PHY 1234 – Introduction à la physique numérique
Laboratoire #12 : Auto-organisation chez les fourmis

Novembre 2005

La cette dernière partie du laboratoire final, nous allons étudier le comportement d'une colonie de fourmis virtuelles lors de la recherche de nourriture. Nous allons étudier la sensibilité des propriétés d'auto-organisation du système aux différents paramètres qui gèrent le comportement de la colonie. Nous discuterons finalement des conditions qui rendent la colonie optimale.

1 Systèmes auto-organisés

Nous avons vu en cours quels sont les ingrédients essentiels à l'apparition d'auto-organisation dans un système complexe. En voici un petit résumé :

1. Rétroaction positive : Il existe des facteurs reforçant certains comportements.
2. Rétroaction négative : Il existe des facteurs inhibant certains comportements.
3. Présence de fluctuations : Des fluctuations aléatoires sont présentes dans le système.
4. Interactions : Il existe des interactions entre les composants individuels.

Dans le cas précis de notre colonie de fourmis, nous voulons favoriser le processus de récupération de nourriture. Les ingrédients de l'auto-organisation prennent donc la forme suivante :

1. Rétroaction positive :
   - Une fourmi qui transporte de la nourriture a tendance à retourner au nid
   - Une fourmi qui transporte de la nourriture émet des phéromones
   - Une fourmi à la recherche de nourriture a tendance à aller dans une direction où il y a beaucoup de phéromones
2. Rétroaction négative :
   - Les phéromones diffusent dans l'air
   - Les phéronomes s'évaporent dans l'environnement
   - Une fourmi n'ajoute pas de phéromones si la concentration locale dépasse un certain seuil
3. Présence de fluctuations
   - Il y a une composante aléatoire dans le mouvement des fourmis
4. Interactions :
   - Les fourmis interagissent uniquement par le biais des phéromones

Le niveau d'organisation du système repose sur l'importance relative attribuée à chacun des facteurs régulant l'auto-organisation. C'est ce que nous étudierons cette semaine.

2 Programme

2.1 Classes

Le coeur du programme de cette semaine sera la classe `Fourmi`, à l'aide de laquelle nous allons implémenter le comportement de nos fourmis virtuelles. Voici les spécifications de cette classe :

Attributs :

- $position$, un array contenant la position de la fourmi
- $positionNid$, un array contenant la position du nid
- $mode$, un entier déterminant l'état de la fourmi. Si $mode == 0$, la fourmi est à la recherche de nourriture, si $mode == 1$ la fourmi transporte de la nourriture

Méthodes :

- $\_\_init()\_\_$, une méthode qui initialise les attributs de la fourmi
- $bouge()$, une méthode qui détermine la nouvelle position de la fourmi. Nous allons supposer qu'une fourmi à la recherche de nourriture a un probabilité proportionnelle à $(n_{i,j} + s_0)^{b_0}$ d'aller vers la case i, j ($n_{i,j}$ est la concentration de phéromones dans cette case). Pour sa part, une fourmi qui transporte de la nourriture a une probabilité proportionnelle à $1/(d_{i,j} + s_1)^{b_1}$ d'aller vers la case i, j. Ici, $d_{i,j}$ est la distance entre la case i, j et le nid. De plus, une fourmi ne devrait pas pouvoir s'aventurer sur les frontières du domaine, ni sortir de ce dernier.
- $modifieEnvironnement()$, une méthode qui détermine comment la fourmi affecte l'environnement. Les effets possibles sont : émission de phéromones, prise de nourriture, dépôt de nourriture au nid

Le programme fera aussi usage de la classe `Domaine` implémentée au dernier laboratoire. Cependant, nous allons lui apporter quelques modifications rendues nécessaires par la présence des fourmis. Ces modifications prennent la forme de trois nouvelles méthodes :

- $ajoutePheromones()$, une méthode qui ajoute une certaine quantité de phéromones à une certaine position
- $environnementLocal()$, une méthode qui retourne une matrice 3×3 contenant la concentration de phéromones autours d'un point donné. Si une position représentée par une case de la matrice est inaccessible aux fourmis, la valeur de cette case devrait être de -1.
- $contientNourriture()$, une méthode qui retourne $True$ si un site donné contient de la nourriture, $False$ sinon

2.2 Méthode principale

La méthode principale sera assez simple. Voici un résumé de ce qu'elle devrait accomplir :

1. Lire les variables nécessaires au clavier
2. Initialiser un objet de type `Domaine` qui représentera l'environnement des fourmis
3. Créer une liste de fourmis
4. Boucle principale :
   - Faire évoluer la concentration de phéromones
   - Faire bouger chaque fourmi
   - Faire interagir chaque fourmi avec l'environnement.

5. Afficher la performance de la colonie en unité de nourriture récoltée par fourmi par unité de temps.

Pour simplifier, nous allons considérer que la variable de type `Domaine` sera une variable globale. Il n'est donc pas nécessaire de la passer en argument aux différentes classes et méthodes. Aussi, nous allons considérer que les sources de nourritures sont inépuisables.

