
Approches multiagents pour l'allocation de courses à une flotte de taxis autonomes

Gauthier Picard, Flavien Balbo, Olivier Boissier

Univ Lyon, MINES Saint-Étienne, CNRS, Laboratoire Hubert Curien UMR 5516 Saint-Étienne, France

{gauthier.picard,flavien.balbo,olivier.boissier}@emse.fr

RÉSUMÉ. Ce travail étudie le problème d'allocation décentralisée de courses à une flotte de taxis autonomes. Classiquement, pour résoudre ce problème, les demandes sont centralisées dans un portail où un dispatcheur alloue les courses aux taxis (idéalement de manière optimale). Ceci nécessite que les taxis aient accès en continu au portail (via un réseau cellulaire). Cependant, avoir accès à une telle infrastructure de communication globale coûte cher à la société de gestion de taxis. L'idée est ici d'utiliser une infrastructure véhicule-à-véhicule, peu coûteuse, pour coordonner les taxis sans infrastructure de communication globale. Notre approche est présentée et évaluée de manière empirique par simulation. Nous avons développé différentes stratégies multiagents, requérant différentes infrastructures de communication et mécanismes de coordination, et les analysons en termes de qualité de service, de satisfaction client, de gain et de robustesse face à la perte de messages.

ABSTRACT. This work is interested in decentrally solving a taxi allocation problem over a fleet of autonomous taxis. Classically, to solve this problem, requests are centralized into a portal where a dispatcher allocates requests to taxis (ideally, in an optimal manner). This requires taxis have continuous access to the portal. However, getting access to such global communication infrastructure is very expensive for taxi companies. The idea here is to use new affordable vehicle-to-vehicle communication technologies to coordinate taxis without global communication infrastructure. Our approach is presented and empirically evaluated via simulations. We have developed different scenarios with different communication infrastructure and coordination mechanisms, and we analyze, their resulting quality of service, user welfare, gain and robustness to message loss.

MOTS-CLÉS : allocation de ressources, taxis autonomes, DCOP.

KEYWORDS: resource allocation, autonomous taxis, DCOP.

DOI:10.3166/RIA.32.223-247 © 2018 Lavoisier

1. Introduction

Le développement de véhicules autonomes capables de communiquer en pair-à-pair et le succès des solutions de transport à la demande (e.g. Uber, Lyft, Heetch) sont les principales motivations de ce travail. En effet, la conjonction de ces deux phénomènes amène à l'automatisation du processus de l'allocation de taxis dans le domaine du transport à la demande.

Dans le cadre d'un partenariat avec le constructeur Renault, nous étudions le paramétrage d'une flotte de taxis électriques pour répondre sans contrôle centralisé à des demandes de déplacement en milieu urbain et avec un coût de communication limité. En effet, les solutions commerciales de gestion de flottes de véhicules avec collecte de données en temps réel et exploitation, notamment dans le cloud, deviennent rapidement coûteuses selon le rythme de collecte et le volume de données à traiter. Par conséquent, développer une solution décentralisée reposant sur de la communication inter-véhiculaire (VANET) avec des performances équivalentes présente un intérêt économique en plus d'un intérêt scientifique.

La modélisation et le développement de systèmes décentralisés est au cœur des travaux multiagents. Le problème d'allocation de taxis peut donc bénéficier de ces recherches comme ce fut le cas pour (Egan, Jakob, 2016; Grau, Romeu, 2015; Maciejewski, Nagel, 2013; Santani *et al.*, 2008; Shen, Lopes, 2015). De plus, si l'objectif est une allocation optimale, les travaux issus du domaine de l'optimisation sous contraintes distribuée constituent une approche pertinente. Nous proposons donc un modèle multiagent du problème d'allocation pour une flotte de taxis (*taxi swarm allocation problem*, ou TSAP). Il s'agit d'une instanciation du modèle d'allocation de ressources en temps réel (*online resource allocation model*, ou OLRA (Zargayouna *et al.*, 2016)) avec l'ajout de contraintes sur la communication (globale ou P2P). Nous étudions les bénéfices et les inconvénients d'une stratégie décentralisée par rapport à une solution centralisée pour résoudre ce problème d'allocation distribuée et dynamique de ressources.

Notre objectif est de concevoir un processus multiagent d'allocation décentralisé capable de rivaliser avec une solution centralisée, tout en palliant l'absence de connaissance sur les demandes (aucune prévision sur l'échéance et la localisation des requêtes) et de communication globale.

Nous commençons par la présentation du problème d'allocation de taxi en section 2 et le traduisons en une série de programmes linéaires en nombres entiers à résoudre, comme fait classiquement dans une allocation de ressources. Nous présentons la contribution principale de ce papier, le modèle multiagent, en section 3, et évaluons ce modèle (section 4) en utilisant le simulateur dédié que nous avons développé. Nous discutons les travaux liés et les approches classiques en section 5, avant de conclure sur quelques perspectives en section 6.

2. Problème d'allocation de taxi (TSAP)

Dans le scénario considéré (cf. figure 1), les demandes (r_j) sont générées de manière non déterministe ¹ par des sources émettrices (S_k) : une source est choisie pour être aléatoirement et ponctuellement sur-représentée durant le scénario comme origine ou destination des demandes au détriment des autres sources. Les taxis (v_i) sont mobiles, distribués et communiquent via un réseau inter-véhiculaire ad-hoc (VANET, représenté en bleu et rouge) ou par un réseau cellulaire (4G, non représenté comme réseau global) pour répondre aux demandes. Les taxis peuvent échanger des informations sur les demandes qu'ils connaissent, leurs statuts ou leurs décisions. L'objectif principal est de trouver une allocation de taxis aux demandes tout en minimisant les coûts opérationnels et en maximisant la qualité des services rendu aux utilisateurs.

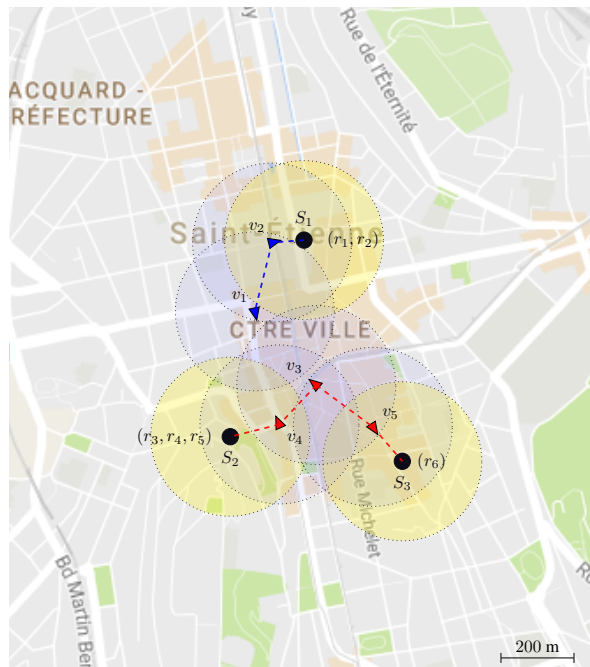


Figure 1. Un problème d'allocation de taxis au temps t : les taxis (v_i) ne peuvent interagir (segments pointillés) que lorsqu'ils sont dans la même sphère de communication (cercles pointillés); les demandes (r_j) sont émises aux sources (S_k)

2.1. Définition du problème

Nous considérons une flotte de taxis notée \mathcal{A} qui se déplace dans une ville représentée par une carte comme définie ci-dessous.

1. L'hypothèse est que nous n'avons pas de modèle probabiliste de la distribution de ces demandes.

DÉFINITION 1. — Une carte est le n -uplet $\langle \mathcal{V}, \mathcal{E}, \mathcal{D}, \mathcal{S} \rangle$, où \mathcal{V} est un ensemble de nœuds, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ est un ensemble d'arcs, $\mathcal{D} = \{d_{ij} \in \mathbb{R}_+ \mid i, j \in \mathcal{V}, \exists e_{ij} \in \mathcal{E}, i \neq j\}$ est l'ensemble des distances entre les nœuds connectés, et $\mathcal{S} \subseteq \mathcal{V}$ est un ensemble de sources.

Une carte est constituée d'un graphe $(\mathcal{V}, \mathcal{E})$ étiqueté par les distances (\mathcal{D}) entre nœuds dont un sous-ensemble, les sources (\mathcal{S}), sont identifiées et représentent les origines et destinations des voyageurs. Sur cette carte des taxis sont positionnés sur des nœuds. Un taxi se déplace de manière discrète de nœud en nœud, i.e sa position sur un arc est celle du nœud i d'origine de l'arc tant qu'il n'a pas atteint le nœud j de destination de l'arc. À partir d'une carte nous définissons différentes fonctions relatives aux déplacements. Ainsi si $i, j \in \mathcal{V}$ alors $dist(i, j)$ retourne la longueur du chemin le plus court de i à j et $travel(i, j, t)$ sa durée au temps t , $\|i - j\|$ est la distance euclidienne séparant les deux nœuds.

Les taxis répondent à des demandes et nous notons \mathcal{R} l'ensemble de ces demandes.

DÉFINITION 2. — Une demande $r \in \mathcal{R}$ est définie par la source de son origine ($origine : \mathcal{R} \rightarrow \mathcal{S}$) et de sa destination ($destination : \mathcal{R} \rightarrow \mathcal{S}$) ainsi qu'une fenêtre temporelle de validité ($tw : \mathcal{R} \rightarrow [T, T]$).

