

**IFT 608/ IFT 702**  
**Planification en intelligence artificielle**

**Devoir3**

*À remettre au plus tard le jeudi 6 avril à minuit par turnin*

**PROFESSEUR**

Froduald Kabanza

[planiart.usherbrooke.ca/kabanza](mailto:planiart.usherbrooke.ca/kabanza)

**INSTRUCTIONS**

Répondez sur ce questionnaire, qui sert en même temps de cahier de réponses, aux endroits indiqués. Vous pouvez étendre les cases de réponses.

Soumettez un **fichier PDF** (pas WORD) portant le nom « Devoir3-NomPrenom » où Nom et Prénom sont les vôtres.

Soyez **le plus clair, le plus concis et le plus précis** possible **dans les limites de l'espace alloué**. Si votre réponse manque à ces critères ou ne suit pas les consignes, elle sera pénalisée.

Écrivez votre nom, prénom et CIP ci-dessous.

**NOM : AMAIRI**

**PRÉNOM : TAHAR**

**MATRICULE : 22 007 908**

**CIP : amat0601**

*Réservé. N'inscrivez rien dans cette table.*

	Q1/12	Q2/6	Q3/6	Q4/6	Total/30
<i>Note</i>					

**(ATTENTION : CERTAINES FONCTIONS DE LA QUESTION 1.A ONT ÉTÉ MODIFIÉES POUR DES RAISONS DE SIMPLICITÉ, BIEN LIRE L'ÉNONCÉ !)**

### Question 1 (12 points) – Planification avec PDDL

Considérons un domaine de planification consistant à contrôler des drones pour livrer des colis à des maisons à partir d'un entrepôt.



Pour modéliser le problème avec le langage PDDL, vous devez utiliser les prédicats suivants :

- **loc( $x,y$ )** : est vrai si  $x$  est à l'endroit  $y$ ,  $x$  pouvant être ~~une maison, un entrepôt~~, un drone ou un colis et  $y$  étant un lieu (maison ou entrepôt) **(j'ai modifié cette fonction pour simplifier le problème : travailler avec le lieu suffit amplement, il n'est pas nécessaire de faire la distinction entre le lieu et sa localisation car nous avons des prédicats statiques).**
- **has( $d,x$ )** : est vrai si le colis  $x$  est à bord du drone  $d$

Selon les questions posées plus bas, vous pouvez introduire d'autres prédicats ou des fonctions (*numerical fluents*), ou faire d'autres ajustements mineurs si nécessaire et sans accroître inutilement la complexité du modèle. Une description inutilement complexe ou suboptimale du domaine ou une description ne respectant pas les critères énoncés sera pénalisée. Un des critères de complexité d'un modèle est le nombre de prédicats, fonctions et variables impliqués. Un des critères d'optimalité est la complexité de l'espace d'états sous-jacent.

On appelle un **prédicat statique**, un prédicat dont la valeur dans l'état initial reste vraie par la suite, dans tous les états, quel que soit le chemin d'exécution. Par exemple, la position des maisons ne changeant jamais,  $\text{loc}(W,P1)$ ,  $\text{loc}(H1,P2)$  et  $\text{loc}(H2,P3)$  sont statiques. Cette notion de « prédicat statique » interviendra dans les questions plus loin.

a. (2 points) Pour un domaine de planification, où il y a un seul drone qui peut transporter un seul objet à la fois, donnez la spécification PDDL pour les actions suivantes :

- **$\text{pickup}(d,c,l)$**  : le drone  $d$  prend le colis  $c$  **depuis l'endroit  $l$**
- **$\text{fly}(d,l1,l2)$**  : le drone  $d$  se déplace de l'endroit  $l1$  à l'endroit  $l2$
- **$\text{deliver}(d,c,l)$**  : le drone  $d$  livre le colis  $c$  **à l'endroit  $l$**

```
(define (domain Q1.a)
  (:types physobj
    drone – physobj
    package – physobj
    location
  )

  (:predicates (loc ?x – physobj ?y – location)
    (has ?d – drone ?x package)
  )
)
```

```

(:action pickup
  :parameters (?d - drone ?c - package ?l - location)
  :precondition (and (not (has ?d ?c)) (loc ?c ?l) (loc ?d ?l))
  :effect (and (has ?d ?c) (not (loc ?c ?l)))
)

(:action fly
  :parameters (?d - drone ?l1 - location ?l2 - location)
  :precondition (and (loc ?d ?l1) (not (loc ?d ?l2)))
  :effect (and (not (loc ?d ?l1)) (loc ?d ?l2))
)

(:action deliver
  :parameters (?d - drone ?c - package ?l - location)
  :precondition (and (has ?d ?c) (not (loc ?c ?l)) (loc ?d ?l))
  :effect (and (has not (?d ?c) (loc ?c ?l)))
)
)

```

- b. (2 points) Soit le but et l'état initial ci-après, où  $W$  désigne un entrepôt,  $H1$  une maison,  $H2$  une maison et  $D$  un drone. Décrivez, de façon précise et complète, les quatre premières transitions (quatre premières actions et états correspondants) le long d'un chemin de l'espace d'états correspondant à une solution optimale. Pour chaque état, décrivez uniquement les prédicats non statiques. Votre description doit être symbolique. Elle ne peut pas être graphique. Si votre domaine de définition a nécessité l'ajout d'autres prédicats ou fonctions, complétez la description de l'état initial en conséquence, de façon à ce qu'elle soit cohérente.

