

# Rapport de projet

**Théo Q, Corto C, Brandon FL**  
Université de Nice  
France

## Abstract

Dans ce projet nous étudions l'impact de la proportion de fourmis chercheuse et ramasseuses dans une fourmilière sur la récolte de nourriture. Nous utilisons pour cela un algorithme de type "Ant Colony Optimization (ACO)" écrit en Netlogo. Notre modèle utilise deux types de fourmis : ramasseuse et chercheuses déposant chacune des phéromones spécifiques et des tas de nourritures à éloignement du nid variable. En faisant varier la répartition des fourmis dans la fourmilière lors de multiples simulations nous avons produits des données expérimentales. Celles-ci ont été étudiées avec le logiciel R pour extraire un modèle mathématique. Finalement l'analyse de la fonction modélisant le temps de récupération de toutes la nourriture en fonction du pourcentage de chercheuse dans la population présente un minimum local en 67,18% ce qui s'approche des observations faites dans la nature.

## 1 Introduction

### 1.1 Contexte

Les "Algorithmes de colonies de fourmis" forment une catégorie d'algorithmes d'optimisation basés sur la modélisation d'une colonie de fourmis. En effet bien qu'ayant des capacités individuelles limitées ces insectes arrivent collectivement à des solutions optimales de problèmes complexes.

Dans le cadre de l'UE "Projet Scientifique Informatique" nous avons choisi le sujet "Colonie de fourmis" d'étude parmi de nombreux proposés. Le modèle de base contenait un seul type de fourmis et trois tas de nourritures fixes. Nous l'avons alors perfectionné.

### 1.2 Etat de l'art

L'ACO est de plus en plus utilisé aujourd'hui dans de très nombreux domaines. Plus particulièrement dans les domaines suivants : planification, routage réseau, affectation, ensembles, conception de circuit nanoélectroniques, traitement d'image etc. Tout les problèmes pouvant se ramener à la recherche d'un chemin optimal dans un graphe en général.

## 1.3 Questions scientifiques

## 2 Modélisation

Pour notre projet nous avons réalisé une simulation via l'outil NetLogo. NetLogo étant un programme de modélisation d'environnement multi-agent. Nous nous sommes appuyé sur un modèle de base déjà existant proposé par le MIT dans le cadre d'un projet de Mathématiques.

### 2.1 Hypothèses simplificatrices

Dans la nature, les fourmis suivent plusieurs critères pour retrouver leur chemin vers le nid. Elles s'orientent notamment en gardant la trace de la direction, la distance parcourue ainsi que des repères visuels pour retrouver le chemin vers leurs nids. Dans notre modélisation, nous prendrons donc en compte uniquement les phéromones pour retourner au nid. De plus, les fourmis qui ne suivent pas de phéromones suivront un mouvement totalement aléatoire jusqu'à trouver des phéromones.

### 2.2 Description du modèle

Dans un premier temps nous avons créé un nouveau type de fourmis, les fourmis chercheuses. Dans la version de base, le modèle présentait qu'un type de fourmis ne reflétant pas la réalité. Or, dans notre nouveau modèle, les fourmis ramasseuses se concentrent maintenant exclusivement sur la collecte de la nourriture pendant que les fourmis chercheuses posent les phéromones. Ce modèle est beaucoup plus proche de la réalité car maintenant, les fourmis se concentrent vraiment sur un tas à la fois en prenant le tas le plus proche.

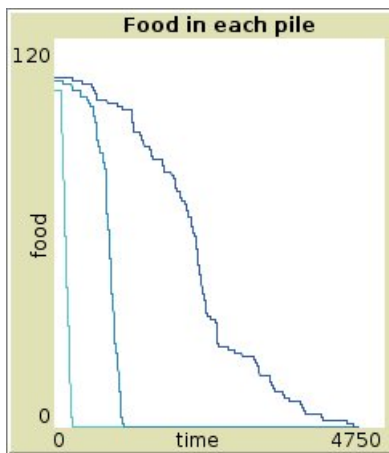


Figure 1: Proportion de nourriture en fonction du temps

De plus, notre modèle comprend 4 variables modifiables dont 2 pour la modification de la population. En effet, la variable "Population" nous permet de modifier le nombre de fourmis au total qui sont présentes dans notre fourmilière pendant que la variable "pourcentage" calcul le nombre de fourmis chercheuses en fonction de la population totale (voir les explications de ce choix ici 3.1). De plus, nous sommes dans la capacité de modifier en direct le pourcentage de diffusion ainsi que le pourcentage d'évaporation des phéromones.

### 3 Simulation

#### 3.1 Cadre expérimental

Toutes la partie simulation de ce projet à été effectué avec le logiciel Netlogo 5.1.0 en utilisant le modèle de base fourni. (Wilensky, U. (1997). NetLogo Ants model. <http://ccl.northwestern.edu/netlogo/models/Ants>. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.)

Voici les paramètres utilisés :

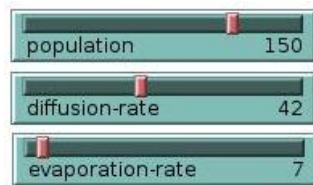


Figure 2: Configuration de la simulation

La population à été déterminée de façon arbitraire, mais telle qu'elle ne soit ni trop basse pour être significative, ni trop haute afin de garder les ressources nécessaires à la modélisation acceptables. Après plusieurs essais, nous avons choisi de garder les paramètres "diffusion-rate" et "evaporation-rate" tel que dans le modèle car ils fournissent une simulation assez plausible : trop de diffusion réduit l'efficacité car la piste laissée est trop large, tandis que trop d'évaporation annule la diffusion. A l'opposé, aucune

évaporation est contre productif car l'espace devient saturé de phéromones ce qui revient au même que de ne pas en avoir.

L'univers est un canvas de dimension 81\*81 aux bords fermés contenant 1 nid et 3 sources de nourritures de (approximation de cercle de rayon 5) placés spécifiquement.

L'étude des données et leur présentation ont été effectuées avec les outils suivants :

1. Python (script) pour le formatage des fichiers .csv
2. Gnuplot pour la représentation des données.
3. R pour l'interprétation mathématique ...
4. L<sup>A</sup>T<sub>E</sub>X pour la présentation et le rapport

Toutes les simulations et la majeure partie du projet ont été réalisées sous Ubuntu sur les machines de la Faculté des Sciences de Nice.

#### 3.2 Protocole expérimental

Pour produire les données permettant de bâtir un modèle nous avons utilisé l'outil BehaviorSpace intégré à Netlogo. Celui-ci permet d'exécuter de manière automatique un nombre déterminé de simulations en faisant varier automatiquement les paramètres voulus.

Afin d'obtenir des mesures les plus objectives possibles nous avons procédé ainsi :

- Faire varier la variable "pourcentage" de chercheuse dans la population dans [1;90] par pas de 5, soit 18 runs.
- Répéter l'expérience précédente pour des sources de nourritures placées à distances proches, moyenne, et éloignées du nid. Soit  $3 \times 18 = 54$  simulations retenues.
- Écrire le nombre de ticks nécessaire à la complétion de la simulation à chaque étape
- Produire une moyenne des 3 fichiers résultats avec python.
- Trouver l'équation approximant le mieux nos résultats par régression linéaire avec R

Nous précisons aussi que les simulations ont été faites consécutivement et non en parallèle dans la configuration de l'outil BehaviorSpace.

### 4 Résultats

### 5 Discussion

### Conclusion et perspectives