L'optimisation du covoiturage dynamique multi-saut

Corwin Fèvre[a] corwin.fevre@univ-lille.fr
Hayfa Zgaya-Biau[a] hayfa.zgaya-biau@univ-lille.fr
Phlippe Mathieu[a] philippe.mathieu@univ-lille.fr
Slim Hammadi[a] slim.hammadi@centralelille.fr

[a]Univ. Lille, CNRS, Centrale Lille, UMR 9189, CRIStAL, F-59000 Lille, France

Résumé

Dans cet article, nous proposons une approche multi-agent pour résoudre le problème du covoiturage dynamique multi-saut. Dans notre système, les passagers et les conducteurs sont représentés comme des agents autonomes et rationnels en perpétuelle interaction pour satisfaire leurs propres objectifs comme leur temps d'attente ou leur temps de trajet par exemple. Dans la solution proposée, les agents conducteurs et passagers ont une perception modélisée dynamiquement en utilisant des R-Arbres. Nous modélisons leurs préférences en matière de détour et de trajet et montrons l'impact de celles-ci sur la résolution d'une instance de covoiturage dynamique. Les résultats présentés montrent que notre système permet de traiter dynamiquement des requêtes complexes de passagers tout en minimisant l'impact du partage de trajet pour les conducteurs, et ce, pour un large spectre de préférences et de comportements.

Mots-clés : *Covoiturage, Simulation, Optimisation, Agents*

Abstract

In this paper, we propose a multi-agent approach to solve the dynamic multi-hop ridesharing problem. In our system, passengers and drivers are represented as autonomous and rational agents in perpetual interaction to satisfy their own objectives such as their waiting time or their travel time. In the proposed solution, driver and passenger agents have a dynamically modeled perception using R-Trees. We model their detour and route preferences and show the impact of these on the resolution of a dynamic ridesharing instance. The presented results show that our system dynamically handles complex passenger requests while minimizing the impact of ridesharing for drivers across a wide spectrum of preferences and behaviors.

Keywords: *Ridesharing, Simulation, Optimization, Agents*

1 Introduction

Le covoiturage est devenu un moyen de transport à part entière, que ce soit pour les trajets quotidiens ou pour les voyages plus longs. Outre son intérêt écologique, il permet aux conducteurs de réduire les coûts liés à leurs déplacements tout en offrant la possibilité aux passagers de voyager avec plus de flexibilité que les moyens de transport conventionnels. Il en résulte une utilisation accrue des véhicules déjà déployés sur la route, une réduction de la congestion du trafic et donc une diminution des émissions polluantes.

Nous proposons dans ce travail d'étudier le covoiturage dynamique entre particuliers. Un système de covoiturage dynamique prends en compte l'état actuel de l'environnement des utilisateurs pour effectuer des associations de covoiturage. Un tel système collecte en temps réel différents flux d'information dans l'environnement comme les itinéraires et temps de trajets des conducteurs en cours de route ainsi que les requêtes des passagers. Le caractère dynamique réside alors dans le flux continu de conducteurs et de passagers entrant, évoluant et sortant du système. Cette forme de covoiturage est majoritairement étudiée à l'échelle de grandes métropoles [10, 29] et dans le cadre de trajets courts et spontanés [30, 21].

La flexibilité d'un système de covoiturage dynamique peut être modélisée et quantifiée par deux opérations. D'une part, les conducteurs peuvent choisir de faire des détours pour prendre ou déposer des passagers, et ainsi élargir l'offre de transport du système de covoiturage. D'autre part, les passagers peuvent choisir de passer d'un véhicule à l'autre, et donc d'effectuer des transferts, afin d'arriver à leur destination en plusieurs étapes, si cela est plus pratique pour eux. Un exemple de ces opérations est illustré dans la Figure 1.

Dans cette figure, deux conducteurs d_i et d_j suivent leur itinéraire, respectivement en jaune et rouge, reliant leur nœud de départ v_s et leur

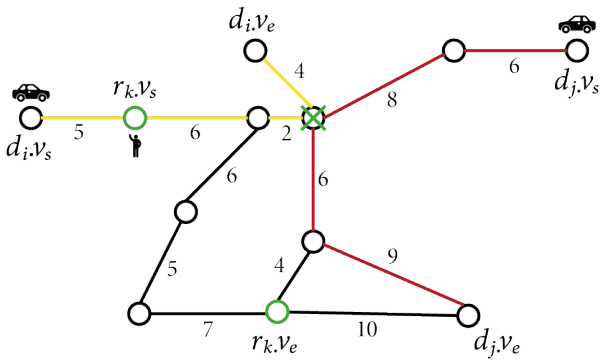

FIGURE 1 – Un exemple de covoiturage multi-saut

nœud d'arrivée v_e. Un passager r_k attend sur son nœud de départ $r_k.v_s$ et veut rejoindre son nœud d'arrivée $r_k.v_e$, tous deux entourés de vert. Les trajectoires des conducteurs se croisent sur le nœud avec une croix verte, un transfert est alors réalisable. Le conducteur d_i peut partager son trajet avec le passager r_k puis, une fois arrivé au nœud de transfert, le passager peut effectuer un transfert avec le conducteur d_j. Le conducteur d_j doit cependant faire un détour pour déposer le passager, ce qui rajoute 5 unités de temps à son temps de trajet.

Le défi d'un tel cadre est d'équilibrer l'impact du covoiturage sur l'offre et la demande. Cet équilibre implique de nombreux paramètres tels que le ratio conducteurs-passagers ou les détours acceptables par les utilisateurs en termes de temps et de distance. Le problème est alors d'associer au mieux passagers et conducteurs en exploitant leur marge de manœuvre afin de satisfaire les deux parties.

Un système de covoiturage peut être naturellement représenté par un système multi-agent, les conducteurs et les passagers étant des entités autonomes en interaction. C'est pourquoi nous proposons dans cet article un système multi-agent permettant de décrire des comportements spécifiques pour chaque utilisateur, garantissant le caractère hétérogène de notre simulation. La limite de perception de ces agents est définie dynamiquement par leurs préférences spatio-temporelles. Ils disposent ainsi d'une rationalité limitée.

