

Projet SMA

Altruisme dans les Systèmes Multi-Agents via la plateforme GAMA

RAGNAUD L. & LEVEQUE D.

Table des matières

Introduction :.....	3
Modèle :.....	4
L'environnement :.....	4
Les agents :.....	4
Interaction avec l'utilisateur :.....	6

Introduction :

Le travail réalisé a pour but d'étudier le comportement altruiste des agents dans le cadre d'un système multi-agents pour la collaboration dans l'exécution de tâches. Afin de modéliser de manière plus concrète le modèle, un plan de fourmis a été adopté. Les agents sont donc des fourmis effectuant des aller-retours entre deux points, la colonie, et la source de nourriture. Notre travail est basé sur le modèle exprimé dans l'article "Implementation and Evaluation of a Satisfaction/Altruism Based Architecture for Multi-Robot Systems" de P. Lucidarme, O. Simonin, A. Liégeois. Le projet a été réalisé sur la plateforme GAMA.

Modèle :

L'environnement :

Les fourmis évoluent dans un environnement quadrillé. La plateforme est de 50x50 pixels, chaque cellule ayant pour attributs deux potentiels en float (sur lesquels nous reviendrons), un booléen pour indiquer si elle est occupée ou non, sa couleur d'affichage, et deux listes de voisins (ceux en position « + » directe, et ceux à 5 cases de distance).

Lors de la création du monde, nous fournissons au programme une image 50x50 px que nous avons nous même faite. Sur celle-ci figurent deux gradients de couleur, un bleu et un vert, et un chemin bloqué par des murs (en rouge). Les gradients de couleur permettent de désigner un centre, dans lequel la couleur sera à 255, et plus faible avec l'éloignement, sans pour autant devenir nulle. Une case (un pixel) aura alors une valeur de bleu, et une valeur de vert. Ces valeurs seront extraites et utilisera la valeur verte pour le potentiel vers le but (la nourriture), et bleue pour la colonie. Le potentiel sera borné entre 0.0 et 1.0. Le potentiel agira donc comme une « odeur » guidant les fourmis vers la source. Le rouge est utilisé comme mur, mettant toutes les cases colorées en rouge à un potentiel à 0.0 pour les deux sources, et les considérant comme « occupées » en sécurité, afin d'éviter que les fourmis n'y accèdent.

Les cases ayant un potentiel de nourriture >0.95 , ou un potentiel de colonie >0.85 seront considérées comme étant la nourriture ou la colonie même. Elles seront listées dans les listes `true_goal` et `true_colony` respectivement. Ces listes serviront donc à initialiser nos agents fourmis aléatoirement sur les cases appartenant à la liste `true_colony`.

Les agents :

Les fourmis disposent de plusieurs attributs que nous prendrons dans l'ordre d'apparition. Tout d'abord, une chaîne de caractère désignant leur but actuel, ainsi qu'un booléen « charged » indiquant si elles sont actuellement en train de porter de la nourriture ou non. Ces valeurs décideront quel potentiel les fourmis regarderont lors de leur choix de déplacement. Un objectif « feed » leur fera regarder le potentiel vers la nourriture de la case, alors qu'un objectif « home » les guidera vers la colonie.

Leurs attributs `size` et `color` servent à l'affichage pour l'utilisateur, qui seront remplacés par une image `my_icon` pour un aspect purement esthétique.

Chaque fourmi garde sa position actuelle dans `my_cell`, référencent la cellule sur laquelle elles sont actuellement présente, ainsi que la localisation précédente dans `prev_loc`, qui n'a pas été utilisée, mais pourra servir à d'éventuelles évolutions.

Enfin, dans l'aspect d'altruisme, chaque fourmi est dotée d'un attribut `satisfaction`, allant de `max_satisfaction` à `-max_satisfaction`. Lorsqu'une fourmi se rapproche de son objectif, sa satisfaction augmente, mais si elle se retrouve bloquée, elle diminue. Ce changement est fait par paliers déterminés par `satisfaction_increment`. `Fleeing`, encore dans l'altruisme, indique si oui ou non, une fourmi a

abandonné sa tâche personnelle afin de laisser la place à une autre fourmi pour le passage. Globalement, nous avons déclaré la variable `ant_threshold_of_dissatisfaction`, désignant le palier de satisfaction à partir duquel une fourmi commencera à manifester son mécontentement.

