Intelligence Artificielle

Devoir 1

Xavier Dubuc
Xavier.DUBUC@student.umons.ac.be

22 octobre 2010

Table des matières

1 Question 1

*Parcourez la littérature portant sur l'***IA** *pour savoir si les tâches suivantes peuvent être actuellement résolues par ordinateur :*

- **jouer un bon niveau au tennis de table,** le robot **Topio** développé par **TOSY** (firme de robotique vietnamienne) permet de jouer au tennis de table de manière assez convaincante, on peut voir la version **2.0** en oeuvre ici : http ://www.youtube.com/watch ?v=SE6tFX5QM_U , on l'aperçoit même en train de battre son adversaire. De plus, on peut apprendre sur wikipédia *(entre autres)* qu'une version **3.0** a été développée et qu'il apprend constamment en jouant, ce qui laisse à penser que cette version joue bien mieux que la version **2.0**. La réponse est donc OUI.
- **conduire à Victorville, en Californie,** Je cite ce pdf : ≪*In 2007 a tricked-out Chevrolet Tahoe nicknamed "Boss" and several similar driverless vehicles successfully navigated through a realistic city streetscape in Victorville, Calif., one complete with other cars and even traffic jams. The autonomous cars and trucks were competing in the Defense Advanced Research Projects Agency's Urban Challenge, arace designed to demonstrate that robot road vehicles can become practical. Soon afterward General Motors CEO G.Richard Wagoner, Jr., predicted his company will market autonomous vehicles within 10 years. That prognostication may be a bit optimistic, but his statement surely points the way to real robotic cars in the not too distant future.*≫ ⇒ La réponse est donc OUI.
- **jouer au Bridge (niveau compétition),** l'article le plus récent, ≪*Search in games with incomplete information : a case study using Bridge card play*≫, écrit par **Ian Frank** et **David Basin**, fait état d'une intelligence artificielle qu'ils ont développé et qui permet de jouer au **Bridge** avec la tactique ≪*Best Defense*≫. On en déduit que le niveau compétition n'est pas atteint par cette intelligence artificielle, la réponse est donc NON.
- **découvrir et prouver de nouveaux théorèmes mathématiques,** malgré de longues recherches, le peu que j'ai pu trouver fut quelques articles des années 70 spécifiant comment prouver/réfuter un large nombre de résolution linéaire *(même avec la VPN je n'ai pas pu avoir accès au fichier PDF)*. Rien n'est trouvable quant à la découverte de théorèmes par contre, ce qui oriente ma réponse vers le NON.

2 Question 2

- *Décrire une fonction pour un agent rationnel dans le cas où chaque mouvement coûte un point (pour le reste les hypothèses sont identiques à celles données au transparent 22 du chapitre 2, y compris le gain d'un point par carré néttoyé). Le programme de l'agent correspondant exige-t-il de conserver un état interne (agent avec modèle) ?*

⇒ **Première ébauche**

```
function MODEL-VACUUM-AGENT ([location,status]) returns an action
    persistent : action, l'action la plus récente (initialement vide)
    if status = Sale then action ← Aspire
    else
        if location = A then
            if action = Gauche then action ← Stop
            else action ← Droite
        else
            if action = Droite then action ← Stop
            else action ← Gauche
    return action
```

Le robot se comporte de la manière suivante :

- si le carré où il se trouve est sale, quoi qu'il arrive, il l'aspire ;
- si ce carré n'est pas sale, il consulte la dernière action qu'il a effectué, s'il vient de se déplacer, c'est que le précédent carré était propre, il peut donc s'arrêter, son travail est terminé ;
- sinon, dans tous les autres cas, (c'est-à-dire que le carré est propre et il ne vient pas de se déplacer), il doit changer de carré.

Avec cette fonction d'agent, le score maximal atteignable est 0 (par exemple, départ en A, il aspire (+1) il va en B (-1), il aspire (+1) puis retourne en A (-1) et s'arrête) ; pour maximiser ce score il faudrait empêcher le retour en A à la fin ; autrement dit stocker l'état du "monde". On va donc stocker

l'état de cette manière : (*statusA*,*statusB*). Ceci nous permettra d'éviter le dernier retour en A car *state* contiendra (*Propre*,*Propre*) (cette variable contient (*Sale*,*Sale*) initialement).

