

TP Algorithmes de fourmi

Fourmiam

Le principe de ce TP va être d'utiliser la programmation d'algorithmes de fourmi (ACO) pour trouver le meilleur itinéraire pour un robot de livraison. Le présent document a pour but de vous aider dans les étapes permettant la réalisation de ce TP.

La présentation graphique de la résolution (l'itinéraire) et des résultats est attendue.

Les éléments à rendre sont les suivants :

- Un rapport présentant cette méthode dans le contexte choisi, en PDF
- La structure de données utilisée
- Une explication complète de la fonction de fitness choisie
- Le code commenté devra être présent sur Gitlab
- Un ou des jeux de test

Pour chaque jeu de test, des données de performance sont attendues.

TP Algorithmes de fourmi

1. Contexte

Ce sujet possède une sensibilité Open Data. Vous travaillerez sur les vraies données de nos communes, mises à jour de façon trimestrielle par Nantes Métropole.

Toutes les données existent, entre autres, sous forme de fichier CSV dont la description vous donne tous les champs. Vous trouverez l'ensemble à l'adresse :

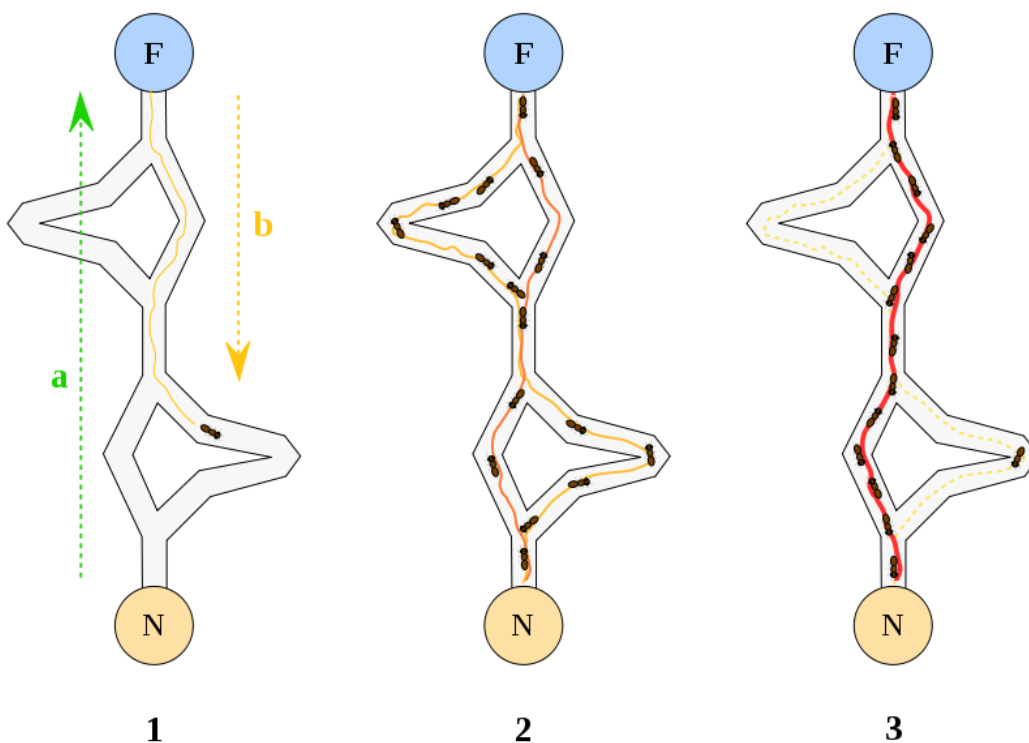
<http://data.nantes.fr/donnees/detail/repertoire-des-voies-de-nantes-metropole/>

Le but est de monter une entreprise de livraison de repas chauds, Fourmiam, dans Nantes Métropole 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.