Un nombre de reflex sont présents dans la déclaration de l'agent. Premièrement `take_putdown` (l.77), indiquant que si la fourmi se trouve sur une case considérée comme « vraie » nourriture ou colonie, elle pourra changer son but et se charger ou se décharger de sa nourriture. Les fourmis ayant le talent « fipa » déclaré, elles peuvent s'échanger des messages. Ainsi, `send_satisfaction` (l.91) permet aux fourmis qui atteignent une satisfaction personnelle inférieure au seuil donné dans `ant_threshold_of_dissatisfaction` de pouvoir informer les autres fourmis de sa propre satisfaction. En parallèle, nous avons le reflex `recieve_satisfaction` (l.102), qui s'active lorsque des messages sont reçus. L'agent calcule donc un coefficient dépendant de sa distance par rapport à l'agent mécontent, et le multiplie par rapport à sa propre satisfaction. Il le compare ensuite avec la satisfaction reçue dans le message par l'autre agent, et si celle-ci est supérieure à la sienne, et que sa tâche n'est pas de rentrer à la colonie, alors il considère que le but de l'agent d'en face est plus important, et se met à fuir (la fonction `flee` (l.205) ne sert qu'à changer l'attribut `flee` à true).

La fonction reflex de base de la fourmi est `move` (l.72), appelant les fonctions `goTo` et `choose_cell`. D'abord, la fonction `goTo` (l.193) prends une cellule en paramètre. Elle fait bouger la fourmi à cette nouvelle cellule, indique la cellule sur laquelle elle était comme libre, et occupe sa nouvelle cellule. Elle réinitialisera également l'attribut `fleeing` de la fourmi.

La fonction `choose_cell` (l.118) est le cœur du programme. En prenant en compte les potentiels des voisins à une case de distance de la cellule actuelle, et regardant le bon potentiel selon le but de la fourmi, nous choisissons la prochaine cellule vers laquelle la fourmi se déplacera, et gardons en mémoire le potentiel trouvé sur celle-ci. Le fait d'avoir indiqué les murs (cases rouge) comme ayant un potentiel 0.0, et ayant sécurisé en les indiquant comme à tout jamais occupées, une fourmi ne pourra jamais choisir un mur comme sa prochaine case. Si la cellule est libre, alors tout se déroule normalement, la satisfaction augmente, et la cellule choisie est retournée pour être utilisée dans `goTo`. Sinon, la fourmi garde en mémoire une liste temporaire de tous les voisins de la case ayant un potentiel plus élevé que la case actuelle, forçant la fourmi à se diriger vers son objectif, et ne pas faire du sur place entre deux cases. Toutes les cases sont essayées, et la première trouvée non occupée coupe la boucle en mettant à jour la satisfaction, et en retournant la cellule. C'est donc le cas non optimal, mais tout de même satisfaisant, car elle continue de se diriger vers son objectif. Cependant, si une fourmi est actuellement en fuite, c'est-à-dire qu'elle a reçu un signal répulsif d'une congénère, alors elle choisira la cellule avec le potentiel maximal opposé à son but, la faisant par ce fait « fuir » dans le sens opposé à là où elle souhaiterait aller. Le fait d'aider une congénère fait cependant augmenter sa satisfaction.

Tous les cas possibles de mouvement ont été traités séquentiellement. Si une fourmi n'est toujours pas sortie de la boucle, cela veut dire qu'elle ne remplis aucune condition, aucun de ses voisins n'aient libre. Elle est donc coincée, et sa satisfaction diminue, renvoyant sa cellule actuelle à `goTo` pour rester en place. Si elle reste coincée trop longtemps, sa satisfaction diminuera trop, et elle enverra un signal répulsif à ses congénères.

Interaction avec l'utilisateur :

Une interaction limitée est possible avec un système comme celui-ci. L'utilisateur aura donc principalement un rôle d'observateur, ce grâce à divers aspects implantés pour les agents, le monde, et les moyens d'affichage. Au lancement, l'utilisateur pourra décider du nombre de fourmis à créer, de leur pas de modification de satisfaction, et de leur maximum et minimum de satisfaction.

Plusieurs fenêtres s'ouvriront lors du lancement, permettant une vision détaillée du travail en cours. Premièrement, l'affichage basique des fourmis se déplaçant sur la grille, deuxièmement, les fourmis représentées par un point, avec leur satisfaction, leur objectif actuel, et l'information pour savoir si elles sont en fuite ou non. Également, un tableau sans les fourmis, indiquant seulement les informations des cases, donc leurs deux potentiels, et leur statut d'occupation, et enfin, un graphe se rafraîchissant tous les 5 cycles, affichant la distribution de satisfaction des fourmis, à titre purement indicatif.