La fonction $origine(r)$ (resp. $destination(r)$) fournit le nœud source (resp. destination) de la demande r . Pour simplifier la présentation, les valeurs minimales et maximales de la fenêtre temporelle $tw(r)$ d'une demande r sont respectivement notées $tw_{min}(r)$, $tw_{max}(r)$.

Cette fenêtre temporelle définit l'intervalle dans lequel la demande doit être satisfaite, comme dans le problème classique Demand-Responsive Transportation systems (DRTS) (Horn, 2002). Une demande est dite *active* au temps t si t est dans la fenêtre de temps et qu'elle n'a pas été *satisfaite*. La demande devient satisfaite si un véhicule se présente pour prendre en charge le client à la source origine de la demande et dans la fenêtre temporelle. Une demande jamais satisfaite dans sa fenêtre temporelle est dite *échouée*. La seule contrainte considérée étant la prise en charge du client, il n'y a pas de fenêtre temporelle pour le dépôt.

Les demandes sont émises via l'infrastructure de communication qui peut être globale (e.g. réseau cellulaire) ou locale. Dans ce dernier cas, une demande est émise depuis la source qui est son origine et est propagée de proche en proche selon les modalités d'un VANET par les taxis qui la reçoivent.

DÉFINITION 3. — Un taxi $v \in \mathcal{A}$ est caractérisé par sa destination courante ($dest : \mathcal{A} \times T \rightarrow \mathcal{V}$) et un rayon de communication fixé² ($rng : \mathcal{A} \cup \mathcal{S} \rightarrow \mathbb{R}_+$) dépendant de l'équipement de communication. Celui-ci n'évolue pas dans le temps.

2. Le rayon de communication est également utilisé pour les sources.

La fonction $dest(v, t)$ donne le nœud de destination du taxi v au temps t . Si $dest(v, t) \notin \mathcal{S}$ alors on considère que le taxi se déplace au hasard et non en fonction des demandes.

Afin d'obtenir le nœud où est positionné une requête ou un taxi, nous utilisons la fonction pos et la fonction $posTravel$ pour obtenir la position d'un taxi en fonction du temps et d'une direction (définition 4).

DÉFINITION 4. — *La fonction $pos : (\mathcal{A} \cup \mathcal{R}) \times T \rightarrow \mathcal{V}$ retourne la position fixe de la source origine d'une demande ou la position d'un taxi en fonction du temps. La fonction $posTravel : \mathcal{A} \times \mathcal{V} \times \mathbb{N} \times T \rightarrow \mathcal{V}$ retourne la position atteignable par un taxi en un temps donné sur le plus court chemin pour atteindre un nœud.*

Ainsi $pos(v_i, t)$ est le nœud où le taxi v_i est positionné au temps t , et le nœud où le taxi v_i sera positionné sur le plus court chemin pour atteindre j après une durée d s'il part à l'horaire t est noté $posTravel(v_i, j, d, t)$. Le déplacement des taxis étant discret, i.e. un arc par cycle, ce temps est calculé comme le nombre d'arcs. Si le temps nécessaire pour parcourir le chemin jusqu'à j est inférieur à la durée d , alors la fonction retourne le nœud destination j .

Selon la définition 5, sources et taxis peuvent communiquer de manière bidirectionnelle s'ils sont suffisamment proches. Selon nos hypothèses, une source ne pourra être qu'émettrice des demandes des voyageurs dont elle est l'origine alors qu'un taxi, sera émetteur et récepteur des différents messages échangés. Ainsi en cas de communication par VANET, les taxis servent de relais à l'échange d'information mais pas les sources. C'est la raison pour laquelle dans la définition 6 seuls les taxis assurent la connexité par transitivité.

DÉFINITION 5. — *Deux composants (taxi ou source) sont connectés à un instant t s'ils sont situés dans la plus petite sphère de communication centrée sur l'un des deux composants.*

$$i, j \in \mathcal{S} \cup \mathcal{A}, connected(i, j, t) = \begin{cases} 1, & \text{si } ||pos(i, t) - pos(j, t)|| \\ & \leq \min(rng(i), rng(j)) \\ 0, & \text{sinon} \end{cases} \quad (1)$$

DÉFINITION 6. — *Un ensemble connecté est l'ensemble des sources et taxis connectés directement ou par transitivité à une source ou un taxi i à l'instant t :*

$$CSet(i, t) = \{j \in \mathcal{S} \cup \mathcal{A} \mid connected(i, j, t)\} \\ \cup \{j \in \mathcal{S} \cup \mathcal{A} \mid \exists k \in CSet(i, t) \cap CSet_{\mathcal{A}}(j, t), k \neq i \wedge k \neq j\}$$

Notons $CSet_{\mathcal{A}}(j, t) = CSet(j, t) \cap \mathcal{A}$ la restriction de $CSet(j, t)$ aux taxis, et $CSet_{\mathcal{A}}(t) = \{CSet_{\mathcal{A}}(i, t) \mid i \in \mathcal{A}\}$ l'ensemble de ces ensembles connectés au temps t .

Remarquons ainsi qu'une source peut apparaître dans plusieurs ensembles connectés distincts, sans les fusionner, puisque la source n'assure pas la transitivité. Cepen-

dant, en cas de rayon de communication infini, il n'y a qu'un seul ensemble connecté : l'ensemble de tous les taxis et de toutes les sources.

EXEMPLE 7. — Sur la figure 1, la carte contient le graphe représentant la ville de Saint-Etienne et trois sources (S_1 à S_3) qui émettent des demandes (r_1 à r_6). Cinq taxis (v_1 à v_5) sont déployés dans la ville pour répondre aux demandes. L'infrastructure de communication choisie est un VANET. Les taxis et sources ont donc un rayon de communication limité (cercle en pointillé). Par conséquent, au temps t deux ensembles connectés sont identifiés ($CSet(v_1, t) = \{v_1, v_2, S_1\}$ et $CSet(v_3, t) = \{v_3, v_4, v_5, S_2, S_3\}$). \square

À partir des concepts et définitions introduits, le problème considéré dans ce papier est défini comme suit :

PROBLÈME 8 (TSAP). — Le *problème d'allocation de Taxi* (TSAP) est un tuple $\langle \langle \mathcal{V}, \mathcal{E}, \mathcal{D}, \mathcal{S} \rangle, \mathcal{R}, \mathcal{A}, cost, qos, [t_{start}, t_{end}] \rangle$ consiste en une affectation continue des taxis de \mathcal{A} (incluant les taxis transportant des passagers) à de nouvelles demandes dans \mathcal{R} , pouvant se déplacer sur \mathcal{M} , tout en minimisant les coûts (donnés par la fonction $cost$) et en maximisant les services aux utilisateurs (donnés par la fonction qos), pour une période donnée $[t_{start}, t_{end}]$, étant donnée une infrastructure de communication. \square

Dans notre étude, les coûts $cost$ correspondent à la *distance* totale parcourue à vide par les taxis ; le service aux utilisateurs qos est évalué par le ratio entre le nombre des demandes servies et le nombre total de demandes et également par le temps moyen d'attente. La période de temps considérée est la période de travail d'un taxi sur une journée (matin au soir).

Le premier problème à considérer dans le cadre de TSAP est le manque de connaissance sur les demandes à venir. C'est la raison pour laquelle le problème consiste en l'affectation continue des taxis aux nouvelles demandes. De manière évidente, il n'est pas possible de définir un plan optimal pour les taxis sur la fenêtre temporelle. Ainsi, le système devra réagir et résoudre les sous problèmes à chaque modification des demandes.

PROBLÈME 9 (TSAP(t)). — Un TSAP(t) consiste en une affectation des demandes actives à un ensemble de taxis de TSAP, au temps $t \in [t_{start}, t_{end}]$. \square

L'idée ici est de résoudre TSAP(t) à chaque modification de la demande et de l'offre de transport. Ainsi à chaque valeur de t , des demandes peuvent ne plus avoir à être considérées car échouées ou satisfaites et de nouvelles demandes peuvent apparaître dans le système. Pour une demande donnée, la composition de l'offre, i.e. l'ensemble connecté de véhicules informés de cette demande, peut être modifiée et donc nécessiter la détermination d'une nouvelle allocation. Notons qu'à la fin d'une période de temps de TSAP, résoudre chaque TSAP(t) peut ne pas conduire à l'optimal théorique. De plus, les critères d'optimisation de TSAP ne peuvent pas être dérivés directement pour un TSAP(t), puisque ce dernier concerne un instant. Notons qu'il est possible de changer l'affectation d'un taxi libre à des demandes tant que ce taxi n'a pas chargé

ses clients, comme dans (Maciejewski, Nagel, 2013 ; Glaschenko *et al.*, 2009 ; Jin, Jie, 2012).

Un autre problème à considérer en résolvant TSAP est que l'infrastructure de communication peut ne pas assurer une connaissance identique entre les taxis. Ainsi en cas de communication par VANET, des taxis dans des ensembles connectés différents n'ont pas les mêmes informations. Les demandes d'une source ne sont connues que des taxis auxquels elle est connectée et les taxis ne peuvent interagir qu'avec les taxis auxquels ils sont connectés. L'approche choisie dans ce papier est de proposer des mécanismes multiagents pour les taxis afin d'échanger les informations utiles à leur coordination pour trouver des solutions aux TSAP(t), dans des configurations totalement décentralisées.