**(Le but et l'état initial ont été modifiés pour respecter la question 1.a : on travaille directement avec les lieux et non les localisations. Les modifications respectent donc l'énoncé initial de la question, en effet :  $P1 == W$ ,  $P2 == H1$  et  $P3 == H2$ )**

But :  $\text{loc}(C1, H1) \wedge \text{loc}(C2, H2)$

État initial :  $\text{loc}(D, H1) \wedge \text{loc}(C1, W) \wedge \text{loc}(C2, W) \wedge \text{loc}(C3, W)$

Action 1 :  $\text{fly}(D, H1, W)$

État résultant :  $\text{loc}(D, W)$

Action 2 :  $\text{pickup}(D, C1, W)$

État résultant :  $\text{has}(D, C1)$

Action 3 :  $\text{fly}(D, W, H1)$

État résultant :  $\text{loc}(D, H1)$

Action 4 : deliver(D,C1,H1)

État résultant : loc(C1,H1)

États finaux (après 4 actions) : loc(D,H1)  $\wedge$  loc(C1,H1)  $\wedge$  loc(C2,W)  $\wedge$  loc(C3,W)

- c. (2 points) En partant de votre définition du domaine à la question (b), donnez une nouvelle définition pour une situation où nous avons plusieurs drones, chacun pouvant transporter plusieurs colis à la fois. Vous devez souligner les nouveaux prédicats, fonctions ou actions introduits s'il y a lieu.

Il n'est pas nécessaire de définir un nouveau domaine si nous incluons plusieurs colis et drones : en effet, les prédicats et les fonctions d'actions prennent en paramètre à chaque fois une variable drone et colis. Cela permet donc de préciser qui transporte quoi et vers où. De plus, le fait de transporter de multiples colis et d'avoir plusieurs drones ne changent en rien les contraintes spécifiées au domaine précédent.

- d. (2 points) Selon votre définition à la question (c), donnez une illustration la plus simple possible d'un plan optimal pour deux drones livrant deux colis, à partir d'un état initial et d'un but que vous avez choisis. Vous devez indiquer d'abord l'état initial, ensuite le but et enfin les actions du plan avec les contraintes d'ordonnancement clairement indiquées. Vous n'avez pas besoin d'indiquer les effets des actions dans le plan.

État initial : loc(D1,W)  $\wedge$  loc(D2,W)  $\wedge$  loc(C1,W)  $\wedge$  loc(C2,W)

But : loc(C1,H1)  $\wedge$  loc(C2,H2)

Plan :

1. pickup(D1,C1,W)
2. pickup(D2,C2,W)
3. fly(D1,W,H1)
4. fly(D2,W,H2)
5. deliver(D1,C1,H1)
6. deliver(D2,C2,H2)

États finaux: loc(C1,H1)  $\wedge$  loc(C2,H2)

- e. (2 points) La question (c) revient à un planificateur centralisé. En moins de 80 mots (plus ou moins cinq lignes), expliquez clairement, mais succinctement, pourquoi ce ne serait pas une bonne solution par rapport à une planification distribuée ?

On le remarque directement sur le plan de la question d), beaucoup d'actions peuvent être distribuées au sein de chaque drone afin de paralléliser le travail. La solution proposée dans la question d) est très séquentielle car le planificateur est centralisé et est donc celui qui distribue les tâches à effectuer pour chaque drone. Avec une planification distribuée, chaque drone aura son « propre » planificateur : cela permettra donc une exécution et une livraison plus rapide.

- f. (2 points) En moins de 80 mots (plus ou moins cinq lignes), expliquez clairement, mais succinctement, une approche de planification distribuée qui serait meilleure selon les faiblesses identifiées à la question précédente.

Comme indiqué précédemment, chaque drone doit posséder son propre planificateur qui lui ordonne les tâches et les actions à effectuer. Cependant, qui dit système d'agent distribué, dit des contraintes supplémentaires à respecter dont : la coordination entre les drones (afin d'éviter des collisions ou bien la récupération d'un même colis), un moyen de communication et une distribution des tâches connues par chaque drone.

## Question 2 (6 points) – Contrôle de la recherche avec les formules LTL

Pour votre définition de domaine à la question 1(a), donnez une formule de contrôle de recherche la plus efficace possible tout en préservant la complétude et l'optimalité.

Définition du domaine : « Un drone D transportant un colis C d'une localisation L1 à une localisation L2 ».

