

République Algérienne démocratique et Populaire

Ministère de l'enseignement supérieur et de la recherche scientifique

Université de 08 Mai 1945 – Guelma

Faculté des Mathématiques, d'Informatique et des Sciences de la Matière

Département d'Informatique



Mémoire de Fin d'étude Master

Filière : Informatique

**Option : Sciences et Technologies de l'information et de la
Communication**

Thème :

**Appariement multi-objectif dans les systèmes de covoiturage
dynamique**

Présenté par : Boussahoul Sara

Membres du jury :

N°	Nom et Prénom	Qualité
1	Mme. Baalia Saida	Président
2	Mme. Abdelmoumène Hiba	Superviseur
3	Mme. Benhamza Karima	Examineur

Juin 2024

Remerciement

*Tout d'abord, je tiens à remercier **Dieu** pour la santé, la passion et la clarté d'esprit ; pour m'avoir guidée dans l'accomplissement de ce travail "Al hamdouli allah".*

Je souhaite exprimer ma gratitude envers mon encadrante, Dr. Abdelmoumène Hiba, pour avoir cru en moi, m'avoir guidée tout au long de ce travail et m'avoir encouragée. Merci pour votre soutien.

Je tiens à remercier les membres du jury d'avoir consacré leur temps précieux et leurs efforts pour lire et évaluer mon travail.

Mes plus grands remerciements vont à ma mère et à mon père, qui ont toujours été à mes côtés. J'espère vous avoir rendus fiers. Merci pour tous vos encouragements et vos prières ; aucun mot ne peut décrire à quel point je vous suis reconnaissante.

Je tiens à remercier aussi ma famille pour son soutien constant et ses prières.

Je remercie chaleureusement mes amies, Manar, Randa et Chayma, avec qui j'ai partagé les bons moments de l'université ainsi que les mauvais.

Enfin, je veux remercier tous ceux qui m'ont encouragée pendant cette période. Votre soutien a été inestimable.

Dédicace

À mes chers parents, dont le soutien indéfectible et les encouragements m'ont guidée tout au long de ce parcours. Vous avez toujours cru en moi, me réconfortant avec les mots :

"Tu y es presque, et tu vas y arriver" – et en effet, j'y suis arrivée.

À toute ma famille, pour avoir été mes côtés à chaque étape de ce parcours.

À mon cher grand-père, bien que tu ne sois pas là pour voir ce moment, tu es toujours présent dans mes pensées.

Aux nuits sans sommeil, alimentées par la caféine et le thé, chacune étant une preuve de dévouement et de détermination.

A moi-même, pour ne jamais abandonner, pour tirer des leçons de chaque échec, et pour persévérer jusqu'à réussir.

Ce mémoire est dédié à tous ceux qui m'ont soutenue et cru en moi.

Résumé

Le covoiturage dynamique représente une solution innovante aux défis de mobilité urbaine moderne. Ce système permet aux utilisateurs de trouver des trajets en temps réel en mettant en relation les conducteurs et les passagers partageant un itinéraire similaire. Les utilisateurs bénéficient ainsi du partage des frais de transport et réduisent l'utilisation des voitures individuelles, contribuant ainsi à une diminution globale de l'utilisation des véhicules.

Dans ce travail, nous nous sommes concentrées sur la problématique complexe de l'appariement dynamique. Pour résoudre ce défi, nous avons opté pour une méthode d'apprentissage par renforcement multi-objectifs. Cette solution prend en compte un ensemble de contraintes cruciales : contraintes spatio-temporelles, contraintes de capacité, contraintes de temps d'attente, contraintes de distance et contraintes de temps de détour. Notre modèle vise à atteindre trois objectifs principaux : minimiser le temps d'attente des passagers, réduire le temps de détour des conducteurs et maximiser l'utilisation des véhicules.

Pour valider notre modèle, nous avons exploité des données réelles issues de l'ensemble de données publiques des taxis de New York City. De plus, nous avons développé un simulateur afin d'évaluer les performances de notre approche. Les résultats obtenus sont prometteurs. Ces résultats positifs soulignent le potentiel de notre système de covoiturage dynamique pour améliorer l'expérience des conducteurs et des passagers, en offrant une solution flexible et robuste.

Mots clés

Covoiturage dynamique, Appariement dynamique, Apprentissage par renforcement multi-objectifs (MORL), Optimisation multi-objectifs, Contraintes.

Abstract

Dynamic ridesharing represents an innovative solution to the challenges of modern urban mobility. This system allows users to find rides in real-time by connecting drivers and passengers sharing a similar route. Users benefit from shared transportation costs and it reduces the use of individual cars, thus contributing to an overall decrease in vehicle usage.

In this work, we focused on the complex issue of dynamic matching. To address this challenge, we opted for a multi-objective reinforcement learning method. This solution takes into account a set of crucial constraints: spatiotemporal constraints, capacity constraints, waiting time constraints, distance constraints, and detour time constraints. Our model aims to achieve three main objectives: minimize passenger waiting time, reduce driver detour time, and maximize vehicle utilization.

To validate our model, we used real data from the public New York City taxi dataset. Additionally, we developed a simulator to evaluate the performance of our approach. The results obtained are promising. These positive results highlight the potential of our dynamic ridesharing system to enhance the experience of both drivers and passengers, providing a flexible and robust solution.

Keywords

Dynamic ridesharing, Dynamic matching, Multi-Objective Reinforcement Learning (MORL), Multi-objective optimization, Constraints.

ملخص

يمثل نظام مشاركة الرحلات الديناميكي حلاً مبتكراً لتحديات التنقل الحضري الحديث. يتيح هذا النظام للمستخدمين العثور على الرحلات في الوقت الفعلي من خلال ربط السائقين والركاب الذين يشاركون طريقاً مشابهاً. يستفيد المستخدمون من تكاليف النقل المشتركة ويقلل من استخدام السيارات الفردية، مما يساهم في انخفاض عام في استخدام المركبات.

في نهجنا، اخترنا التركيز على القضية المعقدة المتمثلة في المطابقة الديناميكية. ولمواجهة هذا التحدي، اخترنا التعلم المعزز متعددة الأهداف. وبأخذ هذا الحل في الاعتبار مجموعة من القيود الحاسمة: القيود الزمانية المكانية، وقيود القدرة، وقيود وقت الانتظار، وقيود مسافة التحويل، وقيود وقت التحويل. يهدف نموذجنا إلى تحقيق ثلاثة أهداف رئيسية: تقليل وقت انتظار الركاب، وتقليل وقت تحويل السائق، وزيادة استخدام السيارة إلى أقصى حد.

للتحقق من صحة نموذجنا، استخدمنا بيانات حقيقية من مجموعة بيانات سيارات الأجرة العامة في مدينة نيويورك. بالإضافة إلى ذلك، قمنا بتطوير جهاز محاكاة لتقييم أداء نهجنا. النتائج التي تم الحصول عليها واعدة بشكل خاص. تسلط هذه النتائج الإيجابية الضوء على إمكانات نظامنا الديناميكي لمشاركة الرحلات في تحسين التنقل الحضري بشكل كبير من خلال تقديم حل مرن واقتصادي وبيئي لتحديات النقل الحديث.

كلمات مفتاحية

مشاركة الرحلات الديناميكية، المطابقة الديناميكية، التعلم المعزز متعدد الأهداف (MORL)، التحسين متعدد الأهداف، قيود.

Table des matières

Introduction générale.....	11
Chapitre 1 : Etat de l'art.....	13
1.1 Introduction	13
1.2 Covoiturage	13
1.2.1 Covoiturage Dynamique.....	15
1.2.1.1 Définition.....	15
1.2.1.2 Composants d'un système de covoiturage dynamique.....	16
1.2.1.3 Variantes du covoiturage dynamique.....	17
1.2.1.4 Contraintes sur le covoiturage dynamique.....	19
1.2.1.5 Objectifs à optimiser.....	21
1.2.2 Problématique de l'appariement dynamique	24
1.3 Apprentissage par renforcement multi-objectifs	25
1.3.1 Composants de l'apprentissage par renforcement multi-objectifs (MORL)	28
1.4 Travaux connexes.....	29
1.5 Conclusion.....	35
Chapitre 2 : Contribution	36
2.1. Introduction	36

2.2. Problématique et objectifs de notre approche	36
2.1.1. Problématique.....	36
2.1.2. Objectifs	37
2.2. Description du problème	38
2.2.1. Offres.....	38
2.2.2. Demandes	39
2.2.3. Contraintes.....	39
2.2.3.1. Contrainte spatio-temporelle.....	39
2.2.3.2. Contrainte de capacité.....	40
2.2.3.3. Contrainte du temps d'attente	40
2.2.3.4. Contrainte du temps de détour	40
2.2.3.5. Contrainte de distance de détour.....	40
2.3. Modélisation proposée	41
2.3.1. Etat.....	42
2.3.2. Observation.....	43
2.3.3. Action	44
2.3.4. Récompense.....	45
2.3.5. Episode	47
2.3.6. Transition d'état.....	47
2.4. Phase d'apprentissage	47
2.4.1. Décision de l'agent	48

2.4.2. Apprentissage	52
2.5. Conclusion.....	53
Chapitre 3 : Implémentation	54
3.1. Introduction	54
3.2. Langages et outils de développement.....	54
3.3. Echantillon de données.....	55
3.3.1. Données des trajets	56
3.4. Apprentissage	58
3.5. Simulateur	60
3.6. Expérimentations.....	67
3.7. Conclusion.....	72
Conclusion générale	73
Bibliographie	75
Webographie.....	81

Liste des figures

Figure 1.1 : Éléments d'un système de covoiturage.	17
Figure 1.2 : Représentation de deux scénarios de covoiturage dynamique : avec détour (scénario à droite) et sans détour (scénario à gauche) (Martins, de la Torre, Corlu, Juan, & Masmoudi, 2021)	21
Figure 1.3 : Objectifs d'optimisation.	22
Figure 1.4 : Comparaison entre MDP et MOMDP.	27
Figure 2.1 : Exemple démonstratif d'un problème d'appariement.	41
Figure 2.2 : Exemple d'observation.	44
Figure 2.3 : Exemple de récompense.	46
Figure 2.4 : Processus de décision de l'agent.	50
Figure 3.1 : Résultat d'une requête de l'API MapQuest Directions.	57
Figure 3.2 : Perte dans l'apprentissage.	58
Figure 3.3 : Hypervolume.	59
Figure 3.4 : Interface du simulateur.	61
Figure 3.5 : Résultat de la simulation.	62
Figure 3.6 : Visualisation des résultats de la simulation.	63
Figure 3.7 : Types des demandes acceptées.	65
Figure 3.8 : Exécution du conducteur 8.	67

Liste des tableaux

Tableau 1.1 : Variantes du covoiturage.	19
Tableau 1.2 : Classification des travaux connexes en se basant sur les problèmes considérés et le type du RL.	33
Tableau 1.3 : Classification des travaux connexes selon le modèle RL proposé.	35
Tableau 3.1 : Exemples des tests.	60
Tableau 3.2 : Paramètres de la simulation.	61
Tableau 3.3 : Résultats de la simulation.	63
Tableau 3.4 : Echantillon des conducteurs.	65
Tableau 3.5 : Echantillon des passagers.	66
Tableau 3.6 : Résultats d'appariement dynamique des échantillons des tableaux 3.4 et 3.5.	66
Tableau 3.7 : Résultats du premier scénario.	69
Tableau 3.8 : Résultats du deuxième scénario.	70
Tableau 3.9 : Résultats du troisième scénario.	71

Introduction générale

Les déplacements quotidiens sont devenus une nécessité incontournable de la vie moderne en raison des différentes activités. L'urbanisation croissante des villes a entraîné un allongement des distances. Face à cette réalité, les personnes doivent utiliser divers moyens de transport, allant des transports publics aux véhicules personnels. Cependant, ces solutions présentent des inconvénients. En effet, les transports publics limitent la liberté des passagers, tandis que les véhicules personnels consomment beaucoup d'argent et contribuent aux embouteillages, ce qui aggrave la pollution atmosphérique et le changement climatique.

Ces dernières années, le covoiturage est devenu une alternative prometteuse, permettant aux passagers de réduire leurs dépenses tout en offrant une certaine liberté de déplacement. De plus, il est également bénéfique pour l'environnement, car il contribue à diminuer le nombre de véhicules sur la route, réduisant ainsi les embouteillages et les émissions de gaz à effet de serre. C'est un mode de transport collaboratif où les individus partagent un même véhicule pour un trajet commun.

Le covoiturage est perçu comme un problème d'optimisation visant à trouver la meilleure combinaison possible entre les individus désireux de partager un trajet, tout en atteignant des objectifs spécifiques et en tenant compte de diverses contraintes telles que des contraintes temporelles et des contraintes spatiales. Pour résoudre ce problème, il est nécessaire d'utiliser des algorithmes sophistiqués qui prennent en compte de diverses variables telles que les lieux, les horaires, la capacité des véhicules et bien d'autres contraintes des conducteurs et des passagers, tout en gérant les données en temps réel et en optimisant un ensemble des objectifs qui peuvent concerner le temps et les distances.

C'est dans ce cadre que s'inscrit notre travail où nous nous concentrons sur le développement d'un système capable d'optimiser les appariements dynamiques entre les offres des conducteurs et les demandes des passagers. Le problème que nous considérons va permettre à un conducteur de s'apparier à plusieurs passagers tout en faisant des détours si c'est nécessaire. Nous visons à optimiser trois objectifs distincts et conflictuels. En effet, notre

solution doit trouver un équilibre entre la minimisation du temps d'attente des passagers, la minimisation du temps de détour des conducteurs et la maximisation de l'utilisation des véhicules. Cet équilibre doit aussi prendre en considération des contraintes temporelles, des contraintes spatiales et des contraintes de capacité.

Pour remédier à cette problématique nous nous sommes orientées vers l'apprentissage par renforcement (Reinforcement Learning – RL) et plus précisément l'apprentissage par renforcement multi-objectifs (Multi-Objective Reinforcement Learning – MORL). Notre choix se justifie par l'aspect dynamique du covoiturage où les offres et les demandes évoluent en temps réel ce qui peut être supporté par MORL qui permet de s'adapter aux changements en apprenant continuellement de nouvelles stratégies basées sur les interactions en temps réel. De plus, MORL est spécifiquement conçu pour traiter des problèmes avec plusieurs objectifs conflictuels. Il permet de trouver un ensemble de solutions optimales de Pareto, offrant ainsi une flexibilité pour équilibrer les différents objectifs. Cela permet d'adopter des politiques qui équilibrent efficacement le temps d'attente, le détour et l'utilisation des véhicules.

Ce mémoire est organisé comme suit :

Chapitre 1 : Ce chapitre propose une vue d'ensemble du covoiturage, en soulignant ses éléments essentiels, ses contraintes et les divers objectifs à optimiser. Nous y exposons également les principes fondamentaux de l'apprentissage par renforcement multi-objectifs et ses composantes clés. Enfin, nous procédons à une revue des travaux connexes.

Chapitre 2 : Ce chapitre présente notre contribution. Nous débutons par exposer notre problématique, nos motivations et les justifications sous-jacentes à nos choix. Ensuite, nous introduisons la modélisation que nous avons proposée et détaillons les différentes étapes de sa mise en œuvre.

Chapitre 3 : Ce chapitre examine en détail l'implémentation de notre système. Nous présentons les choix effectués pour son développement. Ensuite, nous discutons les résultats obtenus à travers nos expérimentations et les évaluations réalisées pour mesurer l'efficacité de notre système.

Chapitre 1 : Etat de l'art

1.1 Introduction

Dans une ère où les déplacements quotidiens sont devenus inévitables, générant d'importants problèmes tels que la congestion, la circulation dense, les émissions de gaz à effet de serre et les coûts individuels, le covoiturage émerge comme une alternative prometteuse et novatrice face aux défis croissants de la mobilité urbaine.

Le covoiturage, également désigné par le terme anglais "*ridesharing*," se profile comme une solution durable (pour atténuer les problèmes liés aux déplacements urbains). Ce concept repose sur le partage de trajets entre des individus se dirigeant vers la même destination, visant ainsi à optimiser l'utilisation des véhicules tout en réduisant le nombre de voitures circulant sur les routes.

Dans ce chapitre, nous allons tout d'abord exposer les aspects théoriques du covoiturage dynamique, ses composantes ainsi que ses caractéristiques. Ensuite, nous examinerons les travaux qui ont traité la problématique de l'appariement dynamique dans le covoiturage à travers l'apprentissage par renforcement.

1.2 Covoiturage

Ballet et al., (2007) définissent le covoiturage comme : « *Le covoiturage consiste en l'utilisation commune d'un véhicule par un conducteur non professionnel et un (ou plusieurs) passager(s) dans le but d'effectuer tout ou une partie d'un trajet commun.* ».

Un système de covoiturage implique plusieurs requêtes de trajet caractérisées par des points de départ et d'arrivée, ainsi qu'un ensemble de véhicules avec leurs trajets. L'objectif est d'optimiser les itinéraires afin de satisfaire au mieux ces demandes système (Silwal, Gani, & Raychoudhury, 2019).

Le covoiturage peut être désigné par les termes anglais "*ridesharing*" ou "*carpooling*". Les deux termes désignent une forme de transport où des conducteurs offrent des sièges libres dans leur véhicule à des passagers qui vont dans la même direction. La principale différence entre ces deux modes est la pré-planification. En effet, dans le *carpooling*, les trajets sont souvent organisés à l'avance avec des participants qui se connaissent ou qui se sont mis d'accord sur les détails du trajet. Le *ridesharing*, peut être plus flexible et dynamique, souvent organisé via des applications mobiles en temps réel. Dans notre étude, par covoiturage nous faisons référence au *ridesharing*.

Selon Edenhofer et al. (2014), les émissions de gaz à effet de serre du secteur des transports ont plus que doublé depuis 1970 et ont augmenté à un rythme plus rapide que tout autre secteur d'utilisation finale de l'énergie, représentant environ 23 % du total des émissions de CO₂ liées à l'énergie. D'après les données de l'Agence Internationale de l'Energie (AIE) pour l'année 2022, les émissions mondiales de CO₂ liées à l'énergie ont enregistré une hausse de 0,9 %, ce qui représente une augmentation de 321 millions de tonnes, atteignant ainsi un niveau record de plus de 36,8 milliards de tonnes [1]. Dans ce contexte, le covoiturage se présente comme une solution potentielle pour atténuer ces émissions en réduisant le nombre de voitures utilisées pour les déplacements individuels. En plus de ses bénéfices environnementaux, le covoiturage contribue à soulager les embouteillages et à atténuer les autres effets négatifs d'une circulation dense. Il offre également des avantages directs aux utilisateurs, notamment en réduisant les dépenses de transport et en diminuant les temps d'attente pour trouver un taxi.

Dans un système de covoiturage, plusieurs approches ont été développées pour créer des systèmes de mobilité performants, offrant aux voyageurs la possibilité de partager leurs trajets et d'améliorer leur expérience de voyage. Cette diversité d'approches permet de classer la planification des trajets en covoiturage en deux catégories principales : le covoiturage "*statique*" et le covoiturage "*dynamique*" (Agatz, Erera, Savelsbergh, & Wang, 2012).

Le covoiturage statique consiste à organiser les conducteurs et les passagers avant le début du trajet. Une fois que le conducteur a entamé son voyage, aucune autre association de passagers n'est possible pour ce trajet où l'itinéraire préalablement calculé est immuable ou sélectionné par le chauffeur lui-même, et les heures de départ et d'arrivée sont également connues à l'avance. En contraste, le covoiturage dynamique se caractérise par la mise en relation en temps réel des conducteurs et des passagers, où les itinéraires sont flexibles et

d'autres passagers peuvent se joindre même après le début d'un voyage, permettant ainsi des modifications instantanées de l'itinéraire (Plate, 2019; Cheikh & Hammadi, 2016).

1.2.1 Covoiturage Dynamique

1.2.1.1 Définition

Le covoiturage dynamique, également connu sous le nom de covoiturage en temps réel ou covoiturage instantané, est un système automatisé qui facilite le partage de trajets entre conducteurs et passagers en favorisant les départs proches de l'heure souhaitée. Cette forme de covoiturage repose sur l'utilisation de la technologie de communication, permettant aux utilisateurs d'organiser rapidement des trajets via une plateforme, offrant ainsi une solution de transport en quasi temps réel (Agatz, Erera, Savelsbergh, & Wang, 2010). Un aspect essentiel du covoiturage dynamique est le partage équitable des coûts entre les participants. Les coûts variables associés au voyage sont répartis de manière à ce que tous les membres bénéficient d'une réduction significative des coûts. Cette pratique encourage la participation des utilisateurs en offrant une alternative économique et efficace aux déplacements individuels (Agatz, Erera, Savelsbergh, & Wang, 2012).

