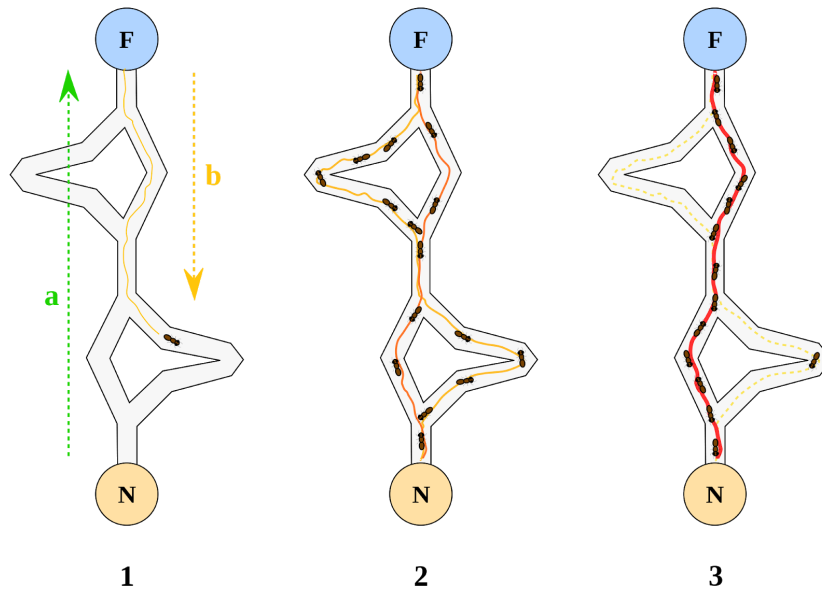


Algorithme Ant Colony Optimization

TP - Ynov Eats



BOUCHAIN Loïc

Bachelor 3 Informatique

Ynov Nantes 2019/2020

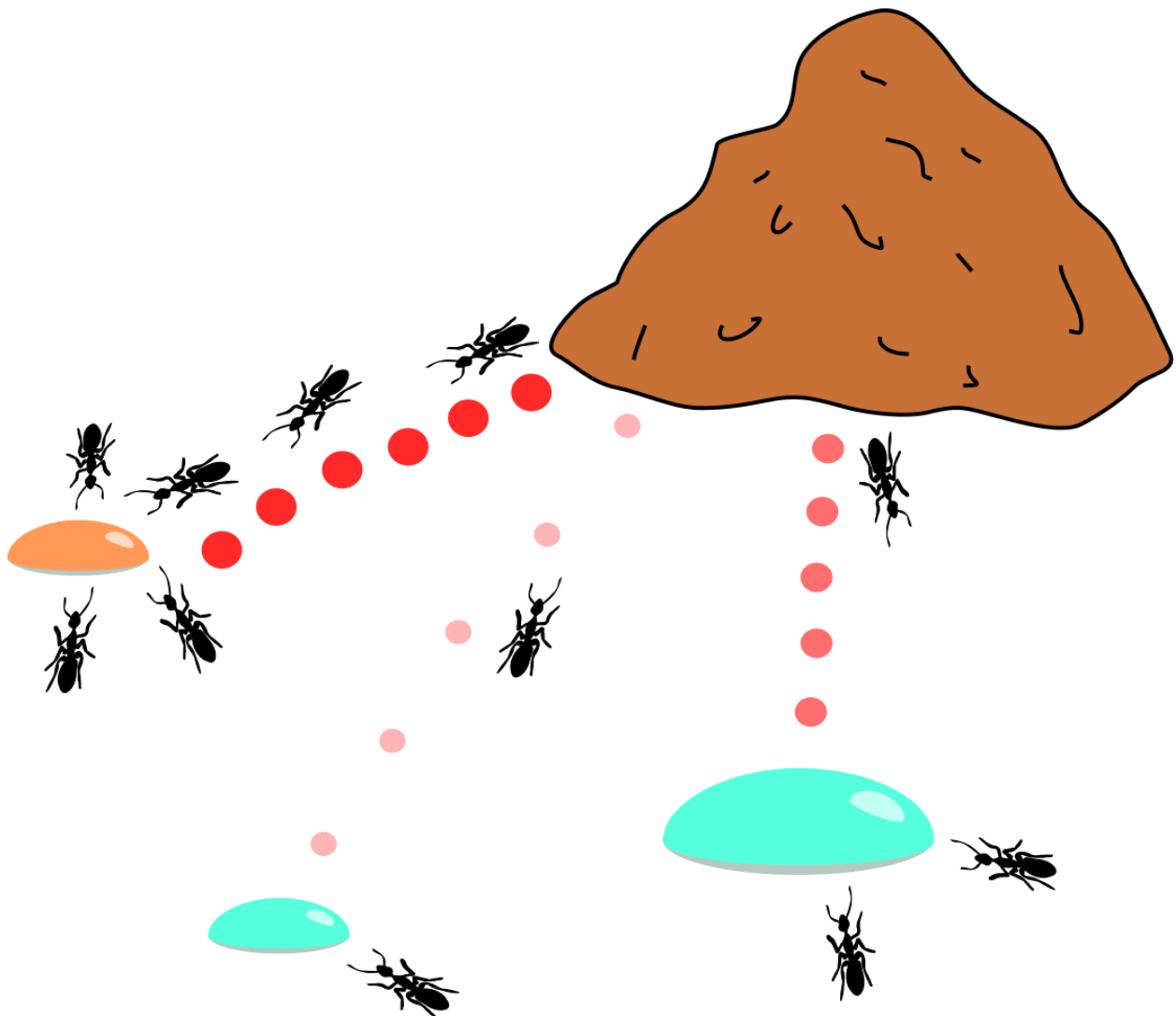


Introduction

1. Contexte
2. Fonctionnement du programme
3. Graphique
4. Autres

Le contexte

Ce sujet aura une carte générée random. Celle ci représentera une ville fictive. Le but est de monter une entreprise de livraison de repas chauds, Ynov Eat, en utilisant des robots qui suivront les itinéraires trouvés par vos algorithmes. Pour rentabiliser l'affaire, il va falloir minimiser les temps de trajet, les clients n'aiment pas manger froid et vous avez besoin que vos robots reviennent au plus vite.



Le fonctionnement

- L'objectif

L'objectif du programme est de déterminer le chemin le plus rapide d'un point A à un point B.

- Comment les routes sont construites ?

Le nombre de routes est définie dans le programme.

Chaque route, (Node 0 à Node 15 par exemple) dispose d'une longueur aléatoire dans une plage de données donnée (1 à 100 de base).

- Comment les phéromones sont déposés ?

Quand une fourmie part d'un point A et arrive au point B, elle doit déposer des phéromones sur son chemin pour avertir les autres fourmies que ce chemin permet d'arriver au point final.

Cependant la quantité de phéromones déposée doit varier en fonction de la longueur du chemin emprunté, en effet un chemin plus court doit être priorisé par rapport à un autre plus long du fait de l'évaporation des phéromones dans l'air et de notre besoins.

Ainsi la longueur du chemin final doit avoir une importance dans le calcul des phéromones déposées.

Le calcul est le suivant :

- On calcule la longueur total du chemin emprunté par la fourmi.
- Et pour chaque route empruntés (Exemple Node 0 à Node 15) on ajoute $1 /$ par la longueur total.

Grâce à cela la prochaine fourmi aura plus de chance de choisir ces routes plutôt que d'autres.



- Comment le programme fonctionne ?

Pour un nombre de fourmis donné, chaque fourmis a pour objectif de trouver le point d'arrivée. Pour cela, il va observer les routes voisines à son point en cours et va en choisir une par rapport à un choix aléatoire.

Ce choix n'est pas totalement aléatoire, en effet les routes voisines ayant des phéromones déposées auront plus de chances d'être choisies, de plus, si une route a déjà été empruntée auparavant ses chances s'amointrissent.

Ce procédé est répété tant que le point final n'a pas été trouvé.

