

N° d'ordre :

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE



UNIVERSITÉ DJILLALI LIABÈS DE SIDI BEL ABBÈS
FACULTÉ DES SCIENCES EXACTES
DÉPARTEMENT D'INFORMATIQUE

THÈSE DE DOCTORAT EN SCIENCES

Domaine : Informatique Filière : Informatique

Spécialité : Informatique

Par

Mr. MAASKRI Moustafa

LES TECHNIQUES BIO-INSPIRÉES ET LA FOUILLE DE DONNÉES POUR LA LOGISTIQUE DES TRANSPORTS

Soutenue le 12/10/2022 devant le jury :

Pr.	BOUKLI HACENE SOFIANE	UDL SBA	Président du jury
Pr.	HAMOU REDA MOHAMED	UTM SAIDA	Directeur de thèse
Dr.	TOMOUH ADIL	UDL SBA	Co-directeur de thèse
Dr.	BENDAOUZ ZAKARIA	UTM SAIDA	Examineur
Dr.	LOKBANI AHMED CHAOUKI	UTM SAIDA	Examineur
Dr.	KHOBZAOUI ABDELKADER	UDL SBA	Examineur

Année Universitaire : 2021/2022

اللهم يَسِّرْ لِي أَمْرِي ...

Dédicace

A mes chers parents

A mon épouse et mes enfants (Youcef, Abdallah et Amina)

A tous mes amis

A toute ma grande famille

Remerciement

Tout d'abord, louange au bon Dieu qui m'a aidé à accomplir ce modeste travail.

J'adresse mes remerciements les plus sincères aux personnes qui m'ont apporté leur aide et qui ont contribué à l'élaboration de cette thèse .

Je tiens à remercier sincèrement Le professeur HAMOU REDA Mohamed, en tant que directeur de thèse pour son encadrement et ses conseils

Toute ma gratitude à Monsieur TOMOUH Adil en tant que co-directeur de thèse.
Je remercie également Messieurs le président et les membres du jury pour avoir bien voulu assister au débat de ma thèse

Ne pouvant pas citer tout le monde et de peur d'en oublier, je souhaite remercier tous ceux qui ont contribué de près ou de loin à l'élaboration de ce travail.

Qu'Allah vous bénisse tous !

Moustafa MAASKRI

Table des matières

Remerciements	iv
Table des Matières	vi
Liste des Figures	viii
Liste des Tableaux	ix
Introduction et Problématique	1
1 État de l’art et travaux connexes	10
1.1 Introduction	11
1.2 Covoiturage avec un problème d’optimisation	11
1.3 Covoiturage avec les graphes	13
1.4 Covoiturage avec un système d’agent	16
1.5 Résumé	20
2 L’optimisation, les méta-heuristiques et les techniques bio-inspirées	22
2.1 Introduction	23
2.2 Problème d’optimisation	23
2.3 Méthodes d’optimisation	24
2.3.1 Les méthodes de recherche locale	24
2.3.2 Méthode de recherche globale	25
2.4 Approches déterministes	26
2.5 Les approches évolutionnaires	27

2.6	Généralités sur les méta-heuristiques	28
2.6.1	Définition	28
2.6.2	Types de méta-heuristiques	30
2.6.3	Classification méta-heuristique	30
2.7	Les Méthodes Bio-Inspirés	32
2.8	Le biomimétisme	33
2.9	Algorithmes évolutionnaires (AEs)	35
2.10	Algorithmes d'Intelligence en Essaims (AIEs)	37
2.11	Conclusion	47
3	L'approche proposée et Conceptions	48
3.1	Introduction	49
3.2	Exigences	49
3.3	Conception du système	50
3.3.1	Conception des véhicules	51
3.3.2	Les demandes ou les requêtes	53
3.4	Définition des actions	54
3.5	Routage	57
3.6	Clustering	58
3.6.1	Introduction	58
3.6.2	K-Means Clustering	60
3.6.3	PSO Clustering	61
3.7	Relocalisation	63
4	Expérimentation et Résultats	68
4.1	Étude de cas (ville de New York)	69
4.1.1	Introduction	69
4.1.2	Data sets utilisées	69
4.1.3	Exploration des données	70

4.1.4	Statut et perception du véhicule	74
4.1.5	Rééquilibrage dynamique sur la base des demandes (DRBD) . .	75
4.2	Évaluation	75
4.2.1	Introduction	75
4.2.2	Mesures d'évaluation	76
4.2.3	Scénarios	77
4.2.4	Configuration expérimentale	78
4.2.5	Évaluation et analyse	79
4.2.6	Conclusion	84
	Conclusion et perspectives	87

Table des figures

1.1	Déplacement des voitures pour améliorer le déséquilibre. À gauche, les déséquilibres initiaux prévus, au centre, la relocalisation des voitures en blocs décalés et enfin la distribution résultante [FKB15].	15
1.2	Architecture proposée par [AGA19].	18
1.3	Système SAMoD [GD18].	20
2.1	Taxonomie des algorithmes méta-heuristiques bio-inspirés	33
2.2	Organigramme de l’algorithme génétique [Gol89]	37
2.3	Architecture générale de l’AOCF [DG97]	40
2.4	Déplacement d’une particule	42
3.1	Schéma d’architecture de notre proposition.	50
3.2	Requêtes actives à l’instant t_v , seules (r3; r4) sont prises en compte lors du rééquilibrage.	54
3.3	Critères de regroupement différent pour le même jeu de données	59
3.4	Flotte déséquilibrée.	64
4.1	Bande horaire d’exploration des données de 7 à 10 heures du matin.	72

4.2	Bande horaire d'exploration des données de 18 à 21 heures.	73
4.3	Représentation de l'emplacement et de la destination du véhicule avec sa perception.	74
4.4	Clusters obtenus via DRBD sur les dernières requêtes pas encore servies.	76
4.5	Comparaison des temps d'attente	80
4.6	Comparaison des nombres des clients servis	82
4.7	Comparaison des distances parcourues	83
4.8	Deux véhicules qui demandent à être relocalisés avec deux zones trouvées par la DRBD.	91

Liste des tableaux

1.1	Résumé des recherches examinées et de notre proposition.	20
3.1	La structure de la base de données	53
4.1	État interne du véhicule avec sa perception. Version numérique de la représentation graphique de la figure 4.3	74
4.2	Tableau récapitulatif des résultats obtenus	85

Introduction et Problématique

Introduction et Problématique

Cette thèse aborde le problème des véhicules partagés, où plusieurs clients partagent le même véhicule pour leur trajet. Le covoiturage a été étudié pour éviter la congestion du trafic et minimiser les coûts. Le covoiturage réduit le nombre de véhicules impliqués, optimisant l'utilisation des véhicules en réduisant le nombre de sièges vides. En outre, grâce au covoiturage, le conducteur peut offrir des coûts réduits aux clients sans affecter ses propres revenus.

Dans ce travail un algorithme a été proposé qui permet à chaque véhicule d'apprendre la politique de prise en charge et de dépose. Ce chapitre traite du domaine retenu et de la motivation de ce travail, en expliquant les objectifs globaux. A la fin de ce chapitre est présentée une feuille de route pour le reste de cette thèse.

Mobilité intelligente

La mobilité intelligente est une branche des villes intelligentes dans laquelle le sujet se concentre sur le transport. La mobilité intelligente signifie une connexion entre différents véhicules qui agissent dans une zone urbaine. Ces véhicules ainsi que l'environnement externe collectent des données grâce à des capteurs, puis ces données sont traitées pour améliorer la qualité des services offerts. La mobilité intelligente couvre plusieurs sujets, depuis les véhicules connectés qui partagent des informations jusqu'à l'allocation efficace des véhicules et des passagers pour un service de transport comme le service de taxi classique. Cette recherche analyse le problème du covoiturage à travers une flotte composée de véhicules dans une zone urbaine. Le but du covoiturage est d'offrir un service de transport en optimisant l'utilisation des ressources.

L'intérêt pour le covoiturage s'est nettement accru ces dernières années, et la création d'entreprises comme uber ou lyft, pour n'en citer que quelques-unes, est un signe tangible de cet intérêt. Ces entreprises investissent d'énormes ressources dans le but de mener la recherche sur les technologies futures dans ce domaine. Les raisons pour lesquelles le covoiturage est si étudié sont multiples et proviennent de

contextes différents. Elles vont de la disponibilité limitée des ressources, comme le pétrole et le gaz, à la congestion du trafic, aux menaces environnementales et aux perspectives économiques. L'objectif du covoiturage pourrait se résumer en deux points essentiels : réduire le nombre de véhicules actifs dans une zone et augmenter le nombre de clients à servir. L'objectif du covoiturage est de trouver un compromis permettant d'atteindre ces deux objectifs. Un système de covoiturage efficace a besoin d'une technologie d'optimisation efficace et fiable pour répartir la demande sur la flotte et éviter que tous les véhicules se rassemblent dans quelques zones.

Le covoiturage doit encore résoudre certains problèmes, notamment celui de la prévision efficace de la demande. La prédiction des demandes futures est un point crucial, la raison principale de son importance étant les bénéfices qui peuvent en découler. S'il était possible de savoir à l'avance où se situent les futures demandes, avec un certain degré de fiabilité, il serait plus facile de concevoir une stratégie qui y réponde.

L'objectif de cette recherche est d'aborder le problème de la relocalisation des véhicules et la prédiction de la demande future avec une particularité. Nous ne nous concentrerons pas sur la prédiction de toutes les demandes de transport futures, mais uniquement sur celles qui ne pourraient pas être satisfaites autrement.

Motivation de la recherche

La recherche dans le domaine des transports est un élément clé de la vie quotidienne de chacun d'entre nous. Chaque jour, nous devons faire face aux embouteillages, certains d'entre nous vivent dans des métropoles où la pollution de l'air ne cesse d'augmenter et où la situation empire d'année en année.

Réduire la pollution tout en augmentant la circulation, en maintenant le niveau du système de transport, est un objectif important pour les sociétés durables et un grand défi. Les scientifiques font des recherches sur les nouvelles technologies qui permettent de surmonter les problèmes présentés ci-dessus. Le sujet auquel ils s'intéressent particulièrement est le covoiturage. En effet, le travail présenté dans cette recherche concerne une flotte de véhicules pour un système de mobilité à la

demande, qui fait du covoiturage une réponse aux demandes de transport.

L'objectif du covoiturage est de permettre au service de transport de transporter plus de passagers sans affecter les performances attendues et en économisant de l'argent pour les clients. Actuellement, une flotte de véhicules, comme les taxis, peut répondre à un nombre limité de demandes. S'ils veulent atteindre plus de clients, ils doivent augmenter leurs ressources en développant la flotte et ainsi de suite. Cependant, tout cela a un prix, et pas seulement sur le plan financier. La qualité de l'air va se dégrader et, à cause de l'augmentation du nombre de véhicules, les embouteillages vont se multiplier, ce qui représente un gaspillage d'argent pour les conducteurs.

