

# Ant colony algorithm

**Description** (le texte en *italique* est copié du lien suivant : [https://fr.wikipedia.org/wiki/Algorithme\\_de\\_colonies\\_de\\_fourmis](https://fr.wikipedia.org/wiki/Algorithme_de_colonies_de_fourmis))

*Ant colony Algorithm est un algorithme d'optimisation appelé ACO initialement proposé par Marco Dorigo en 1990. Cet algorithme est inspiré du comportement des fourmis et dont l'objectif est de trouver des chemins optimaux d'une source (nid) à une destination (nourriture) au sein d'un graphe.*

*L'idée originale provient de l'observation de l'exploitation des ressources alimentaires chez les fourmis. En effet, celles-ci, bien qu'ayant individuellement des capacités cognitives limitées, sont capables collectivement de trouver le chemin le plus court entre une source de nourriture et leur nid.*

*Des biologistes ont ainsi observé, dans une série d'expériences menées à partir de 1989, qu'une colonie de fourmis ayant le choix entre deux chemins d'inégale longueur menant à une source de nourriture avait tendance à utiliser le chemin le plus court.*

## Travail demandé :

Créer une application en C qui va simuler le comportement d'une colonie de fourmis.

Cette application peut se faire en trois étapes principales :

### Etape 1 : créer l'environnement

L'environnement se compose d'un ensemble de points (x,y), une source (nid), une destination (nourriture) et deux fourmis F1 (triangle bleu) et F2 (triangle vert) qui se trouvent initialement dans le nid.

Le nid se trouve au point 31 ( $x=3, y=1$ ) et la nourriture se trouve au point 45

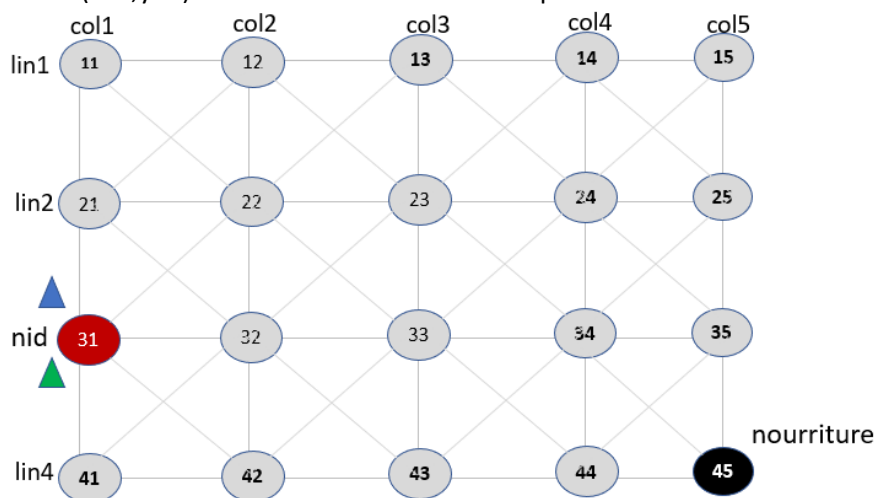


Figure 1

**Question 1** : après définition des structures de données qu'il faut, créer les éléments ci-dessus.

Dans la suite, on considère si une fourmi se trouve à une position elle peut se déplacer dans huit directions.

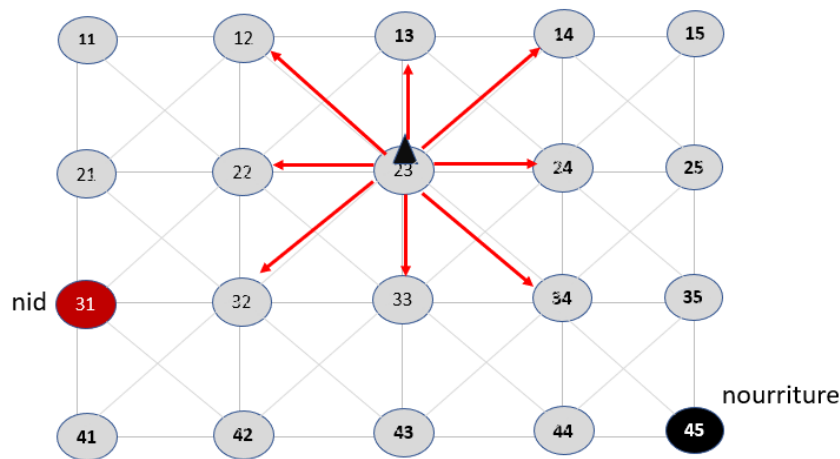


Figure 2

Par exemple, si la fourmi se trouve dans le nœud 23, elle peut aller soit vers 13, ou bien 14, ou bien 24, ou bien 34, ou bien 33, ou bien 32, ou bien 12.

