

Travail Pratique pour le cours des Systèmes Multiagents

But

Le but du présent travail est donner l'opportunité aux étudiants de mettre en pratique les connaissances apprises dans le cours : modélisation multiagent, adaptation des algorithmes d'apprentissage ou de planification pures à un problème réel, aspects de la programmation multiagent et méthodes d'évaluation de la performance des agents.

Méthodologie

Les étudiants sont invités à modéliser et à programmer un environnement multiagent en utilisant un des *frameworks* modernes de la modélisation et de la programmation multiagent (JADE [1] par exemple). Les étudiants doivent ensuite choisir un ou plusieurs algorithmes pour chacun des agents et les programmer. Finalement, les étudiants doivent faire les expérimentations et rédiger un rapport sous forme d'un travail scientifique.

Le problème étudié

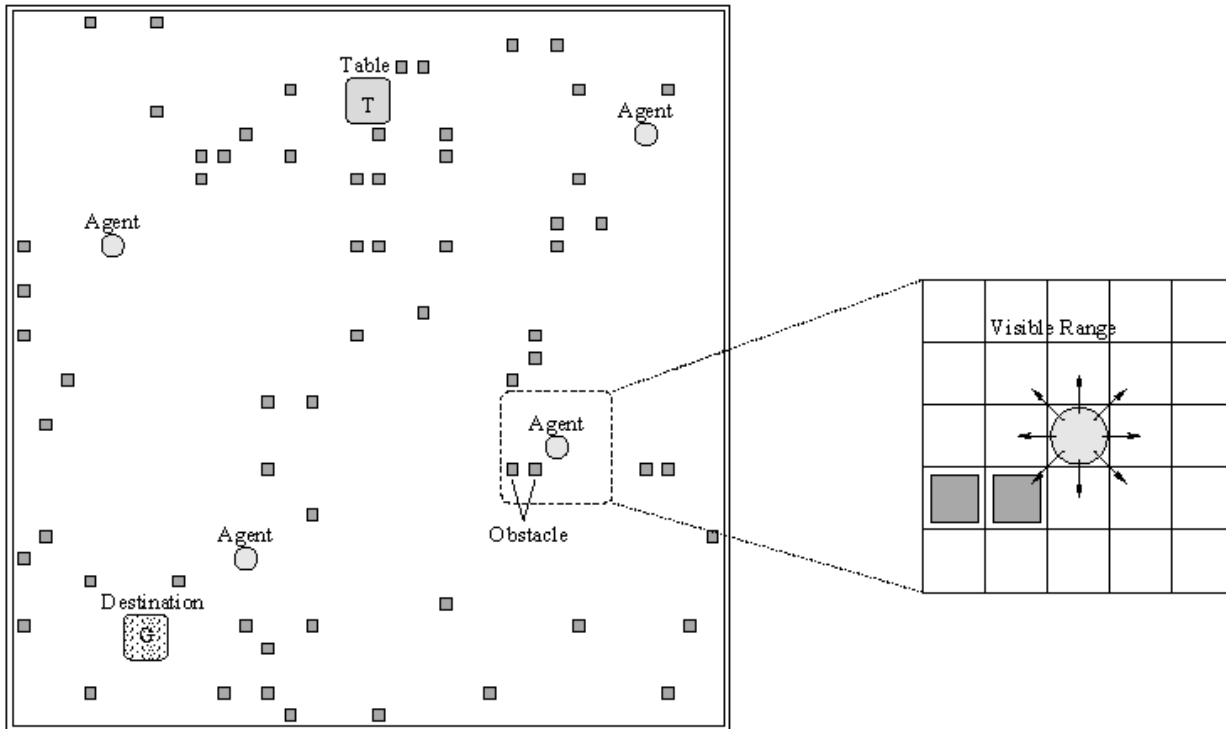


Figure 1

Soit une grille 32x32 cases (Figure 1, à gauche). Sur la grille, il y a 4 agents et 64 obstacles; il y a aussi un endroit spécial appelé « destination » (4x4 case) et une table (4x4 cases). Chaque agent a une vision locale de 5x5 cases autour de lui et est capable d'exécuter 8 différentes actions de déplacement (Figure 1, à droite) ; il y a aussi 8 actions spéciales : « pousser à (direction) » pour pousser la table; et 3 actions « communiquer à X » pour envoyer de l'information à l'agent X. Il faut au moins deux agents pour pousser la table à la case voisine. Trois (quatre) agents peuvent

déplacer la table de deux (trois) cases à la fois. Pour pouvoir pousser la table, l'agent doit se mettre sur une des 4 cases occupées par la table. En tout temps, une case peut être occupée par un seul agent. Pour pouvoir pousser la table dans une direction voulue, tous les agents doivent pousser dans la même direction. Les agents peuvent communiquer entre eux en vue d'échanger n'importe quelle information à l'aide de l'action « communiquer à X ».

