

Master Informatique ISICG, semestre 2

INTELLIGENCE ARTIFICIELLE II

Résolution du problème du voyageur de commerce avec
un algorithme génétique et un algorithme de colonie de
fourmis

Auteur : ROUIJEL Mehdi

Enseignant : M. TAMINE Karim

INTRODUCTION

L'objectif de ce projet était d'apporter une solution au problème du voyageur de commerce en utilisant deux algorithmes : un algorithme génétique et un algorithme de colonie de fourmis.

Le problème du voyageur de commerce est dit « NP-complet », c'est-à-dire qu'une solution proposée peut être vérifiée rapidement, mais on ne sait pas en trouver efficacement¹. Les deux algorithmes que nous utilisons ici sont des approches non déterministes permettant d'obtenir une approximation satisfaisante de la solution au problème.

L'implémentation réalisée ici utilise le langage Processing 3.3 pour l'affichage. Les fichiers sources ci-joints peuvent être exécutés depuis l'application téléchargeable sur le site officiel² (pas d'installation nécessaire).

ENONCE DU PROBLEME

Etant donné un nombre n de villes, le problème consiste à trouver le plus court chemin passant une et une seule fois par chaque ville.

Ici, nous considérons que nous travaillons avec un graphe complet, c'est-à-dire que toutes les villes sont reliées entre elles.

La difficulté est évidente : le nombre de routes possibles va être la factorielle du nombre de villes. Même avec un nombre n relativement faible, le nombre de chemins à étudier va rapidement devenir difficile à gérer.

RAPPELS SUR LES ALGORITHMES

ALGORITHME GENETIQUE

Cet algorithme tient son nom du fait qu'il prend exemple sur le procédé de sélection naturelle

observable chez les êtres vivants pour résoudre un problème donné.

L'algorithme va devoir réaliser plusieurs itérations représentant chacune une « génération » d'individus. Au sein de chaque génération, l'aptitude des individus à résoudre le problème doit être évaluée ; les individus jugés les plus aptes vont se reproduire pour créer de nouveaux individus au patrimoine génétique combinant ceux de ses deux parents. Enfin, une dernière étape consiste à introduire un élément aléatoire - une « mutation » - permettant de distinguer les nouveaux individus de leurs ancêtres.

En pratique, dans cette implémentation, un individu est un chemin contenant toutes les villes. Puisque nous considérons que toutes les villes sont reliées deux à deux, la première génération pourra simplement passer par toutes les villes dans un ordre aléatoire. Le patrimoine génétique d'un individu correspond ainsi à l'ordre dans lequel celui-ci visite les villes.

La sélection des meilleurs candidats à la reproduction est faite sous forme de tournoi : un certain nombre des individus de la génération courante est choisi aléatoirement ; le vainqueur est celui ayant le chemin le plus court.

Une fois deux parents choisis de la sorte, leurs patrimoines génétiques vont être mélangés. Pour cela, nous prenons deux indices aléatoires ; les villes entre ces deux indices dans le chemin du premier parent sont héritées par l'enfant et le second parent transmet les villes restantes.

Enfin, nous faisons subir une mutation à l'enfant en intervertissant deux villes de son chemin.

Les meilleurs chemins sont transmis au fil des générations et les éléments aléatoires qui parsèment l'algorithme font en sorte que certaines arêtes ne soient pas ignorées parce qu'elles ont eu l'infortune de faire partie d'un mauvais chemin.

¹ https://fr.wikipedia.org/wiki/Probl%C3%A8me_NP-complet

² <https://processing.org/download/>

Un dernier point à évoquer est la notion d'élitisme. Celle-ci consiste à sauvegarder le meilleur individu d'une génération pour la génération suivante. C'est une légère optimisation permettant d'accélérer la convergence vers une solution optimale en assurant que chaque nouvelle génération aura au moins un individu au moins aussi bon que le meilleur de ses ancêtres.

COLONIE DE FOURMIS

Cet algorithme se base sur une observation empirique du fonctionnement d'une colonie de fourmis. C'est un système multi-agents qui utilise la notion de connaissance collective.

Les fourmis communiquent entre elles par le biais de phéromones qu'elles laissent sur leur chemin pour signaler la direction d'un danger, d'une source de nourriture, etc., à leurs consœurs.

Si l'on prend l'exemple d'une source de nourriture, après qu'une fourmi l'ait trouvée, elle va retourner au nid en laissant une traînée de phéromones. Si plusieurs fourmis sont parties en même temps, la fourmi ayant emprunté le chemin le plus court sera la première à rentrer au nid, et les fourmis suivantes seront plus enclines à suivre le même chemin.