Le covoiturage dynamique offre une plus grande souplesse par rapport à la méthode conventionnelle ou statique de covoiturage, en faisant ainsi une solution efficace dans le monde moderne. Les systèmes de covoiturage dynamique peuvent être conçus de manière non centralisée, distribuée, ou sous forme de systèmes hybrides. Ces derniers permettent d'optimiser les itinéraires avec un coût minimal pour le fournisseur du système (Silwal, Gani, & Raychoudhury, 2019).

Le covoiturage implique la mise en relation de personnes partageant des horaires similaires afin de partager un même trajet en voiture, tout en respectant des contraintes spatio-temporelles spécifiques. Cette démarche est complexe, car elle nécessite de trouver des correspondances entre conducteurs et passagers, malgré leurs exigences différentes. Les passagers cherchent un trajet précis avec un départ et une arrivée déterminée, tandis que les conducteurs ont un itinéraire fixe et une heure de départ préétablie. Les plateformes de covoiturage doivent donc être flexibles pour permettre des ajustements d'itinéraire afin de prendre et de déposer les passagers. Coordonner efficacement les horaires, les parcours et les

préférences des utilisateurs constitue un défi majeur dans la mise en place et la gestion des systèmes de covoiturage (Martins, de la Torre, Corlu, Juan, & Masmoudi, 2021).

1.2.1.2 Composants d'un système de covoiturage dynamique

Dans un système de covoiturage, un conducteur prévoit de faire un trajet en voiture. Avant de partir, il se connecte à une plateforme de covoiturage en ligne pour partager son offre de trajet. De l'autre côté, plusieurs passagers recherchent des trajets pour des destinations différentes mais situées sur le chemin du conducteur. Ils se connectent également à la plateforme de covoiturage et partagent leurs demandes de trajet. Le système de covoiturage agit comme un intermédiaire efficace, faisant automatiquement correspondre l'offre du conducteur avec les demandes des passagers, en tenant compte des contraintes d'appariement. Ce processus prend en compte diverses contraintes d'appariement, telles que les horaires, ainsi que les préférences des utilisateurs. Une fois que tous les passagers ont été déposés à leurs destinations respectives, le conducteur peut poursuivre son trajet jusqu'à son point d'arrivée final. Cette collaboration mutuellement bénéfique permet non seulement d'optimiser l'utilisation des véhicules, mais aussi de réduire les coûts et l'empreinte carbone associés aux déplacements individuels (Mourad, Puchinger, & Chengbin, 2019). Les principaux éléments d'un système de covoiturage sont :

- **Conducteur (Covoitureur) :** la personne qui met à disposition son véhicule afin de partager un trajet commun avec une ou plusieurs personnes (Sghaier, 2011; Mitropoulos, Kortsari, & Ayfantopoulou, 2021).
- **Offre du conducteur :** est publiée par un conducteur qui souhaite partager son trajet. Elle détaille les paramètres du déplacement, tels que l'origine, la destination, l'heure de départ et le nombre de places disponibles dans le véhicule utilisé, etc.
- **Passager (Covoituré) :** La personne qui émet une demande de covoiturage afin d'être transportée et bénéficie du trajet proposé par le conducteur. En utilisant les services du conducteur, elle se déplace vers une destination commune ou une direction similaire. En contrepartie, le passager partage habituellement les frais de voyage avec le conducteur, ce qui favorise une répartition équitable des coûts (Sghaier, 2011).
- **Demande du passager :** Ce sont des requêtes postées par les passagers souhaitant partager un trajet. Elles contiennent des informations nécessaires pour organiser le

covoiturage, telles que le lieu de départ, la destination, les horaires de départ et d'arrivée, etc.

- **Un moteur d'appariement dynamique :** Pour faire correspondre une demande à une offre, il est nécessaire de respecter un ensemble de contraintes. Ces contraintes concernent principalement l'alignement entre l'offre et la demande en termes d'horaires de départ et d'arrivée, de lieux de départ et d'arrivée, etc. De plus, le déplacement demandé par le passager doit être inclus dans l'itinéraire prévu par le conducteur. Cela n'exclut pas la possibilité d'accepter un léger détour pour les prises en charge ou les déposes, en fonction des préférences du conducteur et des autres passagers (nombre de détours possibles par exemple). En outre, le nombre de places disponibles dans le véhicule de covoiturage doit être suffisant pour accueillir tous les passagers demandeurs du même trajet (Sghaier, 2011; Ting, Lee, Pickl, & Seow, 2021).

La figure 1.1 présente les éléments du covoiturage et la relation entre eux.

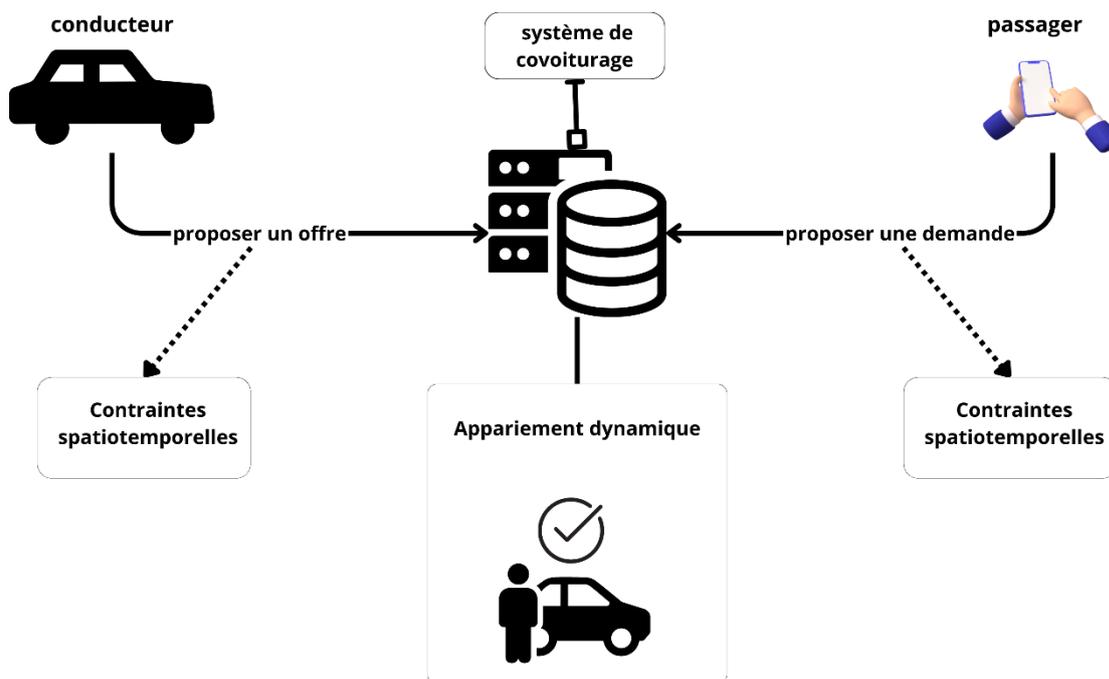


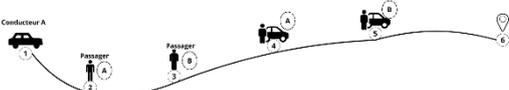
Figure 1.1 : Éléments d'un système de covoiturage.

1.2.1.3 Variantes du covoiturage dynamique

Dans l'appariement dynamique du covoiturage, il existe plusieurs contraintes à respecter, qu'elles soient proposées par le conducteur ou le passager. Une de ces contraintes est

que les conducteurs proposant un trajet peuvent préférer emmener un seul passager ou être disposés à en prendre plusieurs passagers. De même, les passagers demandant un trajet peuvent préférer voyager avec un seul conducteur ou être disposés à partager leur voyage avec plusieurs conducteurs et à passer de l'un à l'autre en cours de route vers leur destination. De cette contrainte découlent quatre variantes du covoiturage (Ting, Lee, Pickl, & Seow, 2021; Plate, 2019; Herbawi, 2013), présentées dans le tableau 1.1 et illustrer avec des exemples.

Dans ces exemples, nous notons l'offre du conducteur par $C_i(a, b)$ où C_i représente le conducteur i avec le point de départ a et le point d'arrivée b . Nous notons aussi la demande du passager par $P_j(c, d)$ où P_j représente le passager j avec le point de départ c et le point d'arrivée d .

Les variantes du système de covoiturage	Exemple
<p align="center">Un conducteur et un passager (1C- 1P)</p> <p>Fait référence à une situation où chaque conducteur du système ne peut prendre qu'un seul passager, et où chaque passager souhaite arriver à sa destination en partageant le trajet avec un seul conducteur (Plate, 2019).</p>	 <p align="center">$C_A(1,4), P_B(2,3)$</p>
<p align="center">Un conducteur et plusieurs passagers (1C- NP)</p> <p>Chaque conducteur a la capacité de répondre à plusieurs demandes de passagers (un nombre spécifique correspondant à la capacité du véhicule). Cependant, chaque passager reste associé à un seul conducteur, même si ce dernier peut satisfaire plusieurs demandes. Les deux problèmes fondamentaux de cette variante sont l'association des passagers aux conducteurs et le choix des itinéraires pour les conducteurs (Plate, 2019; Herbawi, 2013).</p>	 <p align="center">$C_A(1,6), P_A(2,4), P_B(3,5)$</p>

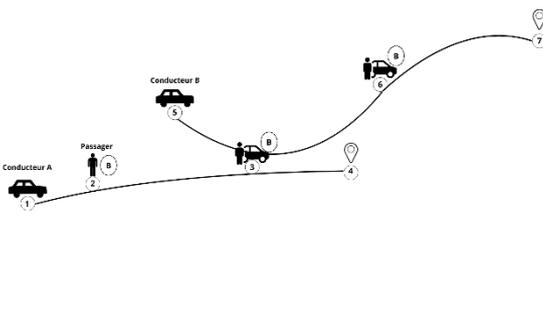
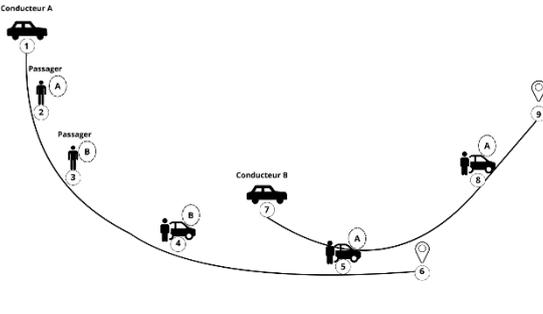
<p>Plusieurs conducteur et un passager (NC- 1P)</p> <p>Il est parfois appelé covoiturage multi-sauts en raison des différentes phases parcourues par le passager avec différents conducteurs. Cette situation décrit la possibilité pour un passager d'arriver à sa destination en partageant son trajet avec plusieurs conducteurs, Chaque conducteur contribue à rapprocher le passager de sa destination. Il est important de noter cependant que chaque conducteur ne prend en charge qu'un seul passager (Plate, 2019).</p>	 <p>$C_A(1,4), C_B(5,7), P_B(2,6)$</p>
<p>Plusieurs conducteurs et plusieurs passagers (NC-NP)</p> <p>Cette configuration implique une combinaison des deux variantes : plusieurs conducteurs pour un passager (NC-1P) et un conducteur pour plusieurs passagers (1C-NP). Dans cette configuration, chaque conducteur a la capacité de prendre en charge plusieurs passagers en partageant le même véhicule et en acceptant des transitions d'un véhicule à un autre (multi-sauts) (Herbawi, 2013; Fèvre, Zgaya-Biau, Mathieu, & Hammadi, 2023).</p>	 <p>$C_A(1,6), C_B(7,9)$</p> <p>$P_A(2,8), P_B(3,4)$</p>

Tableau 1.1 : Variantes du covoiturage.

1.2.1.4 Contraintes sur le covoiturage dynamique

Dans un système de covoiturage, il est important de respecter un ensemble de contraintes afin de mettre en correspondance une offre et une demande. Ces contraintes peuvent être regroupées en deux types : contraintes opérationnelles (contraintes de capacité, etc.) ainsi que des contraintes spatiotemporelles (contraintes de temps, contraintes de détour, etc.). Dans ce qui suit, nous allons présenter quelques contraintes. Il est à noter que cette liste n'est pas exhaustive, mais, elle donne un aperçu des principales contraintes associées au covoiturage dynamique.

- **Contrainte de capacité :** Le nombre de passagers pour un trajet doit être inférieur ou égale au nombre de places disponibles simultanément. Chaque véhicule possède une capacité maximale prédéterminée en termes de passagers (Santos & Xavier, 2015; Fèvre, Zgaya-Biau, Mathieu, & Hammadi, 2023).
- **Contrainte de coût :** permet d'économiser de l'argent pour les participants en répartissant les dépenses liées au voyage, telles que le carburant et les frais de route. Chaque personne contribue en proportion de la distance parcourue par rapport à la distance totale du trajet. Parfois, un utilisateur peut fixer un budget maximal pour le trajet partagé (Agatz, Erera, Savelsbergh, & Wang, 2010; Mourad, Puchinger, & Chengbin, 2019).
- **Contrainte de temps d'attente :** Lorsqu'une demande de transport est reçue, le conducteur doit récupérer le passager dans un délai inférieur ou égale au temps d'attente spécifié (Herbawi, 2013).
- **Contrainte liée aux fenêtres temporelles :** Pour mieux capturer les préférences temporelles des participants, qu'ils soient conducteurs ou passagers, une représentation de fenêtre temporelle peut être utilisée. Cela permet à un participant de spécifier une heure de départ et une heure d'arrivée, offrant ainsi une flexibilité accrue dans la planification des déplacements (Agatz, Erera, Savelsbergh, & Wang, 2010).
- **Contrainte de détour :** Le conducteur peut effectuer un léger détour pour récupérer ou déposer un passager, à condition que ce détour ne viole pas les contraintes du conducteur, telles que la distance préalablement définie, le temps, etc. (Cici, Markopoulou, & Laoutaris, 2015). La figure 1.2 (Martins, de la Torre, Corlu, Juan, & Masmoudi, 2021) présente deux scénarios de covoiturage, avec détour et sans détour.

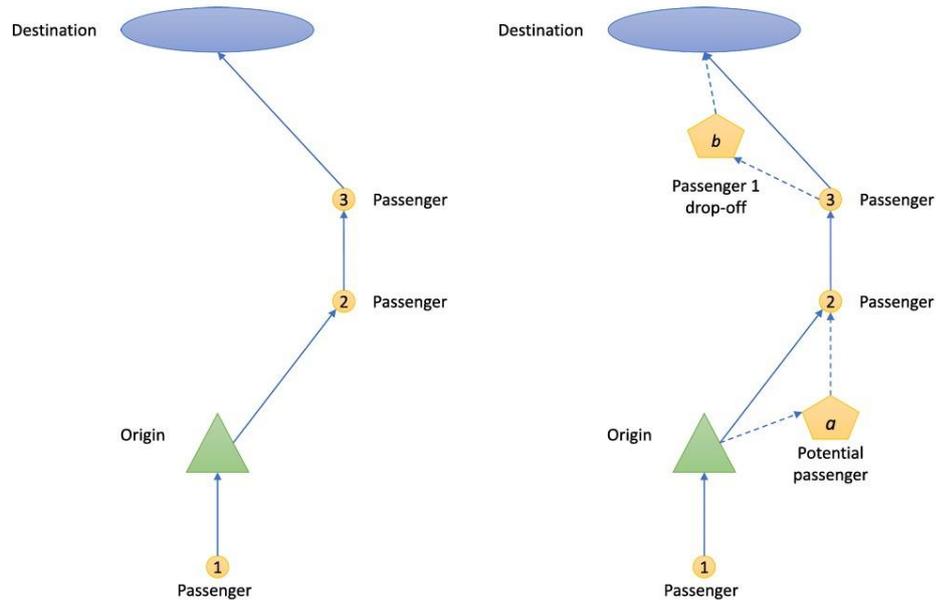


Figure 1.2 : Représentation de deux scénarios de covoiturage dynamique : avec détour (scénario à droite) et sans détour (scénario à gauche) (Martins, de la Torre, Corlu, Juan, & Masmoudi, 2021)

1.2.1.5 Objectifs à optimiser

Dans l'appariement dynamique du covoiturage, il y a plusieurs objectifs à optimiser, qui sont répartis dans certains travaux en deux catégories : objectifs opérationnels et objectifs de qualité (Mourad, Puchinger, & Chengbin, 2019). Dans notre travail, nous allons organiser les critères en quatre catégories distinctes : optimisation temporelle, optimisation spatiale, optimisation environnementale et optimisation de l'efficacité du service. La figure 1.3 représente la catégorisation que nous proposons des objectifs d'optimisation dans le covoiturage dynamique.

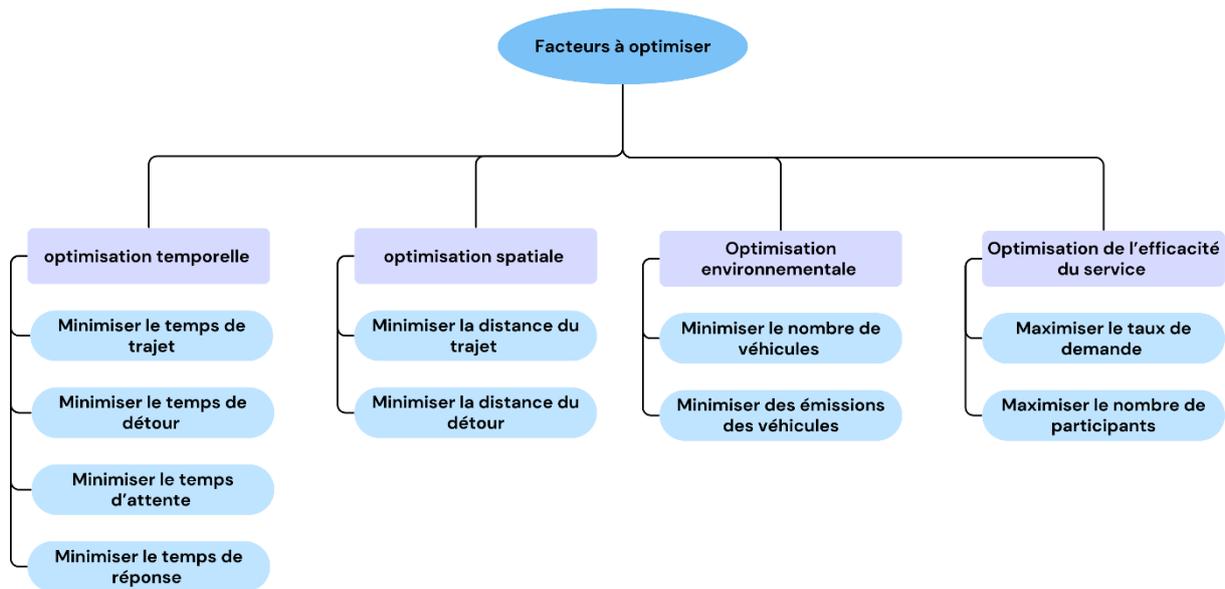


Figure 1.3 : Objectifs d'optimisation.

1.2.1.5.1. Optimisation temporelle

- **Minimiser le temps de trajet :** Réduire le temps de déplacement à l'échelle du système vise à minimiser la durée totale passée par les passagers dans le véhicule pendant leur trajet entre le point de départ et la destination. Ce critère joue un rôle essentiel en termes de commodité pour les participants. De plus, sur le plan sociétal et environnemental, c'est une mesure significative, car les émissions des véhicules dépendent non seulement de la distance parcourue, mais également de leur vitesse (Agatz, Erera, Savelsbergh, & Wang, 2012)
- **Minimiser le temps de détour :** Optimiser les itinéraires afin de réduire le temps additionnel que les passagers passent dans le véhicule lorsqu'ils participent au covoiturage, contribuant ainsi à diminuer la durée totale du trajet (Qin, Zhu, & Ye, 2022).
- **Minimiser le temps d'attente :** Réduire le temps d'attente consiste à diminuer la période pendant laquelle les passagers doivent patienter avant d'être pris en charge par un conducteur. L'optimisation de ce paramètre vise à améliorer la qualité du service en

répondant aux attentes des passagers et à contribuer à la réduction du temps total des trajets.

