

# Projet Complexité

## Master MIAHS 2016-2017

### Colonie de fourmis

#### **1 – Problème n°1 : Choisir le bon trajet entre 2 branches.**

Dans le premier exercice, nous devons créer un simulateur, où des fourmis doivent trouver le plus court des chemins, entre deux chemins pour accéder à la source de nourriture.

Pour cette simulation, nous utilisons des tableaux de différentes longueur pour simuler les chemins. A chaque tour de la simulation les fourmis se déplacent sur les chemins, et donc dans le tableau. A chaque déplacement la fourmi dépose de la phéromone sur le chemin. Pour représenter ce dépôt dans notre simulation, on incrémente un compteur de phéromone pour chaque cellule du tableau (qui représente une portion du chemin). Cette valeur servira par la suite aux fourmis suivantes à choisir quel chemin emprunté.

Au cours de cet exercice, j'ai expérimenté 2 versions différentes de ce simulateur :

##### **1.A) Choix du phéromone max**

Dans cette première version du simulateur, les fourmis devaient choisir de se diriger vers le chemin présentant le plus haut taux de phéromone. Lors du premier tour, lorsque les valeurs de phéromones sont nulles pour les deux chemins, les fourmis choisissent aléatoirement leurs directions entre les deux chemins.

Une fois que de la phéromone a été déposé sur les chemins, la fourmi choisit le chemin avec la valeur de phéromone la plus haute (ou aléatoirement si les valeurs sont égales). A chaque fin de tour, la phéromone s'évapore. La valeur de la phéromone de chaque cellule est soustraite par un entier passer en paramètre de la simulation.

Après avoir effectué un nombre important de simulation avec ce système on observe les résultats suivant :

- En cas de chemins de longueurs égales, peu importe le taux d'évaporation mis en place, le chemin ayant reçu plus de fourmis lors du tirage initial est favorisé, et le second chemin est oublié dès le second tour.
- En cas de chemins de longueurs différentes, la simulation reste toujours très liée au tirage initial. Si le chemin le plus long reçoit plus de fourmis lors du premier tour, il est très compliqué de compenser ce tirage, malgré que les fourmis empruntant le chemin le plus court reviennent plus vite au nid que les fourmis parcourant le chemin le plus long.

En conclusion avec ce choix de la phéromone maximum, on ne trouve la bonne solution que si le chemin le plus court est beaucoup plus court que le chemin le plus long, et que le tirage initial est proche de 50/50 entre les deux chemins. Cet algorithme ne semble pas très approprié pour une résolution efficace du problème.

### **1.B) Choix pondéré**

Dans cette seconde version de la simulation, juste l'algorithme de choix du chemin diffère. Comme précédemment, si la valeur de la phéromone à l'entrée des chemins est nulle pour les deux chemins, ou identique, la fourmi choisie aléatoirement. Si ce n'est pas le cas, désormais, elle effectuera toujours un choix aléatoire, mais cette fois, le chemin ayant une valeur de phéromone plus haute recevra une plus haute probabilité d'être choisie par une fourmi.

De par ces modifications, le problème est moins soumis au premier tirage puisqu'un chemin n'ayant pas reçu une bonne distribution des fourmis ne sera pas complètement ignoré. Enfin une bonne répartition des valeurs : nombre de fourmis injecté à chaque tour et taux d'évaporation permette dans la totalité des configurations possibles de chemins de trouver une convergence vers un chemin.

Le désavantage de cette méthode réside dans les chemins de longueur égales, où de par le caractère aléatoire du choix, il est parfois long d'obtenir une réelle convergence vers un chemin. En effet, vu qu'il n'existe pas un chemin plus favorable que l'autre. Seul le hasard des tirages permettra de favoriser à terme un seul des deux chemins. Cependant, si l'on doit se référer à la réalité, le fait de ne pas pouvoir optimiser réellement un trajet entre deux chemins de longueur égale ne doit pas poser de réels problèmes pour les fourmis.

## **2 – Problème n°2 : Le problème du voyageur.**

Pour cet exercice, nous devons implémenter un simulateur pour la résolution du problème du voyageur. Toujours à l'aide de colonies de fourmis, il faut trouver le plus court chemin dans un réseau de villes reliées entre elles. Ce chemin doit relier toutes les villes du problème et tenir compte du coût de chaque voyage.

Pour notre simulateur, la position et les distances reliant les villes sont tirées aléatoirement, et les informations telles que le coût d'un trajet, et la quantité de phéromone sur chaque trajet entre deux villes sont stockées dans des matrices.