Quand un chemin a été trouvé, pour chaque route emprunté dans ce chemin, on dépose des phéromones, le nombre de phéromones est définie en fonction de la longueur totale du chemin. Ainsi chaque route recevra $1/\text{longueur totale}$ de phéromones en plus.

Une phase d'évaluation du chemin commence, permettant de déterminer si c'est le chemin le plus court trouvé, si oui il est enregistré.

Une fois que toutes les fourmis ont fini de chercher le point final une phase d'évaporation des phéromones s'enclenche, chaque route perd 20% de ses phéromones du fait de l'évaporation dans l'air.

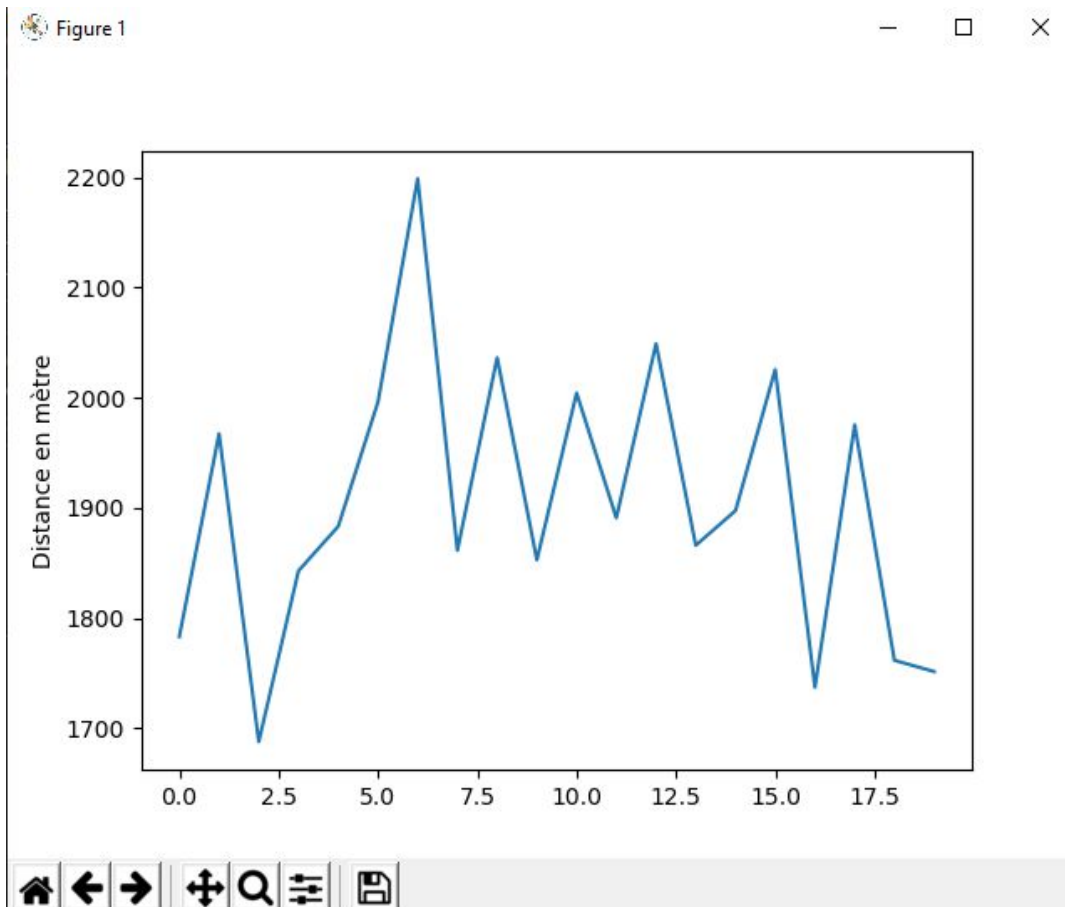
Ce procédé est répété un nombre de fois donné.

Le chemin le plus court est ensuite affiché.

Les graphiques

À noter : chaque lancement du programme le résultat sera différents car les données sont générées aléatoirement.