- **Minimiser le temps de réponse** : l'objectif est de garantir une réponse satisfaisante, sans nécessairement viser l'optimalité, en diminuant le temps entre la demande de covoiturage d'un passager et la confirmation de sa demande par un conducteur. Cela revêt une importance particulière pour les services en ligne (ou en temps réel) où les demandes peuvent survenir peu avant l'heure de départ souhaitée (Plate, 2019).

1.2.1.5.2. Optimisation spatiale

- **Minimiser la distance du trajet** : implique de sélectionner des itinéraires nécessitant le moins de kilomètres possible pour tous les participants se rendant à leur destination. Cet objectif revêt une importance sociétale en contribuant à diminuer la pollution et les embouteillages, tout en étant alignée sur la réduction des coûts globaux de déplacement. (Agatz, Erera, Savelsbergh, & Wang, 2012).
- **Minimiser la distance du détour** : Diminuer la distance totale parcourue par les conducteurs et les passagers lorsqu'ils font des détours en choisissant les meilleurs itinéraires.

1.2.1.5.3. Optimisation environnementale

- **Minimiser le nombre de véhicules** : Cette mesure indique la diminution du nombre de véhicules mobilisés dans le cadre du covoiturage dynamique, ce qui comporte des bénéfices pour l'environnement (Mourad, Puchinger, & Chengbin, 2019) tels que la réduction des émissions de CO₂ et les embouteillages
- **Minimiser les émissions des véhicules** : vise à réduire l'impact environnemental des déplacements. L'objectif est de limiter les émissions de CO₂ (Atahran, Lenté, & T'kindt, 2014).

1.2.1.5.4. Optimisation de l'efficacité du service

- **Maximiser le taux de demande** : Augmenter le taux de demande consiste à stimuler un nombre élevé de requêtes de covoiturage. Cela nécessite la mise en œuvre de stratégies

visant à attirer plus de conducteurs et de passagers, même si des détours mineurs ou une réactivité rapide ne peuvent rendre un algorithme efficace (Plate, 2019)

- **Maximiser le nombre de participants :** Cet objectif vise à augmenter le nombre de participants satisfaits, que ce soient les conducteurs ou les passagers. Cette approche peut être bénéfique pour un opérateur privé de services de covoiturage dont les revenus sont liés au nombre de covoiturages réussis. De plus, le taux de réussite des appariements peut jouer un rôle crucial en tant qu'indicateur de performance pour les utilisateurs d'un service de covoiturage spécifique, et un taux de réussite élevé pourrait stimuler une plus grande adhésion à l'avenir (Agatz, Erera, Savelsbergh, & Wang, 2012).

Le système de covoiturage peut avoir soit un objectif unique (mono-objectif) pour l'optimiser, soit plusieurs objectifs (multi-objectifs). Lorsqu'il y a plusieurs objectifs à optimiser, ceux-ci peuvent être en conflit entre eux.

1.2.2 Problématique de l'appariement dynamique

Le covoiturage dynamique représente une avancée majeure dans le domaine des transports en offrant des solutions de mobilité efficaces. Cette méthode permet aux passagers de partager un trajet avec des conducteurs. Mettre en œuvre un système de covoiturage dynamique nécessite de considérer plusieurs défis impliquant un traitement spécifique, tels que la tarification dynamique, l'appariement dynamique, le routage, la prédiction de la demande, etc. (Qin, Zhu, & Ye, 2022)

L'appariement dynamique dans le covoiturage représente un défi crucial et complexe, étant donné qu'il s'agit de faire correspondre en temps réel les demandes de transport des passagers avec les offres de trajet des conducteurs. Ce processus doit tenir compte de multiples contraintes et objectifs, tels que les fenêtres temporelles, les temps d'attente, les temps de détour, la capacité des véhicules, etc. Un des principaux défis est l'optimisation combinatoire, où les systèmes doivent résoudre des problèmes difficiles dans des délais très courts pour garantir une efficacité opérationnelle. En outre, la volatilité et l'imprévisibilité des demandes de covoiturage exigent des algorithmes adaptatifs capables de réagir rapidement aux changements dans l'environnement urbain.

Pour résoudre efficacement ces défis, il est impératif de mettre en œuvre des algorithmes d'optimisation sophistiqués, capables de gérer simultanément plusieurs objectifs qui peuvent être conflictuels et de s'adapter aux variations continues des données d'entrée. La réussite de l'appariement dynamique repose donc sur des techniques avancées d'optimisation, pour atteindre un équilibre optimal entre efficacité, équité et satisfaction des utilisateurs.

Dans ce travail, nous nous intéressons au problème de l'appariement dynamique sous contraintes, en optimisant plusieurs objectifs conflictuels. Ce défi implique la coordination en temps réel entre les passagers, via leurs demandes, et les conducteurs, via leurs offres, sur une plateforme de covoiturage. Ce problème a été abordé dans de nombreuses études avec différentes approches qui varient des méthodes exactes, (Masoud & Jayakrishnan, 2017; Alisoltani, N.; Leclercq, L.; Zargayouna, M., 2019), aux méthodes heuristiques (Cheikh-Graïet, Dotoli, & Hammadi, 2020; Masmoudi, Hosny, Demir, & Pesch, 2020) à l'apprentissage par renforcement (Qin, Zhu, & Ye, 2022).

Le problème que nous considérons implique plusieurs contraintes (temporelles, spatiales et de capacité) et plusieurs objectifs à optimiser qui rentrent en conflit (minimisation du temps d'attente des passagers, minimisation du temps de détour des conducteurs et maximisation de l'utilisation des voitures). Ce problème va être abordé avec plus de détails dans le chapitre suivant. Nous explorons l'application de l'apprentissage par renforcement multi-objectifs (MORL : Multi-Objective Reinforcement Learning) (Hayes, et al., 2022) comme solution pour notre problématique.

Nous présentons dans la section suivante les concepts de base de MORL.

1.3 Apprentissage par renforcement multi-objectifs

L'apprentissage par renforcement (RL : Reinforcement Learning) est défini comme la discipline visant à élaborer des stratégies de prise de décision optimales. L'essence de cette méthode réside dans la capture des aspects cruciaux des défis auxquels un agent apprenant est confronté lorsqu'il interagit de manière coopérative avec son environnement pour atteindre un objectif spécifique (Joshi, et al., 2021; Sutton & Barto, 1999; Wiering & Van Otterlo, 2012). L'apprentissage par renforcement (RL) permet à un agent cognitif d'apprendre un comportement souhaitable en utilisant une fonction de récompense. Le problème de contrôle

est formulé pour trouver la meilleure politique qui maximise cette fonction de récompense cumulée, tout en prenant en compte les récompenses et les pénalités associées à chaque action. L'objectif principal de l'apprentissage par renforcement est de développer la capacité à prendre des décisions judicieuses dans un environnement incertain à travers l'expérience « essais et erreurs » (Vamvoudakis, Wan, Lewis, & Cansever, 2021).

Dans un système d'apprentissage par renforcement classique, il n'existe qu'un seul objectif à atteindre (SORL : Single-Objective Reinforcement Learning). Cependant, dans de nombreux cas réels, plusieurs objectifs distincts doivent être pris en compte simultanément. On parle alors d'apprentissage par renforcement multi-objectifs (MORL : Multi-Objective Reinforcement Learning). La difficulté réside dans le fait que ces différents objectifs entrent souvent en conflit les uns avec les autres, complexifiant ainsi le processus d'optimisation. Par conséquent, il n'existe pas une solution optimale unique, mais plutôt un ensemble de solutions potentielles formant un compromis entre les différents objectifs (Asl, Derhami, & Yazdian-Dehkordi, 2017). Les algorithmes d'apprentissage par renforcement multi-objectifs cherchent à optimiser conjointement ces multiples critères, en trouvant le meilleur équilibre possible entre eux.

Dans l'apprentissage par renforcement, l'agent cherche à prendre des actions dans un environnement pour maximiser la somme de ses récompenses. Un cadre couramment utilisé pour modéliser ces problèmes est le processus de décision de Markov (MDP : Markov Decision Process). Ce processus est composé d'un ensemble des états, un ensemble des actions, une fonction de transition et une fonction de récompense.

Si on considère plusieurs objectifs, on recourt au modèle MOMDP (Multi-Objective Markov Decision Process). Ce dernier est caractérisé par un ensemble d'états, un ensemble d'actions, une fonction de transition qui décrit l'évolution de l'état de l'environnement en fonction des actions de l'agent, et une fonction de récompense à valeur vectorielle qui évalue l'efficacité des actions entreprises par l'agent (Abels, Roijers, Lenaerts, Nowé, & Steckelmacher, 2019).

La figure 1.4 représente une comparaison entre un processus de décision de Markov (MDP) et un processus de décision de Markov multi-objectifs (MOMDP).

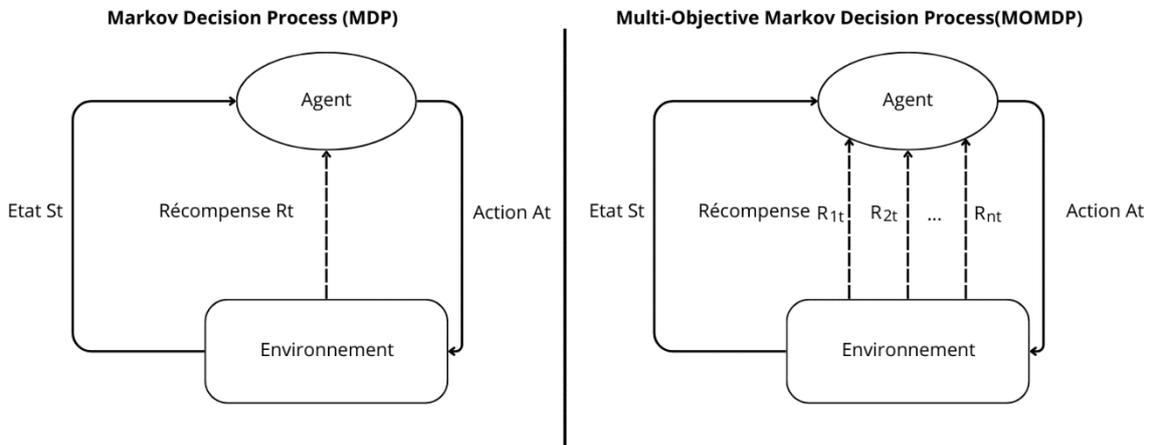


Figure 1.4 : Comparaison entre MDP et MOMDP.

Résoudre un problème multi-objectifs par MORL revient à trouver une politique qui permet d’optimiser simultanément plusieurs fonctions de récompense, chacune correspondant à un objectif spécifique. Cela implique de gérer et de concilier des objectifs potentiellement conflictuels pour parvenir à un compromis optimal. Le but est d’élaborer une stratégie d’apprentissage capable de naviguer efficacement dans un espace d’actions complexe, tout en tenant compte des diverses contraintes et priorités liées aux objectifs multiples du problème. Ceci se fait soit en maintenant un ensemble de politiques Pareto-optimales, soit en réduisant le vecteur multi-objectif en un seul scalaire, puis applique une méthode d’apprentissage par renforcement standard (Wu, Zhang, Yang, & Wang, 2021).

Les techniques de résolution de MORL peuvent être classées en deux catégories : les méthodes hors ligne et les méthodes en ligne. Ces deux catégories déterminent la manière dont les agents apprennent et interagissent avec leur environnement. Le MORL hors ligne (Offline MORL) repose sur un ensemble de données fixe, tandis que le MORL en ligne (Online MORL) interagit avec l’environnement en temps réel. C’est cette dernière catégorie que nous utiliserons dans notre travail : le MORL en ligne (Online MORL) (Abels, Roijers, Lenaerts, Nowé, & Steckelmacher, 2019).

1.3.1 Composants de l'apprentissage par renforcement multi-objectifs (MORL)

Le MORL utilise le processus de décision de Markov multi-objectif MOMDP pour modéliser les problèmes, comme nous l'avons expliqué précédemment. Le MOMDP est défini par les éléments :

- **Agent** : prend toutes ses décisions en se basant sur les récompenses et les punitions associées à chaque action ou série d'actions. (Joshi, et al., 2021).
- **Etat** : L'état du système à l'instant t , qui renferme toutes les informations nécessaires et suffisantes pour calculer les coûts/récompenses, les contraintes et toute autre information requise par la fonction de transition (Vamvoudakis, Wan, Lewis, & Cansever, 2021).
- **Action** : C'est un ensemble restreint d'actions, ce qui sous-entend que toutes les actions que l'agent peut entreprendre sont incluses dans cette liste limitée (Jia & Wang, 2020; Van Moffaert & Nowé, 2014).
- **Transition** : La transition d'état correspond à la probabilité de passer de l'état actuel à l'état suivant (Asl, Derhami, & Yazdian-Dehkordi, 2017).
- **Récompense** : La fonction de récompense à valeur vectorielle fournit un retour numérique distinct pour chacun des objectifs considérés, représentant ainsi la réponse de l'environnement à l'agent. En conséquence, la longueur du vecteur de récompense correspond au nombre d'objectifs à atteindre (Abels, Roijers, Lenaerts, Nowé, & Steckelmacher, 2019; Hayes, et al., 2022).
- **Fonction de valeur** : Il s'agit du gain ou de la récompense que l'agent obtiendra ultérieurement en prenant une action spécifique dans un état donné (Swain, Pattnaik, & Athawale, 2021).

- **Politique** : La politique détermine les actions à entreprendre dans un environnement donné et les exécute pour chaque état de cet environnement (Asl, Derhami, & Yazdian-Dehkordi, 2017).

La seule différence entre les composants du MDP et du MOMDP est le type de la récompense.

1.4 Travaux connexes

Le domaine du covoiturage a suscité une attention considérable ces dernières années. De nombreux travaux ont abordé le covoiturage dynamique en traitant diverses problématiques, variantes et objectifs, en utilisant l'apprentissage par renforcement.

Wang et al. (2018) proposent une approche d'apprentissage par renforcement profond (Deep-RL) pour optimiser la politique de répartition des conducteurs sur les plateformes de covoiturage. Ils modélisent le problème de répartition comme un processus décisionnel de Markov (MDP) et utilisent des réseaux Q-profonds (Deep Q-networks : DQN) avec recherche d'actions pour estimer la fonction de valeur état-action du conducteur, dans le but de maximiser les revenus totaux. Les auteurs ont aussi introduit une nouvelle méthode appelée *Correlated Feature Progressive Transfer (CFPT)* pour permettre le transfert de connaissances à la fois dans les espaces spatiaux et temporels, augmentant ainsi l'adaptabilité et l'efficacité de l'apprentissage. Les agents de répartition sont entraînés et évalués sur des données réelles spatio-temporelles de la plateforme DiDi.

Xu et al (2018) proposent un algorithme de répartition des demandes visant à optimiser l'efficacité à long terme tout en assurant la satisfaction immédiate des clients. Cet algorithme utilise une approche centralisée pour associer les conducteurs aux demandes, en modélisant la répartition comme un problème de prise de décision séquentielle. L'apprentissage se fait en construisant un Processus décisionnel de Markov (MDP), où un agent représente un conducteur individuel.

Ke, Yang, Ye et al. (2020) Proposent une solution au problème de répartition des passagers basée sur l'apprentissage par renforcement (RL) et un processus décisionnel de Markov multi-agents (MMDP). Chaque demande du passager est considérée comme un agent qui décide, à chaque intervalle de temps, s'il doit être retardé jusqu'à l'intervalle suivant. Quatre méthodes d'apprentissage par renforcement ont été utilisées et adaptées : Delayed-DQN, Delayed-PPO (Proximal Policy Optimization), Delayed-A2C (Advantage Actor Critic) et Delayed-ACER (Actor Critic with Experience Replay).

Kullman et al. (2022) Présentent une étude de recherche sur l'optimisation de la gestion d'une flotte de véhicules électriques dans un service de covoiturage en utilisant des techniques d'apprentissage par renforcement profond (Deep-RL). Les chercheurs appliquent le Deep-RL pour élaborer des politiques décisionnelles basées sur des approximations de la valeur Q, obtenues grâce à des réseaux neuronaux profonds. L'objectif principal est de maximiser le profit de l'opérateur en assignant efficacement les véhicules aux demandes entrantes, tout en gérant la recharge et le repositionnement des véhicules en prévision de la demande future.

D'autres travaux abordent plusieurs objectifs simultanément, Parmi lesquels : Jindal et al., (2018) ont développé un système basé sur l'apprentissage par renforcement pour apprendre une politique de covoiturage efficace visant à maximiser l'efficacité du transport. Ils ont élaboré un modèle de réseau neuronal profond appelé ST-NN (SpatioTemporal Neural Network) pour prédire le temps de trajet en taxi à partir des données brutes de trajet GPS. Ensuite, ils ont créé un environnement de simulation de covoiturage pour l'entraînement en RL, utilisant les sorties de ST-NN et l'ensemble de données des trajets en taxi de New York.

Zhou et al., (2019) proposent une méthode décentralisée de répartition des demandes basée sur l'apprentissage par renforcement multi-agents. Contrairement aux solutions centralisées qui nécessitent de considérer toutes les correspondances possibles entre les demandes et les véhicules disponibles, cette approche permet à chaque agent de fonctionner de manière indépendante, guidé par une évaluation commune sans nécessiter de communication ou de coopération explicite entre les agents.

Al-Abbasi et al., (2019) introduisent DeepPool, un modèle distribué et sans modèle qui utilise des techniques de réseau neuronal profond (DQN) pour apprendre des politiques de répartition optimales en interagissant avec l'environnement. Il intègre efficacement les données

sur la demande de déplacement et les modèles d'apprentissage profond pour améliorer la gestion de la répartition des véhicules dans le cadre des services de covoiturage. L'évaluation de DeepPool repose sur l'utilisation d'un ensemble de données réelles sur les trajets en taxi à New York.

Jin et al., (2019) proposent CoRide, une approche innovante combinant apprentissage par renforcement multi-agents et hiérarchique pour optimiser conjointement la répartition des demandes et la gestion de flotte dans les plateformes de covoiturage multi-échelles.

Wang et al., (2023) étudient un modèle de covoiturage avec des stations de transfert permettant le transfert de passagers entre véhicules. Ils proposent une approche combinant l'apprentissage profond par renforcement (Deep-RL) et la programmation linéaire en nombres entiers (ILP : Integer Linear Programming) pour optimiser la répartition des véhicules.

Singh et al., (2021) proposent un algorithme décentralisé sans modèle pour le covoiturage multi-saut (MHRS). Dans ce système, les passagers peuvent être déposés à plusieurs reprises avant d'atteindre leur destination finale. L'algorithme utilise l'apprentissage par renforcement profond, spécifiquement un réseau de neurones profond à double Q (Double DQN), pour apprendre et déterminer les politiques optimales de répartition et d'appariement pour chaque véhicule individuellement.

Le Tableau 1.2 classe les travaux mentionnés précédemment.