Fort des expérimentations réalisées pour la première partie, j'ai adopté le même système de choix de trajet que pour le 1.B mais adapté au nouveau problème. C'est-à-dire que l'on effectue un tirage aléatoire pondéré par la valeur des phéromones des trajets menant à des villes que l'on n'a pas encore parcourues.

Une fois qu'une fourmi a parcouru l'ensemble des villes, elle revient à la ville de départ, et on observe si son trajet est meilleur que le dernier que l'on a stocké. Le meilleur trajet, est celui dont le coût total (la somme des coûts unitaires entre chaque ville visitée) est le plus faible. Dans notre problème il est assez simple de voir si l'on a trouvé la bonne solution. Les poids entre chaque ville sont calculés sur la distance entre chaque ville. Le chemin le plus court réside dans « le contour » du graph.

Dans une première itération du simulateur, j'ai utilisé exactement les mêmes paramètres que pour le premier exercice. C'est-à-dire que chaque fourmi dépose une phéromone par passage, et que à chaque tour le chemin s'évapore d'une valeur entière passée en paramètre de la simulation.

Rapidement on observe que peu importe les valeurs prises sur le nombre de fourmis injectées dans le système chaque tour, et le taux d'évaporation fixé, les résultats semblent très aléatoires. En effet avec cette démarche, on ne prend pas en compte en réalité la longueur du chemin dans le mécanisme de la phéromone. Et c'est une erreur ! Dans le premier exercice, la longueur du chemin était représentée par les cellules que devait parcourir la fourmi pour accéder à la source de nourriture. Et c'est la différence de temps de parcours de la fourmi qui faisait converger vers un résultat. Il me fallait donc inclure un rapport de longueur dans mon nouveau problème.

J'ai donc par la suite modifié la façon dont les fourmis déposaient la phéromone. Ainsi à chaque passage d'une fourmi, la phéromone était recalculée en fonction d'un taux d'évaporation en pourcentage passé en paramètre, et incrémenter par une valeur calculée comme le rapport de la somme totale des poids du système divisé par la somme des coûts de trajet de la fourmi. Ainsi, plus le trajet effectué par la fourmi pour accéder à ce point est court, plus la valeur de phéromone déposée sera élevée.

Avec cette version du simulateur on obtient de très bons résultats sur la simulation. On observe que les couples Nombre de fourmis injecté, et taux d'évaporation sont importants pour une convergence rapide des résultats. Un haut taux de fourmis est préférable pour trouver la bonne solution en un minimum de tour, cependant, cela augmente fortement le temps de calcul du simulateur. Cependant, avec moins de fourmis, des résultats semblables peuvent être obtenus. Il suffit d'affiner le taux d'évaporation en fonction du nombre de villes et d'effectuer un peu plus d'itération et on obtient le résultat dans des temps corrects.

On observe que changer les paramètres en cours de simulation peut-être intéressant. En effet il est souvent intéressant d'utiliser plus de fourmis lors de la première itération (Exploration) pour être sûr de découvrir tous les chemins. Et ensuite de réduire le nombre de fourmis injectées

dans le problème tout en augmentant le taux d'évaporation pour effectuer une convergence plus rapide, et avec de meilleures performances.

Pour conclure, avec ce simulateur, et les choix réalisés dans les algorithmes, il est très rare de ne pas trouver de résultats, il faudra cependant plus ou moins de temps en fonction des paramètres passé en arguments. Enfin on peut mettre en défaut le choix d'une technologie Web-Php pour le temps d'exécution. Le langage est certes pratique à utiliser, mais les performances des serveur locaux sur les machines individuelles peuvent parfois laisser à désirer.

## **Conclusion**

Malgré un temps de calcul important que peu engendré la résolution du problème du voyageur sur un grand nombre de ville dans notre simulateur, on observe que les algorithmes à bases de colonies de fourmis sont très efficaces pour la résolution de ce genre de problème combinatoire. Néanmoins, la qualité des solutions est très dépendante des paramètres passés au simulateur, et des paramètres valables pour une simulations, ne sont pas forcément les paramètres optimaux pour une autre conformation du problème (plus de villes par exemples). Ainsi il faudra apprendre pour chaque problème à étudié à trouver les bons paramètres pour optimiser nos résultats. On pourra imaginer sans mal, que si un déploiement sur plus grande échelle devrais s'effectuer, il pourras être utile d'accompagner le simulateur d'un réseau de neurones, pour optimiser et trouvez tous ces paramètres optimaux.