Quant aux clients, ils seront insatisfaits, car leurs attentes en matière de temps de trajet ne seront pas satisfaites, en raison du temps supplémentaire passé dans les embouteillages. Dans la littérature, concernant le covoiturage, sont utilisées différentes approches. Les approches les plus récentes et les plus importantes traitent le problème du covoiturage de trois manières. La première, consiste à résoudre un problème d'optimisation par la programmation en nombres entiers et la programmation par contraintes. La seconde consiste à considérer les routes de liaison comme les arêtes d'un graphe et les demandes comme des sommets. Ensuite, des algorithmes de graphes sont appliqués pour trouver les correspondances entre les passagers et les conducteurs. La dernière stratégie consiste à orchestrer les véhicules en utilisant un agent qui, grâce à un moteur d'apprentissage par renforcement, peut apprendre le comportement optimal pour répondre aux demandes.

Dans cette thèse, une nouvelle approche est présentée. Elle s'appuie sur un algorithme pour apprendre la politique optimale de prise en charge et de dépose afin de répondre aux demandes de transport en covoiturage. Le but de notre solution est de surmonter les limites des approches disponibles dans la littérature. Ces travaux concernent un cas d'étude spécifique divisant la zone de travail en zones prédéfinies. Leur solution ne peut pas être appliquée à une autre ville, pour cela un pré-traitement de la zone doit être effectué, avant d'appliquer leur approche. Ces modèles ne sont pas en mesure de faire face aux changements de structure de la ville. Si la ville s'agrandit et qu'une nouvelle zone est ajoutée à la zone de travail, tout le processus

doit être refait.

Pour surmonter ces limites, nous présentons une solution flexible, capable de s'adapter à l'évolution de la ville et aborde également le problème de la relocalisation des véhicules. Dans ce but, nous présentons Dynamic Rebalancing Based on Demand (DRBD), qui, en tenant compte des dernières demandes non servies, prédit les zones dans lesquelles il y a un manque de véhicules. Ensuite, les véhicules inactifs sont relocalisés dans ces zones qui ont été calculées.

Défis

La recherche sur les systèmes de transport couvre de nombreux aspects différents. On peut citer, par exemple, les véhicules connectés pour partager des informations ainsi que le contrôle efficace du trafic aux intersections pour améliorer la programmation des feux de circulation. En ce qui nous concerne, dans cette thèse nous nous concentrons sur le véhicule pour le service de covoiturage. Dans ce contexte, il est possible d'identifier plusieurs points qui pourraient être étudiés,

- **Correspondance passagers-conducteur** : Quels sont les principes qui conduisent un véhicule à servir une demande plutôt que d'autres ?
- **Algorithme de routage** : Trouver le chemin optimal est un problème complexe en soi. Si, en plus, nous permettons aux véhicules de faire du covoiturage, la complexité du problème augmente. La question n'est plus seulement liée au chemin le plus rapide en termes de temps ou de longueur du trajet, mais doit également prendre en compte des aspects plus complexes liés aux clients. Une stratégie efficace de routage, faisant du covoiturage, doit prendre en compte la distance globale parcourue. Il faut une stratégie sophistiquée pour trouver le chemin qui maximise les demandes servies tout en minimisant la distance parcourue plutôt qu'un algorithme gourmand pour trouver le chemin le plus rapide.
- **Priorité au covoiturage** : Comme cela a été fait pour les véhicules d'urgence, il est possible de définir une stratégie visant à donner la priorité aux véhicule de covoiturage ou aux véhicules qui transportent le plus de passagers afin d'encourager ces services plutôt que les moyens de transport privés.

- **Prévision de la demande** : Est la tâche d'estimation qui identifie comment et dans quelle mesure les demandes sont réparties dans une zone désignée. Avec une estimation fiable, il est possible de prévoir les zones de pointe et d'orchestrer ensuite une flotte pour répondre aux demandes de la meilleure façon possible.
- **Rééquilibrage** : Organise, en cas de besoin, la flotte pour augmenter l'efficacité du service en déplaçant les véhicules inactifs vers une zone où ils peuvent être plus performants. Cette tâche dépend en partie de celle définie au point précédent. En effet, il devrait être plus facile de relocaliser les véhicules lorsqu'on connaît la répartition des demandes. Cependant, l'estimation de la demande n'est pas la seule tâche impliquée dans ce travail, une stratégie de relocalisation des véhicules est nécessaire pour minimiser le temps passé sans passagers.

Buts et objectifs de la thèse

Cette thèse aborde le problème de servir les clients qui demandent un transport. Les clients veulent aller d'une origine à une destination dans la ville. Il est très probable de trouver des itinéraires communs et similaires entre les usagers. Servir ces demandes en covoiturage permet d'économiser de l'argent à la fois pour les clients et pour les conducteurs. Cela conduit également à d'autres avantages, environnementaux ou sociaux, en réduisant le nombre de véhicules impliqués dans le transport quotidien. Une flotte de véhicules est utilisée pour résoudre le problème décrit. L'objectif de cette proposition est de présenter une solution qui pourrait être réellement employée dans un scénario réel. Les objectifs sous-jacents de la solution peuvent être schématisés comme suit :

- Le système doit être capable de traiter des connaissances partielles Pour fonctionner dans un environnement réel. Il faut mettre en œuvre une solution capable de gérer des connaissances partielles. Si la ville évolue au fil du temps, l'expérience collectée peut servir de base aux nouvelles connaissances et un nouveau départ n'est donc pas nécessaire.
- Le système doit pouvoir être appliqué à toutes les villes sans aucune modification.

Pour être appliqué dans un scénario réel, il doit être conçue comme une solution de haut niveau plutôt que de s'adresser à un scénario ou une ville spécifique. Grâce à une stratégie générale, la même approche pourrait être diffusée dans le monde entier et être utilisée sans qu'il soit nécessaire de modifier le cadre.

- Le système doit pouvoir évoluer en fonction des besoins indiqués ci-dessus, il doit s'adapter aux demandes entrantes, en augmentant ou en diminuant le nombre de véhicules de manière dynamique.

Pour conclure, notre contribution peut être résumée comme suit : proposer un algorithme qui orchestre une flotte composée de véhicules , pour le partage de trajets et le rééquilibrage, afin de maintenir le niveau de service avec une utilisation limitée des ressources. De plus, cette approche est résiliente au changement de ville et peut être reproduite dans d'autres villes. Enfin, la tâche de rééquilibrage calcule les zones dynamiquement, en tenant compte uniquement des demandes disponibles en temps réel.

Hypothèses

Lors de la conception de la proposition, certains aspects ont été omis dans cette thèse. La première hypothèse est que chaque véhicule mis en œuvre est capable de fournir un service continu. Les véhicules sont donc sans défaillance et ne peuvent pas avoir d'accident. Si le temps demandé pour servir un trajet n'est pas fixe, un module ayant été utilisé pour prendre en compte l'état du trafic en fonction de l'heure de la journée, ce dernier module n'a pas été utilisé pour connaître l'impact de ces véhicules sur le trafic. Dans ce travail, chaque véhicule est autorisé à se garer partout sans produire de congestion et la flotte ne contribue pas au trafic urbain. Avant de mettre en œuvre un système tel que celui proposé, il convient d'évaluer l'impact de la flotte sur la congestion et l'emplacement des véhicules en attente du prochain trajet. Ces évaluations ne sont pas faites dans cette recherche.

Contribution

Ce travail aborde un scénario de covoiturage dans lequel une flotte de véhicules doit s'organiser, sans coordination entre eux, pour répondre aux demandes

de transport entrantes. La recherche derrière cette thèse veut démontrer la faisabilité d'un système de covoiturage qui est capable de faire face à un scénario réel, en utilisant une approche bio-inspiré de rééquilibrage des voitures.

Dans l'état de l'art, plusieurs approches combinent l'utilisation de systèmes d'agents et la division statique de la ville en zone. Une revue détaillée de ces approches est faite au chapitre 2.1. En bref, ces solutions répondent à une situation spécifique et sont conçues pour ne traiter qu'une configuration spécifique. Pour donner un exemple, la plupart de ces solutions divisent la zone urbaine en plusieurs régions, qui sont utilisées pour identifier la position d'un véhicule ou l'emplacement d'une demande. En outre, certains de ces travaux disponibles dans la littérature impliquent l'ensemble de la flotte dans le processus d'évaluation de la prochaine action. Cela limite la capacité du système, le rendant incapable d'adapter le nombre de véhicules en temps réel. Le modèle présenté dans ce travail représente une position à l'aide d'une stratégie générique plutôt que d'utiliser une stratégie propre. Grâce à cette représentation globale, il est possible de localiser chaque demande avec plus de précision, et cela permet à l'approche conçue d'être indépendante de la ville.

Le même algorithme peut être appliqué à plusieurs villes dans le monde sans aucune modification de la partie modélisation. La taille de la flotte est capable de s'adapter pour faire face aux demandes entrantes ou pour diminuer le nombre de ressources utilisées dans le cas où le même nombre de clients pourrait être servi avec moins de ressources. Il est ainsi possible de réduire les encombrements inutiles du trafic. Pour que la solution soit évolutive en termes de taille de la flotte, un véhicule ne perçoit que les demandes proches de lui, il n'a donc aucun indice sur les autres véhicules et agit comme s'il était le seul acteur jouant dans l'environnement.

Organisation de la thèse

Le reste de la thèse est organisé comme suit :

— Chapitre 1

Introduit les travaux connexes disponibles dans la littérature, en les comparant sous plusieurs points clés et en discutant les avantages et les inconvénients de

chaque approche.

— **Chapitre 2**

Passe en concepts théoriques de base que le lecteur doit connaître pour bien comprendre et apprécier pleinement le travail proposé.

— **Chapitre 3**

Présente l'idée sous-jacente d'un point de vue abstrait, suivie d'un aperçu de la structure de mise en œuvre présentant les principaux défis auxquels le système doit faire face.

— **Chapitre 4**

Présente une application réelle du système sur une zone restreinte de la ville de New York. puis on montre les résultats obtenus par la mise en œuvre de l'approche proposée pour le problème de covoiturage présenté dans cette thèse. A travers une évaluation de plusieurs mesures, il souligne les avantages et les inconvénients de ce travail en le comparant à d'autres versions de référence.

— **Conclusion et perspectives**

Conclut la thèse en résumant la proposition et en discutant brièvement des questions ouvertes et les perspectives pour les futures travaux.

Chapitre 1

État de l'art et travaux connexes

Ce chapitre examine les approches actuelles disponibles dans l'état de l'art concernant les systèmes de covoiturage. En particulier, nous avons abordé en particulier plusieurs recherches qui, à travers différentes stratégies, atteignent le même objectif.

1.1 Introduction

Le thème du covoiturage, fréquemment désigné par l'abréviation RS (Ride Sharing), est un problème bien connu depuis le 20^{ième} siècle et les chercheurs continuent à l'étudier à ce jour. Le covoiturage pose plusieurs défis, tels que l'appariement des conducteurs et des usagers ou la minimisation de la distance totale parcourue. Chaque sous-tâche du covoiturage peut être liée à d'autres tâches dans différents domaines. Par exemple, dans le cas du vélo en service libre, nous avons une tâche de relocalisation pour déplacer les bicyclettes d'une station à l'autre, ou encore, la tâche d'appariement client-conducteur pourrait être appliquée à un système de transport normal qui sert la demande de transport indépendamment du covoiturage.