2.2. Modéliser TSAP(t) en programme linéaire en nombres entiers

Comme il est classique de procéder pour l'allocation de ressources, TSAP(t) peut être modélisé comme un problème de programmation linéaire en nombres entiers binaires (*PL en 0-1*). Soit v_{ij}^t une variable représentant la décision binaire d'un taxi i d'avoir pour prochaine destination l'origine de la demande j ($v_{ij}^t = 1$) dans les autres cas, i.e. origine d'une autre demande ou encore un déplacement aléatoire, ($v_{ij}^t = 0$) au temps t .

PROBLÈME 10 (*PL en 0-1-TSAP(t)*). — Selon le formalisme *PL en 0-1*, TSAP(t) est défini ainsi :

$$\underset{v_{ij}^t}{\text{minimiser}} \quad \sum_{v_{ij}^t} c_{ij}^t \cdot v_{ij}^t \quad (2)$$

avec

$$\forall i \in \mathcal{A} \quad \sum_{j \in \mathcal{R} \cup \{\emptyset\}} v_{ij}^t = 1 \quad (3)$$

$$\forall j \in \mathcal{R} \quad \sum_{i \in \mathcal{A}} v_{ij}^t \leq 1 \quad (4)$$

□

Alors qu'intrinsèquement TSAP(t) est multi-objectif, nous le transformons en un problème d'optimisation mono-objectif (2), où le coût opérationnel et le critère de service sont tous les deux encapsulés dans la fonction de coût c_{ij}^t du taxi i pour la requête j au temps t . La contrainte (3) assure que chaque taxi décide de son prochain nœud et la contrainte (4) assure qu'une demande ne peut pas être allouée à plus d'un taxi afin d'éviter que deux taxis ne répondent à la même demande.

Nous pouvons noter que d'autres écritures non binaires sont possibles pour cette modélisation de TSAP(t) avec des solutions équivalentes mais une charge en mémoire et en temps de calcul différentes. Par exemple :

$$\underset{v_i^t}{\text{minimiser}} \quad \sum_{v_i^t} c_i^t(v_i^t) \quad (5)$$

$$\text{avec AllDiff}(v_1^t, \dots, v_{|\mathcal{A}|}^t) \quad (6)$$

où les variables v_i^t correspondent à la décision du taxi v_i au temps t et prennent une valeur dans $\mathcal{R} \cup \{\emptyset\}$, $c_i^t(v_i^t)$ donne alors le coût de cette décision pour le taxi v_i au temps t . **AllDiff** vérifie que les décisions des taxis sont différentes (sans considérer le non choix d'une destination de l'origine d'une demande).

Ces différentes modélisations du même problème (comme les deux présentées ci-dessus) impliquent l'usage de différents algorithmes et méthodes, plus ou moins efficaces en fonction de la topologie du graphe de contraintes, comme nous le discuterons dans la section 4.

Classiquement, les systèmes de gestion de taxi résolvent ce problème soit à la main (par un opérateur humain, appelé dispatcheur) soit par un logiciel du marché, comme CPLEX. Bien que simple dans sa formulation, de tels problèmes appartiennent à la famille des problèmes NP-difficile. Les systèmes actuels centralisés sont capables de résoudre de tels problèmes en un temps raisonnable mais ne passent pas à l'échelle d'une ville avec des milliers de demandes et de taxis. Le problème doit être relâché en assignant par exemple des zones aux taxis ou en interdisant de remettre en cause les allocations avec pour conséquence une allocation sous optimale des nouvelles demandes. Notre objectif est de concevoir un modèle multiagent d'allocation décentralisée capable de rivaliser avec une solution centralisée, tout en palliant l'absence de connaissance sur les demandes et de communication globale.

3. Modèle multiagent

Dans cette section, nous optons pour une approche multiagent pour résoudre TSAP. Nous analysons différentes solutions de coordination qui se différencient selon la manière dont les décisions sont coordonnées et l'infrastructure de communication requise.

3.1. Comportement générique d'un taxi

Un taxi exécute en continu une boucle avec à chaque cycle les instructions suivantes :

1. lecture des messages entrants (concernant les autres taxis, les demandes, etc.);
2. mise à jour de ses croyances à propos des demandes et taxis;
3. décision de la prochaine destination;

4. déplacement vers la prochaine position en direction de la destination ;
5. émission de messages sur son état et transmission des messages concernant les demandes et autres taxis.

A chaque fois qu'un taxi disponible (i.e. étant libre de prendre une course à ce moment) obtient de nouvelles informations à propos d'une demande (e.g. transmise par un autre taxi, ou reçu d'un portail centralisé), il met à jour ses croyances. Nous notons $KR(v_i, t)$ l'ensemble des demandes connues à l'instant t par le taxi v_i et $KT(v_i, t)$ l'ensemble des taxis connectés avec v_i à l'instant t . Par extension, nous notons également $KR(C, t) = \bigcup_{v_i \in C} KR(v_i, t)$, l'ensemble des demandes connues par un ensemble de taxis C . Ainsi au sein d'un ensemble connecté, les taxis alignent leur connaissance. Avec ses croyances, un taxi doit décider de son déplacement (étape 3). Ce déplacement peut être :

1. en direction d'une source pour satisfaire une demande ;
2. en direction d'une autre position sur le réseau, pour par exemple anticiper l'arrivée de nouvelles demandes.

La décision est prise selon des critères concernant les demandes ($\kappa : \mathcal{A} \times \mathcal{R} \times T \rightarrow]0, 1]$) qui assignent à chaque demande connue une utilité et qui dépend des choix du concepteur du modèle d'affectation. Nous avons choisi que nos agents soient coopératifs et les critères décrits en section 4.1.1 intègrent la prise en compte du collectif. Une demande non satisfiable par un taxi, i.e. le taxi ne peut atteindre la source origine avant la fin de la fenêtre temporelle, a une utilité fixée à 0. Selon la nature et l'information utilisée dans ces critères, le comportement d'un agent peut être plus ou moins coopératif. Ces critères correspondent à la manière d'évaluer les coûts c_{ij}^t dans $PL \text{ en } 0-1\text{-TSAP}(t)$. La littérature sur le sujet montre principalement l'usage de critères fondés sur la distance ou le temps comme la demande la plus urgente ou la plus proche mais sans échange d'informations additionnelles entre les taxis (Alshamsi *et al.*, 2009 ; Egan, Jakob, 2016 ; Glaschenko *et al.*, 2009 ; Maciejewski, Nagel, 2013 ; Santani *et al.*, 2008 ; Shen, Lopes, 2015).

3.2. Processus d'allocation

Le processus d'allocation pour résoudre le $PL \text{ en } 0-1\text{-TSAP}(t)$ est à réaliser par les taxis eux mêmes ou éventuellement un dispatcheur. Selon la manière dont les décisions sont prises, le processus peut être plus ou moins coordonné. A présent, nous présentons les différentes stratégies de coordination que nous considérons.

1. **Coordination centralisée (c-alloc)** : les agents ne reçoivent pas les demandes mais des ordres de déplacement d'un dispatcheur (comme dans les approches classiques non multiagents) en utilisant une infrastructure globale de communication – i.e. à chaque temps t le dispatcheur collecte les informations pour calculer les coûts, puis résout directement $PL \text{ en } 0-1\text{-TSAP}(t)$ (voir section 2.2) et informe les taxis de la solution.

2. **Coordination via un portail (p-alloc)** : Les taxis accèdent aux demandes via un portail et prennent eux mêmes leur décision en choisissant la requête qui maximise leur utilité (fonction κ). La coordination est réalisée indirectement par la réservation des demandes sur ce même portail. La communication est globale et les conflits sont évités en ne permettant qu'une seule réservation par demande et en interdisant la remise en cause des décisions. Les taxis ne communiquent pas d'informations entre eux (uniquement avec le portail).

3. **Coordination par un DCOP (d-alloc)** : L'environnement diffuse les demandes selon un modèle P2P et les agents prennent leur décision par eux-mêmes. Afin d'éviter les conflits dans un ensemble connecté les agents se coordonnent selon une approche DCOP. Par construction, deux agents dans deux ensembles connectés distincts ne peuvent pas communiquer et il peut donc y avoir des conflits d'affectation d'une même demande entre ensembles connectés (voir section 3.3).

La seule manière d'assurer une allocation optimale pour PL en $0-1$ -TSAP(t) est de centraliser la prise de décision avec une communication globale (cas 1). Les alternatives peuvent conduire à une allocation sous-optimale et le cas 3 peut mener à des conflits résultats d'une coordination incomplète et/ou d'informations manquantes.

3.3. Coordination par un DCOP

Dans cette section, nous nous focalisons sur l'implémentation d'une solution DCOP (**d-alloc**) pour résoudre PL en $0-1$ -TSAP(t) à chaque fois qu'une nouvelle information est obtenue. Ce protocole est réalisé entre taxis au sein d'un même ensemble connecté. Afin d'évaluer le coût pour atteindre l'origine d'une demande, les taxis partagent des informations sur les demandes et les autres taxis.