### Etape 2 : tracer deux chemins de votre choix

Entre le point de départ (nid) et le point d'arrivée (nourriture), on peut avoir plusieurs chemins. Dans un premier temps, pour simplifier et comme montré dans la figure ci-dessous, nous allons définir uniquement deux chemins celui en vert et celui en bleu qui mènent du nid à la nourriture puis revenir de la nourriture au nid.

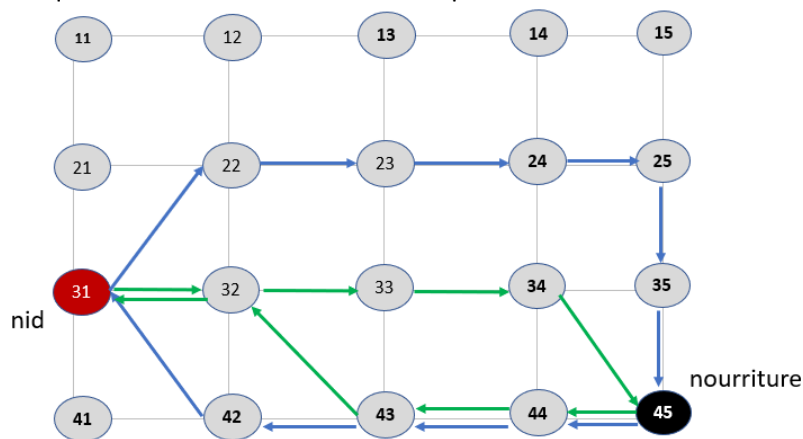


Figure 3

**Question 2 :** après modification des structures de données, si nécessaire, initialiser les deux chemins comme montré dans la figure ci-dessus.

### Etape 3 : emprunter un chemin et mettre à jour la quantité des phéromones sur des arcs

L'objectif principal est de trouver un chemin optimal parmi plusieurs entre deux points (nid et nourriture). Pour simplifier la réalité, dans le schéma ci-dessous, la fourmi F1 (triangle en bleu) a choisi d'emprunter le chemin en bleu tandis que la deuxième fourmi a choisi d'emprunter le chemin en vert.

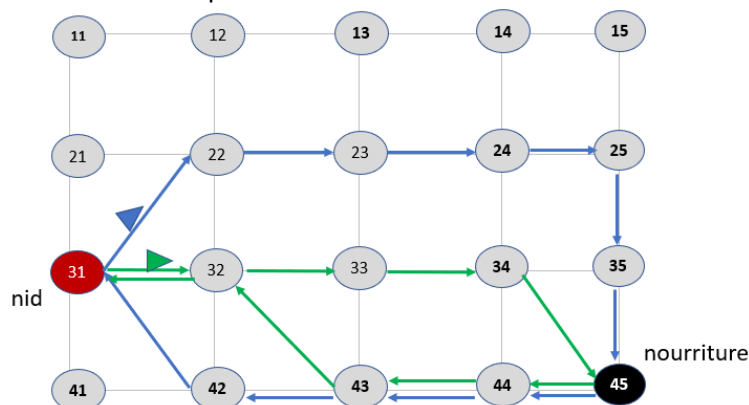


Figure 4

D'après la figure, on constate que le chemin en vert est plus court que celui en bleu.

L'objectif de l'algorithme de colonie de fourmis est de proposer le chemin le plus court pour une nouvelle fourmi.

Dans la suite, nous allons montrer comment, en se basant sur un travail collectif des fourmis, les bout de chemins sont mis à jour jusqu'à identifier le chemin optimal.