Pour traiter l'important flux d'informations généré par un tel système, nous utilisons une structure d'indexation spatiale nommée R-Arbre [22]. Cette structure, et les algorithmes qui y sont associés, permet de stocker efficacement ces données et d'effectuer des requêtes de superposition sur des points et des zones de l'espace. Les R-Arbres nous permettent aussi de réduire la complexité associée à l'identification du nœud de transfert optimal dans le cas d'un covoiturage multi-saut.

Après un état de l'art sur le problème du covoiturage et ses approches d'optimisation et de simulation dans la section 2, nous définissons notre problème de covoiturage en section 3. Nous proposons en section 4 une solution basée sur un système multi-agent dont le comportement des agents est défini par divers algorithmes. Enfin, en section 5 nous étudions différents comportements typiquement observables dans des scénarios réels de covoiturage. Nous les confrontons à notre modèle et étudions les résultats obtenus sur des cartes réelles.

2 Travaux connexes

À l'occasion d'un covoiturage, un passager peut être acheminé par un seul et unique conducteur, c'est ce que l'on nomme **le covoiturage simple saut** (ou *single-hop* en anglais). Le problème du covoiturage simple-saut consiste à identifier le meilleur conducteur pour acheminer un passager entre deux positions. C'est la première forme de covoiturage étudiée dans la littérature [19, 1, 10] de par ses similarités avec les problèmes d'allocation de ressources.

Il est aussi possible pour un passager d'utiliser plusieurs conducteurs, et donc d'effectuer des transferts afin d'arriver à sa destination. Cette variante s'appelle **le covoiturage multi-saut** (ou *multi-hop* en anglais) ou encore le covoiturage avec transferts [8]. Cette méthode permet d'améliorer les résultats obtenus avec le covoiturage simple-saut en augmentant les possibilités de covoiturage [17, 30]. Le covoiturage multi-saut implique une complexité accrue, car il faut cette fois identifier la meilleure combinaison de conducteurs ainsi que le lieu de transfert optimal pour tous les acteurs du covoiturage. Le problème du covoiturage multi-saut a été associé à de nombreux problèmes NP-Durs et NP-Complets de la littérature [7, 21].

Pour gérer cette complexité, l'espace de recherche doit être organisé et exploré de manière optimale. De nombreux travaux de recherche hiérarchisent l'espace [23] ou le divisent en zones pour en réduire la complexité [29, 28]. Ces divisions et réductions de l'espace sont cependant la plupart du temps effectuées de manière empirique et non dynamique forçant

une approximation importante de l'espace. Dans ce contexte, une structure d'indexation spatiale connue sous le nom de R-Arbre (*R-Tree* [15]) permet d'indexer dynamiquement les objets dans un arbre et d'optimiser les requêtes de superposition de zones ou de nœuds de l'espace [22]. Cette structure a notamment été utilisée pour le partage de taxi et s'est révélée être très efficace [30]. Dans nos travaux, nous proposons d'étendre le champ d'utilisation des R-Abres au problème du covoiturage entre particuliers.

Il existe de nombreuses façons de modéliser le problème, notamment en ce qui concerne la granularité de la population. D'une part, l'approche centrée-groupe vise à optimiser des objectifs communs à l'ensemble de la population et donc à privilégier le système par rapport à l'individu [10, 1]. Cette méthodologie permet une simulation moins coûteuse en ressources mais s'avère éloignée de la réalité vis-à-vis du caractère hétérogène de la population. En revanche, l'approche centrée-individu cherche à résoudre la demande de chaque utilisateur, au cas par cas, en tenant compte de ses préférences pour définir les objectifs d'optimisation. Ainsi, chaque individu a son propre comportement et sa propre perception. Dans ce contexte, l'approche multi-agent [12] permet de modéliser un tel système en garantissant l'autonomie de chaque utilisateur et en permettant à la simulation d'être plus proche de la réalité [3, 19, 18, 24, 9].

Le problème du covoiturage a d'abord été étudié de manière mono-objectif avec des approches de programmation par contraintes [1, 10]. Des problèmes de covoiturage multi-objectif ont ensuite été étudiés et résolus à l'aide d'heuristiques évolutionnaires et génétiques [17, 16]. Ces méthodes permettent une optimisation plus rapide - bien qu'approximative - et donc plus applicable dans le monde réel. Avec l'avènement du covoiturage multi-saut et afin d'obtenir une solution exacte et multi-objectif, la recherche s'oriente vers une approche du problème de covoiturage via le problème du ou des plus proches voisins d'un groupe (GNN/ANN) [25]. En effet, puisque le multi-saut implique la recherche du nœud de transfert optimal, des heuristiques basées sur des algorithmes de recherche de plus proches voisins ont été développées [31] et ont été améliorées pour être plus efficaces dans un contexte réel [30, 6]. C'est dans cette perspective que nous proposons d'étendre l'algorithme de référence IER (Incremental Euclidean Restriction) [31] pour évaluer conjointement les solutions à sauts multiples et à sauts uniques de manière multi-objectif.

3 Formulation du problème

3.1 Spécification du réseau routier et des agents du système

Le réseau routier de notre système est modélisé par un graphe orienté et connexe $G = \langle V, E \rangle$ dans lequel V est l'ensemble des nœuds du graphe et E est l'ensemble des arêtes liant ces nœuds. Les arêtes représentent des routes et les noeuds représentent des intersections de routes ou des impasses. Un nœud $v_i = (x, y)$ est référencé spatialement par un couple de coordonnées : une longitude x et une latitude y. Différents poids peuvent être attribués aux arêtes désignant le coût du trajet entre deux nœuds, comme le temps de trajet par exemple.

Nous utilisons une simulation en temps discret, le temps évolue à intervalles constants et le pas de simulation courant est dénoté par la variable $time$. À chaque pas de simulation, tous les passagers et conducteurs présents dans le système peuvent effectuer une action en fonction de leur comportement.