Article	Variante	Objectifs d'optimisation	Type du RL	Points forts	Points faibles
Wang et al., (2018)	1C- 1P	Maximiser le total des gains obtenus (financier)	Deep-RL : - DQN - CFPT	Utilisation de DQN pour optimiser les politiques de répartition des trajets. Utilisation des données réelles de la plateforme DiDi renforçant l'applicabilité des résultats. Introduction des méthodes de transfert d'apprentissage pour améliorer l'adaptabilité et l'efficacité.	Nécessité d'étapes de prétraitement impactant la complexité et la scalabilité.
Xu et al., (2018)	1C - 1P	Maximiser le Brut Volume de marchandises (GMV) Optimiser l'affectation des demandes à grande échelle	Apprentissage hors ligne + Planification en ligne	Prise de décision séquentielle. Utilisation des données historiques. Appariement centralisé et coordonné. Déploiement avec succès dans DiDi.	Dépendance aux données historiques. Complexité et scalabilité.
Ke et al., (2020)	1C - NP	Minimiser la distance de prise en charge. Minimiser le temps de prise en charge.	Deep-MARL : -M-A2C -M-PPO -M-ACER	Equilibre entre le temps de prise en charge, le temps d'appariement et le taux d'appariement. Incorporation de Deep-MARL permet de déterminer de manière dynamique le temps de retard pour chaque demande de passager, améliorant ainsi l'efficacité du système.	Concentration sur l'amélioration de l'efficacité de l'appariement, mais ne prend pas en compte d'autres aspects importants tels que l'équité entre les passagers et les conducteurs.
Kullman et al., (2022)	NC - NP	Maximiser les récompenses totales (profit).	Deep- RL : DQN	Robustesse et capacité d'évolution face à des situations variées.	Nécessiter des ressources computationnelles importantes, limitant ainsi l'applicabilité dans des environnements en temps réel ou à grande échelle.
Jindal et al., (2018)	1C - NP	Maximiser l'efficacité des transports. Minimiser la congestion.	Deep-RL : ST-NN	Intégration des données réelles des trajets de taxis de NYC permet de concevoir des modèles et des simulations fidèles aux conditions réelles. Prédiction précise des temps de trajet.	L'article se concentre sur le covoiturage de taxis à New York, ce qui pourrait limiter la généralisabilité des résultats à d'autres lieux ou modes de transport.
Zhou et al., (2019)	1C - 1P	Maximiser le revenu des conducteurs. Maximiser le taux de réponse.	MARL	Apprentissage centralisée et exécution décentralisée sans communication explicite entre agents. Accélération du processus d'apprentissage grâce à l'optimisation KL.	Absence de communication ou de coopération explicite entre les agents peut limiter la capacité à tirer parti de l'intelligence collective et de la coordination dans certains scénarios complexes.

Al-Abbasi et al., (2019)	1C - NP	Minimiser l'inadéquation entre l'offre et la demande. Minimiser le temps d'attente des passagers. Minimiser le temps de détour. Maximiser le nombre de participant.	Deep-RL : DeepPool	Adaptation rapide à un environnement dynamique. Bonnes performances en termes de taux d'acceptation, de temps d'attente et d'utilisation des véhicules.	Les performances de DeepPool dépendent fortement de la qualité et de la disponibilité des données réels, ce qui pourrait limiter son efficacité dans les zones où les sources de données sont limitées ou peu fiables.
Jin et al., (2019)	1C - 1P	Maximiser le revenu du conducteur et le taux de réponse	MARL hiérarchique	Saisir les impacts des agents voisins et identifier les agents ou régions clés grâce à la hiérarchie.	La modélisation peut ne pas capturer tous les aspects dynamiques du problème en temps réel.
Singh, et al. (2021)	NC - NP	Minimiser le temps d'attente des passagers et le temps d'inactivité des véhicules. Minimiser le nombre de transferts par passager. Réduire la consommation de carburant.	Deep-RL : DDQN	Utilisation du Double DQN permet une gestion efficace des ressources et une amélioration des performances globales du système.	Prise de décision décentralisée peut rendre plus difficile l'obtention d'une solution globalement optimale.
Wang et al. (2023)	1C - NP	Minimiser les temps d'attente et de retard des passagers et les temps morts des véhicules. Maximiser les incitations à servir les demandes.	Deep-RL + ILP (Integer Linear Programming)	Stabilité et efficacité du modèle. Résultats numériques qui surpassent plusieurs autres méthodes.	Complexité élevée due au Deep-RL hiérarchique. Incertitude quant à la généralisation du modèle à d'autres villes

Tableau 1.2 : Classification des travaux connexes en se basant sur les problèmes considérés et le type du RL.

Dans le tableau 1.3, nous allons classer ces mêmes travaux en se basant sur les modèles RL proposé.

Article	Agent	Etat	Action	Récompense
Wang et al., (2018)	Conducteur	Localisation actuelle. Le Temps.	Accepter ou refuser la demande.	Le montant total.
Xu et al., (2018)	Conducteur	Localisation actuelle. Le Temps.	Servir d'une demande Ne rien faire.	Le prix d'une demande
Ke et al., (2020)	Demande	Localisation actuelle. Nombre des sièges disponible. Le temps de prise en charge du passager. La destination de chaque passager.	Accepter ou refuser la demande.	Le temps d'appariement.
Kullman et al., (2022)	Un opérateur central	Le temps actuel. Les coordonnées d'origine et de destination de la demande. Les coordonnées de la position actuelle du véhicule, son état actuel et sa destination.	Servir une nouvelle demande. Repositionner vers une station de recharge. Ne rien faire	Le revenu d'une demande
Jindal et al., (2018)	Conducteur	La Localisation actuelle. Le Temps.	Prendre un/plusieurs passagers.	La distance du voyage.
Zhou et al., (2019)	Conducteur	Le nombre de véhicules inactifs. Le nombre de demandes valides. La répartition des destinations des demandes.	Accepter ou refuser la demande.	Le prix d'une demande
Al-Abbasi et al., (2019)	Conducteur	Localisation actuelle. Nombre de siège disponible. Le temps de prise en charge du passager. La destination de chaque passager.	Accepter ou refuser la demande.	Le temps de détour. Le temps de trajet. Le nombre des passagers servis.
Jin et al., (2019)	Cellule de région	Le nombre de véhicules. Le nombre de demandes. L'entropie et la distribution des caractéristiques des demandes (le prix, la durée, etc.).	Demandes réelles (OD) Fausses demandes (FM)	Le revenu actualisé du conducteur. Le revenu calculé d'une demande.

Singh, et al. (2021)	Conducteur	L'emplacement et l'état actuels des véhicules. Le nombre prédit de véhicules par zone. La demande prédite par zone.	Servir un passager et de choisir la zone vers laquelle se diriger.	L'écart entre l'offre et la demande. Le temps d'appariement/attente pour les passagers et les véhicules Le nombre de sauts pour les passagers. Le temps de trajet supplémentaire des sauts. Le nombre de véhicules utilisés pour répondre à la demande. La consommation de carburant pour chaque véhicule.
Wang et al. (2023)	Conducteur	La position des véhicules. L'état des passagers (assignés à un véhicule ou en attente).	<i>Passer</i> : ne rien faire et maintenir l'état actuel du véhicule. <i>Assigner</i> : attribuer une nouvelle requête au véhicule et mettre à jour son trajet. <i>Repositionner</i> : déplacer un véhicule inactif vers une zone à forte demande.	Temps d'attente. Distance parcourue. Nombre des trajets partagés. Pénalités de transfert.

Tableau 1.3 : Classification des travaux connexes selon le modèle RL proposé.

1.5 Conclusion

Dans ce chapitre, nous avons abordé le covoiturage dynamique en explorant ses concepts, composants, contraintes et objectifs. De plus, nous avons inspecté plusieurs travaux dans ce domaine qui ont utilisé l'apprentissage par renforcement, en proposant également une classification de ces travaux.

Dans le prochain chapitre, nous présenterons notre contribution visant à résoudre le problème d'appariement dans un système de covoiturage dynamique.

Chapitre 2 : Contribution

2.1. Introduction

Dans ce chapitre, nous présentons notre principale contribution, qui consiste en la proposition d'une modélisation du problème d'appariement dynamique en utilisant l'apprentissage par renforcement. Notre approche vise à résoudre le problème d'appariement dynamique en optimisant plusieurs objectifs conflictuels, à savoir : minimiser le temps d'attente des passagers afin de leur offrir une expérience optimale, minimiser le temps de détour des conducteurs et maximiser l'utilisation des véhicules afin de motiver les conducteurs à prendre des passagers qui, leurs trajets, impliquent un détour. Ces objectifs visent à garantir la satisfaction des utilisateurs et à établir un équilibre entre les besoins des passagers et ceux des conducteurs, améliorant ainsi l'efficacité de notre système. Nous prenons en compte plusieurs contraintes spatiales, temporelles et de capacité, de se rapprocher des applications réelles et ajoute de la complexité à notre modèle.

2.2. Problématique et objectifs de notre approche

2.1.1. Problématique

Le développement urbain a entraîné une augmentation significative du trafic routier, exacerbant les impacts négatifs sur l'environnement en raison des émissions de gaz des véhicules. Dans ce contexte, le covoiturage dynamique se présente comme une solution prometteuse pour réduire l'utilisation individuelle des voitures et atténuer les problèmes environnementaux. La popularité du covoiturage a considérablement augmenté ces dernières années, notamment grâce à l'émergence de plateformes technologiques telles qu'Uber¹, DiDi² et Lyft³.

¹ Uber: <https://www.uber.com/>

² DiDi: <https://www.didiglobal.com/>

³ Lyft: <https://www.lyft.com/>

Le covoiturage dynamique consiste à regrouper plusieurs personnes dans un même véhicule pour effectuer des trajets communs, qu'il s'agisse de conducteurs avec leur propre voiture ou de passagers. L'un des principaux défis de ce système est l'appariement dynamique, c'est-à-dire la mise en relation en temps réel des conducteurs et des passagers de manière optimale. Cet appariement vise à trouver les meilleurs passagers pour chaque conducteur et vice versa, tout en optimisant plusieurs objectifs simultanément.

Les défis de l'appariement dynamique incluent la nécessité de respecter diverses contraintes spatio-temporelles et de capacité, telles que le nombre de passagers que le conducteur peut prendre et les délais de prise en charge. Dans cette étude, notre objectif est de résoudre le problème d'appariement dynamique en optimisant simultanément plusieurs objectifs conflictuels sous certaines contraintes. Cela implique la minimisation de certains objectifs, comme le temps d'attente des passagers et le temps de détour des conducteurs, et la maximisation des autres, comme l'utilisation des véhicules.

L'équilibrage de ces objectifs conflictuels est crucial pour développer une politique efficace qui respecte toutes les contraintes imposées. Pour ce faire, nous avons choisi d'utiliser l'apprentissage par renforcement multi-objectifs (MORL), une méthode qui permet de gérer et d'équilibrer plusieurs objectifs conflictuels en même temps.

2.1.2. Objectifs

Notre travail vise principalement à résoudre le défi des appariements dans le covoiturage dynamique, en nous concentrant spécifiquement sur les offres de covoiturage proposées par les conducteurs et les demandes émises par les passagers. Nous proposons la mise en œuvre d'un système visant à optimiser cet appariement entre un conducteur et plusieurs passagers (1C - NP). Nos objectifs principaux sont :

- **Minimiser le temps d'attente des passagers.**
- **Minimiser le temps de détour des conducteurs.**
- **Maximiser l'utilisation des véhicules**, ce qui va inciter les conducteurs à accepter le maximum de demandes et ainsi maximiser le nombre de demandes acceptées dans un détour.

Pour atteindre ces buts, nous devons optimiser ces objectifs conflictuels tout en respectant des contraintes spécifiques telles que **les horaires de départ et d'arrivée des passagers et du conducteur, les points de départ et d'arrivée des deux acteurs, le temps d'attente des passagers pour leur satisfaction, le temps de détour des conducteurs ainsi que la disponibilité des sièges.**

Pour ce faire, nous avons choisi d'utiliser l'Apprentissage par Renforcement Multi-objectifs (MORL), une méthode qui permet de gérer et d'équilibrer plusieurs objectifs conflictuels en même temps. Cette approche offre une solution robuste pour répondre aux défis complexes de l'appariement dynamique dans le covoiturage, en visant à améliorer l'efficacité du système tout en maximisant la satisfaction des utilisateurs. En optimisant ces divers objectifs et en respectant les contraintes spatio-temporelles et de capacité, notre système se rapprochera des applications réelles, offrant une solution efficace et durable au problème de l'appariement dynamique.

2.2. Description du problème

Dans notre problème de covoiturage, nous nous concentrons sur le problème de l'appariement en temps réel des paires (offre, demande). Pour appairer un ensemble d'offres avec un ensemble de demandes, il est nécessaire de prendre en compte un ensemble de contraintes. Les informations relatives à ces éléments seront abordées en détails dans les sections suivantes.

2.2.1. Offres

Pour un instant donné t , nous définissons l'ensemble des offres proposées par les conducteurs comme $C(t) = \{C_1(t), C_2(t), \dots, C_n(t)\}$, où chaque conducteur i propose une offre $C_i(t)$ caractérisée par les éléments suivants : $C_i(t) = (d_{c_i}, a_{c_i}, T_{d_{c_i}}, T_{a_{c_i}}, Sg_{c_i}, dt_max_{c_i}, D_{c_i})$

- d_{c_i} : Le point de départ du conducteur i .
- a_{c_i} : Le point d'arrivée du conducteur i .
- $T_{d_{c_i}}$: Le temps de départ au plus tard du conducteur i .
- $T_{a_{c_i}}$: Le temps d'arrivée au plus tard du conducteur i .

- Sg_{c_i} : Le nombre de sièges disponibles. En d'autres termes, c'est le nombre maximum des passagers que le conducteur i peut accueillir dans son véhicule.
- $dt_{max_{c_i}}$: Le temps maximal de détour du conducteur i .
- D_{c_i} : Le seuil de détour dans lequel le conducteur i peut prendre un passager supplémentaire. En d'autres termes, c'est la distance maximale de détour autorisée par rapport à l'itinéraire initial du conducteur pour prendre un passager.

2.2.2. Demandes

Pour un instant t , nous définissons l'ensemble des demandes soumis par les passagers par : $P(t) = \{P_1(t), P_2(t), \dots, P_n(t)\}$, où chaque passager j dépose une demande $P_j(t)$ caractérisée par les éléments suivants : $P_j(t) = (d_{p_j}, a_{p_j}, T_{d_{p_j}}, T_{a_{p_j}}, wt_{max_{p_j}})$, où :

- d_{p_j} : Le point de départ du passager j .
- a_{p_j} : Le point d'arrivée du passager j .
- $T_{d_{p_j}}$: Le temps de départ au plus tard du passager j .
- $T_{a_{p_j}}$: Le temps d'arrivée au plus tard du passager j .
- $wt_{max_{p_j}}$: Le temps d'attente maximal du passager j .

2.2.3. Contraintes

Les contraintes que nous devons respecter pour résoudre notre problème sont les suivantes :

2.2.3.1. Contrainte spatio-temporelle

Nous visons à mettre en place un système de covoiturage dynamique qui tient compte à la fois des trajets et des horaires des conducteurs et des passagers. Il est impératif que les appariements trouvés respectent les points de départ et d'arrivée des conducteurs et des passagers. De plus, le respect des horaires de départ et d'arrivée est essentiel, que ce soit pour les conducteurs ou pour les passagers.

2.2.3.2. Contrainte de capacité

Chaque conducteur dispose d'un nombre limité de sièges disponibles. Par conséquent, le nombre de passagers appariés à un conducteur doit respecter la capacité maximale du véhicule. Ainsi, le conducteur ne peut pas accepter d'autres passagers s'il n'y a aucun siège disponible.

2.2.3.3. Contrainte du temps d'attente

Chaque passager se voit attribuer un temps d'attente spécifique, qui correspond à la période pendant laquelle il est prêt à attendre le véhicule depuis le début du temps de départ. Ce temps d'attente commence à s'écouler dès que le temps de départ du véhicule est prévu. Si le conducteur n'arrive pas ou s'il n'y a pas d'appariement durant ce laps de temps, le passager ne sera plus disponible pour être pris en charge, car on considère que son temps d'attente a expiré. Cette contrainte est essentielle pour garantir que les passagers ne subissent pas de retards inacceptables.

2.2.3.4. Contrainte du temps de détour

Le détour est un élément essentiel car il influence la durée du trajet et modifie la planification initiale des conducteurs et des passagers en altérant l'itinéraire initial pour prendre ou déposer un passager. Chaque conducteur a un temps maximal autorisé de détour, donc il ne peut pas effectuer un détour qui prendrait plus de temps que celui autorisé.

2.2.3.5. Contrainte de distance de détour

Il s'agit de la distance autorisée au conducteur afin de prendre des passagers qui ne sont pas sur son itinéraire principal, pour lesquels un détour est nécessaire (distance de *pick-up*). Ainsi, dans le cas où le conducteur se trouve sur son itinéraire initial et doit effectuer un détour pour prendre un passager, la distance entre le conducteur et le passager doit être inférieure ou égale à la valeur proposée dans l'offre du conducteur.

Exemple démonstratif

Nous présentons à travers cet exemple un scénario de covoiturage dynamique avec la description que nous avons présentée précédemment. La figure 2.1 présente un scénario où nous avons un conducteur C_1 dont le départ se situe au point 1 et la destination au point 6. L'heure de départ est 9h00 et l'heure d'arrivée au plus tard est 9h45. Le nombre maximal de sièges est de 4, le temps maximal de détour est de 13 minutes et la distance maximale pour prendre un passager dans le détour est de 400 mètres. De plus, nous avons 4 demandes de passagers, où pour chacune des demandes, nous avons le point de départ, le point d'arrivée, l'heure de départ, l'heure d'arrivée au plus tard et le temps d'attente. Le passager P_1 part du point 1, le passager P_2 du point 2 et les passagers P_3 et P_4 du point 7.

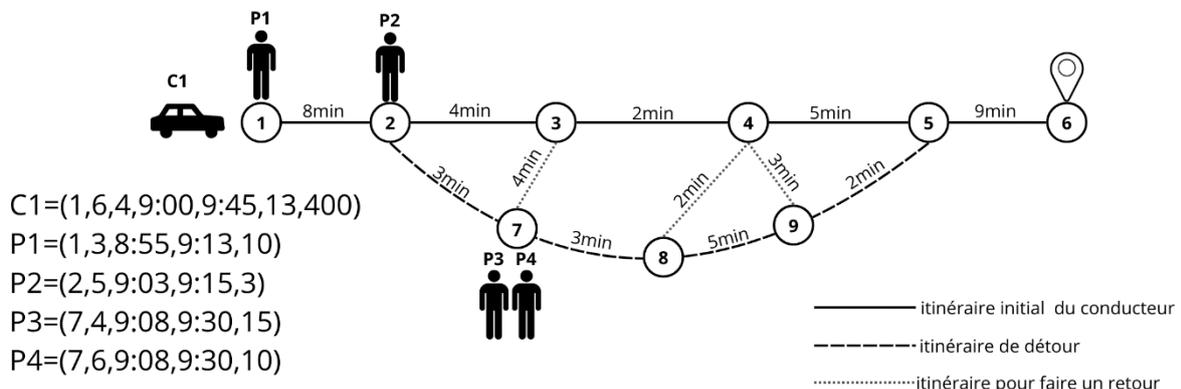


Figure 2.1 : Exemple démonstratif d'un problème d'appariement.

2.3. Modélisation proposée

Le défi majeur de ce travail consiste à appairer les demandes et les offres de trajets qui sont continuellement reçues en temps réel, tout en optimisant simultanément divers objectifs. De nombreux facteurs et variables entrent en jeu, tels que les horaires et les lieux. La modélisation de ce système nécessite une approche capable de gérer la complexité de ces facteurs.

Dans le contexte de notre travail, nous avons choisi d'utiliser l'apprentissage par renforcement multi-objectifs (MORL). Ce type d'apprentissage a comme fondement théorique

le MOMDP (Multi-Objective Markov Decision Process). Contrairement aux processus Décisionnels de Markov (MDP) classiques, les MOMDPs nous permettent de gérer plusieurs signaux de récompenses, où chaque signal interprète un objectif. Cela va nous permettre d'optimiser simultanément des objectifs conflictuels. Les objectifs que nous considérons, dans ce travail, sont : réduire le temps d'attente des passagers, minimiser le temps de détour des conducteurs et maximiser l'utilisation des véhicules.

Dans notre problématique, nous proposons un modèle où un conducteur peut prendre en charge plusieurs passagers (1C - NP). En outre, nous tenons en compte de diverses contraintes, telles que les contraintes spatio-temporelles, les contraintes de capacité et les contraintes de détour. Dans ce qui suit, nous allons décrire en détail la modélisation du MOMDP que nous proposons pour notre système de covoiturage dynamique multi-objectifs, où le conducteur est considéré comme l'agent.

