Algorithmes collaboratifs et applications

Rapport du Projet

le Plus Court Chemin et l'algorithme de colonie de fourmis

Auteurs :
M. ilyas DAHAOUI

Encadrants :
M. Alexandre SAIDI

Version du
28 avril 2024

Table des matières

1 Introduction

L'optimisation combinatoire est un domaine crucial dans lequel de nombreuses problématiques complexes doivent être résolues pour trouver la meilleure solution possible. Parmi ces problèmes, le Problème du Plus Court Chemin (PCC) est d'une importance particulière, car il consiste à trouver le chemin le plus court entre deux points dans un graphe pondéré. Ce problème est essentiel dans de nombreux domaines tels que la logistique, les réseaux de communication et la planification des transports.

Le projet présenté ici vise à résoudre le Problème du Plus Court Chemin en utilisant l'algorithme des colonies de fourmis (Ant Colony Optimization - ACO). L'ACO est une méthode métaheuristique qui s'inspire du comportement des fourmis lorsqu'elles recherchent de la nourriture. Dans ce contexte, chaque fourmi représente un agent qui explore l'espace des solutions en suivant des chemins dans le graphe, en déposant et en suivant des phéromones.

Ce rapport détaillera la conception, l'implémentation et les résultats obtenus grâce à notre mise en œuvre de l'algorithme des colonies de fourmis pour résoudre le Problème du Plus Court Chemin. Nous commencerons par présenter le contexte et les concepts théoriques derrière l'algorithme des colonies de fourmis, en mettant l'accent sur son application au PCC. Ensuite, nous décrirons notre approche de résolution, y compris les choix de paramètres et les structures de données utilisées. Enfin, nous analyserons les résultats obtenus et discuterons des implications de notre travail.

2 Fondements de l'Algorithme des Colonies de Fourmis (ACO)

2.1 Modélisation de la colonie de fourmis

La modélisation de la colonie de fourmis comme un système multi-agents repose sur des équations mathématiques décrivant le comportement des agents (fourmis) dans leur environnement. Soit t le temps discret représentant les itérations de l'algorithme. L'environnement est modélisé comme un graphe $G = (V, A)$, où V est l'ensemble des villes et A est l'ensemble des arêtes représentant les routes entre les villes. Chaque arête (i, j) est caractérisée par le niveau de phéromone $\tau_{ij}(t)$ au temps t et par son attractivité η_{ij}.

Les équations suivantes décrivent les interactions entre les fourmis et leur environnement :

1. **Évaporation de la phéromone** :

$$\tau_{ij}(t+1) = (1-\rho) \times \tau_{ij}(t)$$

   L'équation ci-dessus représente l'évaporation de la phéromone sur chaque arête à chaque itération, où ρ est le taux d'évaporation.

2. **Dépôt de phéromone** :

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \sum_{k=1}^{n} Q$$

   si la fourmi k a emprunté l'arête (i, j), où Q est la quantité de phéromone déposée par une fourmi lors de son passage. Ainsi, après le premier passage des fourmis sur le graphe, l'actualisation se fait selon l'équation suivante :

$$\tau_{ij} = (1-\rho)\tau_{ij} + \frac{1}{N}\sum_{k=1}^{N} \Delta\tau_{ij}^{k}$$

3. **Choix de l'arête par une fourmi** :

$$p_{ij}^{k} = \frac{\tau_{ij}^{\alpha} \times \eta_{ij}^{\beta}}{\sum_{l \in allowed\_edges} \tau_{il}^{\alpha} \times \eta_{il}^{\beta}}$$

   Cette équation calcule la probabilité p_{ij}^{k} pour la fourmi k de choisir l'arête (i, j) en fonction de la phéromone et de l'attractivité des arêtes disponibles, avec α et β comme paramètres de pondération.

4. **Mouvement d'une fourmi** : À chaque itération, si une fourmi est à la ville i, elle choisit l'arête (i, j) avec une probabilité p_{ij}^k.

Ces équations décrivent de manière mathématique les interactions entre les fourmis et leur environnement, permettant ainsi de simuler leur comportement collectif dans la recherche de nourriture, avec chaque fourmi choisit la prochaine ville à visiter en fonction du taux de phéromone τ_{ij} sur l'arête $i - j$ et de l'heuristique η_{ij} associée à cette arête. L'heuristique η_{ij} peut être interprétée comme l'attractivité de l'arête $i - j$, souvent définie comme l'inverse de sa distance d_{ij}, c'est-à-dire $\eta_{ij} = \frac{1}{d_{ij}}$.