Étant donné la nature de notre proposition, nous étudions celles qui abordent le problème de covoiturage d'un point de vue global impliquant plusieurs tâches, et donc celles qui pourraient être appliqués dans un scénario réel. Dans l'état de l'art, différentes approches sont disponibles pour traiter les systèmes de covoiturage. Dans ce qui suit, nous proposons une comparaison entre les multi-agents, les problèmes d'optimisation et les algorithmes qui s'appuient sur les chemins des graphes.

1.2 Covoiturage avec un problème d'optimisation

Les chercheurs de [AB14],[AMB16] ont modélisé le problème du traitement des demandes de transport en covoiturage par la programmation mathématique en nombres entiers. Les acteurs des systèmes appartiennent à différentes catégories. On trouve les personnes censées conduire (conducteurs) et les demandeurs, qui de-

mandent simplement à être servis comme passagers. En outre, il existe une troisième catégorie appelée "shifters", dans laquelle on trouve des personnes qui sont censées conduire mais qui sont prêtes à abandonner leur rôle pour devenir des passagers. Ces travaux traitent d'un scénario dans lequel les conducteurs utilisent leur propre véhicule privé pour atteindre leur destination et, ce faisant, ils prennent en charge des passagers qui correspondent à la destination ou à l'origine.

Pour chaque acteur du système sont définies quelques variables, comme la distance qu'il est prêt à parcourir et la marge de temps. La distance définie indique la distance maximale que chaque passager accepte de parcourir. Elle comprend, soit celle parcourue pour être prise en charge, soit celle parcourue pour atteindre la destination finale. La marge temporelle indique l'intervalle dans lequel un utilisateur peut commencer et le temps dans lequel ce même utilisateur veut atteindre la destination. Selon l'utilisation, il faut définir le cadre de temps entre le départ le plus tôt et l'arrivée la plus tardive qu'il peut tolérer.

Les objectifs sont différents dans les deux travaux. Dans [AB14], l'objectif est de minimiser la distance totale parcourue par les conducteurs. Alors que [AMB16] vise à maximiser le nombre de participants au système de covoiturage. Pour les deux problèmes d'optimisation, certaines contraintes sont imposées. Par exemple, chaque passager doit être le passager de l'un des conducteurs réalisables, et un conducteur qui prend un passager est aussi un conducteur. L'un des sous-objectifs des deux travaux est de maintenir le temps de résolution, nécessaire à l'algorithme pour calculer les correspondances entre les conducteurs et les passagers, aussi bas que possible. Comme résultats, les chercheurs ne présentent pas le temps d'attente ou la couverture de la demande. Ils donnent plutôt des informations sur les utilisateurs, comme le nombre de ceux qui sont concernés par la stratégie et le rôle qu'ils jouent.

1.3 Covoiturage avec les graphes

Une façon intéressante d'aborder le covoiturage est de recourir à la théorie des graphes. Une solution intelligente est disponible dans l'état de l'art. Elle concerne une flotte de véhicules autonomes dans la région d'Austin au Texas [FKB15]. L'objectif de ce travail est de démontrer qu'un véhicule partagé peut remplacer plusieurs voitures conventionnelles, sans affecter le niveau de service. Le modèle peut être considéré comme une composition de quatre sous-modules : **l'affectation conducteur-client, la génération de la flotte, les actions du covoiturage et enfin le rééquilibrage.**

-Tout d'abord, le module **d'affectation des conducteurs** est utilisé pour déterminer quels sont les voitures disponibles les plus proches d'une demande à satisfaire. Ensuite, ce module doit identifier le véhicule le plus approprié pour répondre à la demande. Il est à noter que le modèle donne la priorité aux demandes qui attendent depuis plus longtemps, tandis que le véhicule est choisi comme celui qui a le temps de trajet le plus court, pour atteindre les passagers, parmi l'ensemble des voitures éligibles. Pour identifier le conducteur approprié, on utilise une version modifiée de l'algorithme de Dijkstra, dont l'idée principale est de définir un rayon à partir de l'origine des passagers, fixé à une limite de temps. Chaque véhicule à l'intérieur du cercle est éligible pour répondre à la demande. Si aucun véhicule n'est trouvé, la zone de recherche s'élargit pas à pas jusqu'à ce qu'un véhicule soit trouvé ou que la limite maximale soit atteinte.

-Deuxièmement, **la génération de la flotte** détermine le nombre adéquat de véhicules nécessaires pour faire face à un scénario. La taille de la flotte est réglée par une première exécution de la simulation le premier jour. Un nouveau véhicule est alloué s'il n'est pas possible d'identifier un véhicule approprié qui peut répondre à une demande dans le temps maximum. Une fois qu'un véhicule est attribué, il reste en permanence dans le système. Une fois la session de réglage terminée, aucun véhicule ne peut plus rejoindre la flotte. Toutes les voitures commencent une simulation au même endroit que celui où ils l'ont terminé pendant la session de réglage.

-**Les actions de voitures** concernent les décisions de déplacement des véhicules. Un véhicule peut être assigné à un usager ou être relocalisé. Une fois qu'une action est choisie, le véhicule commence à rouler pour atteindre sa destination. Pendant le trajet, un véhicule suit le chemin prévu, qui est le plus court disponible. La vitesse de déplacement au niveau des liens varie toutes les heures et est déterminée par des simulations.

-Enfin, le dernier module est **le rééquilibrage**. Il déplace les véhicules dans la zone pour répondre à la demande. Cette tâche est cruciale pour la performance globale de cette stratégie. Plus un véhicule est proche de ses futurs usagers, le temps d'attente est court pour être servi. Un inconvénient important de cette tâche concerne les véhicules vides qui circulent. En effet, alors que le temps d'attente est réduit par le rééquilibrage, le temps parcouru par les véhicules vides augmente. Ces véhicules vides contribuent à la congestion du trafic sans apporter aucun bénéfice.

La stratégie de rééquilibrage repose sur un critère de division de la zone. La zone est divisée en blocs, puis pour chaque bloc est calculé son équilibre défini comme suit,

$$Equilibre_de_zone = NV_{Total} \left(\frac{NV_{Zone}}{NV_{Total}} - \frac{Demandes_{Zone}}{Demandes_{Total}} \right). \quad (1.1)$$

ou NV : le nombre de véhicules

D'après l'équation 1.1, un bloc dont le solde est négatif tente de tirer un véhicule du voisinage, tandis que ceux dont le solde est positif tentent de pousser un véhicule vers les zones les plus proches. Le véhicule qui se déplace d'une zone à l'autre est celui qui est le plus proche du centre de la zone de destination.

Les résultats obtenus concernant le temps d'attente sont incroyables. Près de 95% des usagers attendent moins de 5 minutes avant d'être servis. Cependant, le nombre de véhicules impliqués dans les simulations est très élevé, ce qui a un impact sur le réseau routier de la ville. Une fois mis en œuvre dans un cas réel, ce nombre de véhicules affectera le temps d'attente. En outre, [FKB15] étudie l'impact d'une

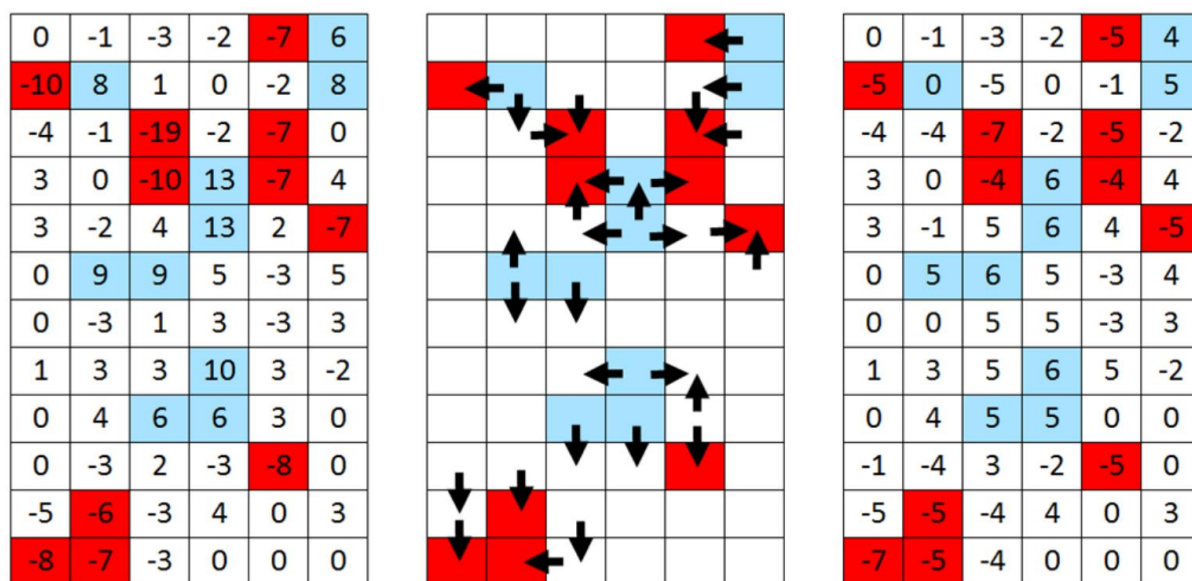


Figure 1.1 – Déplacement des voitures pour améliorer le déséquilibre. À gauche, les déséquilibres initiaux prévus, au centre, la relocalisation des voitures en blocs décalés et enfin la distribution résultante [FKB15].

flotte de voitures sur l'environnement en comparant les résultats avec une flotte de véhicules traditionnels.

Un autre travail, dans lequel il n'est pas explicite qu'une approche par graphe a été utilisée mais qui est très probable, est celui présenté dans [Bat+18]. Ici, l'algorithme commence lorsqu'une demande est générée. Elle est diffusée à tous les véhicules situés dans un rayon préfixé. Ces véhicules évaluent leur éligibilité à la demande et répondent ensuite aux usagers en indiquant leur disponibilité. Une fois que les passagers ont reçu la réponse, ils décident, en renvoyant une confirmation positive, quel véhicule convient le mieux pour leur trajet.

Pour cette approche, les demandes sont pré-traitées et un regroupement est effectué pour définir les demandes dans des régions proches qui peuvent être regroupées. Plusieurs mesures sont proposées pour évaluer l'approche. Premièrement, la couverture de la demande réalisée par les véhicules avec le pourcentage de demandes servies en covoiturage. Ensuite, la facilité d'utilisation des véhicules avec leur kilométrage. Enfin, le temps d'attente des passagers et le coût du trajet pour évaluer l'économie.

1.4 Covoiturage avec un système d'agent

La modélisation du problème par un système d'agents peut inclure différentes approches. La principale différence réside dans les systèmes centralisés et décentralisés. Dans un système centralisé, un noyau prédominant gère l'ensemble de la flotte, en suivant le statut de tous les véhicules et leur perception de toutes les demandes disponibles. D'autre part, une approche décentralisée ne dispose pas d'une unité centrale. Chaque agent, qui agit de manière autonome, est affecté à un véhicule de la flotte.

