

# TP Algorithmes de fourmi : Fourmiale

FLAVIEN LEVESQUE

## ACO-Pants

Pants fournit un moyen de déterminer comment visiter une collection de « nodes » (nœuds) interconnecté d'une manière à minimiser le travail requis. Les nodes peuvent être représentés par n'importe quelle liste de données.

Un « world » est construit à partir d'une liste de nodes et d'une fonction qui retourne la distance entre deux nodes. Ce module peut aller plus loin et utiliser le travail nécessaire pour relier deux nodes plutôt que la simple distance, nommé « edge ».

Les solutions sont trouvées dans un processus itératif.

Pour une itération, plusieurs fourmis sont lâchées et trouvent un chemin. Chacune laisse des phéromones derrière elle. Le chemin le plus court est retenu comme meilleure solution, et des fourmis élite vont venir y renforcer les phéromones. Car plus un chemin possède d'enzyme, plus les fourmis suivantes sont susceptibles de s'y aventurer.

La trace d'une phéromone disparaît au bout d'un certain nombre d'itérations.

Chaque itération doit pousser les fourmis à privilégier le chemin le plus court, tout en essayant de nouvelles solutions.

Le solveur est entièrement paramétrable :

- Importance des phéromones
- Importance de la distance
- Vitesse d'évaporation des phéromones
- Quantité de phéromones
- Nombre d'itérations
- Nombre de fourmis
- Force des fourmis élite

## Exercice

Dans le cadre de cet exercice, nos points de coordonnées sont des bars, nos fourmis des poivrots, et nos phéromones des flaques de bière.

Les données importées sont au format CSV. Chaque ligne comporte les coordonnées géographiques et cartésiennes. Nous utiliserons ces dernières pour le calcul de la distance entre chaque point.

Les coordonnées de chaque bar sont stockées dans une liste sous forme de tuple dans la forme  $(x, y)$  :

```
[(604748.0, 234405.0),  
 (587356.0, 241327.0),  
 (582888.0, 247368.0),  
 (582750.0, 248298.0),  
 (604270.0, 233920.0)]
```

Fonction fitness :

Le calcul de la distance entre deux coordonnées suit la formule suivante :

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Qui s'obtient alors en python :

```
def calcDistance(a, b):  
    d = math.sqrt(math.pow((b[0] - a[0]), 2) + math.pow((b[1] - a[1]), 2))  
    return d;
```

a et b représentant chacun un tuple de coordonnées.

Dépôt github : <https://github.com/LevFlavien/epsi-aco-pants>

Sources :

Documentation aco-pants : <http://aco-pants.readthedocs.io/en/latest/>

Calcul de la distance : <https://www.calculatorsoup.com/calculators/geometry-plane/distance-two-points.php>