2.3.1. Etat

L'état s capture l'ensemble des informations spatiales et temporelles pertinentes à un moment donné. $s_t = (G_c, G_p, G_{nd}, G_{ap}, Sg_t, Ar_p, G_{dr}, T_{ar}, t)$, où :

- G_c : Les coordonnées géographiques (latitude, longitude) du point de départ et d'arrivée du conducteur.
- G_p : Les coordonnées géographiques (latitude, longitude) des points de départ et d'arrivée de la nouvelle demande.
- G_{nd} : L'ensemble des coordonnées géographiques (latitude, longitude) des deux points de destination suivants dans le trajet principal et dans le trajet de détour.
- G_{ap} : L'ensemble des coordonnées géographiques (latitude, longitude) des points d'arrivée de l'ensemble P des passagers à bord.
- Sg_t : Le nombre de sièges disponibles à l'instant t .
- Ar_p : L'ensemble des temps d'arrivées au plus tard de l'ensemble P des passagers à bord.
- G_{dr} : Les coordonnées géographiques (latitude, longitude) des points à partir desquels le conducteur va faire un détour et des points à partir desquels il va retourner à son itinéraire initial. Ces points sont représentés sous forme de deux couples. Le premier

couple concerne le point dans l'itinéraire initial du conducteur à partir duquel il va faire un détour et le point qui lui est le plus proche dans l'itinéraire de détour. Le deuxième couple est représenté par le point dans l'itinéraire de détour à partir duquel le conducteur va retourner à son chemin initial et le point qui lui est le plus proche dans l'itinéraire initial.

- T_{ar} : Le temps d'arrivées au plus tard de la nouvelle demande.
- t : Le temps actuel.

2.3.2. Observation

L'observation, notée O , correspond aux informations extraites de l'environnement représentant l'état actuel à l'instant t . Nous définissons l'observation O_t à l'instant t comme un tuple $O_t = (Sg_t, T_{nd}, wt_t, wt_{max_t}, dt_{max_t}, dt_t, D_t, D_{p_t}, T_{ar_t}, T_{ar_{est_t}})$, où :

- Sg_t : Le nombre de sièges disponibles au moment t .
- T_{nd} : Le temps d'arrivée à la prochaine destination.
- wt_t : Le temps d'attente actuel du passager.
- wt_{max_t} : Le temps d'attente maximum du passager.
- dt_{max_t} : Le temps maximal de détour du conducteur.
- dt_t : Le temps nécessaire pour faire un détour au moment t (à partir du point actuel du conducteur).
- D_t : La distance maximale de détour autorisée pour prendre un passager.
- D_{p_t} : La distance nécessaire pour prendre un passager nécessitant un détour.
- T_{ar_t} : Le temps d'arrivée au plus tard du passager.
- $T_{ar_{est_t}}$: Le temps nécessaire pour l'arrivée du passager à sa destination. Autrement dit, c'est le temps estimé nécessaire pour que le passager atteigne sa destination.

Exemple

En se basant sur l'exemple présenté ci-dessus. Nous supposons que le conducteur est au point 2 et les deux passagers, P_3 et P_4 , sont au point 7. L'observation pour le passager P_3 : $O_t = (4,3,0,15,13,8,400,200,22,5)$, où : 4 est le nombre de sièges disponibles, 3 min est le temps pour aller à la prochaine destination, 0 min représente le temps d'attente consommé, 15 min est le temps d'attente maximal, 13 min est le temps maximal de détour, 8 min est le temps nécessaire pour faire le détour, 400 mètres est la distance maximale pour prendre le passager en compte dans le détour, 200 mètres est la distance pour prendre le passager dans le détour, 22 min est le temps au plus tard pour déposer le passager, 5 min est le temps nécessaire pour déposer le passager. Quant à l'observation du passager P_4 : $O_t = (6,3,0,10,13,13,400,200,22,19)$.

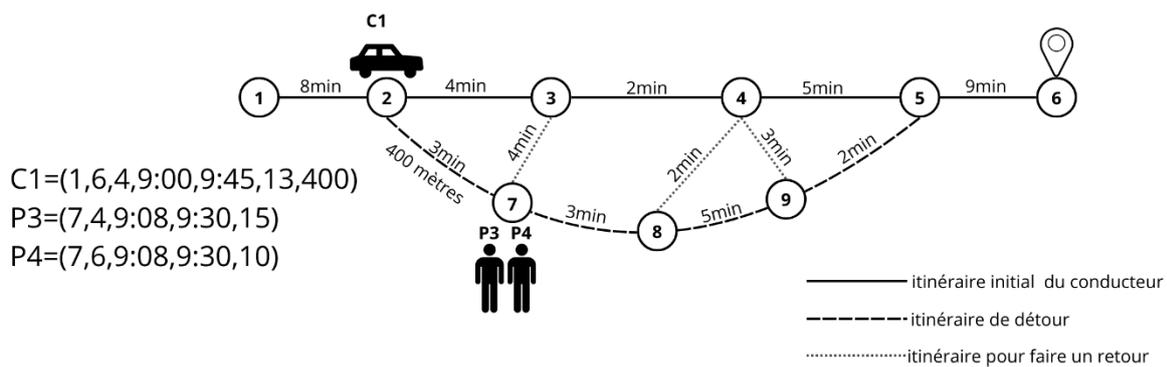


Figure 2.2 : Exemple d'observation.

2.3.3. Action

L'action A représente la décision du conducteur vis-à-vis d'un passager particulier. On considère deux actions :

- $A = 0 \rightarrow$ refuser la demande
- $A = 1 \rightarrow$ accepter la demande

2.3.4. Récompense

Pour atteindre nos principaux objectifs du système, qui consistent à réduire le temps d'attente des passagers, minimiser le temps de détour des conducteurs et maximiser le nombre de demandes acceptées dans un détour. Nous définissons un vecteur de récompense, composé de trois signaux (scalaires) où chaque signal représente un objectif. Il est crucial que nos récompenses fournissent des informations nécessaires pour guider l'action de notre agent. On note, $R = (r_1, r_2, r_3)$.

- Pour notre *premier objectif*, qui est la minimisation du temps d'attente des passagers, la récompense est calculée de la manière suivante :

$$r_1 = wt_max_t - (wt_t + T_{nd})$$

- Où : wt_max_t : Le temps d'attente maximum du passager, wt_t : Le temps d'attente actuel du passager, T_{nd} : Le temps d'arrivée à la prochaine destination.
- La *deuxième récompense* représente l'objectif de minimiser le temps de détour des conducteurs. Nous définissons la récompense r_2 de la manière suivante :

$$r_2 = dt_max_t - dt_t$$

- Où : dt_max_t : Le temps maximal de détour du conducteur, dt_t : Le temps nécessaire pour faire un détour au moment t .
- La *dernière récompense* vise à maximiser l'utilisation des véhicules et ainsi inciter les conducteurs à accepter un maximum de demandes. Cela implique d'encourager le conducteur à prendre des passagers même s'ils ne sont pas sur son itinéraire initial, ce qui permet d'optimiser l'utilisation du véhicule tout en respectant les contraintes. Cet objectif entre en conflit avec les deux premiers objectifs, et sa mesure ne se base pas sur les informations environnementales comme les deux premiers. Nous proposons de calculer cette récompense ainsi :

$$r_3 = \begin{cases} +1 & \text{si } D_{pt} \leq D_t \\ 0 & \text{sinon} \end{cases}$$

○ Où : D_{pt} : La distance nécessaire pour prendre un passager nécessitant un détour.

D_t : La distance maximale de détour autorisée pour prendre un passager.

Dans l'exemple suivant, nous allons motiver notre choix de ces récompenses.

Exemple

Nous allons justifier et motiver notre choix des récompenses à travers un scénario de l'exemple sus-présenté. Nous supposons que le conducteur C_1 est au point 2 et qu'il reçoit simultanément deux demandes, P_3 et P_4 . Le point de départ pour ces deux passagers est identique et correspond à la prochaine destination sur le détour du conducteur. La destination de P_3 est le point 4, situé sur l'itinéraire initial et la destination de P_4 est le point 6, qui se trouve sur l'itinéraire initial du conducteur.

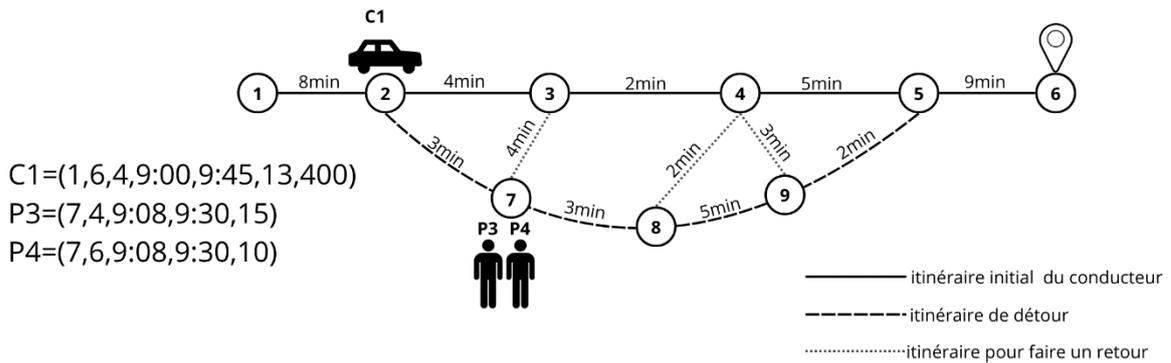


Figure 2.3 : Exemple de récompense.

Si le conducteur décide de prendre le passager P_3 , les récompenses seront : $r_1=15-(0+3)=10$, $r_2=13-(3+3+2)=5$ et $r_3=1$. Et pour le passager P_4 , les récompenses seront : $r_1=10-(0+3)=7$, $r_2=13-(3+3+5+2)=0$ et $r_3=1$.

Les récompenses de P_3 sont : $(10,5,1)$ et de P_4 sont : $(7,0,1)$, donc le conducteur va choisir le passager P_3 .

2.3.5. Episode

Un épisode correspond au moment où l'agent prend une décision. Dans notre contexte, chaque position du conducteur représente un épisode, c'est-à-dire que chaque intervalle de temps entre S_t et S_{t+1} constitue un épisode.

2.3.6. Transition d'état

Quand un passager atteint sa destination, cela entraîne une mise à jour de la liste des destinations des passagers restants. Cette actualisation marque une transition vers un nouvel état. Cet état mis à jour reflète la configuration actuelle du système après le débarquement du passager.

2.4. Phase d'apprentissage

La phase d'apprentissage revêt une importance cruciale dans notre travail, impliquant l'élaboration du modèle mentionné précédemment, son intégration dans l'agent, et lui permettant d'acquérir un comportement décisionnel répondant aux diverses contraintes du problème tout en optimisant les objectifs définis.

Pour élaborer notre modèle proposé, nous devons d'abord fournir des informations correspondant à la description du problème, telles que les points de départ et d'arrivée. Pour cela, nous avons décidé d'utiliser la base de données Uber Pickup dans la ville de New York [2]. De plus, nous avons besoin d'autres informations sur la direction entre les paires de points (départ, arrivée). Nous avons utilisé l'API MapQuest [3] pour obtenir ces informations nécessaires.

Une fois que nous aurons collecté toutes les données nécessaires, nous identifierons les données concernant chaque conducteur, telles que leurs points de départ et d'arrivée, ainsi que leurs contraintes : l'heure de départ et l'heure d'arrivée au plus tard, le nombre de passagers qu'ils peuvent prendre, le temps maximal de détour et la distance maximale pour qu'un passager soit pris en détour. Chaque conducteur dispose de son itinéraire principal ainsi que d'un itinéraire alternatif pour les détours éventuels. Ces parcours sont segmentés en une série de points de contrôle, correspondant à des endroits spécifiques où le conducteur examinera les

demandes de covoiturage disponibles. Il peut consulter les demandes à la fois à son emplacement actuel et au prochain point sur son trajet.

Pour les passagers, nous devons extraire leurs points de départ et d'arrivée correspondants, ainsi que l'heure de départ, l'heure d'arrivée et le temps d'attente. Le temps d'attente commence à diminuer à partir de l'heure de départ. Lorsque le temps d'attente est écoulé, c'est-à-dire lorsque l'heure de départ maximale est dépassée, la demande sera retirée.

2.4.1. Décision de l'agent

Le processus de décision de l'agent se divise en deux étapes principales : la première consiste à décider d'accepter ou de refuser la demande dans l'état actuel ; la deuxième concerne les changements qui se produisent dans l'environnement à la suite de cette décision. Ce processus se répète jusqu'à ce que le conducteur atteigne sa destination finale.

Lors du processus décisionnel, le conducteur peut recevoir deux types de demandes : celles avec des points de départ et de destination appartenant à l'itinéraire initial du conducteur, et celles qui nécessitent un détour.

Dans le cas où les points de départ et de destination se trouvent sur l'itinéraire initial du conducteur, les demandes doivent être évaluées en fonction de plusieurs critères, tels que le temps d'attente actuel du passager, son temps d'attente maximal, le temps restant avant sa prochaine destination et le nombre de sièges disponibles dans le véhicule.

D'autre part, il y a les demandes qui nécessitent un détour, soit pour atteindre le point de départ du passager, soit pour atteindre son point de destination, ou les deux. Ces demandes ont des critères additionnels tels que le temps consommé par le détour, le respect du temps maximal autorisé pour un détour, et il faut également que l'heure d'arrivée prévue du passager et celle du conducteur ne dépassent pas les horaires d'arrivée spécifiés par eux. Dans le cas où le conducteur est sur l'itinéraire principal et que le point de départ du passager nécessite un détour, nous ajoutons un critère supplémentaire, à savoir la distance pour prendre le passager en faisant un détour qui doit respecter la distance maximale autorisée. En se basant sur ces critères, l'agent décide d'accepter ou de rejeter la demande. Une fois qu'une action est sélectionnée, la récompense est calculée en fonction de l'état actuel de l'agent. Cela permet à l'agent de

déterminer si cette action représente une bonne décision ; sinon, l'agent évitera de prendre la même décision s'il se retrouve dans la même situation plus tard.

Après avoir pris une décision, si le conducteur accepte une demande, il doit vérifier si le passager se trouve à la même position que lui pour le prendre ; sinon, il le notifie. Il doit également vérifier si aucun des passagers qu'il a déjà notifiés ne se trouve pas à sa position actuelle pour les prendre. De plus, s'il y a des passagers à bord, il vérifie leurs destinations pour les déposer. En outre, il va vérifier s'il y a de nouvelles demandes ou non.

Dans le cadre de notre étude, nous avons catégorisé les passagers en quatre types, basés sur la concordance de leurs points de départ et de destination avec l'itinéraire initial et les détours potentiels des conducteurs :

- *Type 1* : Passagers dont le point de départ et la destination sont tous les deux situés sur l'itinéraire initial du conducteur.
- *Type 2* : Passagers avec un point de départ sur l'itinéraire initial et une destination située dans un détour.
- *Type 3* : Passagers dont le point de départ se trouve dans un détour, tandis que la destination est sur l'itinéraire initial.
- *Type 4* : Passagers pour lesquels à la fois le point de départ et la destination se situent dans un détour par rapport à l'itinéraire initial du conducteur.

La Figure 2.4 présente le processus de décision de l'agent lorsqu'il reçoit une nouvelle demande.

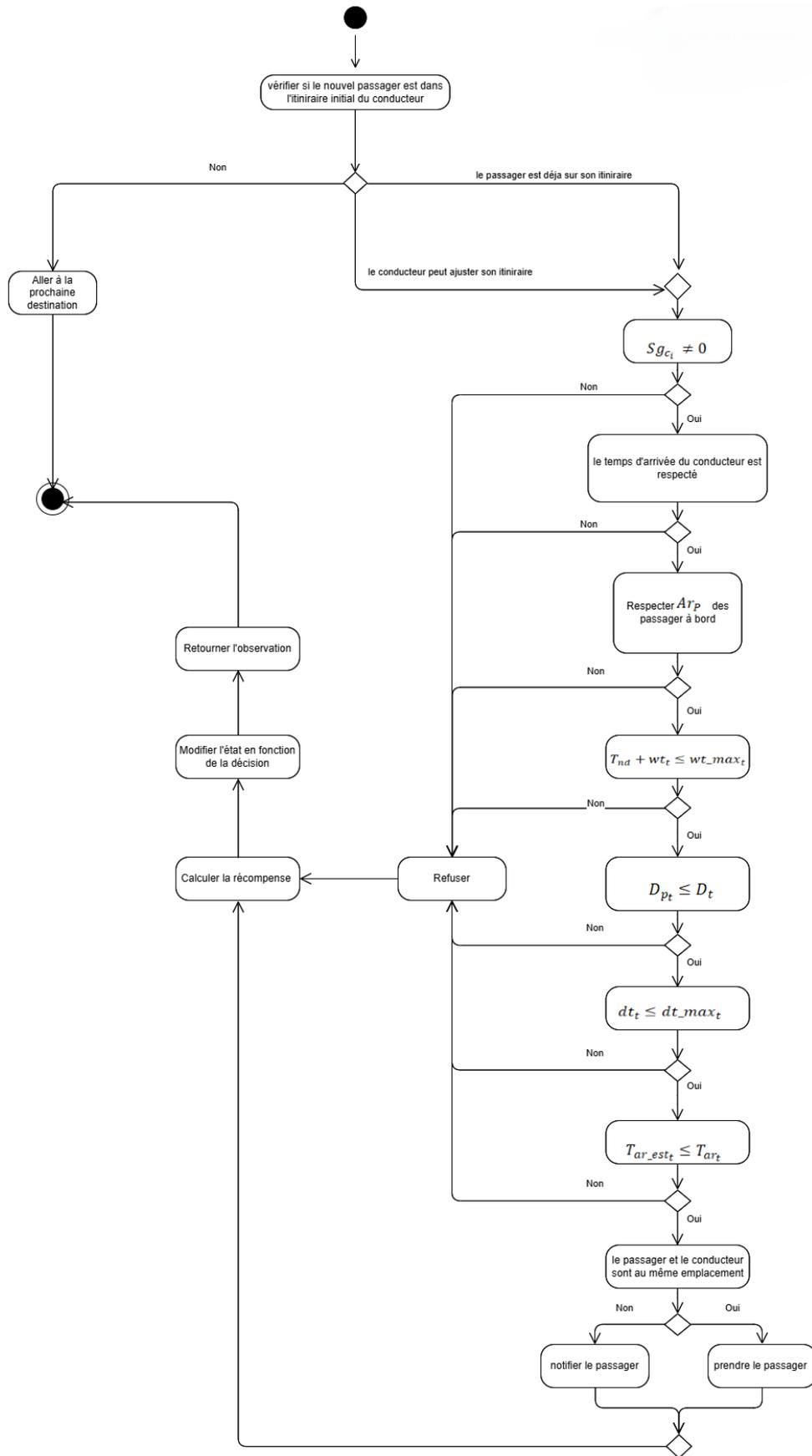


Figure 2.4 : Processus de décision de l'agent.

Dans ce qui suit nous présentons l'algorithme de décision de l'agent que nous proposons.

Algorithme de décision de l'agent
<p>Les entrées : $P_j(t)$</p> <p>Les sorties : (Accepter /Refuser)</p> <p>Si $Sg_{c_i} \neq 0$ alors</p> <p style="padding-left: 2em;">Nv_point_détour \leftarrow obtenir_nv_point_detour ($P_j(t)$, G_{ap})</p> <p style="padding-left: 2em;">Si calculer_temps_d_arrivée_conducteur (G_c, Nv_point_détour) $\leq T_{a_{c_i}}$ alors</p> <p style="padding-left: 4em;">Si verifier_temps_d_arrivée_des_passagers (G_{ap}, Ar_p, Nv_point_détour) alors</p> <p style="padding-left: 6em;">Si $T_{nd} + wt_t \leq wt_{max_t}$ alors</p> <p style="padding-left: 8em;">Si $D_{p_t} \leq D_t$ alors</p> <p style="padding-left: 10em;">Si $dt_t \leq dt_{max_t}$ alors</p> <p style="padding-left: 12em;">Si $T_{ar_{est_t}} \leq T_{ar_t}$ alors</p> <p style="padding-left: 14em;">Retourner (Accepter)</p> <p>Retourner (Refuser)</p>

- « obtenir_nv_point_detour($P_j(t)$, G_{ap}) » est une fonction qui retourne les nouveaux points de détour et de retour choisis en respectant les positions de destination des passagers à bord ainsi que la destination de la demande. Si l'agent décide d'accepter la demande, ces points doivent être affectés à G_{ar} . Les entrées de la fonction sont $P_j(t)$: la demande du passager, et G_{ap} : les coordonnées géographiques des points d'arrivée des passagers à bord.
- « calculer_temps_d_arrivée_conducteur(G_c , Nv_point_détour) » est une fonction qui calcule le temps d'arrivée estimé du conducteur. Les entrées de cette fonction sont les nouveaux points de détour et les points géographiques d'arrivée du conducteur.
- « verifier_temps_d_arrivée_des_passagers(G_{ap} , Ar_p , Nv_point_détour) » est une fonction qui vérifie si le temps d'arrivée des passagers à bord n'est pas violé en respectant l'itinéraire avec les nouveaux points de détour. Elle retourne un booléen *True* ou *False*. Les entrées sont, G_{ap} : les coordonnées géographiques des points d'arrivée des

passagers à bord, Ar_p : l'ensemble des temps d'arrivées au plus tard des passagers à bord et $Nv_point_détour$.