Dans une approche multi-agents distribuée, l'état interne d'un agent décrit le statut du véhicule. Il peut inclure des informations telles que la position du véhicule, le nombre de passagers et leur destination, etc. Alors que la perception de l'agent peut se concentrer uniquement sur quelques requêtes, celles qui sont importantes pour le véhicule analysé. Dans un système MoD, ils ont suggéré d'utiliser une approche décentralisée afin de réduire la complexité du problème et de faciliter l'évolutivité du modèle.

Dans l'état de l'art sont définis plusieurs travaux concernant le dispatching efficace des véhicules ou le problème du covoiturage est appliqué à un système de mobilité à la demande (SMoD). Nous citons ci-après, au meilleur de notre connaissance, deux des principales et récentes solutions disponibles.

La première est [AGA19]. Dans ce travail chaque véhicule a son moteur de décision pour sélectionner l'action optimale en donnant son état et sa perception. L'état d'un agent est représenté par l'heure de prise en charge et la destination de ses passagers, ainsi que le nombre de sièges libres dont il dispose. L'agent perçoit des informations sur les demandes disponibles dans tout l'environnement et la position des autres véhicules qui appartiennent à la flotte. Toutes ces données sont traitées par un algorithme d'apprentissage par renforcement profond qui renvoie les actions optimales à effectuer.

L'objectif principal de cette recherche est d'organiser une flotte de taxis, en

dirigeant les voitures à différents endroits pour répondre aux demandes de transport en covoiturage. Tout en réalisant l'objectif principal, les chercheurs visent à réduire le temps de trajet ainsi qu'à atteindre un bon niveau de service avec un nombre limité de véhicules et de ressources. Grâce à cela, il est possible de réduire les coûts de transport pour les passagers et de diminuer la congestion du réseau routier.

Des mesures de performance ont été définies pour évaluer si, et dans quelle mesure, la portée est atteinte. Les chercheurs ont suivi le temps d'attente, le taux de rejet et, au final, le temps parcouru par les véhicules à vide. Enfin, ils proposent un taux d'utilisation qui décrit le temps pendant lequel un véhicule est occupé.

L'appariement conducteur-clients se fait en fonction de la position de l'agent. Une fois qu'une demande est générée, elle est attribuée à un véhicule de la même zone. S'il n'y a pas de véhicules éligibles, la demande est rejetée. Une fois qu'elle est attribuée à un véhicule, le conducteur doit se rendre au point de prise en charge pour charger les nouveaux passagers. Le cadre comprend un graphe de réseau routier, pour identifier le trajet optimal, et un modèle qui estime le temps nécessaire au voyage. L'estimation du temps s'appuie sur un réseau neuronal.

Compte tenu de la contrainte selon laquelle les passagers ne sont servis que s'il existe un véhicule à proximité d'eux, cette approche nécessite un cadre de prédiction de la demande pour prévoir les demandes futures, puis coordonner les véhicules dans les zones pour anticiper la demande. Un autre réseau neuronal est proposé pour accomplir cette tâche. Il traite la tendance de la demande au cours des 30 dernières minutes pour faire une estimation de la tendance future. En sortie, une carte est produite. Les véhicules sont ensuite relocalisés en fonction de cette carte, ce qui permet de réduire l'écart de distance entre les conducteurs et les clients. Notez que, pour la relocalisation, une voiture peut passer par 7 zones au maximum.

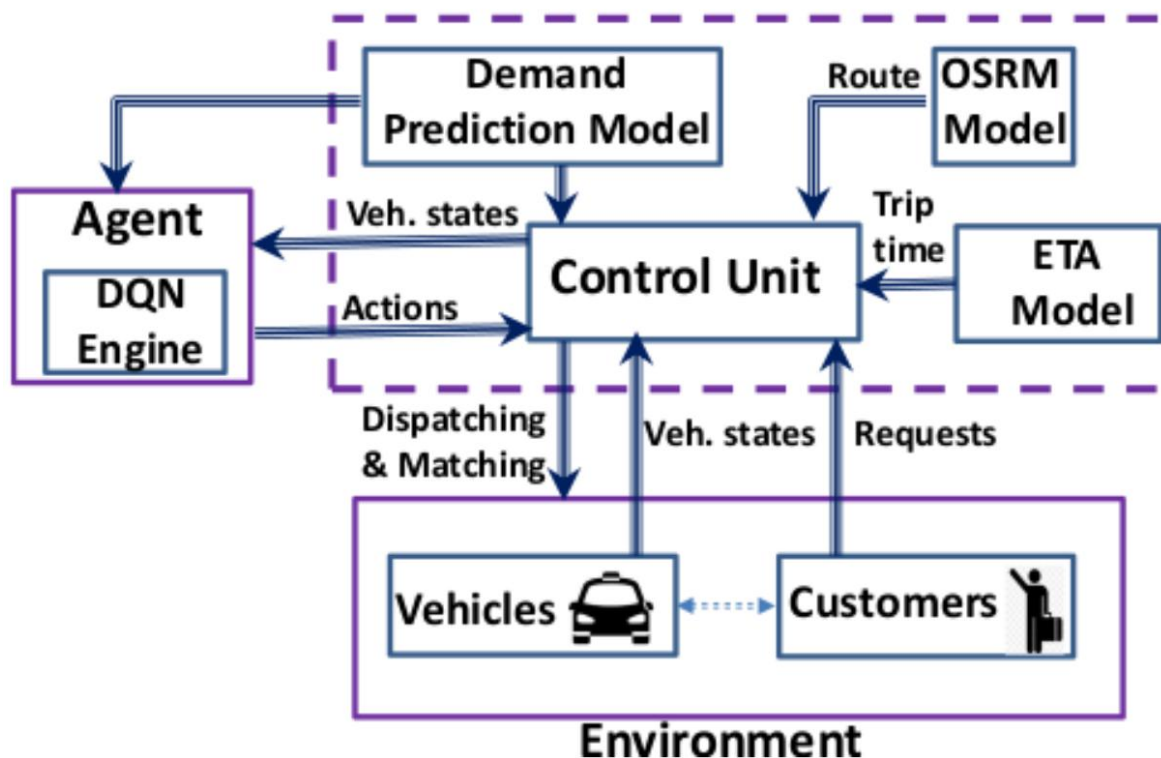


Figure 1.2 – Architecture proposée par [AGA19].

Les recherches présentées abordent le problème du covoiturage dans le contexte multi-agent, l'évaluation des performances ne concernant que quelques mesures.

Pour le deuxième dans l'état de l'art, on trouve des travaux qui analysent plusieurs aspects. L'un d'entre eux est SAMoD [GD18], dans lequel les chercheurs ne se contentent pas d'évaluer la performance de leur approche mais évaluent également l'impact de leur flotte sur le réseau routier.

SAMoD vise à évaluer l'influence du trafic régulier sur le comportement des voitures, ainsi que l'influence des voitures sur le trafic régulier. SAMoD met en œuvre une approche multi-agents distribuée dans laquelle chaque agent dirige un véhicule. Le processus d'apprentissage se fait par Q-learning et l'objectif des agents est de satisfaire les demandes de transport qui sont générées.

A chaque étape, un agent a trois choix : (1) prendre une demande, que ce soit la première ou non ; (2) se rendre à destination pour déposer les usagers ; (3)

se relocaliser vers une nouvelle zone dans laquelle on a prévu qu'il y aurait des demandes à servir.

L'état du véhicule est constitué de 3 variables. Deux sont binaires et indiquent s'il y a des demandes disponibles à servir, respectivement, dans la zone courante et dans les zones proches. La dernière indique si un véhicule a des sièges libres ou non sans en donner le nombre exact, une exception étant faite lorsque le véhicule est vide. Dans ce cas, il est signalé que l'agent dispose de tous les sièges disponibles. La perception des agents est limitée à un sous-ensemble de zones. Il s'agit de la zone dans laquelle se trouve l'agent et des zones directement voisines. À partir de ces zones, l'agent peut percevoir la quantité de demandes disponibles et le nombre d'agents employés dans les zones concernées.

Les véhicules de la flotte ne communiquent pas directement mais se coordonnent à travers l'environnement. Grâce à cela, aucune modification du système n'est nécessaire si la flotte s'agrandit ou si une voiture quitte l'équipe. Si un agent est en train de servir une demande, il ne peut pas évaluer des nouvelles demandes d'autres passagers que si la zone de destination de ces nouvelles demandes correspond à une destination des usagers actuels.

L'algorithme de relocalisation est très sophistiqué dans SAMoD. Chaque véhicule apprend par lui-même quand un déplacement est nécessaire et quelle est la meilleure politique pour le faire. Il n'existe essentiellement que deux stratégies : un véhicule peut décider de se déplacer dans une zone voisine où il y a le plus de demandes en attente ou dans une zone proche où l'écart entre l'offre de véhicules et le nombre de demandes est le plus important. En outre, un agent peut décider d'appliquer les deux stratégies sur la base des données historiques, collectées au fil du temps, ou sur la base des informations en temps réel qu'il peut obtenir.

En plus des mesures d'évaluation concernant la satisfaction des clients et le comportement de l'agent à propos des trajets, les chercheurs proposent une comparaison entre la même implémentation du système avec/sans évaluation de l'impact de leur agent sur le réseau routier. Il est intéressant de voir comment le nombre

de demandes satisfaites diminue lorsque l’on prend en compte la contribution des véhicules à la congestion du trafic.

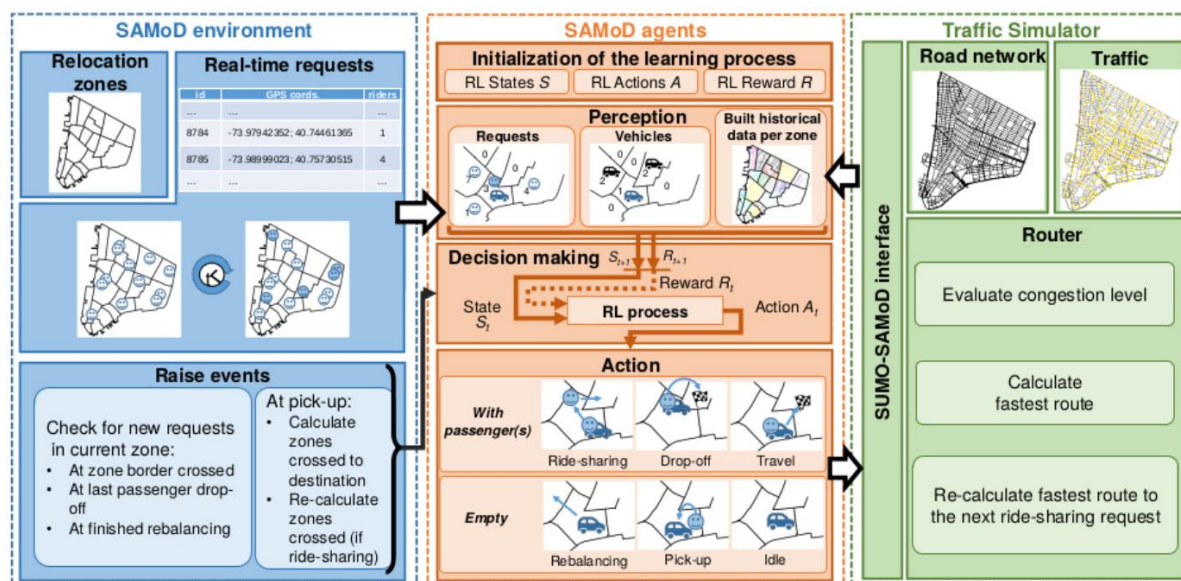


Figure 1.3 – Système SAMoD [GD18].