PL en $0-1$ -TSAP(t) est transformé en un DCOP (distributed constraint optimization problem) $\langle A, X, D, C \rangle$ (Cerquides *et al.*, 2014). Nous optons ici pour un encodage binaire de notre problème, l'obtention de bonnes performances ayant été prouvée pour des problèmes similaires (Pujol-Gonzalez *et al.*, 2013 ; 2015). Les variables dans X sont toutes binaires et ainsi tous les domaines dans D sont réduits à $\{0, 1\}$. Chaque taxi $v_i \in A$ a ses propres variables de décision (v_{ij}^t) correspondant à chaque demande $r_j \in KR(v_i, t)$ connues au temps t , plus la décision vide (\emptyset). L'ensemble de contraintes C contient les coûts de l'équation (2) et les contraintes (3) à (4). Comme indiqué précédemment, plusieurs fonctions de coût peuvent être considérées. Ici, les coûts unaires c_{ij}^t sont dérivés du critère de décision κ :

$$c_{ij}^t(d) = \begin{cases} \frac{1}{\kappa(v_i, r_j, t)}, & \text{si } d = 1 \\ 0, & \text{sinon} \end{cases} \quad (7)$$

Le coût est donc l'inverse de la fonction objectif si la décision du taxi v_i est de choisir la requête r_j au temps t ($d = 1$) et 0 si r_j n'est pas choisie.

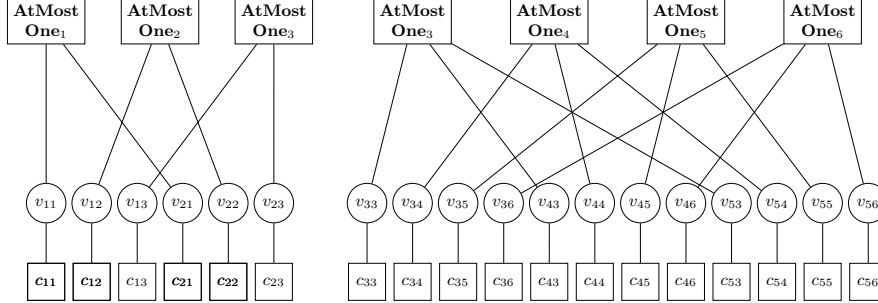


Figure 2. Exemple de modèle DCOP pour un $TSAP(t)$, avec deux composants connectés (un pour chaque ensemble connecté) concernés par la demande r_3

La contrainte (3) impose que tous les agents fassent au moins une décision. Comme les agents peuvent décider de ne choisir aucune requête (\emptyset), cette contrainte est toujours vérifiée. Comme, il n'y a pas de maximum à la fonction de coût qui agirait comme un veto, la décision d'un taxi ne peut être \emptyset que s'il y a plus de taxis que de requêtes atteignables, i.e dont le temps du trajet pour atteindre l'origine ne dépasse pas la valeur maximale de la fenêtre temporelle.

La contrainte (4) est reformulée en contrainte (8) du type "Au plus un", qui est connue comme étant un *tractable higher order potential* (THOP), pouvant être évalué en temps linéaire (Pujol-Gonzalez *et al.*, 2015):

$$\mathbf{AMO}_j^t(v_{1j}^t, \dots, v_{|\mathcal{A}|j}^t) = \begin{cases} 0, & \text{if } \sum_{i \in \{1 \dots |\mathcal{A}|\}} v_{ij}^t \leq 1 \\ +\infty, & \text{otherwise} \end{cases} \quad (8)$$

Dans notre problème, pour chaque ensemble connecté il y a autant de contraintes **AMO** que de r_j demandes connues par les taxis de l'ensemble connecté considéré. Les taxis ne peuvent se coordonner qu'avec les taxis auxquels ils sont connectés, par conséquent tous les agents d'un même ensemble sont connectés aux mêmes contraintes **AMO**. La figure 2 illustre le graphe de contrainte d'un tel DCOP.

Au final, L'objectif de ce DCOP est le suivant :

$$\underset{v_{ij}^t}{\text{minimiser}} \left(\sum_{v_{ij}^t} c_{ij}^t(v_{ij}^t) + \sum_{\substack{C \in CSet_{\mathcal{A}}(t) \\ j \in [1, |KR(C,t)|]}} \mathbf{AMO}_j^t(v_{1j}^t, \dots, v_{|C|j}^t) \right)$$

Notons que deux ensembles connectés peuvent être informés d'une même demande selon leur positionnement respectifs vis-à-vis des sources. Dans un tel cas, chaque ensemble connecté doit satisfaire une contrainte **AMO** différente pour la même demande. C'est par exemple le cas pour la demande r_3 de l'exemple suivant.

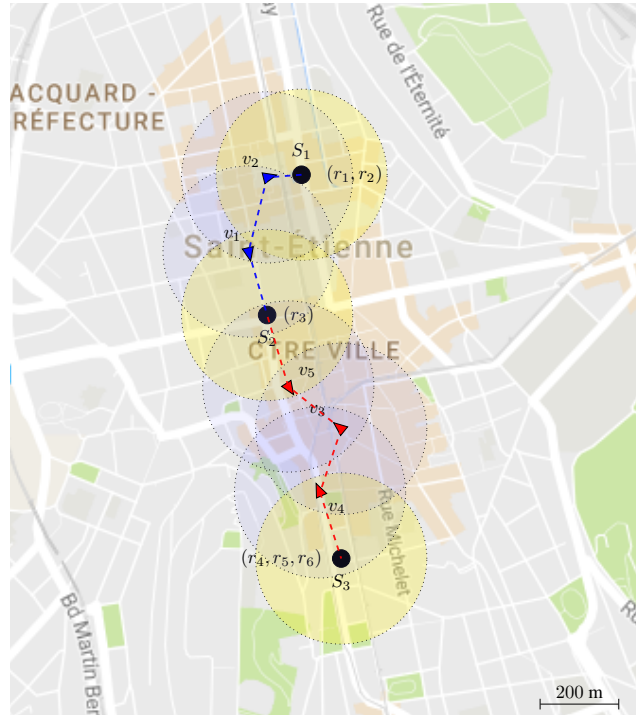


Figure 3. Un TSAP(t), avec une demande partagée entre deux ensembles connectés

EXEMPLE 11. — Soit la carte avec trois sources S_1 (émettant r_1 et r_2), S_2 (émettant r_3) et S_3 (émettant r_4 , r_5 et r_6), illustrée en figure 3. Supposons les deux ensembles connectés suivants $CSet(v_1, t) = \{v_1, v_2, S_1, S_2\}$ et $CSet(v_3, t) = \{v_3, v_4, v_5, S_2, S_3\}$ (et rappelons que S_1 peut être présente dans deux ensembles connectés distincts, voir définition 5). Le modèle DCOP correspondant est illustré dans la figure 2. \square

La conséquence est un potentiel conflit avec l'allocation de deux taxis différents pour la même demande. Sans communication globale et donc la considération d'un seul ensemble connecté, de tels conflits ne peuvent être évités.

Il existe de nombreux algorithmes dans la littérature pour résoudre de tels DCOPs. L'encodage binaire que nous avons choisi a montré de bonnes performances sur des problèmes similaires en utilisant l'algorithme Binary MaxSum (BMS) (Pujol-Gonzalez *et al.*, 2015). Cependant, cette approche ne garantit pas la convergence. Dans le cas d'un non encodage binaire, avec équations (5) et (6), les composants connectés du graphe de facteurs deviennent des arbres (une variable de décision par taxi, toutes connectés à une contrainte **AllDiff** pour chaque ensemble connecté). Ici, tout algorithme d'inférence comme DPOP ou le classique MaxSum convergera vers une solution optimale avec une charge de communication et un temps d'exécution faible. Finalement, une autre alternative est de coordonner les agents de chaque ensemble connecté avec un agent

unique qui calcule la solution optimale à PL en $0-1$ -TSAP(t) à partir des coûts collectés auprès des autres agents puis qui leur diffuse le résultat.

Nous avons implanté l'algorithme MaxSum sur encodage binaire ou non binaire, ainsi que l'approche où un seul agent calcule la solution (pour comparaison), en encodage binaire. Les trois approches fournissent des solutions équivalentes en qualité, mais la plus rapide est, de loin, la troisième solution. Ce sont donc les résultats obtenus avec cette dernière qui seront présentés dans la section suivante.

4. Évaluations

Dans cette section, nous évaluons les performances des trois stratégies de coordination décrites en section 3.2, dans différentes configurations (nombre de taxis, rayon de communication), en utilisant une simulation à temps discret.

Les simulations sont réalisées par un simulateur développé en Java dont les principales fonctions sont d'assurer 1) le déplacement des taxis sur une représentation discrète du réseau de transport, 2) la génération des requêtes par source selon un modèle rendant compte du scénario choisi, 3) diffuser les informations dans un réseau inter-vehiculaire. A chaque cycle, le simulateur génère les requêtes, diffuse les requêtes et communications entre agents selon leur proximité et exécute aléatoirement les taxis qui décident de leur prochaine position. Un démonstrateur est disponible en ligne ³.