Pour répondre à cet objectif, on peut considérer les éléments suivants

Un modèle pour trouver ce chemin optimal et qui explique le comportement collectif des fourmis est le suivant :

1. une **fourmi** (appelée « éclaireuse ») **parcourt** plus ou moins au hasard l'environnement autour de la colonie ;
2. si celle-ci découvre une **source de nourriture**, elle rentre plus ou moins directement au nid, en laissant sur son **chemin** une piste de **phéromones**; (voir figure ci-dessous)
3. ces **phéromones** étant attractives, les fourmis passant à proximité vont **avoir tendance à suivre**, de façon plus ou moins directe, cette piste ;
4. en revenant au nid, ces mêmes fourmis vont **renforcer la piste** ;
5. si deux pistes sont possibles pour atteindre la même source de nourriture, celle étant **la plus courte** sera, dans le même temps, parcourue par plus de fourmis que la **longue piste** ;
6. la **piste courte** sera donc de **plus en plus renforcée**, et donc de plus en plus **attractive** ;
7. la **longue piste**, elle, finira par **disparaître**, les **phéromones étant volatiles** ;
8. à terme, l'ensemble des fourmis a donc déterminé et « choisi » **la piste la plus courte**.

**Question 3** : après modification des structures de données, si nécessaire, ajouter le traitement à votre application qui permet à une fourmi de suivre un chemin

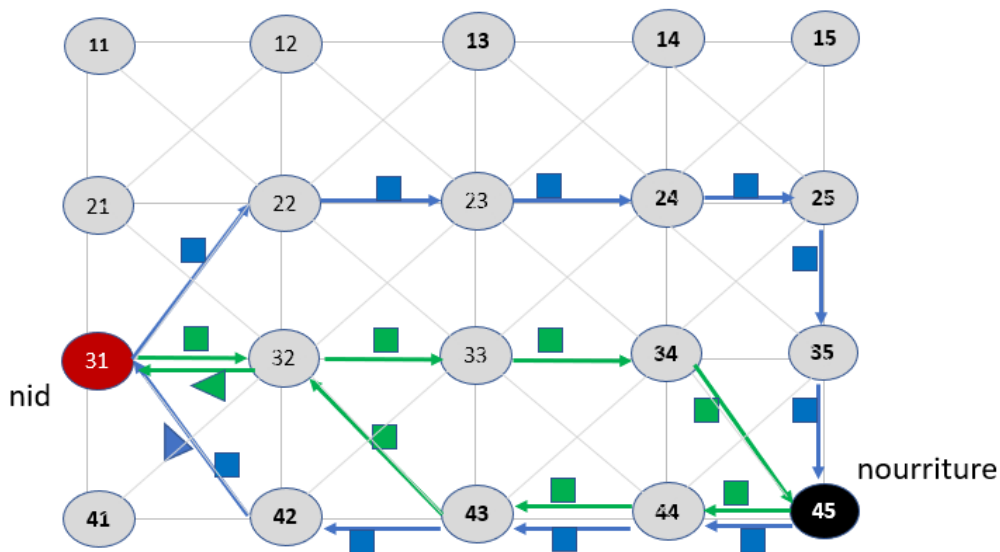


Figure 5

**Quel est la quantité de phéromones déposée par chaque fourmi ?**

On considère qu'une fourmi dépose une quantité de phéromones  $\Delta\tau_{i,j}^k = \frac{1}{L_k}$  (Formule 1) sur chaque arc entre un nœud i et un nœud j.

Où  $L_k$  est la longueur du chemin emprunté par la fourmi k.

On considère que chaque arc a un coût égal à 1

Sur le graphe la quantité de phéromone est représentée par le rectangle bleu et le rectangle vert

Par exemple, la longueur du chemin emprunté par la fourmi verte est  $L_1 = 8$

Tandis que la longueur du chemin emprunté par la fourmi en bleu est  $L_2 = 10$

Donc, la fourmi bleue a déposé sur chaque arc une quantité égale à  $\frac{1}{10}$

Et la fourmi verte a déposé sur chaque arc une quantité égale à  $\frac{1}{8}$

**Question 4 :** après modification des structures de données si nécessaire, ajouter le traitement pour que à chaque déplacement entre deux points, une fourmi  $k$  dépose une quantité de phéromone sur chaque arc traversé (voir figure ci-dessus) et ainsi augmente sa quantité par  $\frac{1}{L_k}$ . Les quantités de phéromones sur les arcs avant qu'ils soient consulté par une fourmi sont initialisées à 0.