Un agent utilisateur du système $u_i = \langle v_s, v_l, v_e, det_{max} \rangle$, avec $u_i \in U$ représentant un passager ou un conducteur dans le système de covoiturage, est initialisé par quatre éléments : son nœud de départ v_s, son nœud de localisation actuel v_l, son nœud d'arrivée v_e et son facteur de détour det_{max}. Le facteur de détour quantifie la volonté de détour d'un agent utilisateur. Ainsi, s'il vaut 0, l'utilisateur ne souhaite pas faire de détour, et s'il vaut 1, l'utilisateur est prêt à doubler son trajet initial pour effectuer des détours. Nous faisons varier ce facteur entre 0 et 1 dans nos expériences. Cet élément permet de définir les limites initiales d'action et de perception d'un agent.

Un agent conducteur $d_i = \langle v_s, v_l, v_e, det_{max}, c_{max} \rangle$, avec $d_i \in D$, est initialisé par cinq éléments. Un conducteur étant un utilisateur, il hérite des quatre premiers paramètres définis précédemment. Le dernier paramètre c_{max} définit le nombre maximal de sièges disponibles dans le véhicule du conducteur. Un conducteur peut effectuer un nombre infini de détours et d'arrêts tant qu'il respecte son objectif d'atteindre sa destination avant son heure d'arrivée la plus tardive. Il suit donc son trajet de nœud en nœud, en prenant systématiquement le chemin le plus court. S'il doit faire un détour pour prendre et

déposer un passager, il doit être capable de déterminer si cette action est possible vis-à-vis de son heure d'arrivée la plus tardive. Pour ce faire, cet agent maintient un ordonnanceur de voyage où il stocke différentes informations telles que ses prochains arrêts $stops$ (nœuds du graphe de route), les temps de trajets minimum arr et maximum ddl, les marges de détours slk ou les capacités restantes c (nombre de sièges disponibles) entre ses arrêts. La modélisation exacte de cet ordonnanceur est détaillée dans nos précédents travaux [13].

Un agent passager $r_i = \langle v_s, v_l, v_e, det_{max}, wt_{max}, pref \rangle$, avec $r_i \in R$, est initialisé par six éléments. Un passager étant un utilisateur du système, il hérite des quatre premiers paramètres définis précédemment. Le paramètre wt_{max} correspond au temps d'attente acceptable au maximum par le passager pour la totalité du trajet. S'il est dépassé, l'agent passager s'impatiente et disparaît du système, ce qui affecte le taux de service global du système de covoiturage. La perception p d'un agent passager représente la zone des conducteurs accessibles. Elle permet d'exclure les conducteurs trop éloignés pour le prendre en charge. Le paramètre $pref$ représente le vecteur de préférence du passager, c'est-à-dire son profil/comportement. Chaque élément $pref_i \in [0, \alpha]$ et la somme des éléments de $pref$ doit être égale à α : $sum(pref_i, ..., pref_k) = \alpha$. On peut alors, par exemple, fixer α à 10 et faire varier la pondération des préférences d'un agent passager entre 0 et 10. De cette manière, nous pouvons attribuer un poids à chaque objectif contenu dans cette liste de préférences et détailler différents profils de passagers. Ceci implique que, si $pref_i > pref_j$, alors le passager accorde plus d'importance à la préférence $pref_i$ qu'à la préférence $pref_j$ lors de l'optimisation de son trajet. Un passager a donc pour objectif d'atteindre sa destination avant son heure d'arrivée la plus tardive tout en sélectionnant le ou les meilleurs covoiturages en fonction de ses préférences.

Un agent service de transport $tsa_i = \langle RT\_p \rangle$ est responsable de la connexion des agents utilisateurs. Il remplit deux fonctions. La première consiste à stocker et à mettre à jour toutes les perceptions des agents conducteurs, à leur demande, lorsqu'un évènement entraîne une modification de leur perception. La seconde fonction est de répondre aux requêtes spatiales des agents passagers du système qui souhaite identifier des candidats au covoiturage. L'agent service de transport est donc implémenté comme un "tableau noir" afin de mettre à jour et de transmettre les informations des agents du système lorsque cela est demandé.

3.2 Contraintes et Formulations

Nous limitons notre étude au covoiturage à un et deux sauts, c'est-à-dire au covoiturage limité à deux conducteurs pour un transfert. En effet, plusieurs articles comme [23] ont montré que l'augmentation du nombre de transferts possibles améliore à peine les performances d'un système de covoiturage tandis que la complexité du problème explose exponentiellement. Dans cet article, chaque requête de covoiturage est associée à un seul et unique passager.

Pour qu'un covoiturage entre un passager et un conducteur soit réalisable, nous définissons plusieurs contraintes spatio-temporelles :

- **Contrainte temporelle :** Les conducteurs et les passagers doivent arriver à leur destination avant leur heure d'arrivée souhaitée. Un passager doit être pris en charge avant que son temps d'attente maximum ne soit atteint.
- **Contrainte de capacité :** La capacité courante d'un conducteur ne doit jamais dépasser la capacité maximale de son véhicule.
- **Contrainte de l'ordre des arrêts :** Un conducteur partageant son itinéraire avec un passager doit d'abord passer par le nœud de prise en charge du passager avant de visiter le nœud de dépôt de ce dernier.

Une association (*"a match"* en anglais) $m \in r_i.M$ est définie par plusieurs variables. Un ou deux conducteurs $d_j, d_k \in D$ (simple-saut : $j = k$, multi-saut : $j \neq k$) ; un nœud de transfert v_tsf s'il y a un transfert ; et enfin quatre critères : un nombre de transferts nb_tsf (0 ou 1 dans cette étude), un temps d'attente total $total\_wt$, un temps d'arrivée arr_time, et une distance additionnelle de détour add_dist. Chaque passager a pour but de choisir l'association la plus optimale dans M selon ses préférences $pref$. Le problème étudié est donc multi-objectif en impliquant la minimisation simultanée de ces quatre critères.

La complexité de notre problème peut être résumée par : quel est le nombre de combinaisons de covoiturage possible ? Avec une approche individu-centrée cette complexité peut être formalisée comme suit :

$$O(R * H^{D*N}) \quad (1)$$

avec H : le nombre de sauts possibles ; D : le nombre d'agents conducteurs ; R : le nombre

d'agents passagers ; N : le nombre de nœuds du graphe de route. Ce problème est ainsi associé aux problèmes de classe de complexité exponentielle : $O(2^n)$.