2.3 Paramètres et unités

Comme la semaine dernière, notre unité de temps sera la seconde et notre unité de longueur le centimètre. Nous utiliserons une constante de diffusion de $0.02cm^2/s$ et une constante d'évaporation de $0.01s^{-1}$. Une fourmi augmente la concentration locale de phéromones de 1 millions de molécules par cm^2 par s. Si la concentration locale excède 10 millions de molécules par cm^2, la fourmi n'en ajoutera pas d'autre. Le pas de temps de l'intégration de l'équation de diffusion sera de $1s$. La taille du domaine sera de $25cm \times 20cm$ et le nid de la colonie sera placé en $(5cm, 5cm)$. Nous supposons que les fourmis prennent $1s$ pour se déplacer à un point (x_i, y_j) voisin de sa position courante. Nous placerons une source de nourriture dans les cases telle que $20cm \leq x_i \leq 24cm$ et $8cm \leq y_j \leq 12cm$. La colonie contient 40 fourmis.

3 Problèmes

3.1 Vérification du programme

Une fois la création de votre programme terminée, comment pouvez-vous vous assurer que le programme fonctionne correctement ? Faites ces tests et présentez les résultats. Vous n'avez pas à tester votre classe `Domaine` puisque vous l'avez fait la semaine dernière.

3.2 Rétroaction positive et importance des fluctuations

Étudions d'abord l'importance de la rétroaction positive sur l'efficacité de la recherche de nourriture. Un des paramètres contrôlant la rétroaction positive lors du processus de recherche de nourriture est le paramètre b_0, l'exposant déterminant à quel point le trajet d'une fourmi est biaisé par les phéromones. Plus cet exposant est grand, plus le trajet de la fourmi sera biaisé. Plus cet exposant est petit, plus le trajet de la fourmi sera aléatoire. L'exposant b_0 contrôle donc à la fois la rétroaction positive et l'importance des fluctuations aléatoires. En posant $s_0 = 1, s_1 = 0.01$ et $b_1 = 8$, variez b_0 entre 0 et 14. Qu'observez-vous ? Discuter le résultat dans la limite b_0 petit et b_0 grand. Quelle est la valeur optimale de b_0 ? À quoi correspond ce maximum ? Imaginez une autre situation où les fluctuations aléatoires seraient importante à la bonne performance de la colonie.

3.3 Rétroaction négative

Nous allons maintenant étudier l'effet de la rétroaction négative sur l'efficacité de la recherche de nourriture. Pour ce faire, nous allons modifier la concentration maximale de phéromones tolérée par les fourmis. Si cette valeur est basse, le sentier de phéromones ne pourra devenir très intense. On se trouve alors dans un régime de forte rétroaction négative. Si ce maximum est grand, le sentier

sera éventuellement très intense et on se trouve dans un régime de faible rétroaction négative. Utilisez des valeurs de maximum entre 0 et 20. Qu'observez-vous ? Est-ce que vous localisez une valeur optimale pour le maximum de phéromones ? Décrivez une situation où une faible rétroaction négative serait un désavantage pour la colonie.

3.4 Adaptabilité

Dans la nature, les sources de nourriture ne sont pas inépuisables et l'environnement est dynamique. La colonie doit donc être en mesure de s'adapter au changement. Nous tenterons ici de mesurer la capacité d'adaptation de notre colonie. Pour ce faire ajoutez, au temps $t = 2000s$, une nouvelle source nourriture à la position $3cm \leq x_i \leq 7cm$ et $17cm \leq y_j \leq 19cm$. Combien de temps est nécessaire avant qu'une fourmi trouve cette nouvelle source plus avantageuse ? Qu'est-ce que ce résultat indique à propos du choix des paramètres ? Étaient-ils réellement optimaux ? Comment améliorer votre choix de paramètres ?

4 Rapport

N'oubliez pas d'indiquer la valeur des paramètres utilisés pour chaque simulation présentée et de joindre les données (sous forme graphique ou numérique, dépendamment des cas), le cas échéant, lorsque vous répondrez aux questions.
N'oubliez pas non plus d'inclure vos programmes commentés en indiquant les parties modifiées.
Le rapport devra avoir la structure suivante :

1. **Introduction** (0.5 pt)— Courte description du problème et des objectifs ;
2. **Méthodologie** (1.5 pts)— Description de la méthode de simulation utilisée et du programme ;
3. **Vérification du programme** (4 pts) — Comment vérifiez-vous le bon fonctionnement du programme, de l'algorithme ? Décrivez les tests faits pour vérifier le bon fonctionnement du programme (inclure les sorties de programme) ;
4. **Problèmes** (13 pts) — Répondre à chaque problème séparément en incluant les sorties et graphiques, s'il y a lieu.
5. **Conclusion** (1 pt) – Qu'avez-vous appris ? Quelles autres questions pourrait-on poursuivre ?

+2 pts pour le travail en classe.