4.1. Configuration des expérimentations

Les paramètres utilisés lors de nos expérimentations sont résumés dans le tableau suivant :

Paramètre	Valeur(s)
nombre d'agents $ \mathcal{A} $	{8, 10, 12, 14, 16, 18}
nombre de sources $ \mathcal{S} $	3
réseau routier	Saint-Étienne (centre)
nombre de nœuds $ \mathcal{V} $	4645
nombre d'arcs $ \mathcal{E} $	503
longueur du réseau $\sum_{d_{ij} \in \mathcal{D}} d_{ij}$	198km
nombre de cycles	3000
temps simulé par cycle	5 secondes
vitesse moyenne des taxis	30 km/h
rayon de communication rng	{50, 100, 200, 250, 300, 350, 400}
requêtes	0 à 2/cycle, plus 0 à 10/100 cycles (cf. texte)
coordination	{c-alloc, p-alloc, d-alloc}

La carte de la ville de Saint-Étienne a été choisie pour la simulation (cf. figure 4). Le réseau est composé de 503 arcs avec 4645 points. Le nombre de points par arc dépend

3. <http://territoire.emse.fr/applications/swarm/stetienne>

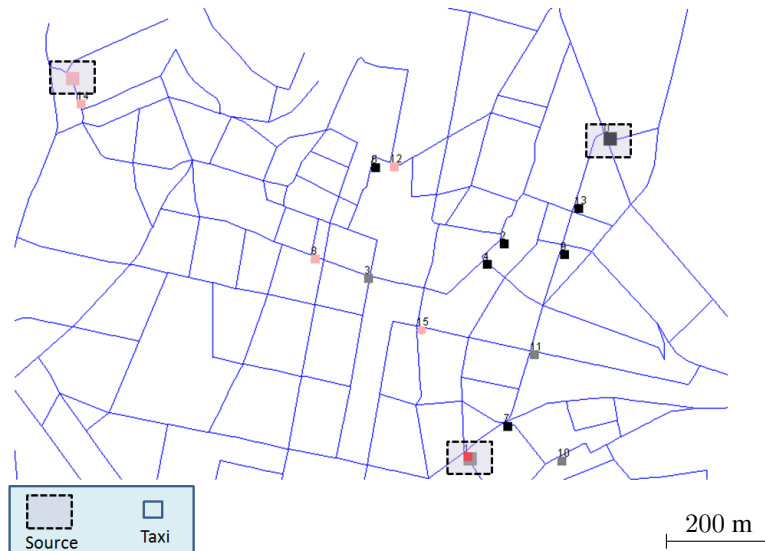


Figure 4. Plan de Saint-Etienne, issu de données OSM, utilisé dans les simulations

de l'information enregistrée dans le fichier Open Street Map (OSM) décrivant la ville de Saint-Etienne. La distance totale du réseau est de 198 km et la distance moyenne entre deux points est de 43 mètres. La position des sources S_1 , S_2 et S_3 est fixée. Ce n'est pas un des paramètres étudié. Les sources sont réparties sur la carte pour pouvoir comparer différents types de courses : longue ou courte distance. La distance entre les sources (calculées en kilomètres à partir de leur coordonnées longitude et latitude) est approximativement de 0,8, 1,5 et 2. La durée d'un trajet est respectivement 19, 36 et 48 cycles. Les taxis se déplacent d'un point à un autre sur le même arc à chaque cycle de simulation. La vitesse moyenne des taxis est de 30 km/h. Un cycle est équivalent à 5 secondes.

Nous utilisons le modèle de distribution de requêtes suivant. À chaque cycle de simulation, 0 à 2 demandes sont générées aléatoirement. Pour chacune d'elles, les sources origine et destination sont générées aléatoirement en suivant la loi espace-temps : les demandes sont concentrées sur la source S_1 de l'espace. Un cycle sur deux (choisi uniformément et de manière aléatoire) l'origine d'une nouvelle demande est S_1 et quelconque sinon. La concentration des demandes varie également au cours du temps (tous les 100 cycles de simulation, un pic de demandes est généré). Le nombre de demandes augmente de manière aléatoire entre 0 et 10 et leur origine est fixée à S_1 . La fenêtre temporelle des demandes est définie comme suit : tw_{min} est initiée avec la valeur du cycle courant et tw_{max} est initié avec la valeur minimale à laquelle est ajoutée une valeur aléatoire dans $[[70, 140]]$. Cet intervalle a été défini à partir de la durée d'une course moyenne (36 cycles), cela permet à un véhicule en course de considérer en moyenne sa prochaine demande à une autre source que celle où il doit

déposer son passager. Une valeur plus grande risque de rendre le nombre d'échec moins significatif, la satisfaction des demandes devenant plus facile.

4.1.1. Fonction de coût

Comme mentionné précédemment, résoudre *PL en 0-1-TSAP(t)* par les approches **c-alloc**, **p-alloc** ou **d-alloc** nécessite de définir les fonctions de coût (c_{ij}^t 's) en s'appuyant sur le critère de décision (κ). Afin de pouvoir comparer les résultats, nous utilisons les mêmes mesures pour chaque stratégie : une mesure fondée sur le temps (κ_{coop}^{time}) et une autre pour l'espace (κ_{coop}^{space}) puis le critère noté κ_{coop} qui combine ces mesures.

Les taxis sont coopératifs dans l'évaluation des différentes demandes. Cela signifie qu'un taxi préfère une demande a une autre si son non choix est mauvais pour le système, i.e. que la demande a plus intérêt à être réalisée par un autre taxi. De plus seuls les taxis libres évaluent les requêtes mais considèrent les états des autres taxis connectés qu'ils soient libres ou non. Pour la mesure spatiale, il s'agit de comparer les positions relatives des taxis vis à vis de la source origine de la demande. En milieu urbain, la notion de proximité est toute relative et dépend notamment du trafic routier ou encore de la signalisation. Par conséquent, nous associons la notion de distance à la notion de temps de parcours. Il faut donc qu'un taxi calcule comment il se positionne vis à vis d'une demande en considérant les autres taxis. Ainsi au temps t , $closerFree(v_i, r_j, t)$ est le nombre de taxis libres et plus proches (en temps de parcours) de r_j que v_i , et la fonction $closerRiding(v_i, r_j, t)$ est le nombre de taxis en course plus proche de r_j que v_i , tout en considérant le trajet à parcourir pour déposer leur client actuel à sa destination. au temps t , si v_i est libre alors $free(v_k, t) = 1$ et 0 sinon.

$$closerFree(v_i, r_j, t) = \sum_{v_k \in KT(v_i, t)} free(v_k, t).closer(pos(v_k, t), pos(v_i, t), origin(r_j), t)$$

$$closerRiding(v_i, r_j, t) = \sum_{v_k \in KT(v_i, t)} (1 - free(v_k, t)).closer(dest(v_k, t), nextPosition, origin(r_j), t + travel(pos(v_k, t), dest(v_k, t), t))$$

avec $nextPosition = posTravel(v_i, origin(r_j), travel(pos(v_k, t), dest(v_k, t), t))$ la position qu'aurait v_i en direction de l'origine de r_j au temps nécessaire à v_k pour atteindre la destination de sa course actuelle.

La fonction *closer* est définie $\forall x, y, z \in \mathcal{V}, t \in T$

$$closer(x, y, z, t) = \begin{cases} 1, & \text{si } x \neq y \text{ et } travel(x, z, t) \leq \\ & travel(y, z, t) \\ 0, & \text{sinon} \end{cases}$$

La mesure relative à la distance est donc l'inverse du nombre de taxis plus proches de l'origine de la demande. La mesure est maximale s'il y a aucun taxi plus proche.

$$\kappa_{\text{coop}}^{\text{space}}(v_i, r_j, t) = \frac{1}{closerFree(v_i, r_j, t) + closerRiding(v_i, r_j, t) + 1} \quad (9)$$

La mesure relative au temps considère comme meilleures les demandes qui ont une durée d'attente plus élevée si elles étaient satisfaites, i.e. selon le temps d'arrivée prévu pour le véhicule.

$$\kappa_{\text{coop}}^{\text{time}}(v_i, r_j, t) = \frac{1}{\sum_{r_k \in KR(v_i, t)} worst(pos(v_i, t), r_k, r_j, t) + 1} \quad (10)$$

La fonction *worst* retourne 1 si la requête r_1 (requête de comparaison (r_k)) dans la fonction 10) a un plus long temps d'attente que r_2 (requête de référence (r_j)) dans la fonction 10) et 0 sinon.

$$worst(x, r_1, r_2, t) = \begin{cases} 1, & \text{if } ((t + travel(x, pos(origin(r_1), t), t)) - tw_{min}(r_1)) \geq \\ & ((t + travel(x, pos(origin(r_2), t), t)) - tw_{min}(r_2))) \\ 0, & \text{otherwise} \end{cases}$$

Ces deux mesures sont agrégées dans la fonction κ_{coop}

$$\kappa_{\text{coop}}(v_i, r_j, t) = \alpha \cdot \kappa_{\text{coop}}^{\text{space}}(v_i, r_j, t) + (1 - \alpha) \cdot \kappa_{\text{coop}}^{\text{time}}(v_i, r_j, t) \quad (11)$$

avec $\alpha \in [0, 1]$ (ici 0.5), $\kappa_{\text{coop}}^{\text{space}}$ mesure préférant les demandes qui ont le moins de chance d'être servies tôt du fait du peu de taxis proches et $\kappa_{\text{coop}}^{\text{time}}$ mesure qui limite le temps d'attente des clients.

4.2. Résultats et analyse

En utilisant les configurations décrites ci-dessus, nous exécutons plusieurs simulations pour une période de 3 000 cycles de simulation (env. 4h de temps simulé).