4 Le système multi-agent proposé

À chaque étape de la simulation, le tour d'un agent utilisateur est divisé en deux phases : la phase de mise à jour de sa perception de l'environnement et la phase de prise de décision.

4.1 La mise à jour de la perception des agents

La perception d'un agent utilisateur représente ce qu'il est capable d'utiliser dans l'environnement pour atteindre son objectif. Dans nos travaux, cette perception est dynamique et doit être mise à jour régulièrement.

Pour établir sa perception, l'agent utilisateur collecte d'abord les nœuds du plus court chemin entre son origine et sa destination. Il récupère ensuite les latitudes et longitudes minimales et maximales de tous ces nœuds pour définir le rectangle de délimitation minimum (MBR) de sa perception. Cette perception initiale est étendue en y ajoutant la marge de temps de détour ou d'attente restante de l'agent afin de former le rectangle englobant (BBOX) de sa perception. Ce processus est basé sur nos travaux antérieurs [13] et est illustré dans la Fig.2.

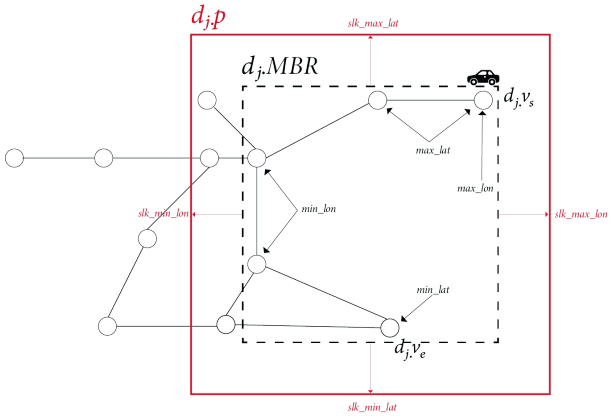

FIGURE 2 – Exemple de calcul de la perception d'un agent conducteur.

Cette méthode est utilisée pour la perception d'un conducteur, entre chaque arrêt de son ordonnanceur, et pour la perception d'un passager, de son point de départ à son point d'arrivée. Cette perception est approximative, ainsi, un test de faisabilité plus rigoureux, basé sur les contraintes de la section 3.2, est nécessaire avant tout covoiturage.

Un agent conducteur ne met à jour sa perception que dans deux cas : (1) s'il atteint un nœud du graphe de route ; (2) si un contrat de covoiturage impliquant un détour est conclu avec un passager. Un agent passager met à jour sa perception à chaque étape de la simulation. Si un nouveau conducteur apparaît dans sa perception, il cherche à déterminer s'il peut lui permettre de rejoindre sa destination.

4.2 L'identification du covoiturage optimal

Le processus d'identification du covoiturage optimal est résumé dans l'algorithme 1. Comme expliqué précédemment, un conducteur peut permettre à un passager de rejoindre sa destination de deux façons : (1) simple-saut : en assurant la totalité du trajet (2) multi-saut : en lui offrant la capacité d'effectuer un transfert avec un autre conducteur. Pour identifier les conducteurs candidats à un covoiturage, un passager effectue des requêtes à l'agent service de transport. Un conducteur candidat au covoiturage est un conducteur dont au moins l'une des perceptions chevauche la perception du passager ainsi qu'au moins l'un de ces deux éléments : le noeud de départ et/ou d'arrivée du passager (PMO L1-3). Ainsi, un conducteur dont la perception chevauche les deux noeuds est candidat au simple-saut, tandis que si sa perception ne chevauche que l'un des noeuds, il est nécessaire de trouver un autre conducteur complémentaire pour espérer effectuer covoiturage multi-saut. Un conducteur est complémentaire à un autre s'il chevauche une partie de sa perception, la zone de rencontre des perceptions des deux conducteurs et du passager est appelée zone de transfert (voir Figure 3).

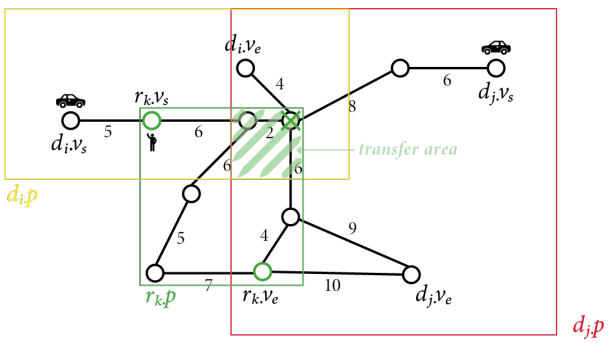

FIGURE 3 – Identification des candidats au covoiturage à l'aide des différentes perceptions.

L'évaluation d'un covoiturage simple saut est peu complexe. Le passager évalue les différents objectifs relatifs au trajet et établit une valeur indiquant la performance de cette solution en fonction de ses préférences (PMO L7-10). La fonction multi-objectif responsable de cette évaluation est détaillée plus tard dans cet article. À contrario, l'évaluation d'un covoiturage multi-saut induit une complexité très importante. Il faudrait effectuer cette même méthode pour chaque nœud de transfert possible pour les conducteurs. Il n'est pas envisageable de couvrir toutes ces possibilités. C'est pourquoi nous avons développé une heuristique, nommée CIER *Constrained Incremental Euclidean Restriction* visant à prioriser l'évaluation des nœuds de transferts les plus proches du passager et des conducteurs (PMO L12-17)[14].

Cette heuristique est basée sur l'algorithme IER (*Incremental Euclidean Restriction*) [26] permettant la résolution du problème des plus proches voisins d'un groupe (ANN/GNN) [25]. Ce problème est semblable au problème de l'identification du nœud de transfert optimal auquel nous sommes confrontés. L'objectif est d'identifier le(s) nœud(s) d'un graphe le(s) plus proche(s) d'un groupe d'autres nœuds de ce même graphe. Un exemple simple est le problème du restaurant : quel est le restaurant le plus proche de trois individus éparpillés dans une ville ? Les nœuds de transferts sont alors les restaurants et les conducteurs et le passager sont les individus. La notion du plus proche peut alors être complexe : minimisant la somme des distances (utilitaire), la moyenne des distances (égalitaire)...

