

INF4215 - Travail pratique #1.1

Hiver 2014

1 Description

Le premier mini-projet réalisé dans le cours permettra d'implémenter plusieurs modèles d'agents intelligents selon différentes contraintes. Dans le cadre de ce premier travail pratique, vous devrez représenter la situation décrite dans un espace d'états et implémenter un agent de base afin de résoudre le problème présenté. Plus précisément, vous devrez implémenter deux types d'agents réflexes : sans mémoire et avec mémoire.

Un robot travaillant dans un entrepôt a comme tâche de déplacer des boîtes d'une section à une autre. Le plan d'un entrepôt est présenté à la figure 1 et consiste de quatre comptoirs (A, B, E et F) et de deux intersections (C et D). Chaque section peut contenir plusieurs boîtes à être transférées et ces dernières possèdent toutes un poids spécifique. De plus, le robot possède une contrainte de poids, c'est pourquoi il est possible qu'il ne puisse recueillir tous les paquets présents dans une section en une seule visite. Ainsi, le robot devra revenir au comptoir ultérieurement afin d'accomplir sa tâche.

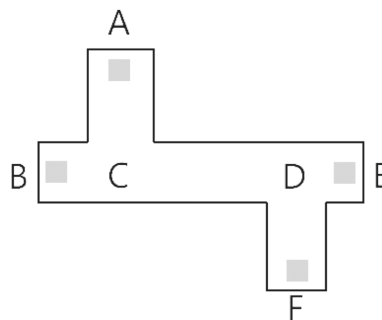


FIGURE 1 – Exemple de problème

2 Représentation du problème en Python

Une partie de la représentation du problème vous est fournie. Voici une brève description de chaque élément :

Agent	<p>Représente un agent abstrait. Les agents que vous implémenterez dériveront de cette classe. La classe possède les implémentations des actions que le robot peut effectuer dans l'environnement telles que <i>take</i>, <i>drop</i>, etc.</p> <p>Toute sous-classe concrète devra définir la méthode <i>chooseAction</i>, qui détermine l'action choisie selon l'interprétation faite de l'environnement, qui lui est passée en paramètre. Un agent avec mémoire devra aussi redéfinir la méthode <i>updateModel</i>.</p>
Environment	Cette classe est la scène où le robot se promène afin d'accomplir sa tâche et contient les méthodes pour manipuler l'environnement. C'est donc dans cette classe que l'agent sera appelé à faire ses actions.
Graph	Cette classe contient les nœuds et les arcs d'un graphe afin de représenter les comptoirs et les couloirs de l'entrepôt. La classe possède des méthodes pour obtenir le chemin entre deux comptoirs et sa distance.
Package	Représente les boîtes à déplacer d'un comptoir à un autre. Un paquet contient principalement une liste d'attributs permettant de savoir son identificateur unique, sa destination et son poids.
Exemple	Une classe contenant un environnement implémenté. Bref, cela vous permettra de tester vos agents. Cette classe peut être considérée comme le point d'entrée du travail pratique.

3 Travail à réaliser

Dans ce travail pratique, vous devrez implémenter deux types d'agents réflexes qui permettront de résoudre le problème du robot. Pour y arriver, vous avez trois tâches principales à réaliser :

1. Compléter la méthode *executeAction* dans la classe *Environment*. La fonction a tout simplement la responsabilité de permettre au robot d'effectuer l'action passée en paramètre.
2. Implémenter un agent réflexe **sans** mémoire *ReflexAgent*. Dans cette implémentation de l'agent, ce dernier ne peut garder aucune mémoire sur les paquets présents dans les comptoirs. Ainsi, le robot ne peut se rappeler qu'il a dû laisser des boîtes dans un comptoir.

3. Implémenter un agent réflexe **avec** mémoire *ModelBasedAgent*. Contrairement à l'implémentation précédente, cet agent peut connaître le nombre de paquets dans tous les comptoirs après les avoir visités. Votre tâche sera de trouver un ensemble de règles permettant au robot d'utiliser efficacement l'information qu'il possède à propos des paquets et des comptoirs. L'agent réflexe avec mémoire doit également implémenter la méthode *updateModel*.

4 Déroulement de la compétition

Il y aura également une compétition entre les équipes afin de déterminer la meilleure implémentation de l'agent réflexe avec mémoire. Cette compétition sera basée sur le nombre d'actions moyen effectué par le robot pour résoudre le problème.

Une implémentation de l'agent avec mémoire offrant le même nombre de coups que l'agent sans mémoire aura une note de 0 pour la compétition. Autrement, la ou les meilleures équipes auront la note maximale de 15. Par la suite, les autres équipes seront divisées en groupes égaux (le plus possible) et auront une note relative à leur position. Par exemple, pour 26 groupes et deux équipes qui se sont démarquées :

15/15	2 équipes
12/15	6 équipes
9/15	6 équipes
6/15	6 équipes
3/15	6 équipes

5 Questions

Question 1 : Expliquez ce que fait le code suivant :

```
def fct(f, g, p, l):  
    return map(lambda x: f(g(x)),  
               [x for x in l if p(x)])
```

Question 2 : Quelles sont les faiblesses de votre implémentation de l'agent avec mémoire ?

6 Directives pour la remise

Le travail sera réalisé en équipe de deux personnes. Vous téléchargerez sur le site du cours un fichier unique contenant les classes mentionnées dans la deuxième section. Un programme pour tester votre agent vous est fourni dans le fichier *example1.py*.

Vous devez uniquement implémenter les fonctions *allDelivered*, *percept* et *executeAction* de la classe *Environment*. De plus, implémentez vos agents dans les fichiers *reflexAgent.py* et *modelBasedAgent.py*. Remettez tous vos fichiers Python dans un fichier compressé (zip, rar, etc.) où ce dernier est nommé comme suit : *matricule1_matricule2.zip*. Vous devez également remettre un fichier (pdf, doc, docx, etc.) contenant une explication de votre implémentation de l'agent avec mémoire ainsi que les réponses aux deux questions.

Tout devra être remis avant **le 2 février à 23h55**. Tout travail en retard sera pénalisé d'une valeur de 20% pour chaque jour de retard. Le barème pour l'évaluation est le suivant :

Environnement	10 %
Implémentation de l'agent réflexe	25 %
Implémentation de l'agent avec mémoire	40 %
Résultat de la compétition	15 %
Questions	10 %

7 Bibliothèque

Pour les étudiants qui travaillent sur leur ordinateur personnel, il est nécessaire d'installer la bibliothèque *python-graph-core* afin de pouvoir exécuter le code donné. Pour ce faire, si vous avez *easy-install*, la commande est tout simplement :

```
easy-install python-graph-core
```

Autrement, il est également possible de télécharger la bibliothèque à l'adresse suivante : <https://code.google.com/p/python-graph/> et exécuter la commande ci-dessous à l'intérieur du dossier *core*.

```
python setup.py install
```