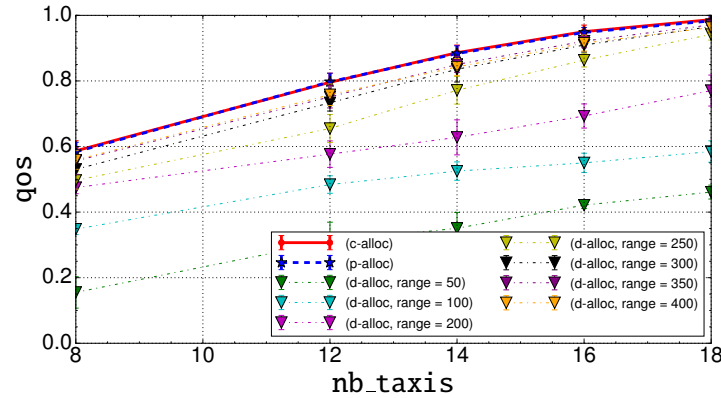


Figure 5. Qualité de service pour les différentes stratégies avec des rayons de communication pour la stratégie décentralisée en utilisant des interactions P2P, pour un nombre croissant de taxis

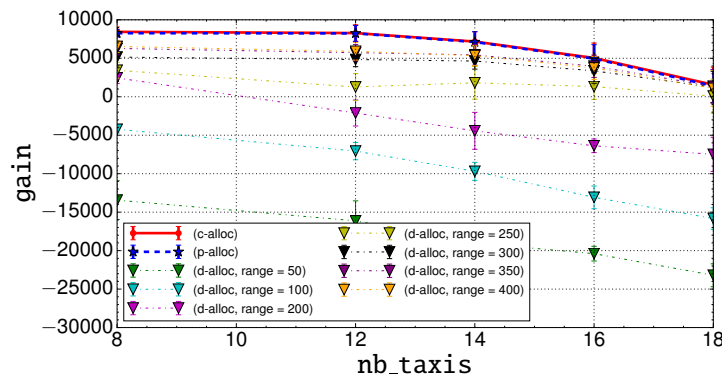


Figure 6. Gain pour les différentes stratégies avec des rayons de communication pour la stratégie décentralisée en utilisant des interactions P2P, pour un nombre croissant de taxis

Regardons en premier la qualité de service (qos) des différentes stratégies, représentée dans la figure 5, qui est le pourcentage de demandes satisfaites. Rappelons qu'une demande est satisfaite si un taxi est à la source origine de la demande avant la fin de la fenêtre temporelle de la demande avec l'objectif de réaliser la course pour le client. Notre objectif est de comparer le nombre de taxis nécessaires pour chaque stratégie afin d'obtenir au moins 90%. Cet objectif est atteint par **c-alloc** et **p-alloc** avec une flotte composée d'au moins 15 taxis. **d-alloc** nécessite 16 taxis et un rayon de communication de 300 ou plus pour la même performance.

La figure 6 représente le gain résultant pour les différentes stratégies, i.e. la différence entre la distance totale des courses réalisés par la flotte et la distance totale

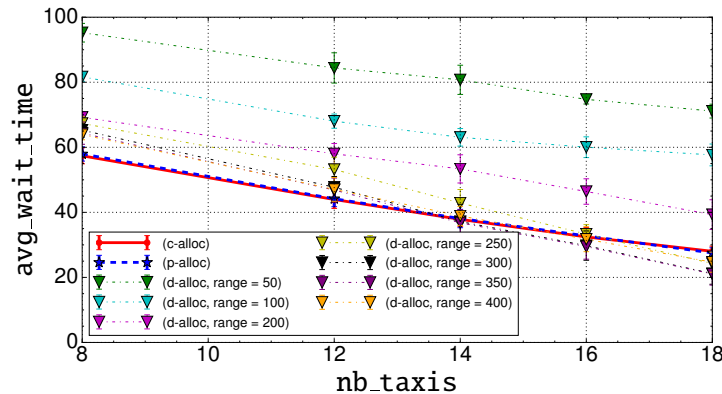


Figure 7. Temps moyen d'attente d'un client pour les différentes stratégies avec des rayons de communication pour la stratégie décentralisée en utilisant des interactions P2P, pour un nombre croissant de taxis

parcourue à vide. Le plus haut est la valeur, meilleur est le résultat puisqu'un taxi rapporte plus à la compagnie. Le gain décroît avec le nombre de taxis puisque à niveau de demande égal, il y a plus de déplacements à vide si l'offre s'approche du niveau de demande optimal. **c-alloc** et **p-alloc** atteignent exactement le même niveau alors que **d-alloc** reste en dessous quel que soit le rayon de communication. Cependant, à partir d'un rayon de 250 les résultats sont équivalents, ce qui signifie que ce rayon est suffisant pour une bonne coordination.

La figure 7 présente le temps moyen passé par les clients à attendre que leurs demandes soient prises en compte. Nous pouvons remarquer que les stratégies décentralisées avec un rayon plus grand que 250 obtiennent de meilleures performances que les stratégies centralisées ou basées sur un portail – elles atteignent l'équivalent mais avec une moins bonne qualité de service.

Ces trois premiers indicateurs sont focalisés sur la qualité des solutions obtenues par les différentes stratégies. Nous constatons une supériorité des approches centralisées du point de vue du gestionnaire de la flotte de taxis (moins de parcours à vide et moins de taxis nécessaires pour atteindre 90 % de qualité de services). Le comportement plus favorable de l'indicateur temps d'attente pour la solution décentralisée au delà de 14 taxis peut s'expliquer par le fait que les clients pris en plus avec une coordination centralisée (meilleur *qos*), le sont à des valeurs temporelles proches de la valeur maximale de leur fenêtre de satisfaction et sont comptabilisés alors qu'ils sont seulement échoués en décentralisés. Lorsque le rayon de communication devient suffisant (supérieur à 300) pour une bonne coordination nous constatons également que l'écart avec les solutions centralisées est faible (env. 3 % *qos* de moins en moyenne).

Les indicateurs suivants considèrent le coût de la mise en œuvre d'une solution. La figure 8 donne le nombre de messages qui transitent par le réseau P2P selon le rayon de communication en décentralisé. Selon le mode de fonctionnement de ces réseaux, tous

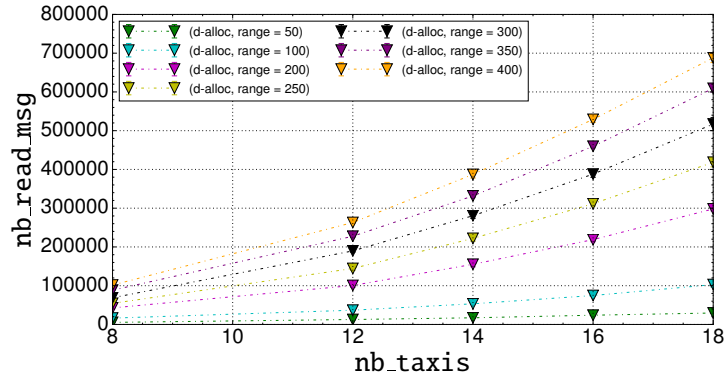


Figure 8. Nombre de messages lus par des taxis pour les différentes stratégies avec des rayons de communication pour la stratégie décentralisée en utilisant des interactions P2P, pour un nombre croissant de taxis

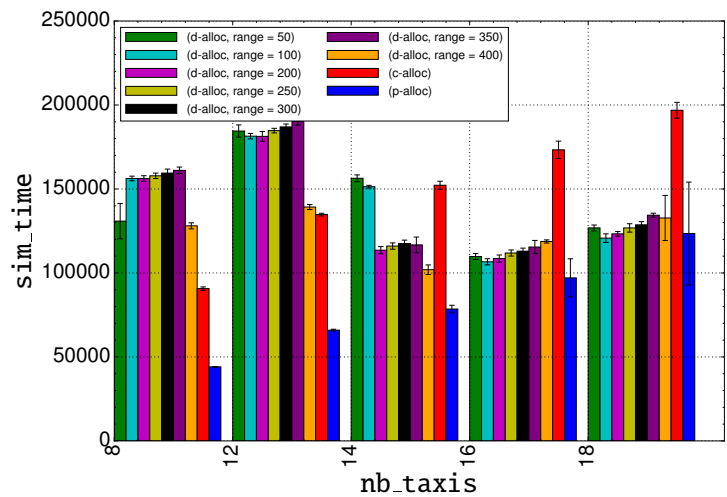


Figure 9. Temps de simulation (en ms) pour les différentes stratégies, pour un nombre croissant de taxis

les messages étant diffusés, nous obtenons un nombre de message total très important, jusqu'à 700 000 pour 18 agents et un rayon de 400. Cependant, cela ne représente par cycle et par taxi que 13 messages. Pour la solution satisfaisant 90 % des demandes en décentralisé (16 taxis et rayon de communication de 300) nous obtenons 400 000 messages soit 8 messages par cycle et taxi ce qui peut être considéré comme faible.

La figure 9 montre le temps nécessaire pour réaliser les simulations sur une seule machine. Le nombre de cycles est identique mais la durée des calculs modifie la durée des cycles. De manière évidente, **p-alloc** nécessite moins de calcul puisqu'une fois les

demandes allouées, elles ne sont pas révisées. Ainsi la **vérification** des demandes est peu déclenchée alors que les autres approches vérifient le statut de la demande à chaque pas de simulation tant qu’elles ne sont pas fermées. Puisqu’il résout implicitement PL en $0-1$ -TSAP(t) à chaque cycle de simulation, **c-alloc** induit un coup de calcul élevé. Dans le cas décentralisé **d-alloc**, le temps de la résolution dépend du rayon et du nombre de taxis qui impactent le nombre d’ensembles connectés et donc le nombre de sous-problèmes à résoudre. Avec peu de taxis et un rayon faible, peu d’ensembles connectés co-existent au même moment, et donc il y a donc moins de calcul nécessaires pour résoudre les sous problèmes. Enfin, rappelons que notre indicateur est le temps total d’exécution des simulations. Pour le cas décentralisé, le temps de calcul pourrait être au maximum (si on ne prend pas en compte les temps de latence dus aux envois de messages) divisé par le nombre de véhicules, la puissance de calcul nécessaire est en effet négligeable pour une résolution locale d’un taxi.