Nous avons choisi une approche utilitariste ayant pour objectif de minimiser la distance additionnelle de détour occasionnée par le covoiturage. L'intérêt de cette approche est de limiter l'impact du covoiturage sur les conducteurs afin de conserver leur marge de détour pour les autres passagers du système. L'étude et la comparaison d'autres politiques feront certainement l'objet d'une étude de notre part. Une fois le nœud de transfert optimal identifié, le passager évalue la solution de la même façon que pour un covoiturage simple-saut.

Pour diversifier l'offre de covoiturage proposée à un passager, nous proposons une solution basée sur le mécanisme de dominance de Pareto [27]. L'idée générale est de maintenir une liste des meilleures solutions de covoiturage $best\_matches$. Une meilleure solution de covoiturage est une solution minimisant au moins un des objectifs présentés dans la section 3.2 (temps d'attente, heure d'arrivée, distance additionnelle et nombre de transferts). Elle est, dans ce cas, Pareto dominante et est ajoutée à la liste $best\_matches$. De cette façon, nous préservons des solutions optimales pour chaque objectif et le passager peut, selon ses préférences, choisir celle qui lui convient le mieux. Si une solution de covoiturage optimise tous les objectifs, alors elle est Pareto optimale, la liste est alors vidée et cette solution lui est ajoutée (PMO, L19-24). Le nombre de solutions optimales est ainsi compris entre 0, quand il n'y a pas de solution, et 4 si il existe une solution pareto dominante par objectif.

En plus de cette liste des meilleures solutions, nous maintenons deux autres listes : la liste des pires valeurs $worst\_goals$ et des meilleures valeurs $best\_goals$ atteintes pour chaque objectif. Ces listes servent d'intervalles de référence à un agent passager pour évaluer la performance d'une solution. En effet, en effectuant une différence proportionnelle pondérée par son vecteur de préférence, l'agent passager obtient un pourcentage de performance pour chaque objectif ainsi qu'en moyenne. Ce pourcentage est calculé à l'aide de la formule 2 :

$$\frac{\sum_i^k (1 - \frac{abs(candidate.goals_i - worst\_goals_i)}{worst\_goals_i - best\_goals_i} * 100) * pref_i}{k + sum(pref)} \quad (2)$$

Avec k le nombre d'objectifs à considérer. De cette façon, un objectif plus important pour le passager a plus d'impact sur l'évaluation finale de la solution. Le passager sélectionne enfin la solution la plus performante (PMO, L28).

5 Expérimentations et résultats

Dans cette section, nous proposons de simuler différents comportements de passagers et de conducteurs à l'aide du modèle agent présenté dans cet article. Les résultats présentés montrent l'impact de ces comportements sur des instances de problème de covoiturage.

5.1 Création des profils utilisateurs

Nous appelons *"un profil utilisateur"* un ensemble de préférences visant à simuler un comportement réel. Pour les conducteurs, nous faisons essentiellement varier la préférence de détour det_{max} entre 0 (c'est-à-dire pas de détour) et

Algorithm 1 PMO - Preferential Matching Optimization

Input : G = road infrastructure graph, r_i = a rider agent
Output : $best\_match$ = the best match

$candidates\_s$: drivers which perception overlaps starting node
$candidates\_e$: drivers which perception overlaps ending node
$best\_add\_dist, best\_wt, best\_arrival\_time, best\_nb\_tsf := \infty$,
$best\_matches := list()$ #$r_i.M$
for all $d\_s \in candidates\_s$ **do**
  for all $d\_e \in candidates\_e$ **do**
    if $d\_s.id = d\_e.id$ and match is feasible **then**
      #single hop
      eval $m.total\_wt, m.arr\_time, m.nb\_tsf, m.add\_dist$
      set match driver $m.d = d\_s$
    else
      #multi hop, CIER heuristic
      Q : set of drivers et rider position nodes
      T : set of candidate transfer nodes
      find : $t \in T$ minimizing aggregated distance from Q and feasible
      eval : $m.total\_wt, m.arr\_time, m.nb\_tsf, m.add\_dist$
      set match drivers : $m.d_0 = d\_s, m.d_1 = d\_e, m.v\_tsf = t$
    end if
    if m minimizes all current best values of the objectives (pareto optimal) **then**
      overwrite $best\_matches := set(m)$ #the only element of the set is now m
      update all objectives : $best\_add\_dist, best\_wt, best\_arrival\_time, best\_nb\_tsf$
    else if m minimizes at least one objective (pareto dominant) **then**
      add m to $best\_matches$
      update concerned objectives
    end if
  end for
end for
$best\_match$ = best match among $best\_matches$ according rider preferences
return $best\_match$

1, correspondant à une capacité de détour de la taille du trajet initial du conducteur. Nous pouvons ainsi générer tout un spectre de comportement de conducteurs, allant du plus égoïste au plus altruiste. Concernant les passagers, nous définissons 5 profils d'utilisateurs présentés dans la Table 1 en fonction des 4 objectifs présentés en section 3.2.

TABLE 1 – Tableau des différents profils des agents passagers

Profil	$total\_wt$	add_dist	arr_time	tsf_nb
Équilibré	3	3	3	1
Pluie	7	1	1	1
Écologique	1	7	1	1
Pressé	1	1	7	1
Confort	1	1	1	7

Chaque valeur de ce tableau représente le poids associé à chaque objectif : plus le poids est élevé, plus le profil a de l'intérêt pour l'objectif concerné. Concrètement, on peut imaginer un curseur ajustant les poids de ces objectifs dans une application smartphone de covoiturage dynamique. Le profil *"équilibré"* représente un passager moyen, sans réelles préférences, et sert de profil de référence. Le profil *"pluie"* représente un agent passager sous la pluie qui souhaite être pris en charge rapidement. Il donne alors la priorité à la minimisation du temps d'attente total du trajet. Le profil *"écologiste"* représente un passager préoccupé par l'impact environnemental lié à la conduite d'une voiture. Ce profil privilégie le partage de trajet qui implique peu de détours et cherche donc à minimiser la distance additionnelle de détour liée au covoiturage. Le profil *"pressé"* représente un passager qui souhaite atteindre sa destination le plus rapidement possible. Ce profil donne la priorité à la minimisation de l'heure d'arrivée à sa destination. Enfin, le profil *"confort"* souhaite éviter le désagrément de devoir changer de véhicule au cours d'un trajet. Ce profil donne la priorité aux covoiturages simple-sauts et minimise donc le nombre de transferts de véhicules.

