

Optimisation d'une livraison de colis par biomimétisme

Mon intérêt pour ce sujet se justifie par l'omniprésence des problèmes modélisés par le problème du voyageur de commerce et par le fait que l'on ne connaisse pas de solution efficace pour résoudre ce problème rendant ainsi sa résolution sur de grandes instances modélisant certaines situations concrètes plus compliqués.

Ce sujet s'inscrit dans le thème de la ville dans la mesure où le problème du voyageur de commerce modélise entre autre les livraisons que peut faire un fournisseur à ses clients qu'il est préférable d'optimiser pour des raisons écologiques et économiques.

Positionnement thématique (ÉTAPE 2) :

- *INFORMATIQUE (Informatique pratique)*
- *INFORMATIQUE (Informatique Théorique)*

Mots-clés (ÉTAPE 1) :

Mots-clés (en français) **Mots-clés (en anglais)**

Méthodes d'approximation *Approximation methods*

Optimisation *Optimization*

Intelligence distribuée *Swarm intelligence*

Algorithme des fourmis *Ant algorithm*

Graphes *Graph*

Bibliographie commentée

Dès le IXe siècle, Rudrata pose le problème du cavalier qui consiste à faire passer un cavalier une unique fois sur chaque case d'un jeu d'échec en utilisant uniquement les déplacements d'un cavalier. Plus tard, au XIXe siècle, William Rowan Hamilton posera le problème de la recherche de cycles Hamiltoniens dans un graphes [1], c'est à dire un cycle passant exactement une seule fois par chaque sommet. C'est par la suite que le problème consistant à trouver le cycle Hamiltonien de poids minimal dans un graphe qui sera nommé pour la première fois dans les années 1930 le problème du voyageur de commerce (TSP)[2]. Ce problème continu à faire l'objet de nombreuses recherches pour les enjeux économiques et écologiques que l'obtention de la tournée de livraison de plus courte distance implique. A l'heure actuelle, un des programmes pouvant résoudre les plus grandes instances du TSP est le concorde TSP solver [3], un programme écrit par David Applegate, Robert E. Bixby, Vašek Chvátal, et William J. Cook qui est capable de trouver le plus court chemin sur des graphes de l'ordre du millier de sommets.

Ce problème montre tout son intérêt par ses diverses applications, premièrement son application directe qui permet d'optimiser le trajet d'une ligne de bus dans une ville, une

ournée de livraison mais aussi ses applications dans l'industrie en permettant par exemple de minimiser le temps que met une fraiseuse à commande numérique pour percer n trous dans une plaque métallique et même en biologie [2]. Cependant bien que très utile ce problème ne dispose pas de résolution efficace. En effet, ce problème étant NP-difficile, il n'existe pas de solution de complexité temporelle polynomiale et donc efficace pour le résoudre (en supposant $P \neq NP$). La résolution devient impossible pour des instances de grandes tailles [2].

C'est donc logiquement, que la recherche s'est orienté vers des algorithmes d'approximations telle que l'optimisation par colonie de fourmis (ACO : ant colony optimization) [4], aux complexités temporelles plus faible [5], pour des instances de plus grande taille. Cet algorithme simule en quelque sorte le comportement d'une colonie de fourmis en utilisant le principe de phéromone [6] des fourmis qui est leur moyen de communication. Les choix des fourmis sont aléatoires mais dépendent de plusieurs paramètres (distance d'une fourmi à un sommet du graphe, taux de phéromone sur les arêtes etc...). Il semble cependant utile d'étudier les caractéristiques de cet algorithme sur des graphes de références pour mieux connaître sa fiabilité car il est probabiliste, mais aussi de connaître au mieux sa vitesse de convergence [7] ce qui permettrait en quelque sorte de caractériser son efficacité en fonctions des nombreux paramètres (le nombre de fourmis, le nombre d'itérations, le taux d'évaporation des phéromones...) entre autre sur des graphes de référence [8] [9].

Problématique retenue

Comment peut-on optimiser les différentes étapes d'une livraison de colis en s'inspirant du comportement des fourmis ?