Dans notre modèle simplifié, la quantité $\Delta\tau_{ij}^k$ est définie comme suit :

$$\Delta\tau_{ij}^k = \begin{cases} Q/L_k & \text{si la fourmi passe sur l'arête } i - j \text{ dans son parcours} \\ 0 & \text{sinon} \end{cases}$$

Où Q correspond à la quantité de phéromones déposée par chaque fourmi, une constante, et L_k est le coût du chemin de la k-ème fourmi, typiquement la longueur totale de son chemin.

2.2 Modélisation UML possible :

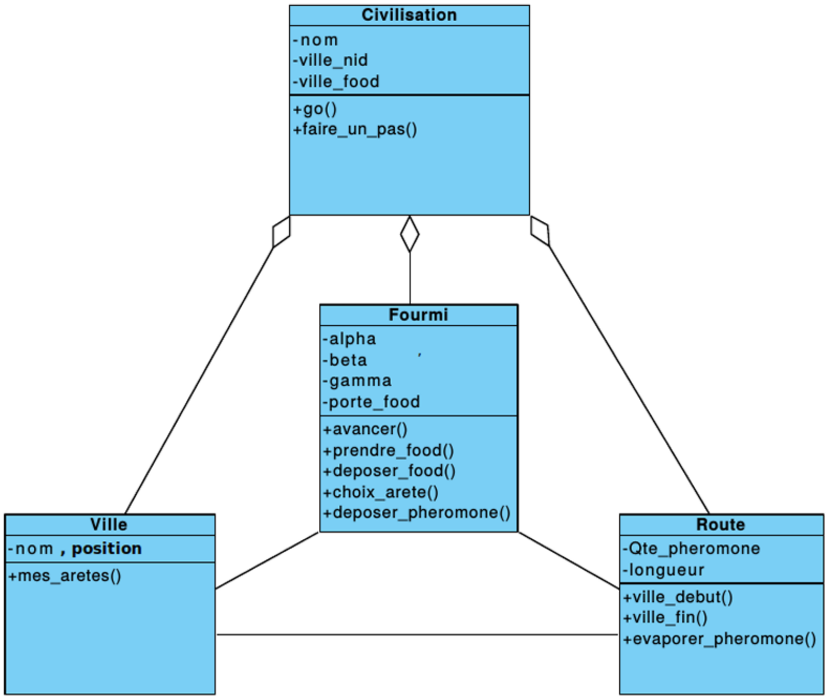

FIGURE 1 – Modélisation Possible du problème

3 Méthodologie

3.1 Description de l'algorithme :

La colonie de fourmis fonctionne de la manière suivante : à chaque étape, chaque fourmi se déplace d'une ville à une autre. À son arrivée dans une ville, elle sélectionne sa prochaine destination en se basant sur l'intensité des phéromones présentes sur chaque route, suivant ainsi un processus décisionnel spécifique décrit précédemment. Cette démarche se poursuit jusqu'à ce que la fourmi découvre la source de nourriture, moment où elle la prend et commence son trajet de retour en suivant la piste de phéromones qu'elle a elle-même laissée. Une fois de retour dans le nid, elle dépose la nourriture et recommence le processus de recherche. Ce processus efficace permet à la colonie de fourmis de trouver et de ramener la nourriture au nid. Cependant, il convient de noter que si une fourmi est la première à explorer le parcours, le taux de phéromones est uniforme sur toutes les routes possibles, la fourmi choisira alors aléatoirement son chemin.

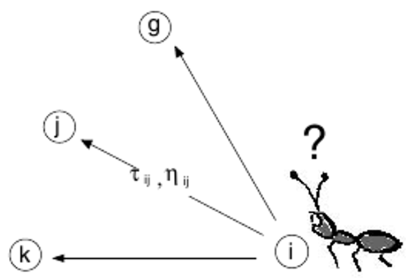

FIGURE 2 – le choix de la nouvelle ville

4 Notre solution du PCC avec l'algorithme de colonie de fourmis :

4.1 interface graphique :