Le terme « enclin » est important, puisque si ce chemin est le plus rapide parmi ceux trouvés par les fourmis parties simultanément, ce n'est pas nécessairement le meilleur chemin possible. Les fourmis ne vont donc pas suivre aveuglément les traînées de phéromones ; ces dernières vont servir à pondérer le choix des chemins. Il est cependant important que certaines fourmis examinent d'autres routes possibles. Si l'une de ces autres routes est meilleure, les quelques fourmis qui vont l'emprunter vont revenir au nid plus rapidement et, au fil des itérations, les phéromones vont y être plus fortes, tandis que celles des autres chemins vont s'évaporer.

En termes d'implémentation, nous avons une fourmi qui est définie par le chemin qu'elle a parcouru ainsi que par sa réserve de phéromones. Lors de son

retour vers le nid, elle va diviser également sa réserve de phéromones entre toutes les routes séparant deux villes qu'elle a empruntées.

A chaque ville qu'elle atteint, une fourmi va examiner les routes possibles ; dans notre cas, elle ne peut pas rencontrer d'impasse mais elle va faire un choix en fonction des différentes quantités de phéromones sur les routes possibles et d'une heuristique ; l'heuristique choisie ici est la longueur de la route. Ce choix est déterminé de façon probabiliste et est pondéré pour favoriser une route courte à une route chargée de phéromones ; cette décision d'implémentation a pour but de faire en sorte qu'un maximum de chemins soit examiné.

La dernière étape importante à chaque itération de l'algorithme est l'évaporation des phéromones. Une valeur « idéale » pour le taux d'évaporation doit être trouvée ; en effet, si les phéromones disparaissent trop vite elles auront un impact moindre à inexistant sur le déroulement de l'algorithme, et si elles mettent trop de temps à s'évaporer, elles risquent de créer un biais trop important.

ANALYSE

La première chose à noter quant à ces deux algorithmes est qu'ils sont non déterministes. Autrement dit, pour un même ensemble de données, le même algorithme a de très fortes chances de donner des résultats différents.

Nombre de villes	Temps d'Exécution Moyen (ms)	
	Algorithme Génétique	Colonie de Fourmis
5	3346	3349
10	3348	3351
15	3352	3347
20	3345	3982
25	4659	7306
50	44813	63256
75	156347	226318

Figure 1: 200 itérations, 20 individus

Les résultats de nos tests tendent à montrer que l'algorithme génétique est plus rapide que la colonie de fourmis, pour un nombre de villes suffisamment grand.

En outre, dans le cas de l'algorithme génétique, chaque génération donne un résultat au moins aussi bon que la génération précédente. La colonie de fourmis, quant à elle, donne des résultats « en dents de scie » d'une itération à l'autre.

La colonie de fourmis apparaît moins flexible en cela qu'elle semble nécessiter plus de « peaufinage » au niveau de ses paramètres (taux d'évaporation des phéromones, biais en faveur des phéromones ou de l'heuristique lors du choix d'une arête), en particulier lorsque le nombre de villes augmente.

CONCLUSION




Ce projet nous aura permis d'étudier un problème intéressant, par le biais de deux algorithmes fondamentaux et couramment utilisés.

Les résultats obtenus sont relativement satisfaisants, bien qu'il soit possible d'apporter plusieurs améliorations à l'implémentation.


Premièrement, l'implémentation a été faite un peu trop rapidement, sans réelle phase de conception, ce qui a amené à une certaine redondance dans les classes créées (classe Ant et Individual qui pourraient être fusionnées, par exemple).

Ensuite, en termes d'usabilité, il aurait pu être intéressant d'intégrer la possibilité de modifier certains paramètres (nombre d'individus, taux d'évaporation des phéromones, ...) depuis l'interface graphique.

Enfin, il aurait pu être intéressant d'explorer d'autres heuristiques, à savoir, avec l'algorithme génétique :

-  Pour la sélection des individus reproducteurs
-  Pour le mélange des gènes des parents
-  Pour la mutation des gènes de l'enfant

Et avec la colonie de fourmis :

-  Pour le choix d'une arête

REFERENCES

ALGORITHME GENETIQUE

<http://antoinevastel.github.io/algorithmes/python/algorithmes%20g%C3%A9n%C3%A9tiques/2016/04/30/probleme-voyageur-commerce.html>

<http://www.theprojectspot.com/tutorial-post/applying-a-genetic-algorithm-to-the-travelling-salesman-problem/5>

COLONIE DE FOURMIS

<http://khayyam.developpez.com/articles/algo/voyageur-de-commerce/colonies-de-fourmis/>

<http://www.theprojectspot.com/tutorial-post/ant-colony-optimization-for-hackers/10>

<https://www.openprocessing.org/sketch/8950#>

<http://stackoverflow.com/questions/6737283/weighted-randomness-in-java>