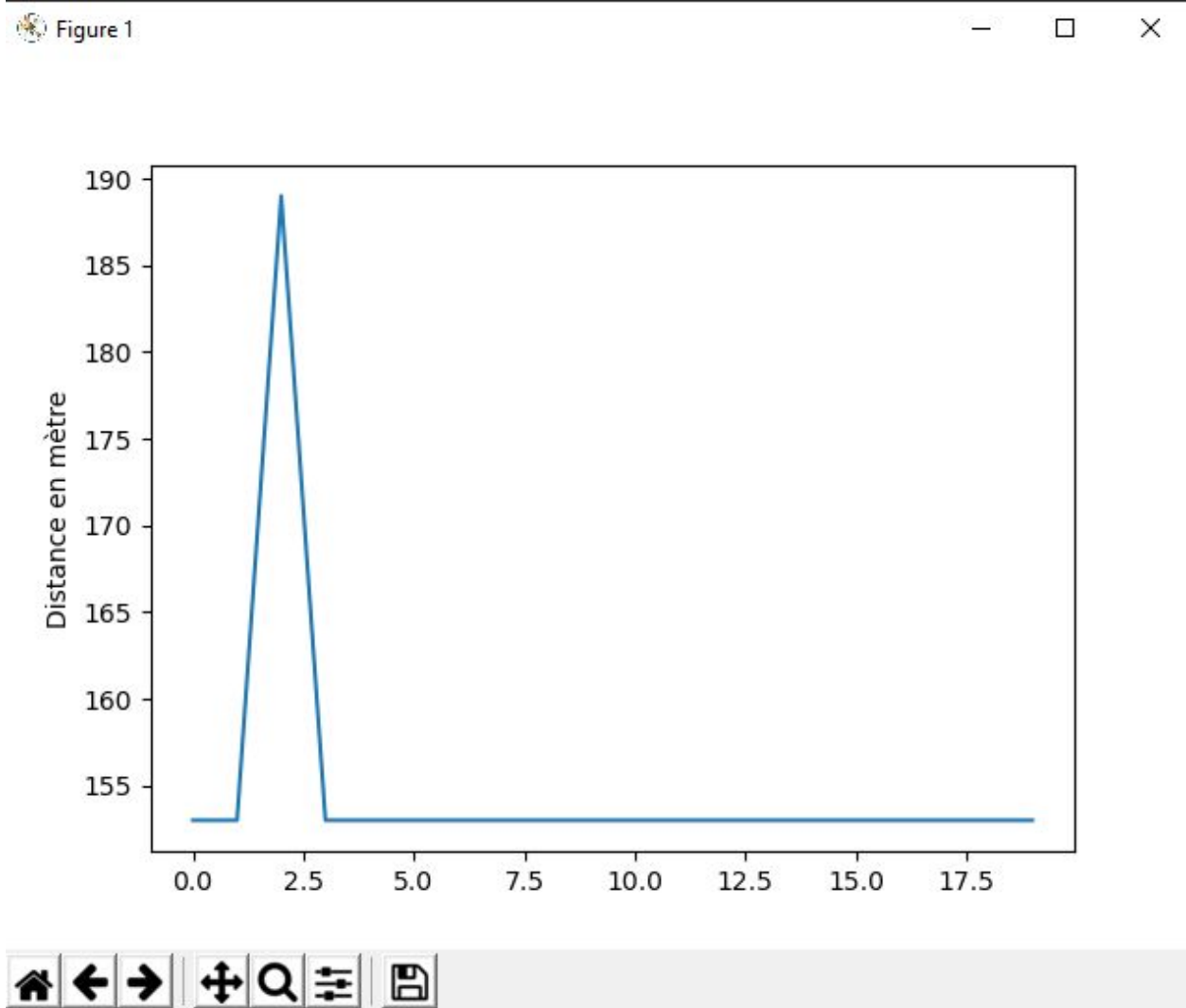
Moyenne des longueurs des chemins pour 20 générations et 100 routes.