2.4.2. Apprentissage

Dans notre approche, nous avons opté pour l'utilisation de l'algorithme d'apprentissage par renforcement appelé *Pareto Conditioned Networks (PCN)* (Reymond, Bargiacchi, & Nowé, 2022), Cet algorithme est employé dans le domaine de l'optimisation multi-objectifs en apprentissage par renforcement. Il s'agit d'une extension de l'algorithme de réseaux neuronaux conditionnels (CNN) qui intègre le concept de dominance de Pareto pour guider l'apprentissage vers des solutions qui équilibrent plusieurs objectifs concurrents.

PCN est conçu pour apprendre efficacement et de manière exhaustive toutes les politiques qui mènent à des solutions Pareto-efficaces. Grâce à un seul réseau neuronal, PCN peut généraliser l'expérience sur l'ensemble des retours multi-objectifs possibles, offrant ainsi une solution stable et évolutive capable de couvrir un large éventail d'objectifs.

PCN fonctionne en associant chaque transition passée à son retour d'épisode, puis en entraînant le réseau de manière à ce qu'il puisse réactiver cette transition lorsqu'il est conditionné à ce retour. Cette approche transforme le problème d'optimisation en un problème de classification, permettant au réseau neuronal de généraliser l'expérience à travers tous les retours multi-objectifs possibles, offrant ainsi une solution stable et évolutive pour couvrir un large espace d'objectifs.

Cet algorithme possède plusieurs avantages. Il permet de trouver des solutions Pareto-optimales qui équilibrent plusieurs objectifs concurrents. PCN utilise des techniques d'apprentissage par renforcement profondes pour améliorer l'efficacité de l'exploration et de l'exploitation de l'espace des solutions. De plus, cet algorithme s'adapte bien à des environnements complexes.

2.5. Conclusion

Dans ce chapitre, nous avons présenté la modélisation que nous avons proposée pour résoudre le problème d'appariement dynamique sous contraintes, en visant à optimiser des objectifs conflictuels avec l'apprentissage par renforcement. Nous avons décrit en détail le problème et expliqué la modélisation de la solution.

Dans le chapitre suivant, nous fournirons des informations détaillées sur l'implémentation de notre application.

Chapitre 3 : Implémentation

3.1. Introduction

Dans ce chapitre, nous allons exposer les aspects techniques relatifs à l'application de notre travail. Tout d'abord, nous décrirons les outils et le langage de programmation utilisés pour l'implémentation. Ensuite, nous présenterons la base de données employée. Nous détaillerons par la suite l'implémentation ainsi que le processus d'apprentissage et de validation de notre système. Enfin, nous allons aborder le simulateur que nous avons développé et les résultats obtenus lors de nos expérimentations.

3.2. Langages et outils de développement

Nous avons utilisé divers outils pour faciliter le développement et la mise en œuvre de notre système. Cette section détaillera chacun de ces outils.

- *Python* : Python est un langage de programmation interprété, orienté objet et de haut niveau avec une sémantique dynamique. Il est idéal pour le développement rapide d'applications grâce à ses structures de données intégrées, son typage dynamique et sa syntaxe simple, qui favorise la lisibilité. Python prend en charge les modules et les packages, ce qui encourage la modularité et la réutilisation du code. De plus, son interpréteur et sa vaste bibliothèque standard sont disponibles gratuitement pour toutes les principales plateformes [4].
- *Visual Studio Code* : est un éditeur de code source léger mais puissant disponible sur Windows, macOS et Linux. Il dispose d'un riche écosystème d'extensions pour les langages et environnements d'exécution tels que JavaScript, C++, C#, Java, Python, etc [5].
- *Les bibliothèques utilisées*
 1. *Numpy* : est la bibliothèque essentielle pour le calcul scientifique en Python, offrant des tableaux multidimensionnels et de nombreuses routines pour des

opérations rapides, couvrant les domaines des mathématiques, la logique, l'algèbre linéaire, les statistiques, etc [6].

2. *Pandas* : est un outil open source d'analyse et de manipulation de données, développé sur le langage de programmation Python. Il est rapide, puissant, flexible et facile à utiliser [7].
 3. *Gymnasium* : est une bibliothèque Python open source dédiée au développement et à la comparaison des algorithmes d'apprentissage par renforcement. Son interface est simple, et capable de représenter des problèmes généraux d'apprentissage par renforcement, tout en offrant un wrapper de compatibilité pour les anciens environnements Gym. Il s'agit d'un fork maintenu de la bibliothèque Gym d'OpenAI [8].
 4. *MORL-Baselines* : est une bibliothèque d'algorithmes d'apprentissage par renforcement multi-objectif (MORL). Ce référentiel vise à contenir des implémentations fiables d'algorithmes MORL en PyTorch [9].
 5. *Tkinter* : est une bibliothèque intégrée à Python permettant de développer des interfaces graphiques utilisateur (GUI) en utilisant le toolkit Tcl/Tk. Elle est compatible avec la plupart des systèmes Unix, macOS, ainsi que sur Windows. Elle permet de concevoir divers éléments GUI tels que des fenêtres, des boutons, des champs de texte, et offre une approche intuitive et pythonique pour leur manipulation [10].
- *API MapQuest* : est un service en ligne qui permet de demander des itinéraires entre différents lieux par une requête HTTP, avec des résultats fournis au format JSON ou XML [3]. Une des fonctionnalités offertes par MapQuest est l'option *AlternateRoutes*, qui permet de récupérer plusieurs itinéraires alternatifs pour un trajet donné.

3.3. Echantillon de données

Pour construire notre modèle, nous avons besoin d'un échantillon de données contenant les informations liées à notre problématique, telles que les trajets, les directions de chaque trajet ainsi que le temps (de départ, d'arrivée, ...).

Pour ce faire, nous avons choisi d'utiliser la base de données *Uber Pickups* dans la ville de *New York* [1], les données de ce répertoire couvrent plus de 4,5 millions de trajets Uber

effectués dans la ville de New York entre avril et septembre 2014. Pour obtenir des informations plus détaillées sur les directions nécessaires, nous avons utilisé l'*API MapQuest Directions*.

3.3.1. Données des trajets

De la base de données *Uber Pickups*, nous avons choisi d'utiliser le fichier « *Federal 02216* ». Nous avons sélectionné aléatoirement un échantillon de données de 50 trajets afin de mener nos expérimentations.

A l'aide de l'*API MapQuest Directions*, nous avons pu obtenir des informations plus détaillées sur les directions nécessaires de cet échantillon. Cela nous a permis de générer un nouvel ensemble de données contenant les informations nécessaires pour pouvoir valider notre travail. Chaque requête envoyée à l'*API MapQuest Directions* renvoie les informations suivantes (figure 3.1) :

- **Info** : contient le copyright.
- **Route** : contient les informations des itinéraires, qui sont :
 - *Time* : la durée de l'itinéraire en secondes.
 - *Distance* : la distance de l'itinéraire en miles ou kilomètres.
 - *Localisation* : c'est une liste qui contient les adresses et les coordonnées GPS des points de départ et d'arrivée.
 - *Legs* : c'est une liste qui contient les informations de l'itinéraire :
 - *Maneuvers* : liste des manœuvres qui contient des informations spécifiques.
 - *Index* : l'indice de la manœuvre.
 - *Time* : le temps entre un point et le point qui le suit.
 - *Distance* : la distance entre un point et le point qui le suit.
 - *StartPoint* : la localisation du point, représenté par des coordonnées GPS.
 - *lat* : latitude.
 - *lng* : longitude.
 - *AlternateRoutes* : c'est une liste qui contient les mêmes informations que *Route*, mais pour une liste de routes alternatives pour le trajet demandé.

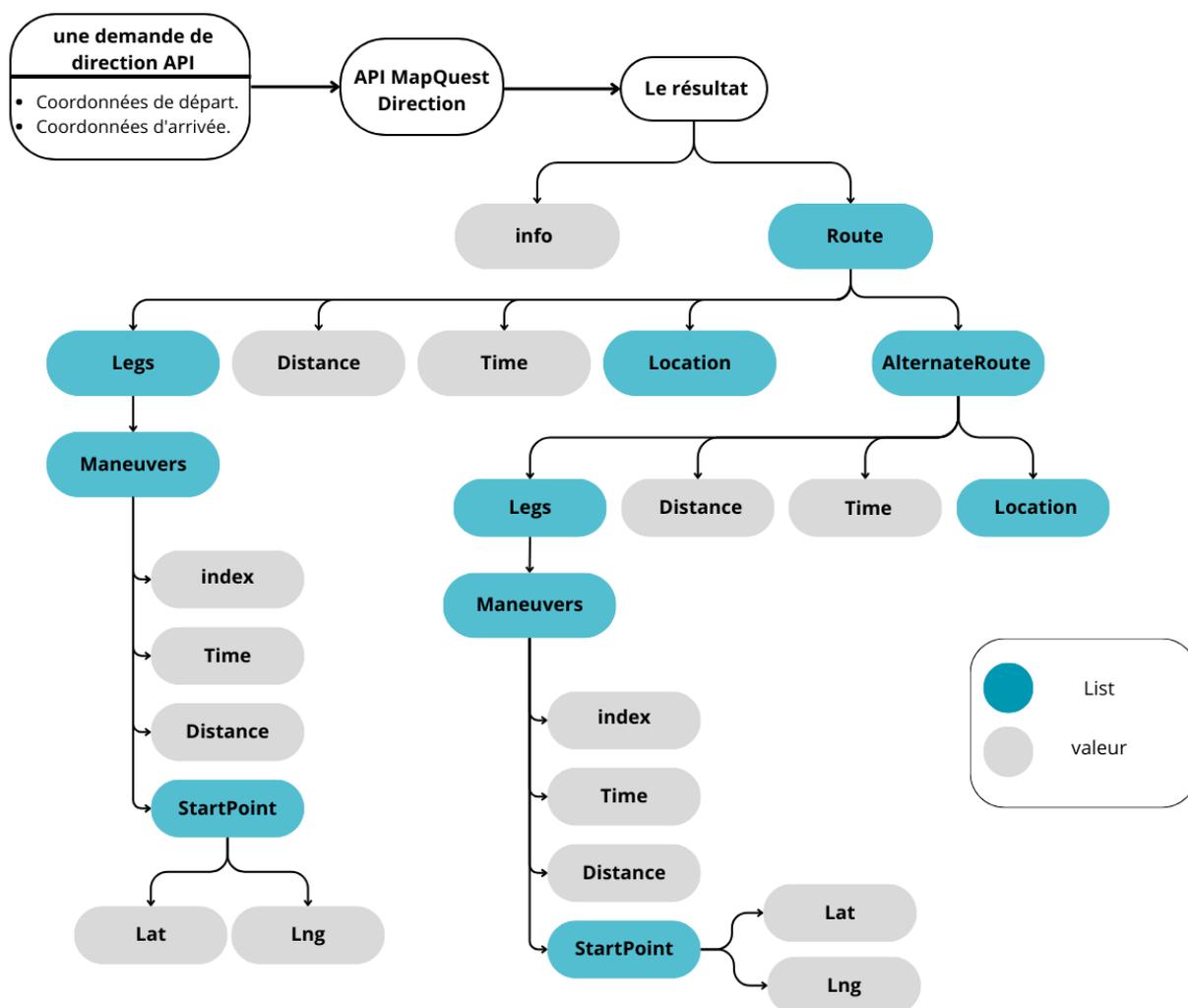


Figure 3.1 : Résultat d'une requête de l'API MapQuest Directions.

A l'aide de l'API MapQuest Directions, on a également extrait la distance et le temps entre les points de la route initiale et de la route alternative, et nous avons regroupé les points ayant la plus petite distance dans un fichier séparé pour les utiliser dans les calculs afin d'obtenir un détour et un retour. Cette opération permet d'identifier les segments de route les plus courts et les plus efficaces, facilitant ainsi une optimisation plus précise des trajets. En regroupant ces points dans un fichier distinct, nous pouvons effectuer des analyses détaillées sur les trajets potentiels, en comparant les distances et les temps de trajet pour chaque segment. Cela nous aide à minimiser le temps de détour pour les conducteurs, à réduire le temps d'attente pour les passagers et à optimiser la distance de pick-up dans un détour.

3.4. Apprentissage

Nous avons entraîné notre agent (le conducteur) avec *Pareto Conditioned Network* (PCN) en utilisant un ensemble de données réelles extraites de la base de données présentée précédemment. Pour évaluer les résultats de l'apprentissage, nous avons considéré le trajet complet d'un conducteur comme un épisode. La durée totale de l'apprentissage s'élève à 900 000 épisodes au cours desquels l'agent cherche à maximiser les récompenses cumulées, avec un paramètre gamma (γ) de 0,99. Ce paramètre permet de pondérer l'importance des récompenses immédiates par rapport aux récompenses futures. Une valeur de γ proche de 1 signifie que les récompenses futures sont presque aussi importantes que les récompenses immédiates, tandis qu'une valeur proche de 0 signifie que les récompenses immédiates sont beaucoup plus importantes.

La figure 3.2 représente la perte pendant l'entraînement (train\loss), qui est la mesure de la différence entre les actions prédites par le modèle et les actions réelles optimales. On remarque que la perte est de plus en plus faible tout au long la phase d'apprentissage ce qui signifie que l'agent apprend correctement à maximiser ses récompenses.

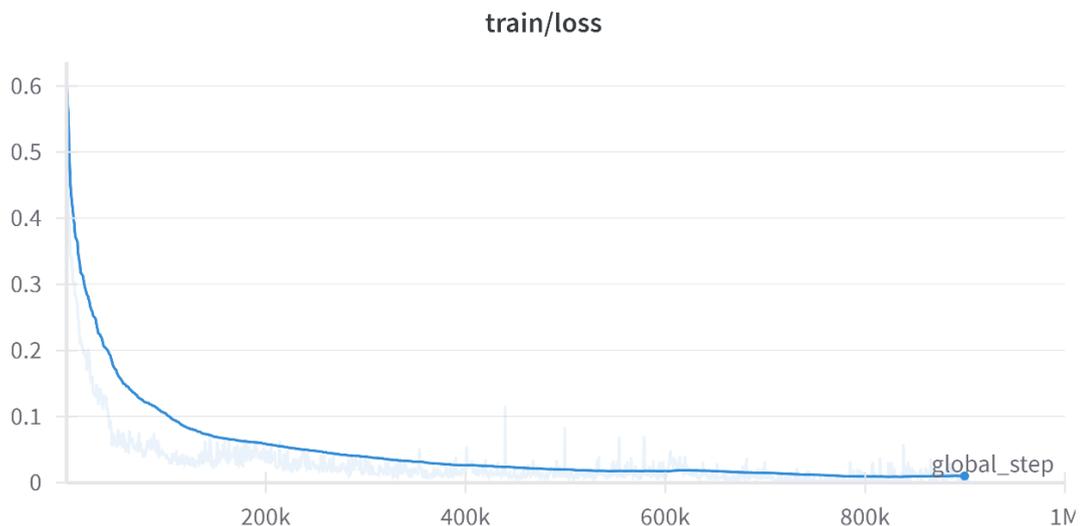


Figure 3.2 : Perte dans l'apprentissage.

Afin d'évaluer la qualité de l'apprentissage et de la décision de l'agent, nous observons aussi la mesure de l'hypervolume (figure 3.3). Cette mesure est un outil essentiel en optimisation multi-objectifs, pour évaluer la qualité globale des solutions obtenues en termes d'équilibre entre les différents objectifs. Un hypervolume plus grand indique que l'ensemble des solutions non dominées (solutions Pareto-optimales) couvre mieux l'espace des objectifs, ce qui reflète une meilleure performance de l'agent.

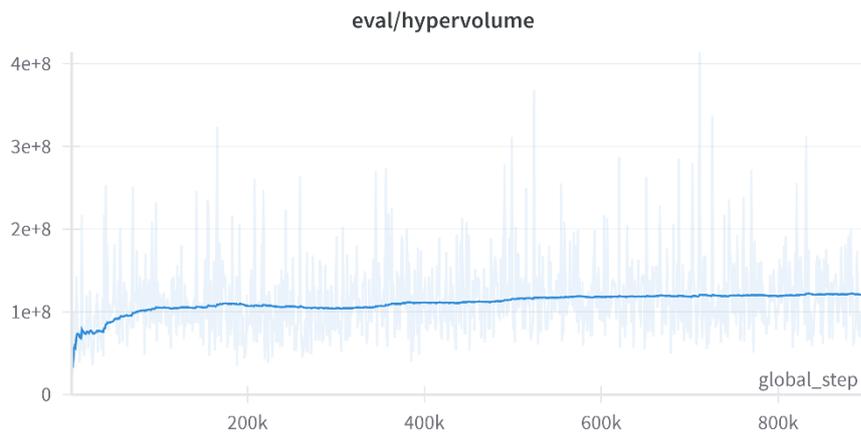


Figure 3.3 : Hypervolume.

Avant de procéder à la simulation en temps réel pour évaluer les décisions de l'agent, nous avons mené des tests préliminaires sur différents scénarios. Ces tests avaient pour but de vérifier si l'agent prend les bonnes décisions dans chaque situation. Nous avons exposé l'agent à une variété de situations d'observation pour tester la fiabilité de ses prédictions dans des contextes diversifiés. Les observations variées fournies à l'agent comprennent un ensemble de facteurs importants qui influencent ses décisions.

Les variations des paramètres (temps d'attente, temps de détour, distance de détour, fenêtres temporelles, itinéraires) sont ajustées de manière à resserrer et alléger les contraintes afin d'évaluer les décisions de l'agent et de tester la robustesse et la flexibilité du système.

Le tableau 3.1 illustre quelques scénarios d'observation et les décisions prises par l'agent dans chaque situation.

Scénarios Observation	Pas de demande	$T_{nd} + wt_t > wt_{max_t}$	$dt_t > dt_{max_t}$	$D_{p_t} > D_t$	Scénarios où les contraintes sont respectées		
					Sans détour	Avec détour	
						Pick-up	Pick-up et drop-off
T_{nd}	0 min	4 min	2 min	3 min	0 min	3 min	2 min
wt_t	0 min	9 min	5 min	2 min	2 min	2 min	5 min
wt_{max_t}	0 min	10 min	9 min	8 min	6 min	8 min	10 min
Sg_t	2	3	2	2	3	3	3
dt_{max_t}	20 min	14 min	12 min	15 min	10 min	15 min	20 min
dt_t	0 min	10 min	16 min	5 min	0 min	5 min	16 min
D_t	400 mètres	800 mètres	1 km	700 mètres	500 mètres	700 mètres	800 mètres
D_{p_t}	0 mètres	400 mètres	700 mètres	850 mètres	0 mètres	530 mètres	400 mètres
T_{ar_t}	0 min	41 min	45 min	25 min	23 min	27 min	35 min
$T_{ar_est_t}$	0 min	30 min	35 min	24 min	23 min	24 min	28 min
Les décisions de l'agent	<i>Refuser</i>	<i>Refuser la demande</i>	<i>Refuser la demande</i>	<i>Refuser la demande</i>	<i>Accepter la demande</i>	<i>Accepter la demande</i>	<i>Accepter la demande</i>

Tableau 3.1 : Exemples des tests.

3.5. Simulateur

Nous avons développé un simulateur pour évaluer notre modèle proposé de covoiturage dynamique. La simulation a été effectuée sur une période de 12 heures, de 9h00 à 21h00. Pendant cette période, toutes les 5 minutes, le système reçoit un nombre aléatoire des offres et des demandes ($\Delta t = 5 \text{ min}$). Les emplacements initiaux pour ces offres et demandes sont générés aléatoirement, en utilisant notre ensemble de données.

Pour les conducteurs, nous avons défini des paramètres clés tels qu'un temps de détour maximal, une distance maximale pour le pick-up, et un nombre de sièges disponibles. Concernant les passagers, un temps d'attente maximal a été attribué. Ces paramètres sont présentés dans le tableau suivant :

Nombre des offres	Aléatoire entre 6 et 9
Nombre des demandes	Aléatoire entre 11 et 15
Temps de détour maximal	Aléatoire entre 15 et 40 minutes
Distance maximale pour un pick-up	Aléatoire entre 500 et 1500 mètres
Nombre de sièges	4
Temps d'attente maximal	Aléatoire entre 15 et 30 minutes

Tableau 3.2 : Paramètres de la simulation.