Voici cette fonction :

```
function MODEL-VACUUM-AGENT ([location,status]) returns an action
    persistent : action, l'action la plus récente (initialement vide)
                 state, contenant un couple représentant l'état de l'environnement
    if status = Sale then action ← Aspire
    else
        if location = A then
            state = (Propre,state[1])
            if state = (Propre,Propre) then action ← Stop
            else action ← Droite
        else
            state = (state[0],Propre)
            if state = (Propre,Propre) then action ← Stop
            else action ← Gauche
    return action
```

Dans ce cas-ci, le robot ne se déplacera que s'il n'est pas sûr que l'autre carré n'est pas propre. Ainsi, dans l'exemple employé précédemment :
– il commence en **A**, *state* = (*Sale*,*Sale*) et *status* = *Sale*, donc il aspire (+1)
– *state* = (*Sale*,*Sale*) et *status* = *Propre*, il met *state* à jour et se déplace en **B** (-1)
– il arrive en **B**, *state* = (*Propre*,*Sale*) et *status* = *Sale*, donc il aspire (+1)
– *state* = (*Propre*,*Sale*) et *status* = *Propre*, il met *state* à jour et s'arrête car *state* = (*Propre*,*Propre*)
⇒ Le robot obtiendra donc un score compris entre -1 (les 2 carrés étaient propres) et 1 (les 2 carrés étaient sales).
Comme l'explication précédente le pointe, le robot nécessite la conservation d'un état interne lui permettant de savoir à tout instant la dernière action effectuée.

– *Discutez des conceptions possibles d'un agent pour les cas où des carrés propres peuvent devenir sales (par exemple, pour chaque unité de temps, chaque carré propre à 10% de chances de devenir sale) et où la géographie de l'environnement est inconnue. Est-il judicieux pour l'agent d'apprendre de son expérience dans ces cas ? Si oui, que devrait-il apprendre ? Si non, pourquoi pas ?*

⇒ Il est très judicieux d'apprendre de son environnement, ne fusse que pour la géographie, le fait de savoir comment se découpe son environnement peut l'amener à optimiser sa performance. Si on prend le même exemple que ci-dessus, sans connaître la géographie de l'endroit où il évolue, ses décisions ne peuvent être aussi précises, il devra scruter son environnement à l'aveugle. Quant au fait que les carrés redeviennent propres à 10% de chances, il peut, en apprenant ça se dire qu'il ne revisite un carré que si ça fait $n \geq 10$ unités de temps qu'il ne l'a pas visité, minimisant ainsi ses déplacements (s'il y a 3 carrés et qu'il revisite sans cesse les carrés pour vérifier qu'ils sont toujours propres, il fait énormément de déplacements inutiles, alors que s'il aspire les 3 à la suite toutes les 10 unités de temps, il ne fait pas de déplacements inutiles).

3 Question 3

Considérons un simple thermostat qui allume un four lorsque la température est d'au moins 3 degrés en dessous du réglage, et qui l'éteint lorsque la température est d'au moins 3 degrés au dessus du réglage. Un thermostat est-il une instance d'un simple agent réflexe, d'un agent réflexe basé sur un modèle ou d'un agent basé sur des objectifs ? Justifiez.

⇒ Il s'agit d'un simple agent réflexe, sa décision ne dépend que de la perception courante. En effet, à chaque unité de temps, il obtient une perception de la température du four, si elle est supérieur au réglage il coupe le four, si elle est inférieur il l'allume sinon il ne fait rien.

4 Répartition du temps

1. entre 1 et 2h, certains sujets sont difficiles à trouver et les articles sur le bridge et Topio m'ont assez bien captivés, j'ai donc passé pas mal de temps à les lire (ou à regarder des vidéos).
2. 1h environ, en comptant la mise en page LaTeX.
3. 5 minutes.