2	a3	a4	a5	a6	a7	a8	a9
cost(\cdot)	1	1	1	1	1	1	1	1	1	1

Fonction heuristique h pour L-RTDP à la question 5e :

État :	s0	s1	s2	s3	s4	s5	s6	s7	s8	s9
$h(\cdot)$	2	1	1	0.5	3	3	0	2	2	0

/***** Fin de l'examen *****/