2. Implémentation

Il vous est demandé d'utiliser l'algorithme des colonies de fourmis (ACO) pour optimiser cette tournée de l'Europe. Si on résume sur un dessin (merci wikipedia)

- 1) la première fourmi trouve la source de nourriture (F), via un chemin quelconque (a), puis revient au nid (N) en laissant derrière elle une piste de phéromone (b).
- 2) les fourmis empruntent indifféremment les quatre chemins possibles, mais le renforcement de la piste rend plus attractif le chemin le plus court. (ah, tiens, une roue biaisée)
- 3) les fourmis empruntent le chemin le plus court, les portions longues des autres chemins perdent leur piste de phéromones.



Les robots utilisés embarquent un compartiment pour tenir au chaud les plats mais en conséquence vont assez lentement et suivent les rues.

- On utilisera comme longueur de liens entre les noeuds la longueur des rues en utilisant `BI_MAX`.
- Nous ne mettrons pas ici de limite de distance à la livraison (la gestion de la batterie est donc une option).
- Leur vitesse est de 16 km/h mais nous ne nous en servons pas ici directement, encore une fois pour ne pas ajouter au problème (le temps de parcours est une option).
- Pour vous faciliter le travail, vous pourrez vous contenter de ne considérer que les rues de Nantes dans un premier temps. La construction du graphe de la métropole étant plus complexe, gardez le pour un second temps.

Le sujet est inspiré de l'article suivant. Pour augmenter le réalisme, piochez dedans :

<https://arstechnica.com/tech-policy/2017/03/i-am-robot-heres-your-pizza/>

Pour représenter des livraisons à des clients multiples, nous allons y aller progressivement. Il y a donc deux parties à ce TP.

- Tout d'abord, il s'agit d'implémenter l'ACO pour aller de votre restaurant (adresse à définir) vers votre client. L'algorithme devra trouver le plus court.
- Dans un second temps, imaginez que ce robot possède plusieurs compartiments et livre donc plusieurs clients, les uns après les autres. Il s'agit d'utiliser l'ACO précédent pour livrer au mieux ces clients.

Vous utiliserez pour votre TP le Python. Pour rappel vous rendrez :

- Un rapport présentant cette méthode dans le contexte choisi, en PDF
- La structure de données utilisée
- Une explication complète de la fonction de fitness choisie
- Le code commenté devra être présent sur Gitlab
- Un ou des jeux de test

L'affichage graphique montrera le graphe et l'itinéraire mais aussi les informations les plus utiles pour déterminer si l'ensemble converge (ou non) (variance, moyenne... tout ce qui vous semblera utile).

3. Complément

Pour les données, il va donc vous falloir utiliser les informations de Nantes Métropoles pour construire le graphe représentatif des voies de circulation de l'agglomération.

Attention, toutefois, lors de votre construction à éviter certains pièges. Voici un extrait du fichier en CSV qui démontre la multiplicité des cas à imaginer.

CATEGORIE	LIBELLE	MOT_DIRECTEUR	STATUT	COMMUNE	RIVOLI	TENANT	ABOUTISSANT	BI_MIN	BP_MIN	BI_MAX	BP_MAX
Allée	Allée Joachim du Bellay	Bellay	PUBLIQUE	REZE	2000						
Rue	Rue Gendron	Gendron	PUBLIQUE	REZE	1510	REZE Rue Julien Douillard	REZE Rue Aristide Briand	3	2	5	6

En effet, si l'on prend la rue Gendron, elle commence rue Julien Douillard et finit rue Aristide Briand. Les cas suivants peuvent se présenter :

- La rue Gendron coupe la rue Douillard, créant un nouveau nœud dans le graphe
- La rue Gendron coupe la rue Aristide Briand créant un nouveau nœud
- La rue Gendron fait suite à la rue Douillard et leurs nœuds d'extrémités sont confondus, cas à traiter
- La rue Gendron fait suite à la rue Aristide Briand et leurs nœuds d'extrémités sont confondus, cas à traiter

Vous noterez aussi le cas de l'allée du Bellay qui n'a ni tenant ni aboutissant... Bref, attention aux pièges.