Le dernier indicateur est la robustesse, que nous évaluons comme la capacité à fournir une solution malgré des perturbations dans les communications. Nous utilisons ici le rayon de communication fournissant les meilleures performances d’après les expérimentations précédentes, à savoir 300. Ainsi les figures 10 à 12 compare le résultat des différentes stratégies pour les trois indicateurs de qualité étudiés (qualité de service, gain, temps attente) selon le niveau de perturbation. Par exemple, une perturbation de 0.5 signifie que 50 % des messages ne sont pas reçus en décentralisé ou que 50 % des demandes de connexion en centralisé échoue. La première constatation est que d’un point de vue globale, l’introduction de perturbation dans les communications ou les connexions perturbent peu les résultats. En effet, figures 10 à 12 l’écart entre les courbes sans perturbations et avec un maximum de perturbation (0.5) est faible. Cette observation est particulièrement vraie pour la qualité de service (figure 10) avec au maximum 4 points d’écart pour **c-alloc** et **p-alloc** et 3 points pour **d-alloc**. Le calcul systématique de l’allocation à chaque pas de temps t lié à la dynamique de la loi espace-temps de génération des requêtes et de la possibilité de remettre en cause une allocation compense la perte des informations entre deux cycles pour un véhicule.

La figure 11 montre le résultat a priori surprenant que le gain peut être amélioré dans certain cas par la perte des messages si celle-ci est faible. L’explication est que les véhicules sur-informés peuvent remettre en cause leur décision et éventuellement osciller entre différentes destinations ce qui entraîne des coûts de déplacement supplémentaires. La perte de messages introduit alors une pseudo stabilité dans les décisions. **p-alloc** échappe à ce mécanisme et est la plus affectée par les perturbations puisque les décisions ne sont pas remises en cause. Un taxi peut donc se rendre à une source pour un client alors que sa réservation sur le portail a échoué et que ce dernier a été attribué à d’autres. Dans ce cas la qualité de service est peu affectée (le client est servi) mais le gain est dégradé puisque plusieurs taxis sont affectés à la même demande.

Enfin figure 12, le temps d’attente pour **c-alloc** et **p-alloc** est aussi amélioré par les perturbations alors qu’il est dégradé pour **d-alloc**. La baisse de la qualité de service observée figure 10 indique que chacune de ces stratégies satisfait d’autant moins de clients que le niveau de perturbation est élevé. Donc les clients attendent plus

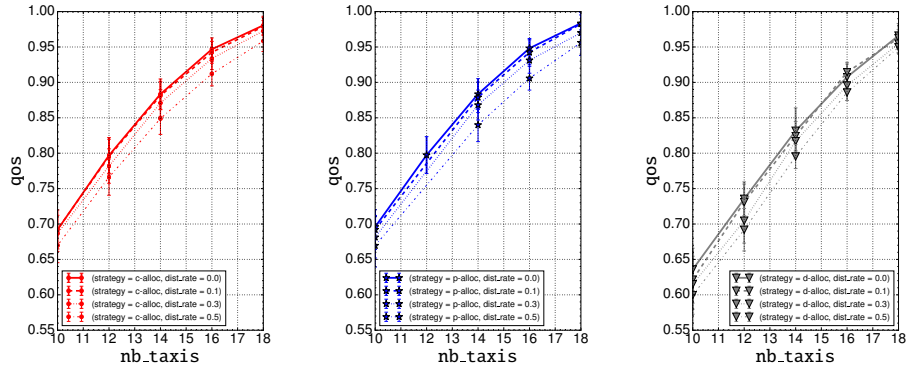


Figure 10. Qualité de service pour les différentes stratégies, pour un nombre croissant de taxis et différents taux de perturbation, portée de communication de 300

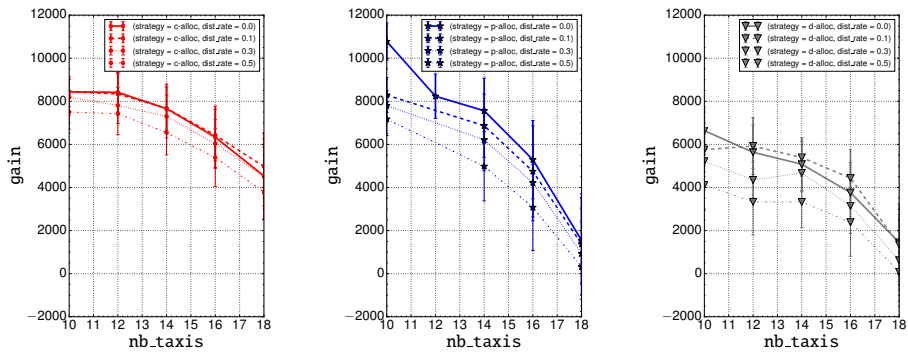


Figure 11. Gain pour les différentes stratégies, pour un nombre croissant de taxis et différents taux de perturbation, portée de communication de 300

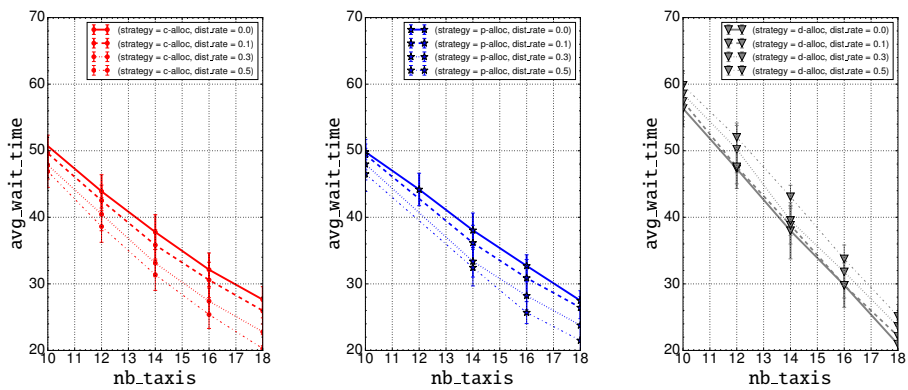


Figure 12. Temps moyen d'attente pour les différentes stratégies, pour un nombre croissant de taxis et différents taux de perturbation, portée de communication de 300

longtemps et ceux qui étaient servis à un horaire proche de la limite de leur fenêtre temporelle de validité sans perturbation ne le sont plus. **c-alloc** et **p-alloc** sont les plus pénalisés en termes de qualité de service et donc plus de clients « limites » ne sont plus comptabilisés ce qui provoque la diminution du temps d'attente moyen alors que **d-alloc**, qui est moins pénalisée, perd moins de clients « limites » et ne compense pas la dégradation du temps d'attente moyen.

Pour résumer, une stratégie fondée sur l'utilisation d'un portail se comporte aussi bien qu'une stratégie centralisant le processus d'allocation. Ceci signifie que résoudre *PL en 0-1-TSAP(t)* n'est pas nécessaire pour atteindre de très bonnes solutions à *TSAP(t)*. Laisser les taxis prendre leurs décisions eux-mêmes et les coordonner par un simple portail est suffisant. Cependant, alors que **c-alloc** et **p-alloc** nécessitent plus de messages (linéaire en nombre de taxis), les deux souffrent d'un goulot de centralisation, qui peut mener à l'échec. Au contraire, **d-alloc** nécessite moins de messages comme représenté sur la figure 8, parce que l'infrastructure P2P nécessite beaucoup de relais de messages. En fait dans les réseaux VANET, les messages sont diffusés impliquant un nombre important de traitement de messages à chaque nœud. Cependant, puisque les agents se coordonnent localement, perdre une information majeure est moins probable et impacte moins le système dans sa globalité.

5. Travaux liés

Dans (Santani *et al.*, 2008), les auteurs analysent les activités d'un dispatcheur et mettent en évidence l'impossibilité de satisfaire à la fois les taxis (critère du taux d'utilisation des taxis) et les clients (critère du temps d'attente à la fois dans les périodes de pointe ou pas). Leur conclusion est que les solutions décentralisées multiagents peuvent améliorer la performance de ces solutions de transport. Notre proposition évalue en plus les bénéfices d'un processus d'allocation centralisé.

La centralisation du processus d'allocation avec un dispatcheur automatique est assez courante dans les approches multiagents (Egan, Jakob, 2016; Grau, Romeu, 2015; Maciejewski, Nagel, 2013; Shen, Lopes, 2015). Dans (Maciejewski, Nagel, 2013), un dispatcheur central met en œuvre trois stratégies d'allocation. Elles utilisent plus ou moins d'information temps réel sur l'ordonnancement des taxis résultat d'interactions avec des agents taxis qui ne prennent pas part au processus de décision. L'utilisation d'un dispatcheur central avec des agents taxis qui donnent l'information sur leur état courant est aussi proposée dans (Grau, Romeu, 2015; Shen, Lopes, 2015). Dans (Egan, Jakob, 2016), une approche d'allocation fondée sur un marché est proposée. Cette solution décentralisée est traitée entre des agents clients et des agents fournisseurs du service de taxi. La décision d'allocation mise en œuvre par le fournisseur du service de taxi est cependant centralisée. Ces propositions évaluent leur propre stratégie d'allocation sans comparaison avec des alternatives décentralisées.