5.2 Protocole expérimental

Nous avons privilégié l'étude de petites instances à de nombreuses reprises afin d'obtenir des résultats reproductibles et fiables. De cette manière, les biais liés aux paramètres du système sont contrôlés. Nous effectuons une simulation avec une génération continue de conducteurs nous permettant de reproduire une situation réelle où il y a un flux entrant et sortant de conducteurs dans le système. En variant la densité de ce flux, nous pouvons simuler tout un panel de trafics routiers et évaluer les comportements en situation de rareté ou d'abondance de l'offre de covoiturage. La résolution de l'instance se termine lorsque tous les agents passagers ont quitté le système : en atteignant leur destination ou en s'impatientant.

Nos instances sont composées d'un graphe de 150 nœuds provenant d'un réseau routier réel de la ville de San Francisco, extrait d'Open Street Map et modélisé à l'aide de la bibliothèque OSMNX [5]. Ce graphe est illustré dans la Figure 4. Sur ce graphe en grille, 30 conducteurs et 20 passagers interagissent. Leurs nœuds de départ et de destination sont générés de manière aléatoire et uniforme. Les itinéraires initiaux des conducteurs sont dérivés du plus court chemin entre leurs nœuds de départ et d'arrivée. Nous supposons que les passagers ont un temps d'attente maximal de 3 minutes et un facteur de détour fixé à 0,2 pour créer une situation difficile.

5.3 Résultats

Dans cette sous-section, nous détaillons les résultats obtenus avec notre modèle. L'objectif est

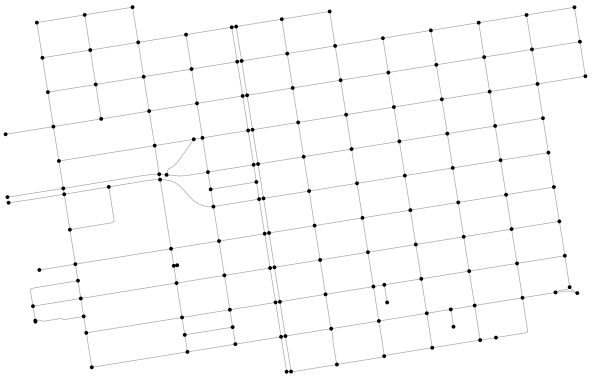

FIGURE 4 – Le graphe étudié : une section du côté nord de la ville de San Francisco

de mettre en évidence l'impact des différentes stratégies sur des indicateurs tels que : le temps d'attente moyen, le taux de service (succès du covoiturage), la distance additionnelle moyenne et le nombre moyen de véhicules nécessaires au covoiturage. Chaque figure comprend plusieurs courbes correspondant aux profils des passagers mentionnés précédemment. Chaque point d'une courbe est le résultat de 50 expériences moyennées. Ces expériences correspondent aux 50 mêmes expériences pour chaque profils de passager. Enfin, l'axe des abscisses est indexé par le facteur de détour des conducteurs.

L'impact du profil "écologiste". La Figure 5 montre comment le profil "écologiste" (courbe verte/triangles) minimise la distance additionnelle moyenne de détour des conducteurs par rapport aux autres profils. Ce phénomène est amplifié par l'augmentation du facteur de détour des conducteurs. On observe ainsi des détours 2 à 3 fois moins importants avec ce profil qu'avec les profils concurrents. En conséquence, les autres objectifs sont très largement négligés par ce profil (Figures 6, 7, 8).

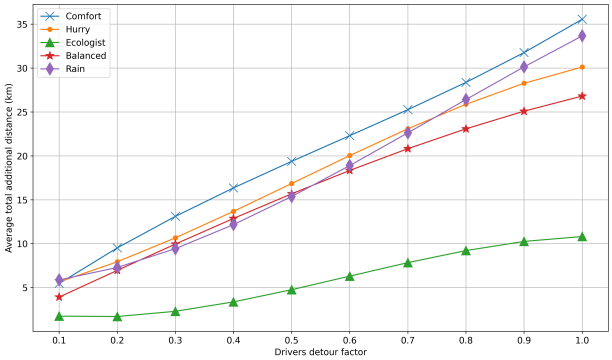

FIGURE 5 – Distance additionnelle moyenne de détour des conducteurs

L'impact du profil "confort". Le profil "confort" (courbe bleue/croix) dans la Figure 6, vise à minimiser le nombre de transferts (changements de véhicules) pour atteindre la destination cible. On constate que ce profil est capable de minimiser sa cible de plus en plus (environ de 1.4 à 1.1 véhicule impliqué en moyenne) avec l'augmentation du facteur de détour des conducteurs. Il est également bien meilleur que les autres profils en montrant entre 20% et 70% de performance supplémentaire sur cet objectif. En revanche, il est le plus faible pour la distance de détour supplémentaire (Figure 5) et est similaire à la majorité des autres profils pour les autres objectifs (Figures 7 et 8). Nous pouvons conclure ici que la réduction du nombre de transferts entre véhicules et la favorisation du covoiturage simple-saut augmente la distance de détour.