Le temps est divisé en des *étapes* discrètes. À chaque étape, une seule action peut être exécutée par un agent (soit une action de déplacement, soit une des actions « pousser à (direction) » ou une des actions « communiquer à X »).

Une *simulation* est composée de plusieurs étapes. Elle commence par des positions aléatoires (et inconnues aux agents) des obstacles, de la table et de la destination. Les positions initiales des agents sur la grille sont aussi aléatoires, mais chaque agent connaît en tout moment sa propre position sur la grille.

Durant une simulation, les agents doivent découvrir les positions de la table et de la destination et rapporter la table à la destination à l'aide des actions disponibles tout en évitant les obstacles. Une fois la table mise à la position « destination », la simulation courante se termine et une nouvelle simulation commence, avec les nouvelles positions de tous les éléments.

Le problème consiste à résoudre 100 simulations en utilisant le minimum d'étapes. Chaque étape ajoute un point au compte de l'équipe. L'équipe qui aura le moins de points à la fin de la centième simulation gagne et obtient un bonus de 5 points à la fin du trimestre. Les simulations doivent être lancées à la fin du trimestre devant le professeur. À cette fin, le programme doit avoir une interface graphique qui visualise en temps réel l'état de l'environnement et les points.

Approche

Chaque équipe doit résoudre ce problème indépendamment des autres équipes. Les étudiants doivent d'abord modéliser l'environnement en utilisant le framework JADE. Ensuite, ils doivent choisir les rôles pour chacun des agents, les algorithmes d'apprentissage ou de la planification, les stratégies de communication, etc. Par la suite, les étudiants doivent programmer les quatre agents et lancer plusieurs simulations pour mesurer les performances des agents en fonction de différents paramètres des algorithmes choisis.

Les étudiants sont libres d'utiliser dans leur travail tous les algorithmes appris dans le cours ainsi que d'autres algorithmes d'apprentissage ou de la planification qu'on peut trouver dans la littérature. Google Scholar (scholar.google.com) est un des moteurs de recherche des publications scientifiques. Les mots clés pourraient être : path planning, path finding, multiagent problem solving, multiagent exploration, multiagent coordination, multiagent planning. **Un des critères importants d'évaluation de ce travail pratique sera la qualité de sa partie scientifique.**

Toutes les décisions clefs prises lors des phases de modélisation et d'implantation, ainsi que les résultats obtenus doivent être décrites dans un rapport. Ce rapport devrait inclure les parties suivantes : Introduction, Préalables théoriques, Modélisation de l'environnement, Modélisation des agents, Expérimentations, Conclusion.

À titre d'exemple de décisions qu'une équipe pourrait prendre au cours de ce travail, on peut supposer que : deux agents sur quatre font l'exploration de l'environnement en se déplaçant de façon aléatoire sur la grille. Les deux autres agents poussent la table une fois connue la position de la table et de la destination. Pour la communication, on pourrait considérer un agent dispatcher à qui tous les autres communiquent; il lui revient ensuite de distribuer l'information aux autres. Pour trouver le plus court chemin entre deux cases, les agents utilisent l'algorithme A* [5]. Dans cet exemple, il reste à préciser la politique de la coordination multiagent et la politique de la communication (quand et quoi communiquer) pour chacun des agents.

Programmation

Le travail doit être effectué en utilisant le langage de programmation Java avec JADE [1] (un environnement de programmation multiagent basé sur Java). Pour s'inspirer de la technique de programmation, les étudiants peuvent voir la page

<http://www.cs.uu.nl/docs/vakken/mas/Pract/exercise2.html>.

Elle reflète un exemple d'environnement multiagent programmé avec JADE. Il y a plusieurs tutoriels au sujet de JADE disponible sur Internet, les exemples incluent :

<http://www.iro.umontreal.ca/~vaucher/Agents/Jade/JadePrimer.html>,

<http://www.ryerson.ca/~dgrimsha/courses/cps720/JADEAdmin/startJade.html>

et

<http://www.jade.co.nz/jade/tutorials.htm>.

Une autre option est d'utiliser un autre framework de la programmation multiagent, NetLogo [4]. Plusieurs tutoriels et exemples peuvent être trouvés sur les pages suivantes : <http://ccl.northwestern.edu/netlogo/models/> et <http://jmvidal.cse.sc.edu/netlogomas/>.

Liens

[1] JADE: Java Agent DEvelopment Framework, <http://jade.tilab.com/>.

[2] Developing Multi-agent Systems with JADE, <http://www.dia.fi.upm.es/~phernan/AgentesInteligentes/referencias/bellifemine01.pdf>.

[3] Block World for JADE, <http://www.cs.uu.nl/docs/vakken/mas/Pract/exercise2.html>.

[4] NetLogo, <http://ccl.northwestern.edu/netlogo/>.

[5] S. Russell and P. Norvig. « Artificial Intelligence: A Modern Approach », Prentice Hall, 2003.