



## Allocation en ligne de tâches

projet à réaliser en binômes

On se propose d'étudier un problème d'allocation multi-agents de tâches dans des flottes de taxis. Les trajets (tâches) à réaliser par les taxis arrivent "en ligne" c'est-à-dire au cours de l'exécution et ne sont pas connus a priori. Ce problème d'allocation a été récemment décrit sous différentes variantes dans la littérature, par exemple dans [1, 2].

On supposera que les véhicules se déplacent dans un plan orthonormé et que les positions de départ et d'arrivée des trajets correspondent à des positions valides dans ce plan. Les tâches de trajets consistent en une position de départ et une position d'arrivée. Chaque tâche a un coût correspondant à la distance euclidienne entre la position de départ et la position d'arrivée.

Tous les  $T$  pas de temps, les taxis reçoivent un nouvel ensemble de tâches (trajets) à se répartir. Les véhicules peuvent bien entendu ne pas avoir terminé l'exécution des trajets déjà acceptés précédemment. Le coût de l'exécution d'un ensemble de trajets pour un taxi correspond à la somme des distances de ces trajets à laquelle s'ajoute la somme des distances parcourues entre chaque trajet pour relier une position d'arrivée d'un trajet à la position de départ du trajet suivant.

L'objectif des agents est de minimiser la somme des coûts des trajets et des déplacements entre les trajets tout en garantissant l'exécution de toutes les tâches.

Dans ce projet nous étudierons et comparerons différentes méthodes décentralisées de résolution du problème d'allocation en lignes des tâches entre les agents (taxis).

La première partie de votre travail consistera en l'implémentation d'un environnement de tests générant des problèmes d'allocation de tâches pour des taxis dans un espace de taille finie. La seconde partie du travail consistera en la modélisation et la résolution par des DCOP du problème d'allocation des tâches entre les agents. La troisième partie du travail consistera en l'étude de différents protocoles de coordination multi-agents pour résoudre ce même problème. Cette partie est liée à la troisième séquence de cours en CoCoMa qui commencera le 17 décembre.

## 1 Partie 1 : Génération de tâches en ligne

Dans cette partie, vous développerez les briques nécessaires à la génération d'instances du problème d'allocation en ligne de trajets entre des taxis. Vous devrez pouvoir générer des environnements en configurant le nombre d'agents (taxis), la taille de l'environnement, la fréquence d'arrivée des tâches ( $T$ ), la génération aléatoire de nouvelles tâches.

Votre programme permettra de simuler l'exécution du système en particulier la génération de nouvelles tâches tous les  $T$  pas de temps, l'allocation de ces tâches et leur exécution.

Vous devrez également développer un algorithme de planification qui étant donné un ensemble de trajets calcule le meilleur ordonnancement de ces tâches et le coût associé (tenant compte des coûts des tâches et des déplacements entre les tâches).

L'implémentation d'une interface graphique permettant la visualisation des allocations et de l'exécution des trajets donnera lieu à des points bonus dans le barème d'évaluation.

## 2 Partie 2 : Optimisation de contraintes distribuées

Le problème de coordination multi-agents entre taxis peut-être modélisé par un DCOP. La bibliothèque Pydcop utilisée en TP peut être utilisée afin de résoudre ces problèmes formalisés comme des DCOP.

Dans cette partie, vous développerez les briques nécessaires à la résolution, avec des DCOP, du problème d'allocation des tâches reçues à un instant  $T$ .

- A partir d'un ensemble de tâches reçues à un instant  $T$ , modélisez le problème de l'allocation de ces tâches en utilisant un DCOP. Vous pourrez vous inspirer de la modélisation proposée dans [3] pour des tâches d'observations à des satellites. Pensez à tenir compte de la date de disponibilité de chaque agent (certains agents peuvent avoir des tâches à terminer avant de pouvoir en commencer de nouvelles!). On rappelle qu'on souhaite minimiser la somme des coûts des agents pour la réalisation de l'ensemble des tâches reçues à un instant  $T$ .
- Implémentez une fonction permettant de générer automatiquement un fichier dcop à partir d'un ensemble de tâches reçues à un instant  $T$ .
- En faisant appel à la bibliothèque Pydcop, permettez la résolution, avec des DCOP, des problèmes d'allocation des tâches générés à la question précédente. Vous choisirez l'algorithme qui vous semble le plus adapté.
- Dans l'environnement développé dans la partie 1, permettez la résolution des problèmes d'allocation générés en utilisant Pydcop.
- Comparez expérimentalement le temps de résolution et la qualité des solutions retournées par les différents algorithmes de la bibliothèque Pydcop. Vous ferez varier la taille et la complexité des problèmes étudiés.

## 3 Partie 3 : Négociation

Cette partie vise à implémenter des protocoles de négociation permettant aux agents de coordonner l'allocation des tâches. Vous étudierez les performances des approches à base d'enchères vues en cours. Plus spécifiquement :

- les enchères *parallèles* (PSI), où chaque agent fait des offres en parallèle sur l'ensemble des items ;
- les enchères *séquentielles* (SSI), où les offres sont réalisées itérativement sur les items ;
- les enchères *séquentielles basées sur le regret* ;
- (optionnel) l'algorithme de consensus CBBA (seule la version CBAA a été détaillée en cours, voir la référence pour le détail de la construction des bundles).

La méthode de calcul des coûts marginaux nécessaire pour le calcul des offres des agents consiste à comparer le plan actuel et le plan qui résulterait de l'ajout de l'item.

Vous reprendrez les instances obtenues à l'aide de votre générateur de la partie 1, et vous comparerez les performances des différentes approches basées sur les enchères. Ces protocoles seront ensuite comparés expérimentalement à l'approche DCOP.

## 4 Rendu du projet

Le projet sera à rendre pour le 10 février au plus tard. Vous devrez transmettre par mail (aurelie.beynier@lip6.fr, nicolas.maudet@lip6.fr) le code de votre projet ainsi qu'un rapport décrivant le travail réalisés et les résultats obtenus.

Nous vous recommandons de traiter les deux premières parties du projet avant les vacances de fin d'année.

Vous devrez rendre votre projet sous forme d'un notebook Python commenté et accompagné d'un rapport au format PDF (LateX recommandé!). Vous expliquerez vos choix (algorithmes de planification, protocoles de coordination...) et détaillerez votre étude expérimentale détaillant les différentes méthodes de résolution du problème.

## Références

- [1] John P. Dickerson, Karthik A. Sankararaman, Aravind Srinivasan, and Pan Xu. Allocation problems in ride-sharing platforms : Online matching with offline reusable resources. *ACM Trans. Econ. Comput.*, 9(3), June 2021.
- [2] Déborah Conforto Nedelmann, Jérôme Lacan, and Caroline P. C. Chanel. Skate : Successive rank-based task assignment for proactive online planning. *Proceedings of the International Conference on Automated Planning and Scheduling*, 34(1) :396–404, May 2024.
- [3] Gauthier Picard. Auction-based and distributed optimization approaches for scheduling observations in satellite constellations with exclusive orbit portions. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '22, page 1056–1064, Richland, SC, 2022. International Foundation for Autonomous Agents and Multiagent Systems.