La figure 3.4 présente l'interface principale de notre simulateur.

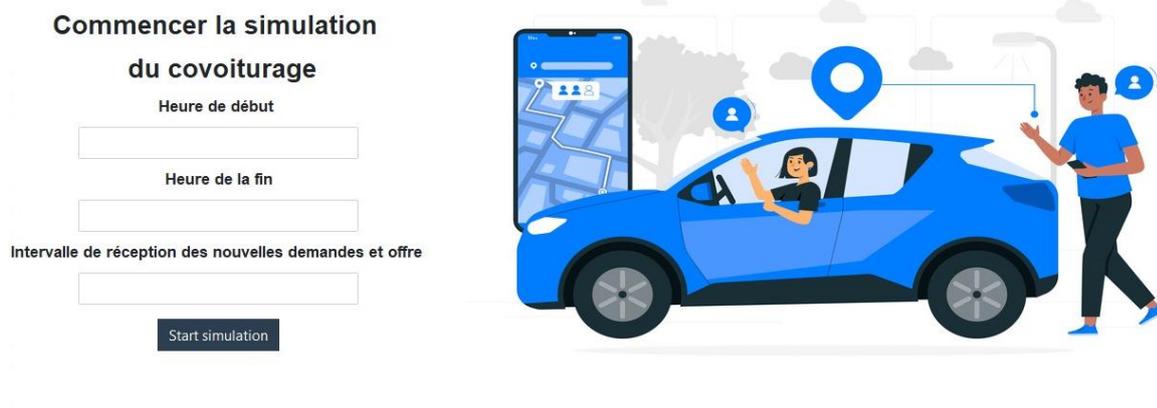


Figure 3.4 : Interface du simulateur.

Les résultats de la simulation sont affichés à travers une interface qui est présentée dans la figure 3.5.

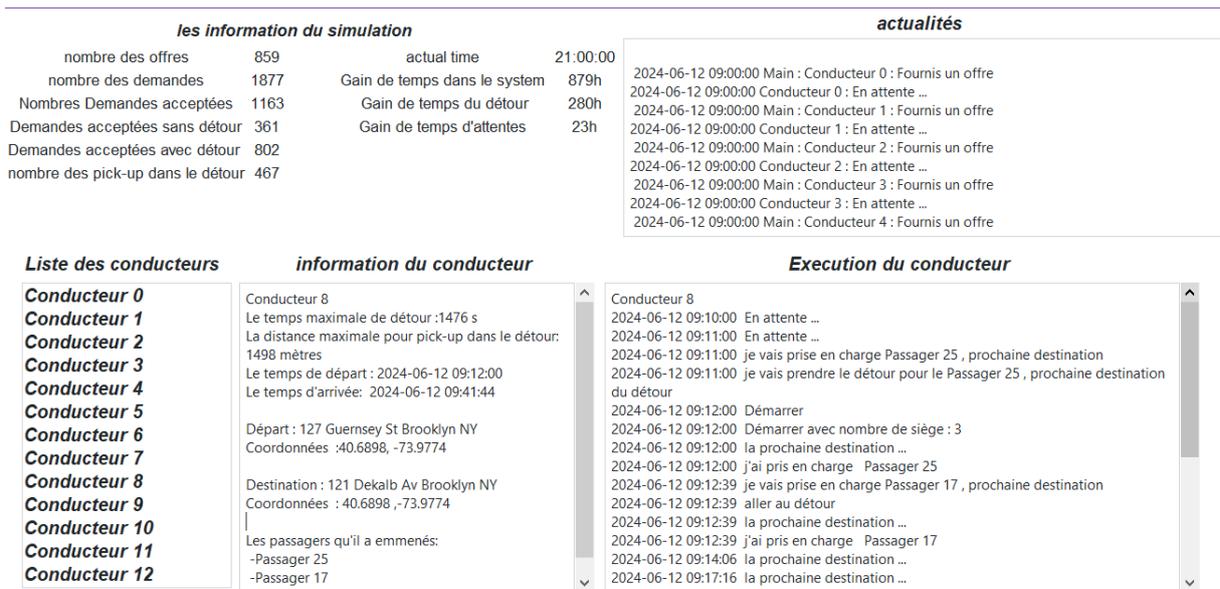


Figure 3.5 : Résultat de la simulation.

Dans notre interface, il est possible de voir tous les conducteurs dans le système et de sélectionner n'importe quel conducteur pour consulter les informations de son offre, telles que l'heure de départ et l'heure d'arrivée au plus tard, les adresses et les coordonnées de départ et de destination, ainsi que les passagers avec lesquels il a été apparié, et l'exécution de son trajet.

Il est également possible de voir certaines statistiques, telles que le nombre des offres et des demandes reçues ainsi que le nombre des demandes acceptées. Les demandes acceptées peuvent être classées en deux catégories :

- Les demandes des passagers qui ne nécessitent pas de détour : leur départ et leur destination se trouvent sur l'itinéraire initial du conducteur.
- Les demandes des passagers qui nécessitent un détour. Ce détour peut être soit pour un pick-up (prendre en charge le passager), soit pour un drop-off (déposer le passager), soit les deux (pick-up et drop-off dans le trajet de détour).

En outre, l'interface propose d'autres indicateurs pertinents relatifs à la performance de notre modèle.

Les statistiques de la simulation sont présentées dans les Figure 3.6, Figure 3.7 et le Tableau 3.3.

Nombre des offres	859
Nombre des demandes	1877
Nombre des demandes acceptées	1163
Nombre des demandes acceptées sans détour	361
Nombre des demandes acceptées avec détour	802
Nombre des demandes acceptées avec un pick-up dans le détour	467
Gain du temps dans le système	879h
Gain en temps de détour	280h
Gain en temps d'attente	23h

Tableau 3.3 : Résultats de la simulation.

Ces résultats sont illustrés dans la figure 3.6 à travers un histogramme.

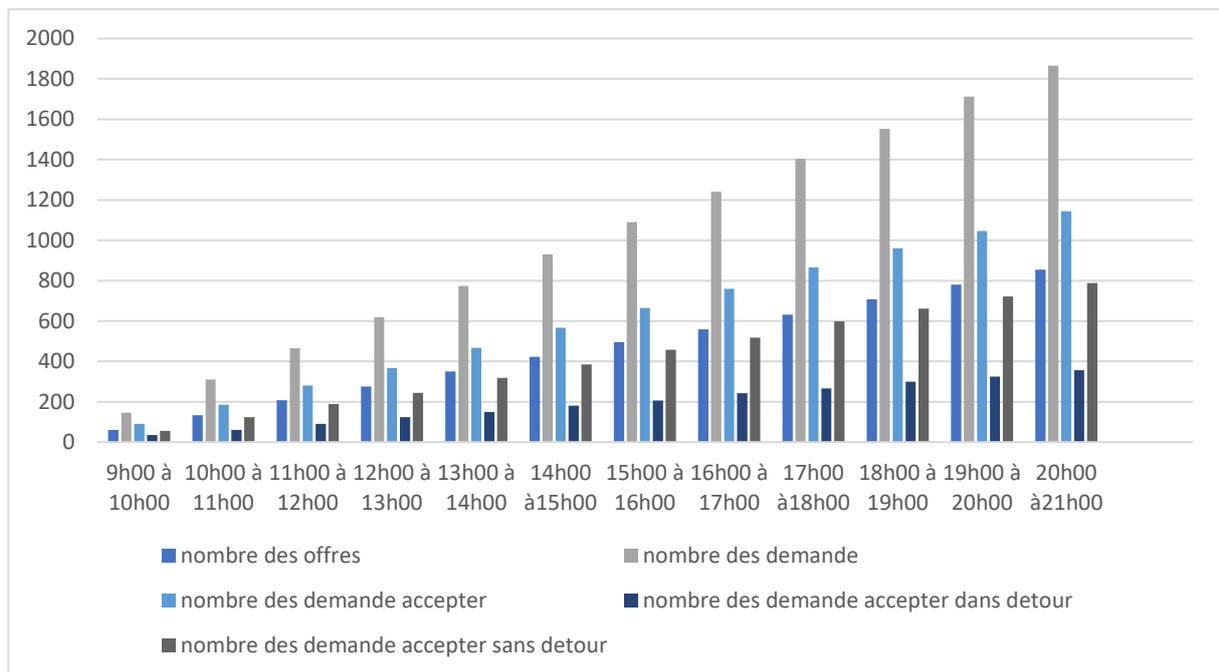


Figure 3.6 : Visualisation des résultats de la simulation.

Les résultats de notre simulation, présentés dans les figures 3.5, 3.6 et le tableau 3.3, valident l'efficacité de notre modèle de covoiturage proposé. L'analyse des données indique que les conducteurs acceptent 61 % des demandes reçues, parmi ces demandes, 68 % impliquent un détour. Notamment, la simulation a révélé que les passagers ont économisé 23 heures de temps d'attente. De plus, nous avons enregistré 280 heures de détour non consommées par les conducteurs. Ces conducteurs ont aussi gagné un total de 879 heures grâce à des arrivées plus précoces que prévues. Ces économies de temps illustrent non seulement l'efficacité de notre système, mais aussi son potentiel à améliorer significativement l'expérience de covoiturage pour tous les participants.

L'efficacité de notre modèle de covoiturage est encore soulignée par le fait que les conducteurs n'ont utilisé que 39 % de leur temps de détour autorisé, ce qui indique une planification efficace des itinéraires. De même, les passagers ont bénéficié d'un temps d'attente moyen représentant seulement 13 % du temps d'attente maximal alloué. Ces résultats démontrent que notre modèle atteint ses objectifs principaux : réduire le temps d'attente et le temps de détour, tout en maximisant le nombre de demandes acceptées.

La figure 3.7 présente la variété des types de demandes acceptées ainsi que le pourcentage de chacun de ces types. Ces types sont : des demandes **sans détour**, des demandes avec **détour pour pick-up** (prendre un passager), des demandes avec **détour pour drop-off** (déposer un passager) et des demandes avec **détour pour pick-up et drop-off** (prendre et déposer).

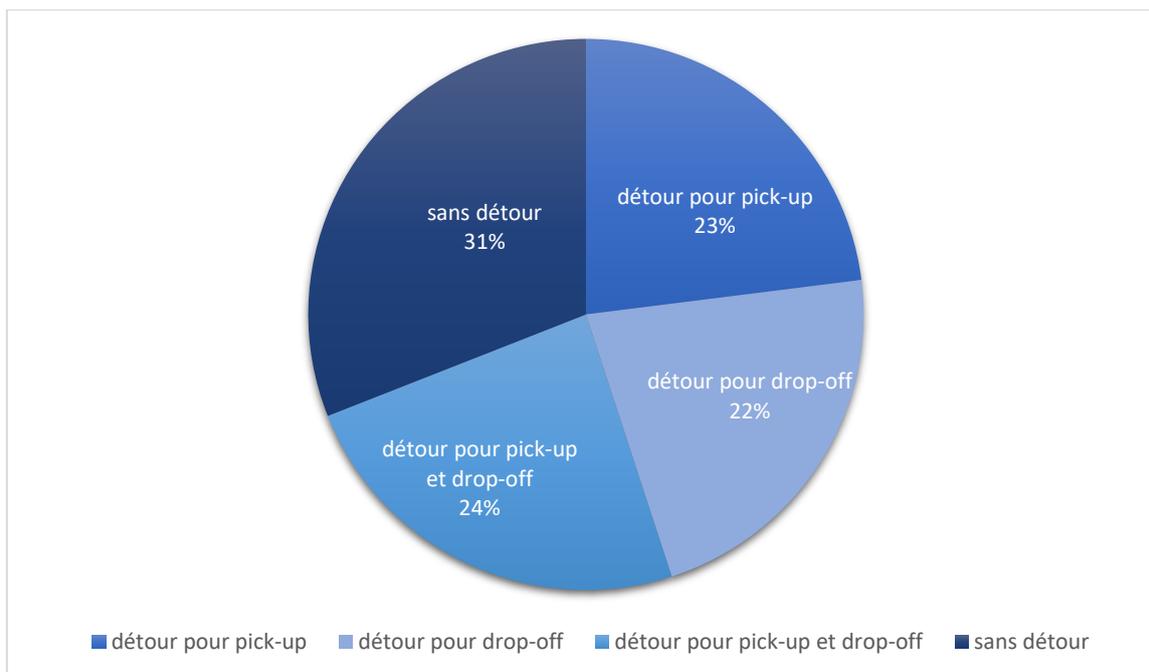


Figure 3.7 : Types des demandes acceptées.

Le tableau 3.4 présente un échantillon des offres des conducteurs, le tableau 3.5 présente un échantillon des demandes des passagers et le tableau 3.6 présente le résultat d'appariement entre les échantillons des offres du tableau 3.4 et les échantillons des demandes du tableau 3.5.

<i>Offres</i>	<i>Temps de détour maximal</i>	<i>La distance maximale pour un pick-up</i>
Conducteur 8	24 min	1498 mètres
Conducteur 7	33 min	1203 mètres
Conducteur 399	20 min	1013 mètres
Conducteur 431	22 min	1270 mètres
Conducteur 854	34 min	808 mètres
Conducteur 167	25 min	1125 mètres

Tableau 3.4 : Echantillon des conducteurs.

<i>Demandes</i>	<i>Temps d'attente</i>	<i>Détour pour drop-off</i>	<i>Détour pour pick-up</i>	<i>Détour pour pick-up et drop-off</i>	<i>Sans détour</i>
Passager 25	25 min	X			
Passager 17	16 min		X		
Passager 20	20 min				X
Passager 24	17 min				X
Passager 890	20 min		X		
Passager 954	15 min			X	
Passager 946	24 min				X
Passager 1866	20min			X	
Passager 1871	15 min		X		
Passager 362	28 min	X			
Passager 410	25 min				X

Tableau 3.5 : Echantillon des passagers.

<i>Conducteur</i>	<i>Passager</i>	<i>Heure d'appariment</i>
Conducteur 8	Passager 25 et passager 17	À 09:12 :00 et à 09:12:39
Conducteur 7	Passager 20 et passager 24	À 09:14:00 et à 09:14:00
Conducteur 399	Passager 890	À 14:41:42
Conducteur 431	Passager 954 et Passager 946	À 15:06:00 et à 15:20:36
Conducteur 854	Passager 1866 et Passager 1871	À 21:03:00 et à 21:03:00
Conducteur 167	Passager 362 et Passager 410	À 11:23:00 et à 11:41:17

Tableau 3.6 : Résultats d'appariement dynamique des échantillons des tableaux 3.4 et 3.5.

<i>Execution du conducteur</i>	<i>Execution du conducteur</i>
Conducteur 8	2024-06-12 09:12:39 la prochaine destination ...
2024-06-12 09:10:00 En attente ...	2024-06-12 09:12:39 j'ai pris en charge Passager 17
2024-06-12 09:11:00 En attente ...	2024-06-12 09:14:06 la prochaine destination ...
2024-06-12 09:11:00 je vais prise en charge Passager 25 , prochaine destination	2024-06-12 09:17:16 la prochaine destination ...
2024-06-12 09:11:00 je vais prendre le détour pour le Passager 25 , prochaine destination du détour	2024-06-12 09:17:19 la prochaine destination ...
2024-06-12 09:12:00 Démarrer	2024-06-12 09:23:35 la prochaine destination ...
2024-06-12 09:12:00 Démarrer avec nombre de siège : 3	2024-06-12 09:26:54 la prochaine destination ...
2024-06-12 09:12:00 la prochaine destination ...	2024-06-12 09:27:22 la prochaine destination ...
2024-06-12 09:12:00 j'ai pris en charge Passager 25	2024-06-12 09:27:43 la prochaine destination ...
2024-06-12 09:12:39 je vais prise en charge Passager 17 , prochaine destination	2024-06-12 09:27:43 Passager 25 arrivé à sa destination
2024-06-12 09:12:39 aller au détour	2024-06-12 09:28:45 la prochaine destination ...
2024-06-12 09:12:39 la prochaine destination ...	2024-06-12 09:30:09 la prochaine destination ...
2024-06-12 09:12:39 j'ai pris en charge Passager 17	2024-06-12 09:31:10 retourner à l'itinéraire initial
2024-06-12 09:14:06 la prochaine destination ...	2024-06-12 09:31:10 la prochaine destination ...
2024-06-12 09:17:16 la prochaine destination ...	2024-06-12 09:31:10 Passager 17 arrivé à sa destination
	2024-06-12 09:33:31 Destination final

Figure 3.8 : Exécution du conducteur 8.

La Figure 3.8 présente l'exécution du conducteur 8 présentée dans le tableau 3.7. Le conducteur 8 a déposé son offre à 9h10 et est resté en mode d'attente, attendant son heure de départ programmée. Une minute plus tard, il a reçu une demande de prise en charge pour le passager 25, nécessitant un détour pour le déposer à sa destination. Il a confirmé la prise en charge et le détour prévu. À 9h12, il a démarré avec trois sièges disponibles et a confirmé avoir pris en charge le passager 25. Peu après, à 9:12:39, à la prochaine destination, il a également accepté de prendre en charge le passager 17. Le conducteur a ensuite suivi l'itinéraire prévu et effectué le détour, marquant chaque prochaine destination jusqu'à ce que le passager 25 soit déposé à sa destination à 9:27:43. Après avoir continué son trajet, le conducteur est retourné à l'itinéraire initial à 9:31:10, moment auquel le passager 17 a également atteint sa destination. Finalement, le conducteur 8 a atteint sa destination finale à 9:33:31.

3.6. Expérimentations

Afin d'évaluer de manière approfondie et efficace le modèle de covoiturage que nous avons développé, nous avons procédé à une série d'expérimentations en utilisant différentes configurations. Nous avons élaboré plusieurs scénarios, tous couvrant le même intervalle de temps, de 9h à 13h, et les emplacements des offres et des demandes ont été choisis aléatoirement à partir de notre ensemble de données. Et nous avons choisi des intervalles de réception des nouvelles demandes et des nouvelles offres de 5 minutes ($\Delta t = 5 \text{ min}$).

Dans nos expérimentations, nous nous sommes basées sur trois paramètres principaux : nombre de demandes, temps de détour, et temps d'attente, pour modéliser divers scénarios de covoiturage dynamique. Pour chaque paramètre, nous avons élaboré plusieurs scénarios afin de tester la robustesse de notre modèle face à des situations réalistes.

- **Premier scénario :** Par rapport au nombre de demandes, nous avons testé deux cas distincts. Dans le premier cas, 70 % des demandes reçues ont leur point de départ situé sur le détour du conducteur. Dans le second cas, seulement 30 % des demandes ont leur point de départ sur le détour. Ces deux situations nous permettent d'évaluer l'efficacité de notre modèle dans des contextes où les besoins en détour varient.

Le tableau 3.7 présente les résultats obtenus des deux cas.

Les horaires	9h00-10h00	10h00-11h00	11h00-12h00	12h00-13h00
Premier cas				
Nombre des offres	66	139	214	220
Nombre des demandes	148	305	457	468
Nombre des demandes acceptées sans détour	32	67	98	106
Nombre des demandes acceptées avec détour	61	128	196	216
Nombre des pick-up dans le détour	21	58	100	113
Gain en temps du détour	1381 min	2738 min	4294 min	4373 min
Gain en temps d'attente	109 min	252 min	365 min	387 min
Gain en temps dans le système	957 min	4693 min	9732 min	13755 min
Second cas				
Nombre des offres	64	139	206	214
Nombre des demandes	145	308	453	467
Nombre des demandes acceptées sans détour	73	155	245	276
Nombre des demandes acceptées avec détour	26	53	75	81
Nombre des pick-up dans le détour	8	24	40	44
Gain en temps du détour	1418 min	3084 min	4655 min	4850 min

Gain en temps d'attente	154 min	300 min	511 min	586 min
Gain en temps dans le système	2383 min	5271 min	9327 min	11568 min

Tableau 3.7 : Résultats du premier scénario.