Les quantités de phéromones  $\tau_{i,j}$  sont cumulables dans chaque arc  $(i,j)$  selon la formule suivante :

$\tau_{i,j} = \sum_{k=1}^m \Delta\tau_{i,j}^k$  (Formule 2), à chaque passage d'une fourmi  $k$  sur un arc  $(i,j)$  on augmente sa quantité de phéromones par  $\Delta\tau_{i,j}^k$

Dans la réalité, les phéromones qui sont des substances chimiques subissent à la vaporisation selon la formule suivante :

$$\tau_{i,j}(t) = (1 - p)\tau_{i,j}^k(t-1) + \sum_{k=1}^m \Delta\tau_{i,j}^k(t) \quad (\text{Formule 3})$$

$\tau_{i,j}(t)$  quantité de phéromones sur l'arc  $(i,j)$  à l'instant  $t$

$\tau_{i,j}(t-1)$  quantité de phéromones sur l'arc  $(i,j)$  à l'instant  $t-1$

$\sum_{k=1}^m \Delta\tau_{i,j}^k(t)$  les quantités de phéromones déposées sur l'arc  $(i,j)$  par les fourmis  $1...k$  à l'instant  $t$

$p$  est un facteur de vaporisation compris entre 0 et 1 pour la régulation de la vaporisation

**Question 5 :** ajouter le traitement pour la mise à jour des quantités de phéromones sur les arcs selon la formule 3.

### Etape 3 : comment une nouvelle fourmi peut choisir le chemin optimal ?

Supposons qu'une nouvelle fourmi (triangle jaune) sort de son nid(31) et elle doit décider est ce qu'elle va aller vers le nœud 22 ou vers le nœud 32. En général, une fourmi a tendance de choisir l'arc ayant la quantité de phéromone la plus importante.

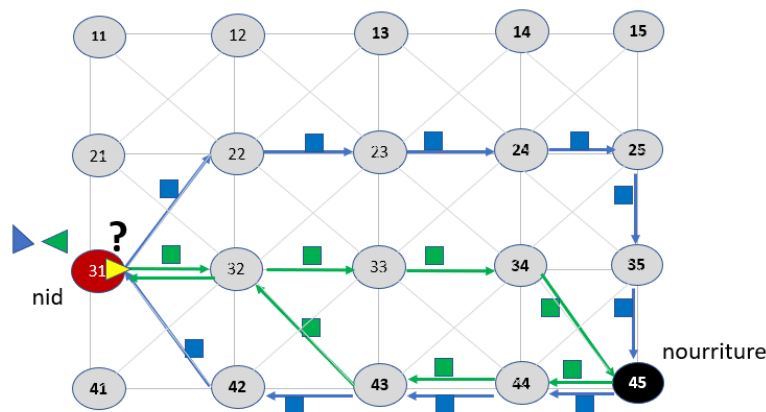


Figure 6

**Question 6 :** ajouter le traitement pour permettre à une nouvelle fourmi d'emprunter le plus court chemin pour aller à la nourriture et revenir vers le nid.

**Question 7 :**

Mettre en place un scénario d'exécution qui simule le comportement des trois fourmis

**Question 8 :** donner la possibilité à plusieurs fourmis de se déplacer dans votre graphe

**Question 9 :** changer le scénario de manière à assurer les étapes suivantes :

- initialiser le graphe (figure 1)
- sans mentionner aucun chemin initial, donner la possibilité à des fourmis (taille  $m$ ) de se déplacer librement tout en choisissant le chemin optimal
- afficher le chemin optimal identifié à la fin de déplacements possibles pour l'ensemble des fourmis

**Question 10** : interface graphique

En utilisant SDL2 (ou autre bibliothèque), réaliser une démonstration pour montrer le fonctionnement de Ant Colony algorithm

**Références**

[https://fr.wikipedia.org/wiki/Algorithme\\_de\\_colonies\\_de\\_fourmis](https://fr.wikipedia.org/wiki/Algorithme_de_colonies_de_fourmis)