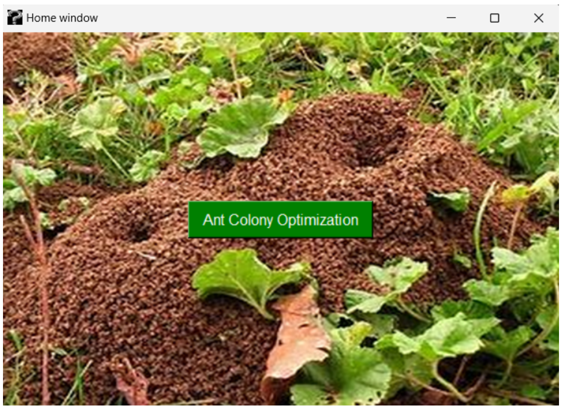

FIGURE 3 – Fenêtre d'accueil

4.2 Comment ça marche ! :

Cliquez une fois sur le bouton Ant colony optimisatoin pour lancer le processus

FIGURE 4 – veuillez construire le graphe

Nous souhaitons vous demander de construire le graphe. Vous avez la liberté de positionner et de connecter les villes comme bon vous semble. Nous

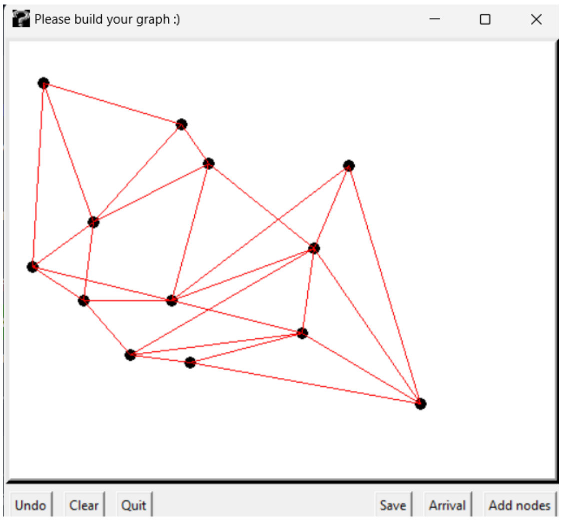

FIGURE 5 – exemple de graphe

allons choisir le point de départ et d'arrivée dans le graphe en utilisant le bouton Arrivée/Départ.

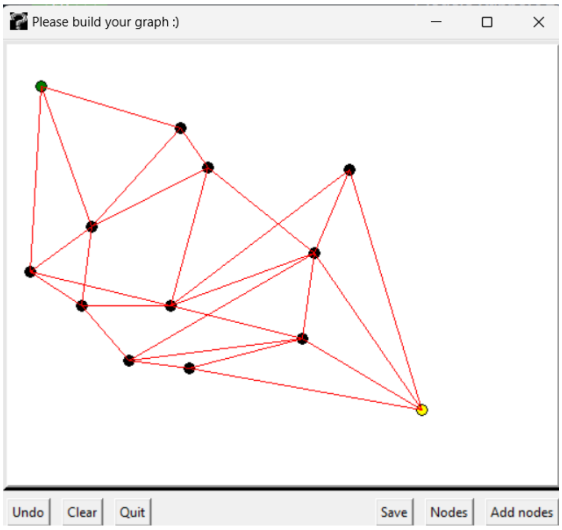

FIGURE 6 – départ et arrivée du voyageur

Vous allez sauvegarder le graphe afin de pouvoir ajuster les paramètres de l'algorithme plus tard.

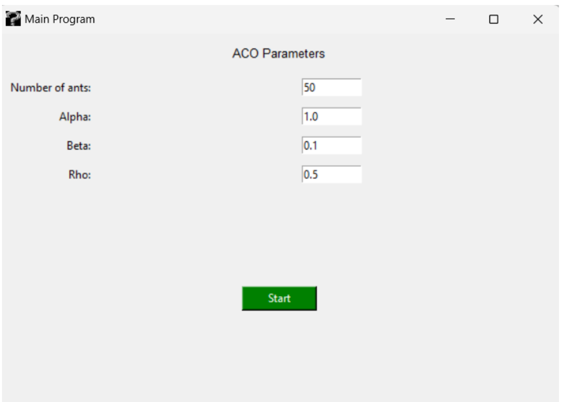

FIGURE 7 – Enter Caption

Vous allez cliquer sur Start, puis vous choisirez entre faire un seul pas ou un aller-retour pour résoudre le problème en question.

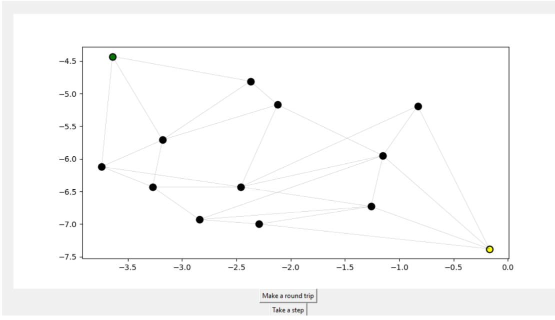

FIGURE 8 – Enter Caption

Enfin, nous trouvons le chemin le plus optimal à l'aide de l'algorithme de colonie de fourmis.