On peut voir que la distance moyenne diminue en fonction des générations.

On peut donc imaginer qu'avec l'aide des phéromones les fourmis trouvent plus facilement leur chemin.

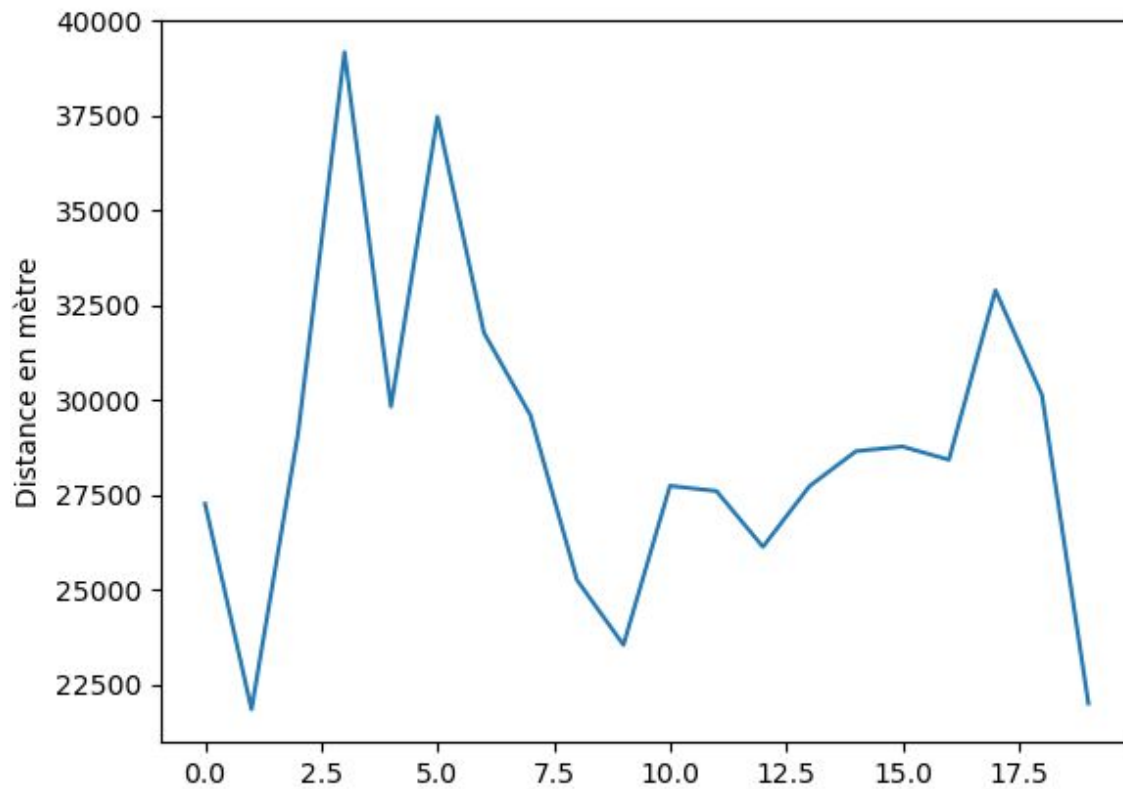
Chemin le plus court par génération



On peut voir ici que le chemin le plus court a été trouvé dès la 3ème génération.

Chemin le plus long par génération

Figure 1



On peut voir ici que les chemins les plus longs sont de plus en plus court en fonction des générations, voulant dire que les fourmis trouvent plus rapidement leur chemin en fonction des générations.



Divers

Lien gitlab : https://gitlab.com/loicbouchain/aco_tp.git

Tutoriel

Changer :

- nb_fourmi pour changer le nombre de fourmis,
- NB_NODE_MAX pour changer le nombre de routes
- NB_LONG_MAX pour changer la longueur maximum possible
- ville_fin pour changer le point final
- ville_depart pour changer le point de début
- nb_essai pour changer le nombre d'essai

Installation des libs sans pip en var global : `python -m pip install -r requirement.txt`

Execution du projet : `python aco.py`