Formule LTL :

$$\Diamond (\text{loc}(D, L1) \wedge \text{loc}(C, L1) \rightarrow \text{pickup}(D, C, L1) \rightarrow (\text{has}(D, C) \wedge \Box \text{not loc}(C, L1) \wedge \text{fly}(D, L1, L2)) \rightarrow \Diamond \text{loc}(D, L2) \rightarrow \text{deliver}(D, C, L2) \rightarrow \Box (\text{loc}(C, L2) \wedge \text{not has}(D, C)))$$

Explications : Lorsque le colis et le drone se trouvent éventuellement au même endroit L1, cela implique que ce dernier peut récupérer le colis. Cette action implique par la suite que le colis est avec le drone, que celui-ci ne sera plus jamais à la localisation L1 et qu'on peut voler de cette localisation à la L2. Cela implique qu'un moment donné le drone sera à la localisation L2 et dans ce cas, le drone peut déposer le colis. Finalement, cette livraison implique pour toujours que le colis sera à la localisation L2 et que le drone ne pourra pas le récupérer.

### Question 3 (6 points) – Contrôle de recherche avec HTN

Donnez une définition du domaine HTN, pour un planificateur HTN, qui planifierait pour un domaine de planification équivalent à celui de la question (a).

Définition du domaine : « Un drone D transportant un colis C d'une localisation L1 à une localisation L2 ».

Etat initial :  $\text{loc}(C, L1) \wedge \text{loc}(D, L3)$

But :  $\text{loc}(C, L2)$

Task : Livraison du colis C à L2

Method : Le colis C est à L1 et le drone D est à L3

Precond :  $\text{loc}(C, L1) \wedge \text{loc}(D, L3)$

Subtasks :

Subtask 1 : Aller à L1

Action :  $\text{fly}(D, L3, L1)$

Precond :  $\text{loc}(D, L3)$

Effects :  $\text{loc}(D, L1)$

Subtask 2 : Récupérer le colis C

Action :  $\text{pickup}(D, C, L1)$

Precond :  $\text{loc}(C, L1) \wedge \text{loc}(D, L1)$

Effects :  $\text{has}(D, C) \wedge \text{not loc}(C, L1)$

Subtask 3 : Aller à L2

Action :  $\text{fly}(D, L1, L2)$

Precond :  $\text{loc}(D, L1)$

Effects :  $\text{loc}(D, L2)$

Subtask 4 : Déposer le colis C

Action :  $\text{deliver}(D, C, L2)$

Precond :  $\text{has}(D, C) \wedge \text{loc}(D, L2)$

Effects :  $\text{not has}(D, C) \wedge \text{loc}(C, L2)$

La tâche associée à notre domaine peut-être décomposée en 4 sous-tâches ordonnées :

1. Aller à L1
2. Récupérer le colis C
3. Aller à L2
4. Déposer le colis C

Chacune de ces sous-tâches contient une action primitive :

- pickup
- fly
- deliver

#### Question 4 (6 points) – Planification de trajectoire

Le drone devra pouvoir éviter des obstacles (arbres, lignes de transport d'électricité, bâtiments en hauteur).

- a. (3 points) En moins de 200 mots, expliquez clairement l'algorithme de planification de déplacements que vous utiliseriez (le nom de l'algorithme, ses entrées et sorties, ses modèles et paramètres) et justifiez votre choix par rapport à deux autres alternatives, qui seraient, dans l'ordre de préférence, les plus pertinentes.

Au vu de notre environnement très complexe, notre drone se doit être capable d'éviter les collisions et de naviguer intelligemment. Pour cela, je propose RDT+ (A Parameter-free Algorithm for Exact Motion Planning, Nikolaus Vahrenkamp et al, 2011), un algorithme de planification de mouvement se basant sur l'algorithme RDT (Rapidly-exploring Dense Tree). Avec RDT+, les paramètres de détection des collisions d'un arbre dense à exploration rapide sont automatiquement ajustés jusqu'à ce qu'une solution exacte puisse être trouvée. Pour une planification efficace, des routines discrètes de détection des collisions sont utilisées dans le planificateur RDT et, en vérifiant les résultats avec des méthodes exactes de détection des collisions, RDT+ permet de calculer des mouvements qui sont garantis sans collision. En sortie, nous obtenons un plan de mouvement pour le drone.

Logiquement, nous pouvons citer aussi RDT, cependant l'inconvénient avec celui-ci, c'est qu'il faut fournir les paramètres de détection des collisions d'un arbre dense à exploration rapide.

Finalement, il y a aussi l'algorithme A\* permettant de rechercher un chemin entre deux points. Cependant, celui-ci est adapté pour un environnement statique et non dynamique. Il performera donc très mal dans notre cas d'étude.

- b. (3 points) En moins de 150 mots, expliquez clairement comment vous intégreriez l'algorithme de planification de tâches (PDDL ou HTN) avec l'algorithme de planification de déplacements et joignez un diagramme d'architecture à votre explication.

On remarque que chacune des sous-tâches décrites dans notre algorithme de planification de tâches (HTN) nécessite un déplacement. Par conséquent, il est nécessaire de connaître tout d'abord les tâches à effectuer pour savoir les contraintes qui y sont impliquées au niveau de chaque déplacement. Il faut donc initier tout d'abord les tâches puis en extraire les contraintes nécessaires qui alimenteront notre algorithme de planification de déplacements.