Dans (Seow *et al.*, 2010), un protocole de coordination décentralisé entre des agents taxis disponibles est proposée. Un sous ensemble de demandes est affecté à un sous-ensemble (de taille équivalente) d'agents taxi disponibles. Les agents taxis

échantent en P2P leur évaluation des demandes afin d'arriver à un consensus. Dans (Glaschenko *et al.*, 2009), les auteurs proposent un processus de négociation fondé sur une estimation du coût et du temps de transport, l'allocation est calculée par un agent dédié à chacune des demandes client. Ces solutions agents illustrent la faisabilité des allocations décentralisées dans un environnement dynamique. Cependant, il s'agit de solutions *ad-hoc* qui ne s'appuient pas sur des protocoles prouvés à la différence du domaine DCOP (Cerquides *et al.*, 2014). De plus ces solutions sont fondées sur une hypothèse de connexion totale entre les agents, ce qui implique des coûts importants.

Dans la limite de notre connaissance, la seule proposition qui compare allocation centralisée et décentralisée est (Jin, Jie, 2012) qui propose une alternative hybride. L'un des objectifs de ce travail est de diviser l'espace de recherche avec des agents nœuds stations qui ne peuvent interagir qu'avec les agents taxis dans leur voisinage direct. Ces derniers planifient des propositions pour les demandes client et les agents nœuds stations appliquent des politiques de filtrage. Les expérimentations montrent des meilleurs résultats qu'une solution pleinement décentralisée avec des communications complètes des demandes aux agents taxi. Une explication possible est que le partitionnement de l'espace de recherche améliore la qualité des résultats et réduit le temps de calcul.

La limitation de l'espace de recherche est implicite dans des solutions d'allocation décentralisées. Dans les solution centralisées, le réseau est décomposé en espace où l'allocation est réalisée et est étendue à l'espace le plus proche si aucune allocation n'est trouvée (Alshamsi *et al.*, 2009 ; Shen, Lopes, 2015). Dans une solution décentralisée, la même approche est proposée par (Jin, Jie, 2012) avec les nœuds stations pour lesquels le nombre d'agents taxis est limité de manière arbitraire (Glaschenko *et al.*, 2009 ; Seow *et al.*, 2010). Dans notre proposition, la limitation l'espace de recherche n'est pas définie *a priori* mais est le résultat de la connexion des véhicules aux sources où les demandes sont émises.

6. Conclusions

Nous avons étudié l'impact de la décentralisation dans le processus d'allocation de demandes à une flotte de taxis autonomes. Nous avons modélisé le problème d'allocation de taxis (TSAP) comme un problème d'optimisation sous contraintes. Afin de le résoudre, nous avons proposé trois stratégies : coordination centralisée par un dispatcheur (avec communication globale), coordination via un portail (avec communication globale et locale), coordination par un DCOP (avec communication locale). La première approche correspond à un système classique d'appariement des demandes aux taxis. La seconde correspond à une solution de décision distribuée, mais coordonnée grâce à un portail de publication et de réservation. Enfin, la troisième ne fait appel qu'à des communication pair-à-pair sans coordination globale. Ces approches ont été évaluées par simulation, sur un scénario réaliste dans le centre-ville de Saint-Étienne, avec des flottes de 8 à 16 taxis devant répondre à des demandes sur une durée simulée de 4h. Comparée aux approches centralisées (dispatcheur ou portail),

la coordination par un DCOP montre des résultats proches de l'optimal en termes de qualité de la solution (ratio entre le nombre de clients servis et le nombre de demandes total) tout en assurant une mise en œuvre qui peut passer à l'échelle et qui est plus robuste. En effet, la solution décentralisée permet d'obtenir des indicateurs de performance équivalents aux autres solutions (qualité de service, gain, temps d'attente) à partir d'une certaine taille de flotte (16 taxis) et d'un certain rayon de communication pair-à-pair (300 m). Il est intéressant de noter que la solution décentralisée exécutée par notre simulateur nécessite moins de temps de calcul que les autres solutions, lorsque le nombre d'agents augmente. En effet, les problèmes à résoudre sont locaux, donc moins coûteux à résoudre (au prix d'une sous-optimalité, cependant limitée). Enfin, nous avons mesuré les conséquences de la perte de messages de coordination, pour évaluer la robustesse des approches. La coordination par DCOP obtient une qualité de service équivalente, mais avec un temps d'attente légèrement plus long.

Cet article n'a présenté qu'une partie limitée de l'espace de paramétrage du TSAP. De nombreuses pistes sont envisageables à la fois sur l'étude de nouvelles stratégies de coordination ou encore l'intégration des contraintes énergétiques des véhicules mais également de nouvelles conditions de simulation intégrant, par exemple, la prise en compte d'un temps de parcours dynamique. De plus, il serait intéressant d'étudier l'impact de la position des sources, sur les indicateurs de performance. Nous lançons actuellement des travaux sur ces pistes d'investigation. Une autre perspective est l'étude d'autres critères d'évaluation. En effet, ce n'est pas le sujet principal de cet article mais le critère d'évaluation des demandes à un grand impact sur la qualité des résultats et doit être étudié en profondeur, en prenant en compte à la fois le point de vue des clients et des taxis. De même, le couplage de ces approches dans un système hybride (couplant centralisation quand possible, et décentralisation quand nécessaire) pourrait également fournir des solutions pertinentes. Enfin, notre étude a mesuré les temps de simulation des différentes approches exécutées sur une même machine de calcul. Nous souhaitons maintenant évaluer notre approche dans un contexte distribué réel (sur différentes machines) ou simulé (en étendant notre simulateur), pour mesurer des temps de calcul prenant en compte les temps incompressibles de diffusion de messages et les latences associées. Ceci nous permettra de juger du réel potentiel de déploiement et de réactivité de notre approche sur une infrastructure de communication distribuée réaliste.

Remerciements

Ces travaux ont été en partie financés par Renault Innovation.

Bibliographie

- Alshamsi A., Abdallah S., Rahwan I. (2009). Multiagent self-organization for a taxi dispatch system. In *8th international conference on autonomous agents and multiagent systems*, p. 21–28.
- Cerquides J., Farinelli A., Meseguer P., Ramchurn S. D. (2014). A tutorial on optimization for multi-agent systems. *The Computer Journal*, vol. 57, n° 6, p. 799–824. Consulté sur

<http://dx.doi.org/10.1093/comjnl/bxt146>

- Egan M., Jakob M. (2016). Market mechanism design for profitable on-demand transport services. *Transportation Research Part B: Methodological*, vol. 89, p. 178–195.
- Glaschenko A., Ivaschenko A., Rzevski G., Skobelev P. (2009). Multi-agent real time scheduling system for taxi companies. In *8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2009), Budapest, Hungary*, p. 29–36. Consulté sur <https://pdfs.semanticscholar.org/dc25/22cfba5bd99e107c93feb4d75d593327a753.pdf>
- Grau J. M. S., Romeu M. A. E. (2015). Agent based modelling for simulating taxi services. *Procedia Computer Science*, vol. 52, p. 902–907.
- Horn M. E. (2002). Fleet scheduling and dispatching for demand-responsive passenger services. *Transportation Research Part C: Emerging Technologies*, vol. 10, n° 1, p. 35–63.
- Jin X., Jie L. (2012). A study of multi-agent based model for urban intelligent transport systems. *International Journal of Advancements in Computing Technology*, vol. 4, n° 6, p. 126–134.
- Maciejewski M., Nagel K. (2013). The influence of multi-agent cooperation on the efficiency of taxi dispatching. In *International conference on parallel processing and applied mathematics*, p. 751–760.
- Pujol-Gonzalez M., Cerquides J., Farinelli A., Meseguer P., Rodriguez-Aguilar J. A. (2015). Efficient inter-team task allocation in robocup rescue. In *Proceedings of the 2015 international conference on autonomous agents and multiagent systems*, p. 413–421. Richland, SC, International Foundation for Autonomous Agents and Multiagent Systems. Consulté sur <http://dl.acm.org/citation.cfm?id=2772879.2772933>
- Pujol-Gonzalez M., Cerquides J., Meseguer P., Rodríguez-Aguilar J. A., Tambe M. (2013). Engineering the decentralized coordination of uavs with limited communication range. In *15th conference of the spanish association for artificial intelligence, caeipia 2013*, p. 199–208. Springer. Consulté sur http://dx.doi.org/10.1007/978-3-642-40643-0_21
- Santani D., Balan R. K., Woodard C. J. (2008). Spatio-temporal efficiency in a taxi dispatch system. In *6th international conference on mobile systems, applications, and services, mobisys*.
- Seow K. T., Dang N. H., Lee D.-H. (2010). A collaborative multiagent taxi-dispatch system. *IEEE Transactions on Automation Science and Engineering*, vol. 7, n° 3, p. 607–616.
- Shen W., Lopes C. (2015). Managing autonomous mobility on demand systems for better passenger experience. In *International conference on principles and practice of multi-agent systems*, p. 20–35.
- Zargayouna M., Balbo F., Ndiaye K. (2016). Generic model for resource allocation in transportation. application to urban parking management. *Transportation Research Part C: Emerging Technologies*, vol. 71, p. 538–554.