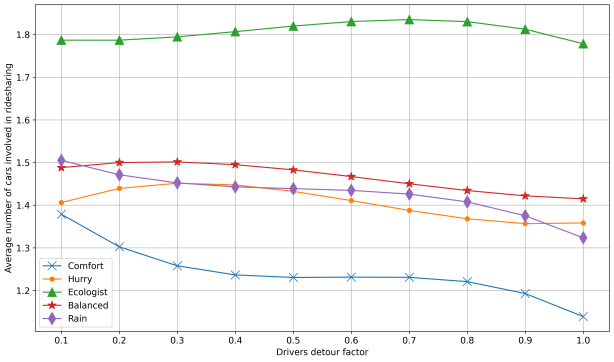

FIGURE 6 – Nombre moyen de véhicules impliqués dans le covoiturage

L'impact des profils "pluie", "pressé" et "équilibré". Nous avons choisi de traiter ces trois profils conjointement, car, dans la majorité des cas, la tendance des courbes et leurs résultats sont similaires. Cela s'explique par la difficulté de dissocier le temps de parcours, le temps d'attente et la distance de parcours. Ces derniers sont en effet étroitement liés et interdépendants. Néanmoins, le profil "pluie" (courbe mauve/losanges) réussit à réduire son temps d'attente (Figure 7) quand les conducteurs sont peu disposés à effectuer des détours (facteur inférieur à 0.5). Il se confond ensuite avec les autres profils.

Le profil "Pressé", cherchant à arriver le plus rapidement possible, vise indirectement à réduire son temps d'attente. C'est pourquoi nous le voyons se positionner très légèrement comme le leader de la minimisation du temps d'attente sur la Figure 7 pour un facteur de détour du conducteur de 0.5 à 0.9. Notre profil "équilibré"

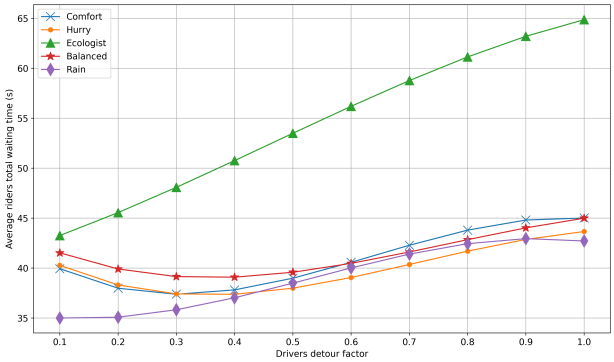

FIGURE 7 – Temps d'attente moyen d'un passager

remplit parfaitement son objectif d'être le juste milieu entre les différents profils et leurs objectifs, comme on peut le voir sur la totalité des figures.

La performance globale des profils : le taux de service. Pour conclure sur l'analyse de ces résultats, nous pouvons considérer la performance des profils du point de vue du taux de service moyen (Figure 8). Le taux de service représente la part des passagers qui ont réussi à se rendre à leur destination en covoiturant.

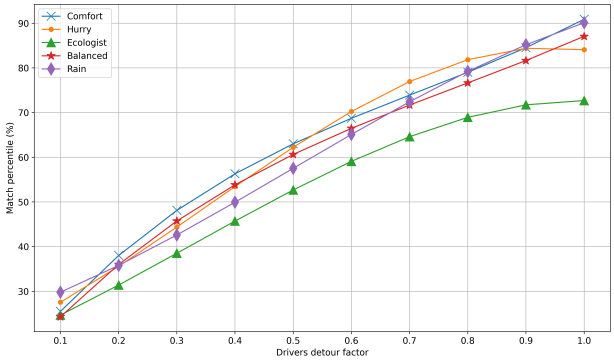

FIGURE 8 – Taux de service moyen du système

On constate tout de suite que plus le facteur de détour augmente, plus il existe de solutions potentielles et donc plus le taux de service augmente. Ce phénomène est normal et démontre le bon fonctionnement de notre modèle. Nous remarquons que le profil "écologiste" (triangles verts) présente un taux de service relativement faible par rapport au reste des profils avec une différence moyenne de 5 à 15%. Aussi, il représente le pire taux de service atteint avec environ 72% pour un facteur de détour des conducteurs de 1. Si on émet l'hypothèse que les utilisateurs qui souhaitaient être passagers ont finalement pris leur véhicule pour rejoindre leur destination, il se peut que ce profil ne permette pas de réduire, autant qu'espéré, les émissions globales du système, curieux paradoxe !

6 Conclusion et perspectives

Dans cet article, nous proposons un système multi-agent pour optimiser le covoiturage dynamique en fonction des préférences des conducteurs et des passagers. Nous modélisons les différents acteurs d'un système de covoiturage et leur spectre de comportements possibles. Nous apportons une approche originale, basée sur des techniques d'indexation spatiale, pour modéliser la perception de nos agents et pour réduire le coût computationnel associé à la recherche du covoiturage optimal. Les expériences présentées ont montré que l'approche proposée permet une optimisation individuelle et efficace.

Cette approche révèle certaines limitations pour les profils favorisant le temps d'attente et le temps de trajet. En effet, l'optimisation par agrégation des pondérations semble insuffisante lorsque les objectifs sont interdépendants. Pour répondre à ce constat, d'autres techniques peuvent être envisagées, comme l'agrégation par l'intégrale de Choquet, qui s'est avérée efficace pour l'optimisation multicritère [4]. Une autre perspective consiste à rendre le comportement des agents dynamique. En effet, le comportement des agents est pour l'instant défini à leur apparition et ne change pas au cours du temps. En fonction de l'état de leur environnement (trafic, météo, pénurie d'offres ou de demandes, etc.), leur comportement pourrait évoluer et les mener à changer leurs objectifs principaux [2]. En étendant notre approche à des agents au comportement dynamique, nous nous rapprocherions encore davantage du monde réel et de ses caractéristiques.

Des critères et mécanismes tels que le nombre d'arrêts maximum souhaité par un conducteur ou encore la possibilité de réserver un seul covoiturage pour plusieurs personnes sont essentiels dans le cadre d'une mise en application de notre solution. Il seront inclus et étudiés dans nos prochains articles.

Enfin, il serait fructueux de considérer également la complémentarité avec d'autres modalités de transport, en particulier les transports publics tels que le bus et le métro [20, 11]. Nous passerions alors de la mono-modalité à la comodalité, c'est-à-dire la multimodalité associée au covoiturage.

Références

[1] N. Agatz, A. L. Erera, M. W. Savelsbergh, and X. Wang. Dynamic Ride-Sharing : A Simulation Study in Metro Atlanta. *Procedia - Social and Behavioral Sciences*, 17 :532–550, 2011.