Les deux travaux analysés ont en commun le concept de zone. La zone de travail a été divisée en zones pour identifier la position des véhicules ainsi que l’origine et la destination des demandes.

1.5 Résumé

Pour conclure ce chapitre, nous présentons un résumé des recherches examinées et une comparaison avec notre proposition.

	[AGA19]	[GD18]	[AB14],[AMB16]	[FKB15]	Ce travail
Environnement	discretisé en zones	discretisé en zones	discretisé en zones	discretisé en zones	continu sur les coordonnées
Applicable à (sans modifications)	Manhattan	Manhattan	Monde	Austin	Monde
Évaluer les conditions de trafic	✘	✔	✘	✘	✔
Évaluer l’impact sur le trafic	✘	✔	✘	✘	✘
Taille de la flotte	statique	dynamique	statique	dynamique	dynamique
Relocalisation des véhicules	✔	✔	✘	✔	✔

Table 1.1 – Résumé des recherches examinées et de notre proposition.

Le tableau ci-dessus met en évidence deux des principales contributions de notre proposition à l’état de l’art. La nouveauté est triple. Tout d’abord, notre propo-

sition peut être appliquée dans toutes les villes du monde sans nécessiter aucun changement de la structure . Cependant, notre modèle s'appuie sur la norme du système de coordonnées géographiques *WGS84*, et ne nécessite donc pas une discrétisation en zones. Grâce à cette norme, notre algorithme est flexible face aux changements de ville qui peuvent survenir au fil des temps.

Enfin, notre proposition comporte une autre contribution qui ne ressort pas du tableau 1.1. Dans ce travail, nous introduisons DRBD (Dynamic rebalancing based on demand). DRBD définit une stratégie de relocalisation dynamique des véhicules qui repose sur un clustering dynamique effectué sur une portion restreinte de données. L'objectif de DRBD est de prédire les zones où les véhicules doivent être déplacés pour anticiper la demande future. Les zones sont identifiées en utilisant uniquement les demandes en temps réel qui n'ont pas encore été servies. Les clusters sont alloués dynamiquement. Ainsi, le nombre de zones dans lesquelles un véhicule peut se déplacer n'est pas fixé et sa position peut changer dans le temps.

Le travail proposé par cette thèse évalue les conditions de circulation tout en estimant le temps nécessaire pour se rendre d'un point à un autre. La plupart des travaux présentés estiment le temps par le biais d'une méthode, à savoir un réseau neuronal. Dans ce travail, nous utilisons un cadre qui, étant donné une zone et les données associées, reproduit les mêmes conditions décrites par les informations données. Ensuite, il est possible d'interroger ce cadre pour obtenir le temps et la distance d'un trajet.

Le chapitre suivant introduit l'architecture de notre proposition, en présentant en détail comment les avantages présentés ci-dessus sont assurés par le modèle.

Chapitre 2

L'optimisation, les méta-heuristiques et les techniques bio-inspirées

2.1 Introduction

La résolution de problèmes d'optimisation est devenue un sujet central en recherche opérationnelle, le nombre de problèmes d'aide à la décision pouvant être formalisés sous la forme d'un problème d'optimisation étant en forte croissance. Les problèmes d'apprentissage de réseaux de neurones, de planification des tâches ou encore d'identification sont, par exemple, des problèmes communs d'optimisation. En outre, l'être humain cherche à améliorer sa vie quotidienne, il aime la perfection et sans qu'il s'en rende compte, il essaye de minimiser ses charges, son loyer ou la consommation de sa voiture. Il tente toujours d'optimiser : que ce soit minimiser ses dépenses ou maximiser ses recettes.

Actuellement, l'optimisation est devenue un domaine indispensable pour résoudre plusieurs problèmes que se soit dans l'industrie ou d'autres secteurs. En effet, nous avons assisté ces dernières années à une croissance très rapide des travaux utilisant les méthodes d'optimisation. Cette tendance peut être observée dans tous les domaines de la science. Nous aborderons dans ce chapitre les définitions générales des méthodes d'optimisation qui se divisent sous deux volets déterministes et non déterministes.

2.2 Problème d'optimisation

La formulation des problèmes d'optimisation reste très ambiguë à cause de la diversité des vocabulaires et des confusions éventuelles que cela pourrait engendrer. Nous avons convenu d'adopter le vocabulaire qui suit [Hac13] :

- Un **problème d'optimisation mono-objectif** est défini par un ensemble de variables, une fonction objective et un ensemble de contraintes.
- Un **problème d'optimisation multi-objectif** est défini par un ensemble de variables, un ensemble de fonctions objectif et un ensemble de contraintes.
- L'**espace de recherche**, appelé aussi domaine de recherche, est l'ensemble des

domaines de définition des différentes variables du problème.

- **Les variables du problème** dites aussi variables de conception ou de décision peuvent être de nature diverse (réelles, entières, booléennes. etc.) et exprimer des données qualitatives ou quantitatives.
- **La fonction objectif** ou encore (fonction de coût) définit le but à atteindre, minimiser ou maximiser celle-ci.
- **Une fonction multimodale** présente plusieurs minimas (locaux et globaux). Tandis qu'une fonction uni-modale n'a qu'un minimum, le minimum global.
- **L'ensemble des contraintes** est en général un ensemble d'égalités ou d'inégalités que les variables de l'espace d'état doivent satisfaire. Ces contraintes limitent l'espace de recherche.

Les méthodes d'optimisation recherchent un point ou un ensemble de points dans l'espace de recherche qui satisfont l'ensemble des contraintes, et qui maximisent ou minimisent la fonction objectif.

2.3 Méthodes d'optimisation

Sur les grilles de calcul, la réplication peut être statique ou dynamique [BOU17b]. Dans la réplication statique, les répliques sont créées, gérées ou supprimées manuellement. Par conséquent, la réplication statique a le problème de ne pas pouvoir s'adapter aux changements de comportement des utilisateurs. Dans un scénario réel où les données sont mesurées en pétaoctets et où il existe des centaines de communautés d'utilisateurs dans le monde, la réplication statique peut ne pas être réalisable. D'autre part, elle est utilisée dans les systèmes orientés pour partager des données.

2.3.1 Les méthodes de recherche locale

La recherche locale est une méthode utilisée pour résoudre des problèmes d'optimisation difficiles. La recherche locale peut être utilisée dans les problèmes de

recherche d'une solution qui maximise un critère parmi un ensemble de solutions candidates. Les algorithmes de recherche locale se déplacent d'une solution à l'autre dans l'espace des solutions candidates (l'espace de recherche) jusqu'à une solution considérée lorsque le temps optimal est trouvé ou que le temps imparti est dépassé. D'autre part, optimiser localement signifie rechercher une solution à un problème proche d'une solution de départ (optimisation locale), mais meilleure en termes de coût (fonction objectif). Pour ce faire, une meilleure solution est recherchée par itérations successives, cette classe de méthodes peut être déterministe ou non déterministe.

— Les limites de l'optimisation local

Dans la catégorie de problèmes que nous avons présentés ci-dessus, les méthodes cherchent le minimum d'une fonction en se basant sur la connaissance d'une direction de recherche. Bien sûr, ces méthodes seront toujours applicables et même recommandées lorsque la solution cherchée est réputée proche de la solution connue ou si la fonction objectif est convexe, en particulier, si le calcul n'est pas coûteux. Toutefois, elles ne sont pas indiquées pour les problèmes multimodale où le risque de rester bloqué dans un optimum local est fort probable. Cela suffit pour illustrer l'enjeu des problèmes d'autres types : L'optimisation globale. Quand un utilisateur se rend compte qu'il trouve des minimums locaux différents en exécutant son code d'optimisation, il est tout à fait légitime qu'il songe à un code qui serait capable d'en tenir compte et de donner le meilleur de tous [[Hac13](#)].

2.3.2 Méthode de recherche globale

Ces dernières années, l'optimisation globale a fait l'objet de plusieurs études en raison de nouveaux acquis théoriques, de fortes demandes dans divers domaines dont les applications industrielles, et du développement des ressources de calcul. C'est ainsi que plusieurs articles ont été publiés sur le sujet, reflétant une richesse d'approches et de motivations. Citons [[Tsoo8](#)],[[YHLo8](#)], [[LT08](#)],.. etc. Contrairement

à ce que l'on pourrait être tenté de croire, l'optimisation numérique globale n'a pas les facilités pour hériter (du moins pas systématiquement) des techniques d'optimisation numérique locale. En fait, comme on peut le voir, ce dernier est utilisé pour l'essentiel de la direction de descente (conditionnée au calcul du gradient ou à son approximation), ce qui permet de converger naturellement vers un point minimal local.

Or, pour être précis, l'optimisation globale évite de rester sur ces points. Au lieu de cela, il devra s'en échapper. C'est pourquoi de nombreuses méthodes ont été utilisées pour tenter de résoudre ce problème. Il s'agit de trouver l'état minimum et de ne s'arrêter que lorsqu'il est le meilleur (optimum global). Pour illustrer cette variété, nous citons ci-dessous les méthodes les plus utilisées parmi elles et les différents principes sur lesquels elles reposent. Bien qu'il existe de nombreuses méthodes proposées, nous pouvons trouver leurs caractéristiques en fonction de leurs méthodes. Par conséquent, nous donnons les classifications suivantes : Selon qu'il s'agisse ou non de processus probabilistes, il existe deux approches : déterministe et non déterministe.

2.4 Approches déterministes

Ces méthodes ont d'abord été introduites pour résoudre précisément des problèmes spécifiques, tels que les problèmes continus et linéaires sous contraintes linéaires (algorithme du simplexe de Dantzig); ces méthodes sont également étendues aux cas discrets et mixtes, mais limitées aux cas linéaires. La principale caractéristique des méthodes globales déterministes est qu'elles ne nécessitent pas de point de départ. Contrairement aux méthodes aléatoires, ces méthodes gèrent bien les contraintes et peuvent être appliquées à des problèmes mixtes (variables réelles, entières et catégorielles).

elles garantissent des solutions globales aux problèmes. Cependant, il convient de noter que les méthodes déterministes globales peuvent toujours être utilisées tant

que le nombre de variables ne devient pas trop important. Au-delà de vingt variables, elles atteignent leur limite. Ces méthodes permettent d'obtenir une solution exacte au problème d'optimisation considéré à la convergence, avec des garanties absolues : en utilisant l'arithmétique des intervalles d'arrondis telle que définie dans [Gar+08].