Objectifs du TIPE du candidat

Mon premier objectif est d'écrire un programme reposant sur le principe d'ACO afin de trouver des solutions approchés au TSP. Puis, par la suite de tenter d'adapter ce principe au problème du bin packing. J'étudierais ensuite les caractéristiques des programmes obtenues telles que leurs vitesses de convergence mais aussi les résultats obtenus sur des instances de références. Enfin, j'analyserais les différents paramètre de l'ACO que sont : le nombre de fourmis, le taux d'évaporation des phéromones et le nombre d'itération)

Références bibliographiques (ÉTAPE 1)

[1] WIKIPEDIA CONTRIBUTORS : Travelling salesman problem : *Wikipedia, the free encyclopedia, 2022. [Online ; accessed 9-January-2023]. https://en.wikipedia.org/wiki/Travelling_salesman_problem*

[2] AHMED HAMMOUCH, BELAÏD AHIOD, SALMA MOULINE, SÉBASTIEN VEREL ET MOHAMED EL HASSOUNI : Adaptation de métaheuristiques pour résoudre des problèmes d'optimisation combinatoire liés aux transports. : https://www.elkrari.com/docs/thesis/these_elkrari.pdf

[3] MICHEL DEUDON, PIERRE COURNOT, ALEXANDRE LACOSTE, YOSSIRI ADULYASAK ET LOUIS-MARTIN ROUSSEAU : Learning heuristics for the tsp by policy

gradient : *In International conference on the integration of constraint programming, artificial intelligence, and operations research*, pages 170–181. Springer, 2018.

[4] ALBERTO COLORNI, MARCO DORIGO, VITTORIO MANIEZZO : Distributed optimization by ant colonies : *In Proceedings of the first European conference on artificial life*, volume 142, pages 134–142. Paris, France, 1991. https://www.researchgate.net/publication/216300484_Distributed_Optimization_by_Ant_Colonies

[5] MUNAN LI : Efficiency improvement of ant colony optimization in solving the moderate tsp. : *Journal of Systems Engineering and Electronics*, 26(6):1300– 1308, 2015. Efficiency improvement of ant colony optimization in solving the moderate tsp. <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7375763>

[6] MARCO DORIGO, ERIC BONABEAU ET GUY THERAULAZ : Ant algorithms and stigmergy. : *Future generation computer systems*, 16(8):851–871, 2000. <http://lia.deis.unibo.it/Courses/AI/fundamentalsAI2006-07/papers/stigmergy.pdf>

[7] THOMAS STÜTZLE, MARCO DORIGO : A short convergence proof for a class of ant colony optimization algorithms. : *IEEE Transactions on evolutionary computation*, 6(4):358–365, 2002. https://www.researchgate.net/publication/3418698_A_short_convergence_proof_for_a_class_of_ACO_algorithms

[8] GERHARD REINELT : Tsplib a traveling salesman problem library : *ORSA journal on computing*, 3(4):376–384, 1991.

[9] ERIC BONABEAU, MARCO DORIGO, GUY THERAULAZ : Swarm intelligence : from natural to artificial systems. : Numéro 1. Oxford university press, 1999.

DOT

[1] : Septembre 2022 : Recherches générales sur le principe de l'algorithme des fourmis, du problème du voyageur de commerce et de leurs applications

[2] : Octobre 2022 : Implémentation d'une première version simple de l'optimisation par colonie de fourmis en Python.

[3] : Fin Novembre 2022 : Premiers tests sur des graphes de référence de la base de donnée TSPLIB, les résultats peu satisfaisants me mènent à chercher des améliorations possibles

[4] : Novembre à Décembre 2022 : Tests concernant l'importance des paramètres alpha et beta, difficultés rencontrées quant au choix des valeurs des nombreux paramètres puis prise de contact avec Guy Théraulaz, chercheur dans le domaine de l'intelligence par essaim. Il m'a conseillé de lire le livre "Ant Colony Optimization" de Marc Dorigo mais également "Swarm intelligence" dont il est entre autre l'auteur.

[5] : Janvier à Février 2023 : Après la lecture des ouvrages et d'autres thèses, je découvre plusieurs procédés applicables à une optimisation par colonie de fourmis. Je réalise donc une nouvelle série de tests afin d'identifier les procédés donnant les meilleurs résultats.

[6] : Mars à Avril 2023 : Après avoir sélectionné les procédés donnant les meilleurs résultats qui n'augmentent pas trop le temps de calcul, je décide d'implémenter une nouvelle version de l'optimisation de colonie de fourmis en C pour que l'exécution du programme et les tests soient plus rapides.

[7] : Mai 2023 : Obtention et analyse de résultats satisfaisants, comparaisons de ceux-ci avec ceux de la première version de l'algorithme, test sur un plus grand nombre de graphes.