Département d'informatique Intelligence Artificielle, 3ème année

Rapport de TP
Algorithme de Recherche

Optimisation par colonie de fourmi pour le TSP

Oumaima Najib

Encadrant :
Mr. Laurent Simon

Table des matières

Résumé

L'algorithme d'optimisation de colonies de fourmi est inspiré de la biologie des colonies de fourmis. Etant donné un ensemble d'endroits N et un point de départ, les fourmis recherchent le plus court chemin pour atteindre la source de nourriture. Les travaux de recherche ont montré qu'en s'inspirant de la génétique des fourmis, il est possible de trouver le plus court chemin hamiltonien dans un graphe et de résoudre des problèmes tels que le problème du voyageur de commerce Travel Salesman Problem.

Dans le présent rapport, une application de l'optimisation par colonie de fourmis pour résoudre le problème du voyageur de commerce est présentée.

1 Introduction

Un nombre de fourmis créent une intelligence collective en déposant des phéromones, celles-ci construisent une solution à un problème d'optimisation par le biais d'un mécanisme de communication.

Le problème du voyageur de commerce est un problème d'optimisation qui consiste à trouver le plus court circuit étant données un ensemble de N villes et les distances entre toutes les paires de villes qui passe par chaque ville une et une seule fois. Dans un premier temps, l'algorithme *Ant Cycle* est présenté, ensuite les résultats et les performances du modèle sont présentés, finalement une conclusion et les possibilités de développement sont décrites.

2 Principe de l'algorithme

L'algorithme est inspiré des travaux [DMC96] ; le principe du système de fourmi *Ant System* est le suivant :
Dans chaque ville $i \in \{1, \ldots, n\}$, on dépose un nombre de fourmis $b_i(t)$ tel que $\sum_{i=0}^{n} b_i(t) = m$, m : le nombre total de fourmis. La fourmi est un agent qui choisit la prochaine ville à visiter selon deux facteurs :

1. La visibilité de la ville $\eta_{i,j}$ telle que $\eta_{i,j} = \frac{1}{d_{i,j}}$, $d_{i,j}$ étant la distance entre la ville source i et la ville destination j.
2. Les pheromones $\tau_{i,j}(t)$ présents dans l'arc (i, j) au temps t.

À la fin de son tour, la fourmi dépose des phéromones sur l'ensemble des arcs des villes visitées.

La quantité de phéromones déposée par la fourmi entre la ville i et j, $(i, j) \in$`to_visit` au temps t est $\tau_{i,j}(t) = \frac{Q}{L_k}$; L_k la distance totale parcourue par la fourmi pendant le tour courant, ainsi la quantité de phéromones déposés dans l'arc (i, j) est inversement proportionnelle à la distance parcourue.

Un cycle est réalisée lorsque l'ensemble des fourmis m ont visité les n villes disponibles dans l'intervalle (t, t+1), la quantité de phéromones déposée est stockée dans une matrice `mat_pheromone` et la quantité de phéromones déposée pendant le cycle courant est stockée dans la matrice `mat_pheromones_inter`.

L'intensité d'un circuit est calculé selon la formule de l'équation 1.

$$\tau_{i,j}(t+n) = \rho.\tau_{i,j}(t) + \Delta\tau_{i,j} \tag{1}$$

ρ : le coefficient d'évaporation, $\tau_{i,j}(t)$ la quantité de phéromones dans l'arc (i, j) au temps t, $\Delta\tau_{i,j}$: la quantité de pheromones déposée par l'ensemble des fourmis à la fin du cycle n avec $\Delta\tau_{i,j} = \sum_{i=0}^{n} \Delta\tau_{i,j}^{k}$, $\tau_{i,j}^{k}$: la quantité de phéromones déposée par la fourmi k dans l'arc (i, j) à la fin du cycle.

La classe `fourmi` prend en argument les villes visitées et à l'initialisation de celle-ci est calculée la liste des villes à visiter `self.to_visit`, cette liste est mise à jour au fur et à mesure du parcours de

la fourmi.