Aucune erreur numérique cachée ne peut éloigner un tel algorithme de la solution optimale, ne le ralentissant que dans le pire des cas. Ces algorithmes sont appelés : algorithmes de Branch-and-Bound par intervalles. Les plus intéressants d'entre eux sont :

- Méthodes à Recherche par Quadrillage (Grid Search Methods);
- Méthodes des Trajectoires (Trajectory Methods);
- Méthodes de séparation-évaluation (Branch-and-Bound).

2.5 Les approches évolutionnaires

Ces méthodes non déterministes utilisent le tirage de nombres aléatoires. Ils permettent d'explorer plus efficacement l'espace de recherche. Dans ce qui suit, nous nous intéresserons plus particulièrement aux méta-heuristiques. Le mot méta-heuristique vient d'une combinaison de deux mots grecs :

- heuristique vient du verbe heuriskein (euriskein), qui signifie trouver;
- meta C'est un suffixe qui signifie au-delà, à un niveau supérieur.

Les premières méta-heuristiques datent des années 1980.elles sont généralement utilisées lorsque les méthodes conventionnelles échouent.

Le terme méta-heuristique est utilisé par opposition à heuristique. En effet, la méta-heuristique, peut être utilisé pour de nombreux types de problèmes, tandis que l'heuristique s'applique à un problème donné. Les méta-heuristiques ont une caractéristique commune de par leur nature stochastique, c'est-à-dire que cette partie de la recherche se fait de manière aléatoire, elles s'inspirent d'analogies avec la réalité : physique (recuit simulé, etc.), biologie (algorithmes évolutifs, recherche de

tabous, etc.) ou éthologie (colonies de fourmis...). En plus de cette base stochastique, les méta-heuristiques sont généralement itératives, c'est-à-dire qu'elles appliquent plusieurs fois le même schéma de recherche au cours du processus d'optimisation, et directes, c'est-à-dire qu'elles n'utilisent pas les informations de gradient de la fonction objectif. Elles ont été particulièrement intriguées par leur capacité à éviter les optima locaux, soit en acceptant la dégradation de la fonction objectif au cours de leur progression, soit en utilisant des groupes de points comme méthode de recherche. En raison de la prolifération des recherches dans ce domaine, un grand nombre de telles méthodes existent [Tsoo8].

2.6 Généralités sur les méta-heuristiques

Les méta-heuristiques sont des heuristiques modernes dédiées à la résolution d'un large éventail de problèmes, en particulier des problèmes d'optimisation bien connus. Pour les difficultés sans méthodes classiques efficaces, les méta-heuristiques visent à atteindre un optimum global, souvent caché au milieu de nombreux optima locaux. Ils fournissent des solutions de qualité dans un délai raisonnable. En général, les méta-heuristiques sont des algorithmes stochastiques itératifs qui atteignent un optimum global en évaluant une fonction objectif. Le développement de machines dotées d'une énorme puissance de calcul démontre l'intérêt croissant pour les méta-heuristiques [Bou17].

2.6.1 Définition

Dans la littérature, on trouve plusieurs définitions qui tentent d'expliquer clairement le concept de méta-heuristique, dont aucune n'est généralement acceptée, dans :

- **Définition 1** : Un processus itératif d'heuristiques de subordination et d'amorçage qui combine intelligemment plusieurs concepts pour explorer et utiliser l'ensemble de l'espace de recherche. Stratégies d'apprentissage pour la construc-

tion d'informations pour trouver efficacement des solutions optimales ou quasi optimales [AO04].

- **Définition 2** : Les méta-heuristiques sont généralement des algorithmes stochastiques itératifs qui progressent vers un optimum global, l'extremum global d'une fonction objectif [Pha+07].
- **Définition 3** : Une méta-heuristique est formellement désignée comme un processus de génération itérative qui guide une heuristique subordonnée en combinant intelligemment des concepts différents pour explorer et exploiter l'espace de recherche. Des stratégies d'apprentissage sont utilisées pour structurer l'information afin de trouver efficacement des solutions quasi-optimales [OL96].

2.6.2 Types de méta-heuristiques

-Métaheuristique de voisinage

Les méta-heuristiques les plus classiques sont celles basées sur le concept de voisinage. elles partent d'une solution initiale obtenue de manière exacte, ou par tirage aléatoire, et s'en éloignent progressivement sur un espace de recherche local adjacent à chaque itération. Ensuite, comparez-le avec la meilleure valeur. L'identification du concept de quartier est nécessaire. Les métaheuristiques les plus célèbres de cette classe sont le recuit simulé et la recherche taboue [Bou17].

-Métaheuristique de population

Manipuler un ensemble de solutions en parallèle à l'aide d'une famille méta-heuristique de concepts de population. Les algorithmes inclus dans cette catégorie comprennent les algorithmes génétiques et les algorithmes de colonies de fourmis [Bou17].

-Métaheuristiques pour les méthodes implicites

Lors de l'utilisation de la méthode implicite, la distribution de probabilité est Inconnue ou inutilisée; le choix de la solution initiale et du modèle de la diversification entre itérations ne suit pas une loi donnée, elle est soumise au hasard [Bou17].

-Métaheuristiques avec méthodes explicites

Méthodes explicites métaheuristiques utilisant des distributions de probabilité sélectionnées à chaque itération. C'est le cas des algorithmes d'estimation de distribution, comme son nom l'indique, il est estimé par la distribution à chaque itération Espace de recherche local optimal probabiliste [Bou17].

2.6.3 Classification méta-heuristique

Il existe plusieurs façons de catégoriser et de décrire les méta-heuristiques, plusieurs classifications sont possibles [BR03]

-Inspiré par la nature Intuitivement, on peut classer les méta-heuristiques en

séparant celles qui s'inspirent de phénomènes naturels de celles qui ne le sont pas. Les algorithmes génétiques et les algorithmes de colonies de fourmis sont inspirés par la nature [EK95].

-Pas inspiré par la nature L'algorithme de recherche tabou et l'algorithme de recherche locale itérative ne sont pas inspirés par la nature. Cependant, certains chercheurs, notamment [BR03], trouvent que cette classification n'est pas significative pour les raisons suivantes :

- De nombreux algorithmes hybrides récents ne conviennent pas aux deux classes (les deux classes en même temps).
- Difficulté à classer les méthodes dans certaines situations, comme l'utilisation de la mémoire dans la recherche tabou qui n'est pas d'inspiration naturelle.

-Basées sur la population des solutions

Les méta-heuristiques ont un grand nombre de solutions, qui incluent des algorithmes évolutionnaires, recherche de dispersion, optimisation d'essaims de particules.

-Basées sur la solution unique

Les métaheuristiques à solution unique où on trouve Recherche Locale, Recherche Tabou et Recuit Simulé.

-Fonction objectif dynamique et fonction objectif statique

Étant donné un problème d'optimisation impliquant la minimisation d'une fonction sur l'espace des solutions, nous pouvons adopter une classification alternative où les méthodes qui agissent directement sont appelées statiques, tandis que les autres méthodes sont dites dynamiques.

-Utilisation de la mémoire à long terme et à court terme

On peut aussi profiter de la propriété d'utiliser l'historique pour la classification méta-heuristique, les algorithmes sans mémoire étant des processus de Markov, car l'action à effectuer est complètement déterminée par la situation actuelle : L'accumu-

lation de paramètres synthétiques dans l'étude, qui sont appelés métaheuristiques à mémoire à court terme. L'autre est pour l'heuristique de la mémoire à long terme et garde une trace des mouvements récents [Bou17].

2.7 Les Méthodes Bio-Inspirés

Aujourd'hui, il est bien mérité de démontrer leur supériorité grâce à la modélisation de l'intelligence biologique résultant en ce que nous appelons des algorithmes méta-heuristiques d'inspiration biologique qui recherchent constamment la meilleure solution à une variété de problèmes complexes.

Ces algorithmes comprennent les réseaux de neurones artificiels, l'informatique évolutive, l'intelligence collective, les systèmes immunitaires artificiels et les systèmes d'organes humains. Ils fournissent des résultats performants dans les études expérimentales et peuvent souvent être appliqués sans connaissance du problème, ce qui les rend idéaux pour une variété de situations. D'autre part, il est souvent difficile de comprendre pourquoi ils travaillent dans un contexte particulier [BOU17a].

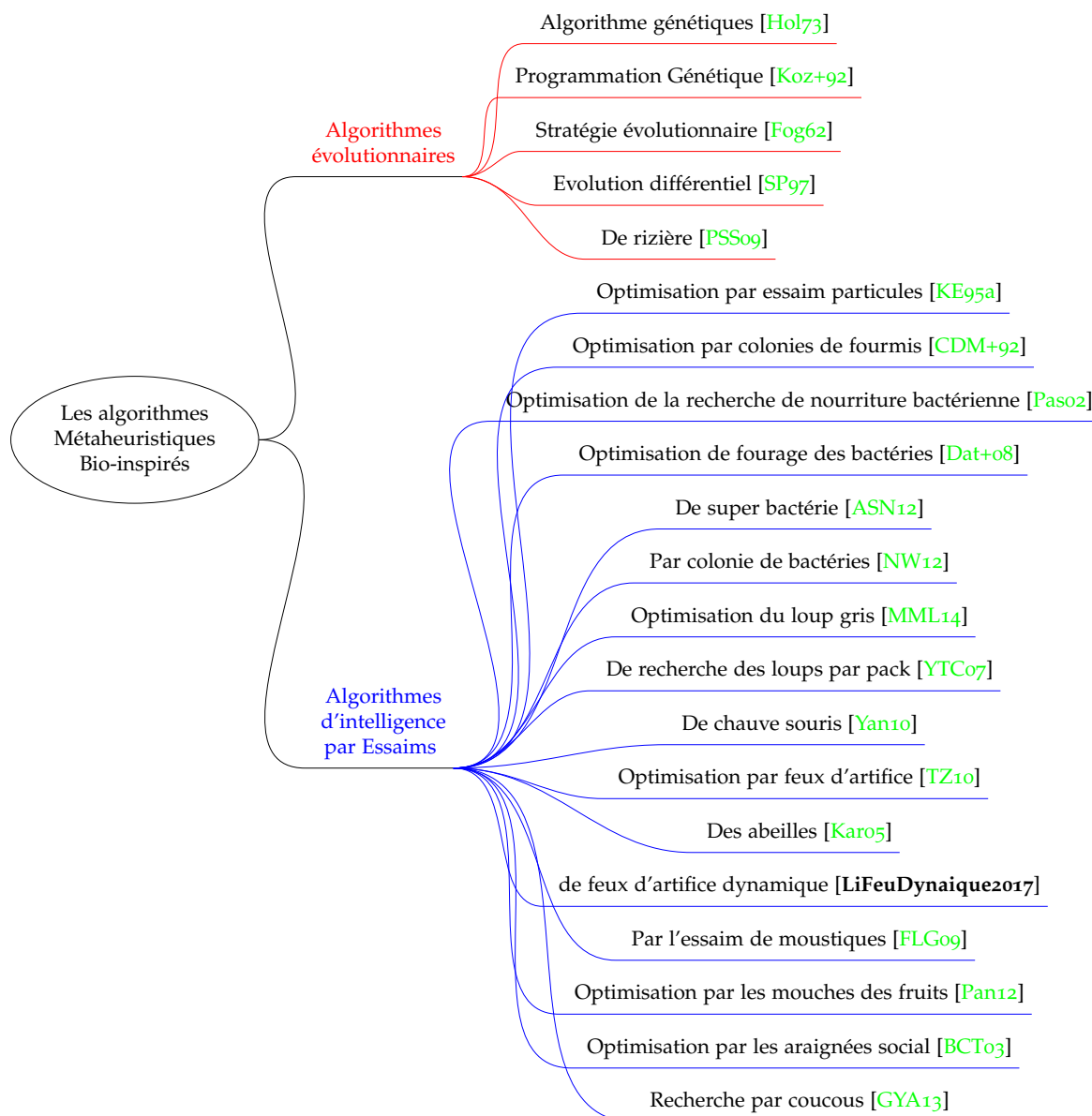


Figure 2.1 – Taxonomie des algorithmes méta-heuristiques bio-inspirés

2.8 Le biomimétisme

La nature est une puissante source d'inspiration pour les prospecteurs car elle offre des phénomènes incroyablement divers, dynamiques, puissants, complexes et fascinants. elle trouve toujours des solutions aux problèmes et maintient un équilibre parfait entre ses composants. Les organismes ont réussi à survivre sur terre au cours des quatre derniers milliards d'années. La principale raison de ce succès est, bien sûr,