[2] F. Balbo and S. Pinson. Dynamic modeling of a disturbance in a multi-agent system for traffic regulation. *Decision Support Systems*, 41(1) :131–146, 2005.

[3] H. A. N. C. Bandara and D. Dias. A multi-agent system for dynamic ride sharing. In *2009 International Conference on Industrial and Information Systems (ICIIS)*, page 199–203. IEEE, Dec 2009.

[4] S. Ben Cheikh-Graiet, M. Dotoli, and S. Hammadi. A Tabu Search Based Metaheuristic for Dynamic Carpooling Optimization. *Computers & Industrial Engineering*, Feb. 2020.

[5] G. Boeing. OSMnx : New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Computers, Environment and Urban Systems*, 65 :126–139, Sept. 2017.

[6] Z. Chen, B. Yao, Z.-J. Wang, X. Gao, S. Shang, S. Ma, and M. Guo. Flexible Aggregate Nearest Neighbor Queries and its Keyword-Aware Variant on Road Networks. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1, 2020.

[7] J.-F. Cordeau and G. Laporte. The dial-a-ride problem : Models and algorithms. *Ann Oper Res*, page 18, 2007.

[8] C. Cortés, M. Matamala, and C. Contardo. The pickup and delivery problem with transfers. *European Journal of Operational Research*, 200(3) :711–724, Feb. 2010.

[9] A. Daoud, F. Balbo, P. Gianessi, and G. Picard. Un modèle agent générique pour la comparaison d'approches d'allocation de ressources dans le domaine du transport à la demande. In *JFSMA29*, pages 127–136. Cépaduès, 2021.

[10] A. Di Febbraro, E. Gattorna, and N. Sacco. Optimization of Dynamic Ridesharing Systems. *Transportation Research Record : Journal of the Transportation Research Board*, 2359(1) :44–50, Jan. 2013.

[11] A. O. Diallo, G. Lozenguez, A. Doniec, and R. Mandiau. Usage des parkings relais dans les comportements de déplacements intermodaux. In *JFSMA29, June 28-30, 2021*, pages 83–92. Cépaduès, 2021.

[12] J. Ferber. Multi-agent systems : An introduction to distributed artificial intelligence, 1999.

[13] C. Fevre, H. Zgaya-Biau, P. Mathieu, and S. Hammadi. Multi-agent Systems and R-Trees for Dynamic and Optimised Ridesharing. In *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 1352–1358, Oct. 2021.

[14] C. Fevre, H. Zgaya-Biau, P. Mathieu, and S. Hammadi. Preferential Optimization of Multi-hop Dynamic Ridesharing based on R-Trees and Multi-Agent Systems. *Under review*, 2022.

[15] A. Guttman. *R Trees : A Dynamic Index Structure for Spatial Searching*, volume 14. Jan. 1984.

[16] W. Herbawi and M. Weber. Evolutionary Multiobjective Route Planning in Dynamic Multi-hop Ridesharing. In *Evolutionary Computation in Combinatorial Optimization*, pages 84–95, Berlin, Heidelberg, 2011. Springer.

[17] W. Herbawi and M. Weber. Modeling the Multi-hop Ridematching Problem with Time Windows and Solving It Using Genetic Algorithms. *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI*, 1 :96, Nov. 2012.

[18] K. Jeribi, H. Mejri, H. Zgaya, and S. Hammadi. Distributed graphs for solving co-modal transport problems. In *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 1150–1155, Oct. 2011.

[19] E. Kamar and E. Horvitz. Collaboration and Shared Plans in the Open World : Studies of Ridesharing. In *IJCAI International Joint Conference on Artificial Intelligence*, page 187, Jan. 2009.

[20] J. Lin, S. Sasidharan, S. Ma, and O. Wolfson. A Model of Multimodal Ridesharing and Its Analysis. In *2016 17th IEEE International Conference on Mobile Data Management (MDM)*, pages 164–173, Porto, June 2016. IEEE.

[21] S. Ma and O. Wolfson. Analysis and evaluation of the slugging form of ridesharing. In *SIGSPATIAL'13*, pages 64–73, Orlando, Florida, 2013. ACM Press.

[22] Y. Manolopoulos, Y. Theodoridis, and V. J. Tsotras. Spatial Indexing Techniques. In *Encyclopedia of Database Systems*, pages 1–7. Springer New York, New York, NY, 2014.

[23] N. Masoud and R. Jayakrishnan. A real-time algorithm to solve the peer-to-peer ride-matching problem in a flexible ridesharing system. *Transportation Research Part B : Methodological*, 106 :218–236, Dec. 2017.

[24] P. Mathieu and A. Nongaillard. Effective evaluation of autonomous taxi fleets. In *ICAART (1)*, pages 297–304, 2018.

[25] D. Papadias, Y. Tao, K. Mouratidis, and C. K. Hui. Aggregate nearest neighbor queries in spatial databases. *ACM Transactions on Database Systems*, 30(2) :529–576, June 2005.

[26] D. Papadias, J. Zhang, N. Mamoulis, and Y. Tao. Query Processing in Spatial Network Databases. In *Proceedings 2003 VLDB Conference*, pages 802–813. Elsevier, 2003.

[27] V. Pareto. *Cours d'économie politique*. Librairie Droz, 1964.

[28] Shuo Ma, Yu Zheng, and O. Wolfson. T-share : A large-scale dynamic taxi ridesharing service. In *ICDE29'*, pages 410–421, Brisbane, QLD, Apr. 2013. IEEE.

[29] A. Tafreshian and N. Masoud. Trip-based graph partitioning in dynamic ridesharing. *Transportation Research Part C : Emerging Technologies*, 114 :532–553, May 2020.

[30] Y. Xu, L. Kulik, R. Borovica-Gajic, A. Aldwyish, and J. Qi. Highly Efficient and Scalable Multi-hop Ride-sharing. In *Proceedings of the 28th International Conference on Advances in Geographic Information Systems*, pages 215–226, Seattle WA USA, Nov. 2020. ACM.

[31] M. Yiu, N. Mamoulis, and D. Papadias. Aggregate nearest neighbor queries in road networks. *IEEE Transactions on Knowledge and Data Engineering*, 17 :820–833, July 2005.