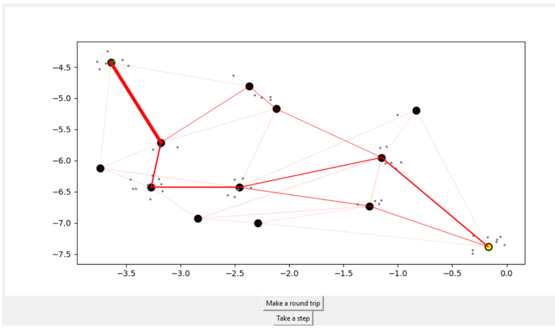

FIGURE 9 – Enter Caption

4.3 test de l'algorithm sur un graphe plus complexe :

Nous procédons à un test de l'algorithme sur un graphe plus complexe pour évaluer sa performance dans des scénarios plus exigeants. Le graphe présenté ci-dessous illustre une configuration plus dense et variée, mettant ainsi à l'épreuve la capacité de l'algorithme à trouver une solution optimale dans des environnements plus challengers.

Après un certain nombre d'itérations, l'algorithme converge vers une solution optimale, comme illustré dans la figure suivante :

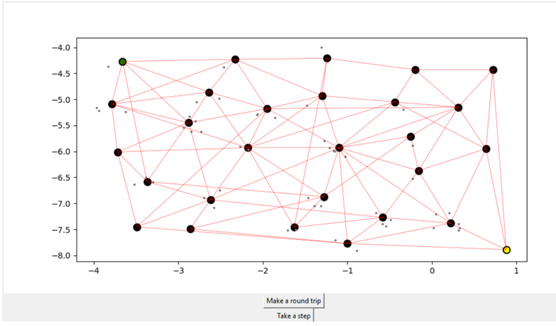

FIGURE 10 – un graphe plus complexe

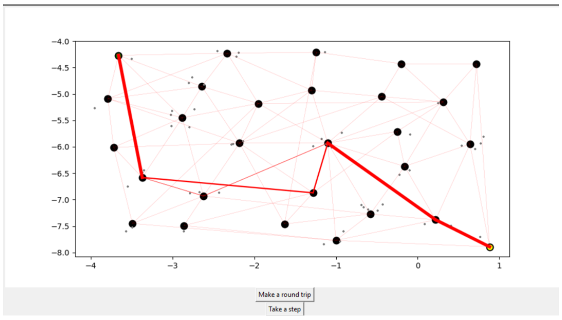

FIGURE 11 – la solution

En analysant la performance de l'algorithme, nous observons que, même sur des graphes complexes, il parvient à trouver rapidement une solution optimale. Ce résultat démontre l'efficacité de l'approche de colonie de fourmis dans la résolution de problèmes d'optimisation combinatoire.

5 Conclusion

En conclusion, notre projet d'implémentation de l'algorithme des colonies de fourmis pour résoudre le Problème du Plus Court Chemin (PCC) a été une expérience enrichissante et fructueuse. Nous avons réussi à concevoir et à mettre en œuvre un système capable de trouver des solutions efficaces et souvent optimales à des instances du PCC, en utilisant une approche inspirée du comportement des fourmis lorsqu'elles cherchent de la nourriture.

Notre travail a permis de démontrer l'efficacité de l'algorithme des colonies de fourmis dans la résolution de problèmes d'optimisation combinatoire, notamment dans le contexte du PCC. En exploitant les interactions entre les agents (les fourmis) et leur environnement (le graphe), nous avons pu trouver des chemins courts et efficaces entre les points de départ et d'arrivée dans divers scénarios.

+

Mots clés : Colonie de fourmis, Problème du Plus Court Chemin, Optimisation combinatoire, Algorithme des colonies de fourmis, Métaheuristique, Graphe, Phéromones.

École centrale de Lyon
36, Avenue Guy de Collongue
code du labo concerné
69134 Écully

6 Réferences :

Shtovba, S. D. (2005). Ant algorithms : Theory and applications. Programming and Computer Software, 31(4), 167-178.

Shtovba, S. D. (2005). Ant algorithms : Theory and applications. Programming and Computer Software, 31(4), 167-178.

Guntsch, M., & Middendorf, M. (2002). Applying population based ACO to dynamic optimization problems. In International Workshop on Ant Algorithms (pp. 111-122). Springer, Berlin,