leur capacité à s'adapter à l'évolution des circonstances défavorables. Ils ont une incroyable capacité à apprendre de situations inconnues, à adapter leur comportement et leur forme aux changements environnementaux, à se reproduire et à se réparer. Au cours des dernières décennies, de nombreux efforts de recherche se sont concentrés sur ce domaine, et une nouvelle ère s'est ouverte dans les algorithmes inspirés de la nature (bioniques).

L'inspiration biologique est l'imitation de processus biologiques, tels que des métaphores ou des phénomènes naturels, afin de développer de nouveaux algorithmes. Ce domaine d'étude combine vaguement un ensemble de sous-domaines liés aux thèmes du connexionnisme, du comportement social et de l'émergence. Il est souvent étroitement lié au domaine de l'intelligence artificielle, car nombreuses de ses activités sont liées à l'apprentissage automatique. Les algorithmes bioniques se sont révélés plus robustes et adaptables que les algorithmes traditionnels. Ainsi, les notions de robustesse, d'émergence, d'auto-organisation, d'adaptabilité, de réactivité et de distribution sont le fondement de ces algorithmes, voire une partie de leur fondement. La première étape dans la construction d'un algorithme biomimétique consiste à construire des composants qui imitent le comportement de leurs homologues biologiques. Ces composants essaient ensuite d'atteindre les objectifs globaux qui leur sont fixés [BS11]

2.9 Algorithmes évolutionnaires (AEs)

La théorie d'évolution

L'évolution des AEs résout les problèmes NP-difficiles en imitant la métaphore de l'évolution biologique proposée par Darwin dans son livre de 1860 sur l'origine des espèces par sélection naturelle, ou la lutte pour l'existence naturelle. Ils disposent en entrée d'un ensemble de solutions initiales qui évolueront d'une génération à l'autre, suivant un ensemble de règles d'élevage jusqu'à ce qu'une solution optimale soit atteinte [BHS97]

Algorithme génétique

En 1975, J. Holland a développé la première version de l'algorithme génétique (GA), qui a été amélioré par D.E. Goldberg en 1989. Ce sont les algorithmes évolutionnaires les plus connus.

En fait, le professeur J.H Holland de l'Université du Michigan a lancé un projet de recherche sur les systèmes adaptatifs avec ses doctorants en 1975 [Hol73], et leur idée principale était d'imiter et de simuler les principes de Darwin (choix, intersection, mutations, chromosomes, gènes). Le projet a finalement jeté les bases de l'algorithme génétique et développé les étapes de l'algorithme et ses principes de codage.

En raison des performances limitées des ordinateurs, les algorithmes génétiques n'ont pas été envisagés pour des problèmes pratiques à grande échelle. David E. Goldberg [Gol89] a repris le travail, très basique au début, et il a amélioré l'encodage en utilisant une représentation du génotype, utilisant d'abord l'encodage binaire pour les vecteurs, puis il a encodé les caractères dans une chaîne. Ce travail a été vérifié par le mathématicien Koza. Bien que les algorithmes génétiques soient théoriquement utilisés pour l'optimisation stochastique, ils peuvent également être utilisés et adaptés pour d'autres tâches plus ou moins complexes, comme l'appren-

tissage automatique.

Aujourd'hui, les algorithmes génétiques occupent une place importante dans de nombreux domaines tels que le milieu universitaire et l'industrie. [SVA05] Les algorithmes génétiques s'inspirent de la génétique classique et utilisent le même vocabulaire.

Avant d'entrer dans le fonctionnement des algorithmes génétiques, il faut définir les termes associés à la génétique utilisés pour décrire les algorithmes génétiques :

1. **Gène** : C'est une séquence de bases azotées qui contient le code d'une protéine donnée (adénine (A), cytosine (C), guanine (G) et thymine (T)). Dans un algorithme génétique, un gène est une suite de symboles qui codent la valeur d'une variable selon un codage (représentation des données).
2. **Un chromosome** : est la séquence de gène d'un nombre fini sélectionnée en fonction du problème à résoudre.
3. **Individus** : Les individus sont généralement représentés par un chromosome, représentant une solution potentielle.
4. **Population** : Une collection (d'individus) de solutions potentielles.
5. **Génération** : Un ensemble d'opérations qui peuvent être transmises d'une population à une autre. Ces opérations sont : la sélection d'individus de la population actuelle, l'application d'opérateurs génétiques, l'évaluation d'individus de la nouvelle population.
6. **La fitness ou fonction d'évaluation** : C'est une fonction d'évaluation d'un individu, c'est le cœur du processus d'optimisation de l'algorithme génétique, qui permet à l'algorithme génétique de s'adapter à un problème donné.

Le schéma suivant montre de manière abstraite le principe et le fonctionnement de l'algorithme génétique.

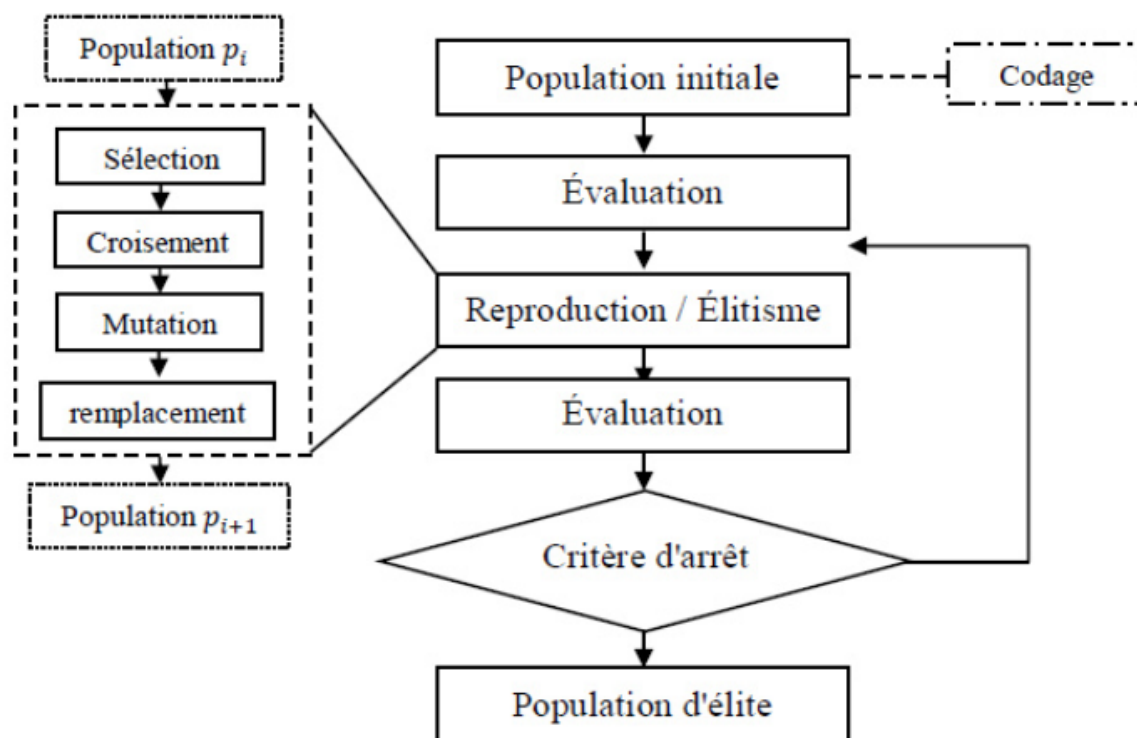


Figure 2.2 – Organigramme de l'algorithme génétique [Gol89]

2.10 Algorithmes d'Intelligence en Essaims (AIEs)

L'intelligence en essaim (SI : Swarm Intelligence) est le résultat de plusieurs efforts de modélisation mathématique et informatique de phénomènes biologiques rencontrés en éthologie. Selon les auteurs de [Bon+99], toute tentative de conception d'algorithmes distribués ou de dispositifs de résolution de problèmes inspirés du comportement collectif d'insectes sociaux ou d'autres sociétés animales relève du cadre de l'intelligence en essaim. Selon Gerardo Beni et Jing Wang, l'intelligence en essaim est le comportement collectif décentralisé et auto-organisé des systèmes naturels ou artificiels.

Les sources d'inspiration viennent souvent de la nature, en particulier des systèmes biologiques, et de l'observation du comportement des insectes grégaires, les individus sont moins complexes et simples compte tenu de la richesse de leurs

interactions et des environnements à travers lesquels ils sont modélisés.

L'ensemble des algorithmes d'intelligence en essaim est basé sur un ensemble simple d'agents qui interagissent entre eux localement d'une part, et leur environnement d'autre part, sans aucun contrôle centralisé pour guider le comportement des agents individuels. Seules les interactions locales entre agents conduisent à une sorte de l'émergence d'un comportement collectif global auto-organisé. Notez que les capacités individuelles des agents sont très limitées et qu'ils peuvent effectuer conjointement certaines tâches ou des tâches simples pour survivre. Les quatre principes régissant l'intelligence en essaim :

1. **Feedback positif** : il renforce les meilleurs choix dans le système,
2. **Rétroaction négative** : elle vous permet d'ignorer et de supprimer les mauvais choix dans le système,
3. **L'aspect stochastique** : il permet de bien explorer l'espace des solutions de manière indépendante de la qualité, en privilégiant le principe de diversification,
4. **L'interaction multiple** qui permet la construction des meilleures solutions et choix.