La probabilité de passage d'une ville courante à la prochaine ville est calculée par l'équation 2.

$$p_{i,j}^{k}(t) = \frac{[\tau_{i,j}(t)]^{\alpha}.[\eta_{i,j}(t)]^{\beta}}{\sum_{k \in to\_visit}[\tau_{i,k}(t)]^{\alpha}.[\eta_{i,k}(t)]^{\beta}} \tag{2}$$

$p_{i,j}^{k}(t)$ est la probabilité de passage au temps t de la fourmi k de la ville courante i à la prochaine ville j parmis les villes non visitées.

2.1 Algorithme

Au début de l'algorithme, m fourmis sont déposées dans les n villes. Un cycle N est achevé lorsque toutes les fourmis ont visités les n villes. L'algorithme développé est inspiré de l'algorithme ANT-CYCLE [DMC96] et est défini dans 1.

Algorithm 1 ANT-CYCLE Algorithm

Require: $N_C, N_b, villes, pheromones, visibility, evaporation$
  $N_c \leftarrow 0$
  $t \leftarrow 0$
  for tout arc (i,j) **do** $\tau_{i,j}(t) = c$ et $\Delta\tau_{i,j}(t) = 0$
  Placer les m fourmis dans les n villes

  for `j` = 1,2,...,N_c **do**
    for `s` = 1,2,...,n **do**
      for `fourmi` = 1,2,...,m **do**
        Choisir la prochaine ville à visiter avec la propabilité de la formule 2
        Mise à jour de la ville de la fourmi Mise à jour du chemin
      end for
    end for
    for `fourmi` = 1,2,...,m **do**
      Déplacer la fourmi de la ville n-1 à la ville `to_visit`(0)
      Calculer la distance du tour L_k parcouru par la fourmi k
      Mise à jour du plus court chemin
    end for
    for tout arc (i,j) **do**
      for k=1,...,m **do**
        $\Delta\tau_{i,j} = \frac{Q}{L_k}$ si (i,j) dans la liste parcourue par k
        $\Delta\tau_{i,j} = \Delta\tau_{i,j} + \Delta\tau_{i,j}^{k}$
      end for
    end for
    for tout arc (i,j) **do**
      $\tau_{i,j} \leftarrow evaporation \times \tau_{i,j} + \Delta\tau_{i,j}$
      $N_c \leftarrow N_c + 1$
      $t \leftarrow t + 1$

      $\Delta\tau_{i,j} \leftarrow 0$

3 Résultats

L'algorithme développé est testé sur 14 villes, les positions des villes sont définies par leurs latitudes et leurs longitudes. Les paramètres du modèle sont fixés à :

- NBFOURMIS = 10
- NBEPOCHS = 100
- $\alpha = 1$
- $\beta = 5$
- NBVILLES = 14
- EVAPORATION = 0.5
- Q = 100
- $c = 0.2$

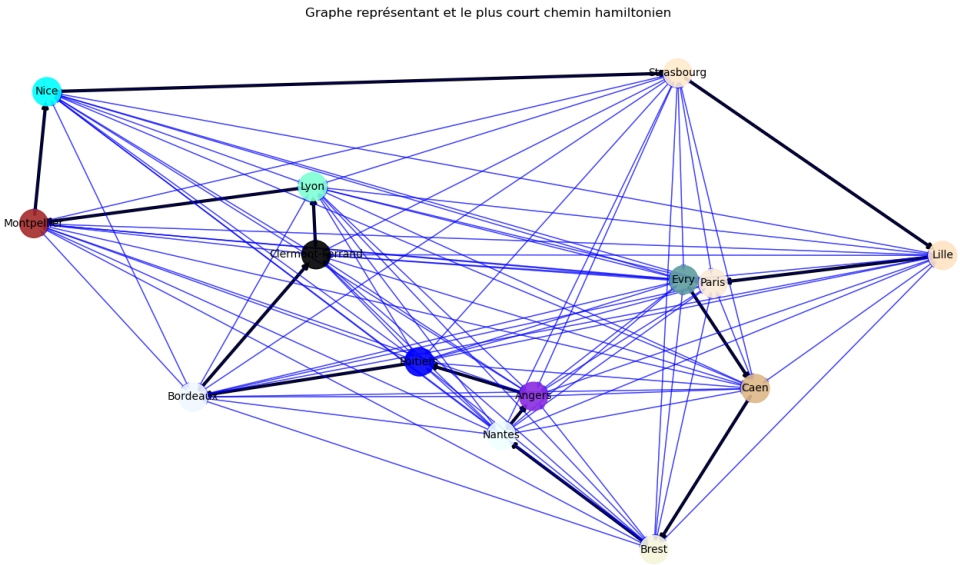

FIGURE 1 – Graphe représentant le plus court chemin trouvé par l'algorithme Ant Cycle