Dans le premier cas, 68% des demandes ont été acceptées, et 67% de ces demandes acceptées nécessitent un détour. Parmi les demandes reçues nécessitant un détour, 64% ont été acceptées. De plus, 78 % des demandes reçues ne nécessitant pas de détour ont été acceptées. La consommation du temps de détour autorisée pour les conducteurs est de 39 %, et le temps d'attente consommé par les passagers est de 14 %. Dans le second cas, 76 % des demandes ont été acceptées, et 22 % de ces demandes acceptées nécessitent un détour. Parmi les demandes reçues nécessitant un détour, 62 % ont été acceptées. De plus, 81 % des demandes reçues ne nécessitant pas de détour ont été acceptées. La consommation du temps de détour autorisé pour les conducteurs est de 28 %, et le temps d'attente consommé par les passagers est de 51 %.

Alors, dans le premier cas où 70% des demandes reçues nécessitent un détour, nous constatons que 64% de ces demandes nécessitant un détour ont été acceptées, et le taux de consommation du temps de détour autorisé pour les conducteurs est de 39%. Même lorsque les demandes reçues nécessitent un détour sont à 30% des demandes reçues, le taux reste faible.

- **Deuxième scénario :** Par rapport au temps de détour maximal, nous avons testé deux cas distincts. Dans le premier cas, nous avons fixé le temps de détour maximal de manière aléatoire entre 5 et 10 minutes. Dans le second cas, nous avons fixé le temps de détour maximal de manière aléatoire entre 20 et 35 minutes.

Le tableau 3.8 présente les résultats obtenus des deux cas.

Les horaires	9h00-10h00	10h00-11h00	11h00-12h00	12h00-13h00
Premier cas				
Nombre des offres	59	126	193	198
Nombre des demandes	148	303	459	473
Nombre des demandes acceptées sans détour	31	59	89	98
Nombre des demandes acceptées avec détour	32	63	93	100
Gain en temps du détour	365 min	796 min	1220 min	1232 min
Gain en temps d'attente	90 min	154 min	269 min	297 min
Gain en temps dans le système	2188 min	5509 min	9845 min	12012 min
Second cas				
Nombre des offres	62	127	202	271
Nombre des demandes	137	290	444	595
Nombre des demandes acceptées sans détour	24	56	87	116
Nombre des demandes acceptées avec détour	65	126	178	270
Gain en temps du détour	1300 min	2836 min	4542min	6098 min
Gain en temps d'attente	110 min	174 min	245 min	311min
Gain en temps dans le système	2936 min	5708 min	10219 min	13665 min

Tableau 3.8 : Résultats du deuxième scénario.

Dans le premier cas, nous avons un taux d'acceptation des demandes de 41%. Parmi ces demandes acceptées, 50% nécessitent un détour, avec un taux de consommation de temps de détour autorisé de 41%. Dans le deuxième cas, le taux d'acceptation des demandes est de 64%. Parmi ces demandes acceptées, 66% nécessitent un détour, avec un taux de consommation de temps de détour autorisé de 38%.

D'après les résultats, il est observé que lorsque le temps de détour maximal est fixé faible ou grand, le pourcentage de consommation du temps de détour autorisé reste toujours faible. Cela démontre que l'objectif de réduire le temps de détour est toujours atteint.

- **Troisième scénario** : Par rapport à la distance maximale pour un pick-up dans le détour, nous avons testé deux cas distincts. Dans le premier cas, nous avons fixé la distance

maximale pour un pick-up de manière aléatoire entre 50 et 200 mètres. Dans le second cas, nous l'avons fixée de manière aléatoire également, entre 400 et 2500 mètres.

Le tableau 3.9 présente les résultats obtenus des deux cas.

Les horaires	9h00-10h00	10h00-11h00	11h00-12h00	12h00-13h00
Premier cas				
Nombre des offres	72	145	216	220
Nombre des demandes	142	294	462	477
Nombre des demandes acceptées sans détour	28	53	83	95
Nombre des demandes acceptées avec détour	34	62	99	109
Nombre des pick-up dans le détour	9	14	25	29
Gain en temps du détour	1461 min	3190 min	4688 min	4729 min
Gain en temps d'attente	81 min	177 min	251 min	294 min
Second cas				
Nombre des offres	72	136	207	214
Nombre des demandes	138	291	455	466
Nombre des demandes acceptées sans détour	31	60	81	86
Nombre des demandes acceptées avec détour	54	128	199	217
Nombre des pick-up dans le détour	34	76	106	113
Gain en temps du détour	1440 min	2641 min	3987 min	4065 min
Gain en temps d'attente	142 min	284 min	375 min	397 min

Tableau 3.9 : Résultats du troisième scénario.

Dans le premier cas, 42% des demandes ont été acceptées, et 53% de ces demandes acceptées nécessitent un détour. De plus, parmi les demandes acceptées dont le point de départ se trouve dans le détour, 76% concernent un détour pour prendre le passager (faire un pick-up dans le détour). La consommation du temps de détour autorisé pour les conducteurs est de 41%, et le temps d'attente consommé par les passagers est de 17%. Dans le second cas, 65% des demandes ont été acceptées, et 71% de ces demandes acceptées nécessitent un détour. De plus, parmi les demandes acceptées dont le point de départ se trouve dans le détour, 86% concernent

un détour pour prendre le passager (faire un pick-up dans le détour). La consommation du temps de détour autorisé pour les conducteurs est de 39%, et le temps d'attente consommé par les passagers est de 15%.

D'après les résultats, nous constatons que, malgré un taux d'acceptation de 71% et 53% dans les deux cas, le modèle réussit à minimiser le taux de consommation du temps de détour autorisé et le temps d'attente consommé par les passagers.

Suite à nos expérimentations, nous pouvons affirmer que notre modèle atteint ses objectifs de manière efficace dans une variété de scénarios testés. Les résultats démontrent que le modèle parvient à équilibrer les différents objectifs conflictuels : la minimisation du temps d'attente des passagers, la minimisation du temps de détours pour les conducteurs et la maximisation de l'utilisations des véhicules, ce qui confirme son adaptabilité et sa robustesse face aux différentes situations rencontrées dans le système de covoiturage dynamique.

3.7. Conclusion

Dans ce chapitre, nous avons détaillé l'implémentation de notre approche et de notre simulateur, en présentant non seulement les résultats de la simulation, mais aussi une série d'expérimentations conçues pour évaluer les performances de notre système. Ces expérimentations ont permis de tester notre modèle dans divers scénarios, en tenant compte de différents paramètres clés. Les résultats obtenus sont prometteurs et viennent appuyer nos choix ainsi que notre modélisation. Ils démontrent la robustesse et l'efficacité de notre modèle de covoiturage dynamique pour tous les utilisateurs.

Conclusion générale

Dans un monde où la flexibilité et l'efficacité sont essentielles, le covoiturage dynamique se distingue par sa capacité à connecter en temps réel des conducteurs et des passagers aux itinéraires compatibles. Contrairement au covoiturage statique, cette approche utilise des plateformes sophistiquées pour identifier et associer automatiquement les trajets compatibles, en fonction des positions actuelles et des fenêtres temporelles. La complexité de ces systèmes repose sur l'appariement d'individus sous contraintes spatio-temporelles, qui doivent être clairement définies par les conducteurs et les passagers pour assurer la réussite du trajet.

Dans ce travail, nous nous sommes intéressées au problème d'appariement dynamique d'un conducteur à plusieurs passagers sous contraintes spatio-temporelles et de capacité. Nous avons proposé une modélisation des composants de ce problème par un processus décisionnel de Markov multi-objectifs (MOMDP), qui constitue la base théorique de l'apprentissage par renforcement multi-objectifs (MORL). Notre modélisation se concentre sur l'optimisation d'objectifs conflictuels, à savoir : minimiser le temps d'attente des passagers, minimiser le temps de détour des conducteurs, tout en maximisant l'utilisation des véhicules. En atteignant ces objectifs, nous sommes en mesure de satisfaire à la fois les passagers et les conducteurs.

La validation de notre modèle proposé a été réalisée à l'aide d'un simulateur que nous avons développé pour capturer la dynamique du système. Les résultats obtenus à partir des diverses expérimentations ont démontré l'efficacité et la robustesse de notre système, ainsi que sa capacité à répondre à différentes demandes soumises à diverses contraintes en temps réel. Notre modèle est capable d'atteindre ses objectifs, en satisfaisant les passagers par la réduction de leurs temps d'attente, et les conducteurs par la minimisation de leurs temps de détour et la maximisation de l'utilisation des véhicules. Il offre ainsi une solution qui satisfait tous les individus impliqués dans le covoiturage.

Pour les perspectives futures, plusieurs axes d'amélioration peuvent être envisagés pour enrichir notre modèle de covoiturage dynamique. Premièrement, l'intégration de poids pour les différents objectifs permettrait de prioriser certains aspects par rapport à d'autres, selon les besoins spécifiques des utilisateurs ou des conditions opérationnelles. Deuxièmement, la

gestion des préférences personnelles ou sociales des conducteurs et des passagers pourrait être ajoutée pour offrir une expérience plus personnalisée et agréable. Enfin, la prise en compte des aspects financiers, tels que le coût et la tarification, pourrait non seulement optimiser l'efficacité économique du système mais également inciter davantage de participants à utiliser le service.

Bibliographie

- Abels, A., Roijers, D., Lenaerts, T., Nowé, A., & Steckelmacher, D. (2019). Dynamic weights in multi-objective deep reinforcement learning. *In International conference on machine learning*, 11-20.
- Agatz, N., Erera, A., Savelsbergh, M., & Wang, X. (2010). Sustainable passenger transportation: Dynamic ride-sharing.
- Agatz, N., Erera, A., Savelsbergh, M., & Wang, X. (2012). Optimization for dynamic ride-sharing : A review. *European Journal of Operational Research*, 223(2), 295-303. doi:10.1016/j.ejor.2012.05.028
- Al-Abbasi, A. O., Ghosh, A., & Aggarwal, V. (2019). Deeppool: Distributed model-free algorithm for ride-sharing using deep reinforcement learning. *In IEEE Transactions on Intelligent Transportation Systems*, 20(12), pp. 4714-4727.
- Alisoltani, N.; Leclercq, L.; Zargayouna, M. (2019). Real-time ride-sharing systems performance considering network congestion. *In hEART 2019, 8th Symposium of the European Association for Research in Transportation*, (p. 6).
- Asl, Z. D., Derhami, V., & Yazdian-Dehkordi, M. (2017). A new approach on multi-agent Multi-Objective Reinforcement Learning based on agents' preferences. *In 2017 Artificial Intelligence and Signal Processing Conference (AISP)* (pp. 75-79). IEEE. doi:10.1109/AISP.2017.8324111
- Atahran, A., Lenté, C., & T'kindt, V. (2014). A Multicriteria Dial-a-Ride Problem with an Ecological Measure and Heterogeneous Vehicles. *Journal of Multi-Criteria Decision Analysis*, 21(5-6), 279-298.
- Ballet, J.-C., & Clave, R. (2007). Le covoiturage en France et en Europe: état des lieux et perspectives. (Doctoral dissertation, Centre d'études sur les réseaux, les transports, l'urbanisme et les constructions publiques (CERTU)).

- Cheikh, S. B., & Hammadi, S. (2016). Multi-Hop Ridematching optimization problem: Intelligent chromosome agent-driven approach. *Expert Systems with Applications*, 62, 161-176. doi:10.1016/j.eswa.2016.06.001
- Cheikh-Graiet, S. B., Dotoli, M., & Hammadi, S. (2020). A Tabu Search based metaheuristic for dynamic carpooling optimization. *Computers & Industrial Engineering*, 140, 106217.
- Cici, B., Markopoulou, A., & Laoutaris, N. (2015). Designing an on-line ride-sharing system. *Proceedings of the 23rd SIGSPATIAL international conference on advances in geographic information systems*, (pp. 1-4). doi:10.1145/2820783.2820850
- Edenhofer, O., Pichs-Madruga, R., Sokona, Y., Farahani, E., Kadner, S., & Seyboth, K. e. (2014). *Climate change 2014: Mitigation of climate change*. UK and New York: Working group III contribution to the fifth assessment report of the Intergovernmental Panel on Climate Change.
- Fèvre, C., Zgaya-Biau, H., Mathieu, P., & Hammadi, S. (2023). L'optimisation du covoiturage dynamique multi-saut. *Trente-et-unièmes journées francophones sur les systèmes multi-agents (JFSMA)* (pp. 139-148). Cépaduès.
- Hayes, C. F., Rădulescu, R., Bargiacchi, E., Källström, J., Macfarlane, M., Reymond, M., . . . al., a. (2022). A practical guide to multi-objective reinforcement learning and planning. *Autonomous Agents and Multi-Agent Systems*, 36(1), 26.
- Herbawi, W. (2013). Solving the ridematching problem in dynamic ridesharing (Doctoral dissertation, Universität Ulm).
- Jia, J., & Wang, W. (2020). Review of reinforcement learning research. *2020 35th Youth Academic Annual Conference of Chinese Association of Automation (YAC)* (pp. 186-191). IEEE. doi:10.1109/yac51587.2020.9337653
- Jin, J., Zhou, M., Zhang, W., Li, M., Guo, Z., Qin, Z., . . . Ye, J. (2019). Coride: joint order dispatching and fleet management for multi-scale ride-hailing platforms. *In Proceedings*

of the 28th ACM International Conference on Information and Knowledge Management, (pp. 1983-1992).

- Jindal, I., Qin, Z. T., Chen, X., Nokleby, M., & Ye, J. (2018). Optimizing taxi carpool policies via reinforcement learning and spatio-temporal mining. *2018 IEEE International Conference on Big Data (Big Data)*, (pp. 1417-1426).
- Joshi, D., Kale, I., Gandewar, S., Korate, O., Patwari, D., & Patil, S. (2021). Reinforcement Learning: A Survey. *Machine Learning and Information Processing: Proceedings of ICMLIP 2020*, 297-308. doi:10.1007/978-981-33-4859-2_29
- Ke, J., Xiao, F., Yang, H., & Ye, J. (2020). Learning to delay in ride-sourcing systems: A multi-agent deep reinforcement learning framework. *IEEE Transactions on Knowledge and Data Engineering*, 34(5), 2280-2292.
- Kullman, N. D., Cousineau, M., Goodson, J. C., & Mendoza, J. E. (2022). Dynamic ride-hailing with electric vehicles. *Transportation Science*, 56(3), 775-794.
- Martins, L. D., de la Torre, R., Corlu, C. G., Juan, A. A., & Masmoudi, M. A. (2021). Optimizing ride-sharing operations in smart sustainable cities: Challenges and the need for agile algorithms. *153*(107080).
- Masmoudi, M. A., Hosny, M., Demir, E., & Pesch, E. (2020). Hybrid adaptive large neighborhood search algorithm for the mixed fleet heterogeneous dial-a-ride problem. *Journal of Heuristics*, 26, 83-118.
- Masoud, N., & Jayakrishnan, R. (2017). 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.
- Mitropoulos, L., Kortsari, A., & Ayfantopoulou, G. (2021). A systematic literature review of ride-sharing platforms, user factors and barriers. *European Transport Research Review*, 13(1-22). doi:10.1186/s12544-021-00522-1

- Mourad, A., Puchinger, J., & Chengbin, C. (2019). A survey of models and algorithms for optimizing shared mobility. *Transportation Research Part B: Methodological*, 123, 323-346. doi:10.1016/j.trb.2019.02.003
- Plate, O. (2019). Ridesharing with Multiple Riders (Doctoral dissertation, Karlsruhe Institute of Technology).
- Qin, Z. T., Zhu, H., & Ye, J. (2022). Reinforcement learning for ridesharing: An extended survey. *Transportation Research Part C: Emerging Technologies*, 144, 103852.
- Reymond, M., Bargiacchi, E., & Nowé, A. (2022). Pareto conditioned networks.
- Santos, D. O., & Xavier, E. C. (2015). Taxi and ride sharing: A dynamic dial-a-ride problem with money as an incentive. *Expert Systems with Applications*, 42(19), 6728-6737.
- Sghaier, M. (2011). Combinaison des techniques d'optimisation et de l'intelligence artificielle distribuée pour la mise en place d'un système de covoiturage dynamique. (Doctoral dissertation, Ecole Centrale de Lille, France).
- Silwal, S., Gani, M., & Raychoudhury, V. (2019). A Survey of Taxi Ride Sharing System Architectures. *2019 IEEE International Conference on Smart Computing (SMARTCOMP)* (pp. 144-149). IEEE. doi:10.1109/smartcomp.2019.00044
- Singh, A., Al-Abbasi, A. O., & Aggarwal, V. (2021). A distributed model-free algorithm for multi-hop ride-sharing using deep reinforcement learning. *IEEE Transactions on Intelligent Transportation Systems*, 23(7), 8595-8605.
- Sutton, R. S., & Barto, A. G. (1999). Reinforcement learning: An introduction. *Robotica*, 17(2), 229-235.
- Swain, D., Pattnaik, P. K., & Athawale. (2021). Machine Learning and Information Processing. *Advances in Intelligent Systems and Computing*. doi:10.1007/978-981-33-4859-2

- Ting, K. H., Lee, L. S., Pickl, S., & Seow, H. V. (2021). Shared mobility problems: a systematic review on types, variants, characteristics, and solution approaches. *Applied Sciences*, *11*(17), 7996.
- Vamvoudakis, K. G., Wan, Y., Lewis, F. L., & Cansever, D. (2021). Handbook of Reinforcement Learning and Control. *Studies in Systems, Decision and Control*. doi:10.1007/978-3-030-60990-0
- Van Moffaert, K., & Nowé, A. (2014). Multi-objective reinforcement learning using sets of pareto dominating policies. *The Journal of Machine Learning Research*, *15*(1), 3483-3512.
- Wang, D., Wang, Q., Yin, Y., & Cheng, T. C. (2023). Optimization of ride-sharing with passenger transfer via deep reinforcement learning. *Transportation Research Part E: Logistics and Transportation Review*, *172*(103080).
- Wang, Z., Qin, Z., Tang, X., Ye, J., & Zhu, H. (2018). Deep reinforcement learning with knowledge transfer for online rides order dispatching. *2018 IEEE International Conference on Data Mining (ICDM)* (pp. 617-626). IEEE.
- Wiering, M. A., & Van Otterlo, M. (2012). Reinforcement learning. *Adaptation, learning, and optimization*, *12*(3), 729.
- Wu, R., Zhang, Y., Yang, Z., & Wang, Z. (2021). Offline constrained multi-objective reinforcement learning via pessimistic dual value iteration. *Advances in Neural Information Processing Systems*, *34*, 25439-25451.
- Xu, Z., Li, Z., Guan, Q., Zhang, D., Li, Q., Nan, J., . . . Ye, J. (2018). Large-scale order dispatch in on-demand ride-hailing platforms: A learning and planning approach. *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, (pp. 905-913).
- Zhou, M., Jin, J., Zhang, W., Qin, Z., Jiao, Y., Wang, C., . . . Ye, J. (2019). Multi-agent reinforcement learning for order-dispatching via order-vehicle distribution matching. *In*

Proceedings of the 28th ACM International Conference on Information and Knowledge Management, 2645-2653.

Webographie

- [1] *CO2 Emissions in 2022 – Analysis – IEA* [En ligne]. Available : <https://www.iea.org/reports/co2-emissions-in-2022/>. [Accès le 19-05-2024]

- [2] Kaggle [En ligne]. Available : <https://www.kaggle.com/datasets/fivethirtyeight/uber-pickups-in-new-york-city/>. [Accès le 07-06-2024].

- [3] Direction API mapQuest [En ligne]. Available : <https://developer.mapquest.com/documentation/directions-api/route/get/>. [Accès le 07-06-2024].

- [4] Python [En ligne]. Available : <https://www.python.org/doc/essays/blurb/>. [Accès le 07-06-2024].

- [5] visual studio [En ligne]. Available : <https://code.visualstudio.com/docs/>. [Accès le 07-06-2024].

- [6] Numpy [En ligne]. Available : <https://numpy.org/doc/stable/>. [Accès le 07-06-2024].

- [7] Pandas [En ligne]. Available : <https://pandas.pydata.org/>. [Accès le 07-06-2024].

- [8] Gymnasium [En ligne]. Available : <https://github.com/Farama-Foundation/Gymnasium/>. [Accès le 07-06-2024].

- [9] morl-baselines [En ligne]. Available : <https://lucasalegre.github.io/morl-baselines/>. [Accès le 07-06-2024].

- [10] Tkinter [En ligne]. Available : <https://docs.python.org/3/library/tkinter.html/>. [Accès le 07-06-2024].