Un tracé de la courbe du plus court chemin 2 trouvé durant un cycle montre que le modèle converge au bout du 50ème cycle. Le plus court chemin diminue globalement pour atteindre une borne minimale de 37.90. Le choix des paramètres est justifié par les paramètres ayant eu les meilleurs résultats de [DMC96].
En modifiant le nombre de ville à 30 et les paramètres du modèle à :

- NBFOURMIS = 100
- NBEPOCHS = 100
- $\alpha = 1$

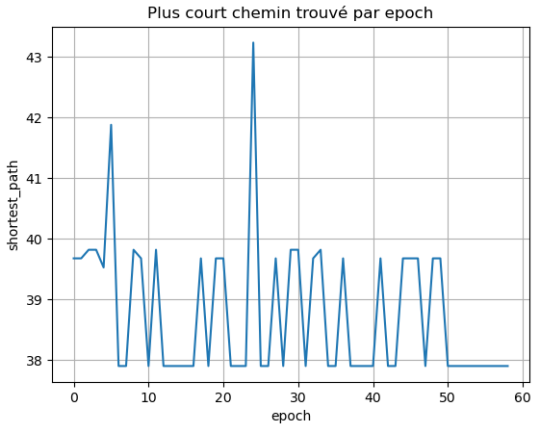

FIGURE 2 – Courbe du plus court chemin trouvé selon l'epoch - NBVILLES = 14

— $\beta = 5$
— NBVILLES = 30
— EVAPORATION = 0.5
— Q = 100
— $c = 0.2$

La courbe de la figure 3 montr que l'algorithme ANT-CYCLE converge vers le plus court chemin trouvé au 35ème cycle avec des fluctuation de l'ordre de 10^{-1}.

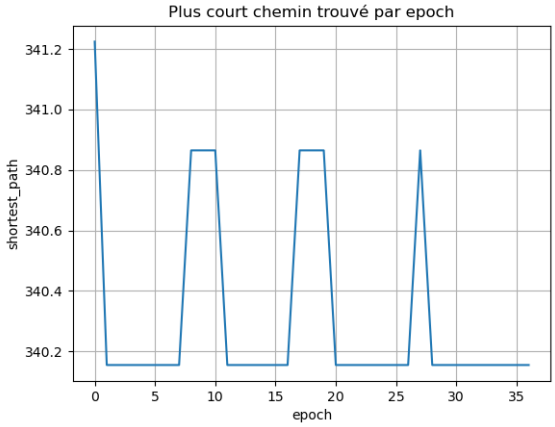

FIGURE 3 – Courbe du plus court chemin trouvé selon l'epoch - NBVILLES = 30

4 Conclusion

En conclusion, l'algorithme de colonies de fourmis Ant-Cycle est une heuristique performante pour trouver le plus court chemin hamiltonien dans un graphe et résoudre le problème du voyageur de commerce. Cependant, le choix des paramètres est déterminant et représente une bonne partie de la résolution du problème. Il existe différentes variantes de l'optimisation par colonie de fourmi parmis lesquels ant-quatity et ant-density à explorer. De plus, les algorithmes de l'optimisation par colonie de fourmi sont des méta-heuristiques d'optimisation applicable à divers problèmes de recherche du plus court chemin dans une graphe tels que la localisation GPS, la conduite des voitures autonomes, etc.

Références

[DMC96] M. Dorigo, V. Maniezzo et A. Colorni. "Ant system : optimization by a colony of cooperating agents". In : *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 26.1 (1996), p. 29-41. DOI : 10.1109/3477.484436.