Optimisation par colonies des fourmis AOCF

L'AOCF a été introduit par l'Italien Marco Dorigo vers les années 1990 dans sa thèse de doctorat sur l'optimisation et les algorithmes naturels [DOR92]. Cet algorithme a été initialement proposé pour résoudre des problèmes d'optimisation discrets, comme le problème du voyageur de commerce dans [Han+17]. Plusieurs versions d'AOCF ont été proposées dans la littérature en tant qu'algorithme de système de fourmis.

L'idée principale d'AOCF est de modéliser le problème à résoudre comme un problème de recherche de chemins optimaux dans un graphe pondéré, et d'utiliser des fourmis artificielles pour trouver des chemins de qualité parmi tous les chemins disponibles.

Processus général de l'AOCF

L'architecture globale de l'AOCF est illustrée à la figure 2.3, et chaque étape de son fonctionnement est détaillée comme suit :

1. **Codage** : L'espace des solutions des problèmes est représenté sous la forme d'un graphe, où chaque solution est un chemin.
2. **Création de solutions heuristiques** : Cette étape assure que N solutions initiales sont créées en utilisant des heuristiques basées sur des ressources externes. elle génère des valeurs pour les arcs reliant les nœuds. Initialement, les trajectoires de phéromones de toutes les trajectoires du graphe sont initialisées avec des valeurs constantes.
3. **Construction de la solution** : les fourmis artificielles se déplacent de manière itérative à travers les nœuds adjacents du graphe selon des règles probabilistes, permettant à chaque fourmi de sélectionner des composants (nœuds) de manière séquentielle, utilisant la solution pour l'intensité de phéromone de chaque piste.
4. **Évaluation** : chaque solution construite est évaluée à l'aide d'une fonction fitness à déterminer la quantité de phéromone à déposer sur chaque piste.
5. **Mise à jour de la phéromone** : Après avoir construit la solution complète, mettez à jour la trajectoire des phéromones. Cette étape comprend également l'évaporation des phéromones (réduction des traces de phéromones) pour aider les fourmis à oublier les mauvaises solutions précédemment instillées. La valeur de la trajectoire des phéromones d'un arc visité augmente inversement avec son coût de visite.

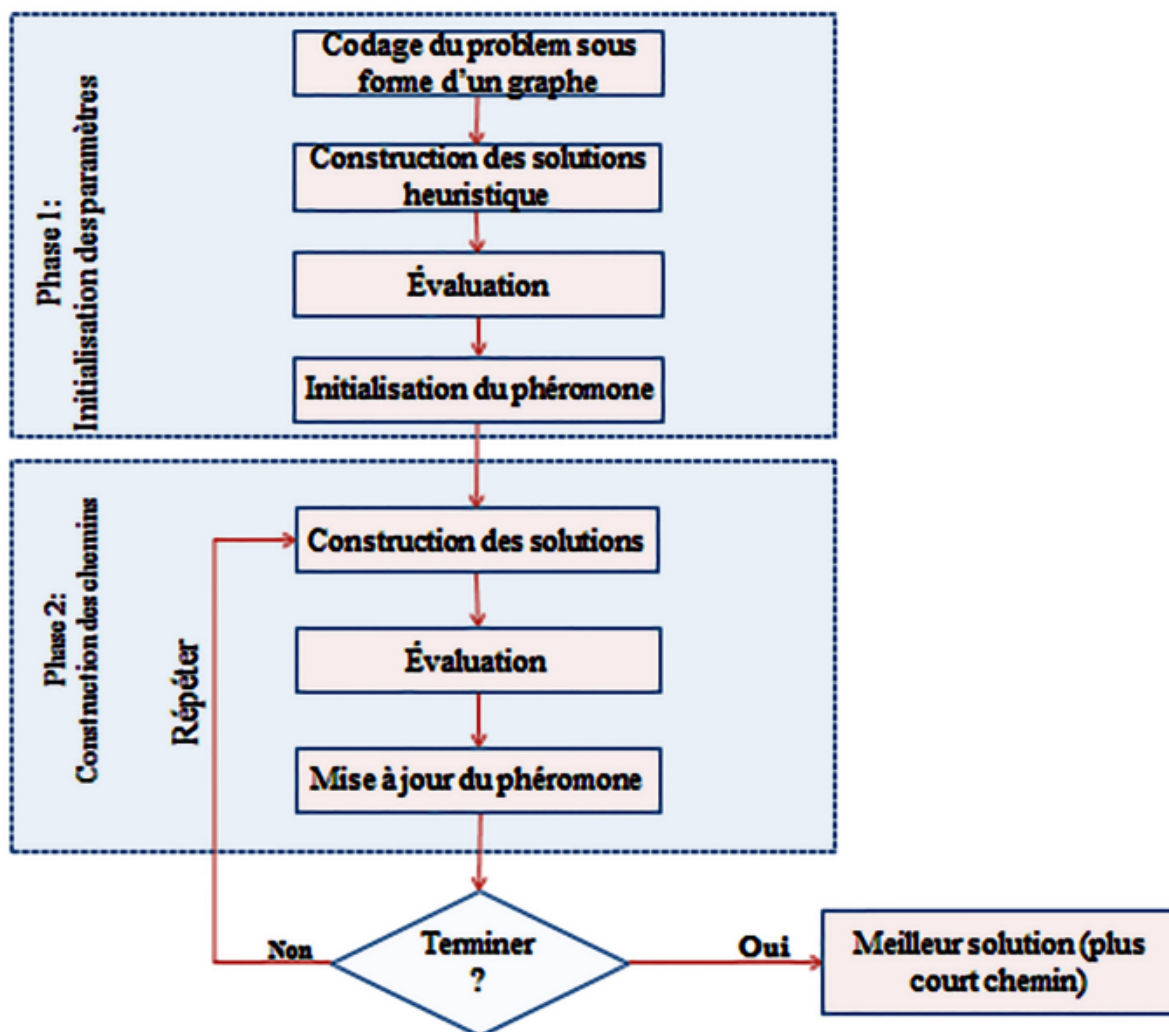


Figure 2.3 – Architecture générale de l'AOCF [DG97]

Optimisation par Essaim Particulaire PSO

Principe général L'optimisation par essaim particulaire (OEP), ou Particle Swarm Optimization (PSO) en anglais, est un algorithme évolutionnaire qui utilise un ensemble de solutions candidates pour développer une solution optimale à un problème. Cet algorithme est proposé par Russel Eberhart (ingénieur en électricité) et James Kennedy (socio-psychologue) en 1995 [KE95b]. Il s'est d'abord inspiré du monde vivant, et plus précisément des comportements sociaux qui évoluaient dans les groupes d'animaux, comme les bancs de poissons et d'oiseaux. En effet, on observe chez ces animaux des dynamiques motrices relativement complexes, alors que

chaque individu a une "intelligence" limitée et une connaissance partielle de sa situation dans le groupe. Les informations locales et la mémoire de chacun sont utilisées pour décider où se déplacer. Des règles simples comme "rester proche des autres", "se déplacer dans le même sens" ou "se déplacer à la même vitesse" est suffisant pour maintenir la cohésion du groupe et permettre l'exécution de comportements collectifs complexes et adaptatifs.

Un essaim de particules correspond à un groupe d'agents simples appelés particules. Chaque particule est considérée comme une solution au problème, elle a une position (vecteur solution) et une vitesse. De plus, chaque particule possède une mémoire qui mémorise ses meilleures performances (position et valeur) et les meilleures performances obtenues par les particules "voisines" (informationnelles) : Chaque particule a un ensemble d'informateurs, historiquement connu comme son voisinage. Un essaim de particules, qui sont des solutions potentielles à un problème d'optimisation, « survole » l'espace de recherche, à la recherche d'un optimum global. Le mouvement des particules est affecté par les trois composants suivants :

1. **Une composante inertielle** : les particules ont tendance à suivre leur direction de mouvement actuelle
2. **Une Composante Cognitive** : La particule a tendance à se déplacer vers la meilleure position par laquelle elle est passée
3. **Une Composante sociale** : les particules ont tendance à s'appuyer sur l'expérience de leur propre espèce et, par conséquent, à se déplacer vers les meilleurs endroits que leurs voisins ont déjà atteints.

La stratégie de mouvement des particules est illustrée à la figure [2.4](#)

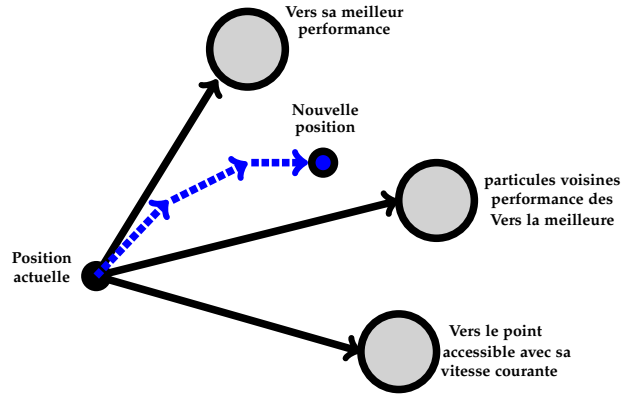


Figure 2.4 – Déplacement d'une particule

Formalisation

Dans un espace de recherche de dimension D , chaque particule i est représentée par les caractéristiques suivantes :

— Vecteur de position : $\vec{x}_i = (x_{1i}, x_{2i}, \dots, x_{iD})$

— Vecteur de vitesse(ou vitesse) : $\vec{v}_i = (v_{1i}, v_{2i}, \dots, v_{iD})$

— Vecteur de meilleure position visitée par i :

$\overrightarrow{Pbest}_i = (pbest_{1i}, pbest_{2i}, \dots, pbest_{iD})$ dans le temps $t + 1$ est calculée par l'équation suivante :

$$\overrightarrow{Pbest}_i(t+1) = \begin{cases} \overrightarrow{Pbest}_i(t), & \text{si } f(\vec{x}_i(t+1)) \geq \overrightarrow{Pbest}_i(t) \\ \vec{x}_i(t+1), & \text{sinon.} \end{cases} \quad (2.1)$$

— La meilleure position du voisinage (toute la population) de la particule i , $\overrightarrow{Gbest}_i = (gbest_{1i}, gbest_{2i}, \dots, gbest_{iD})$ obtenue par :

$$\overrightarrow{Gbest}_i(t+1) = \begin{cases} \overrightarrow{Pbest}_i(t) = \arg \text{Min}_{\overrightarrow{Pbest}_i} f(\overrightarrow{Pbest}_i(t+1)) \\ 1 \leq i \leq P \end{cases} \quad (2.2)$$

Le vecteur de vitesse et le vecteur de position sont mis à jour à partir des équations (2.3) et (2.4) respectivement :

$$v_{i,j}^{t+1} = wv_{i,j}^t + C_1R1_{i,j}^t[Pbest_{i,j}^t - x_{i,j}^t] + C_2R2_{i,j}^t[gbest_{i,j}^t - x_{i,j}^t], \quad j \in \{1, 2, \dots, D\} \quad (2.3)$$

$$x_{i,j}^{t+1} = x_{i,j}^t + v_{i,j}^{t+1} \quad j \in \{1, 2, \dots, D\} \quad (2.4)$$

où

- $v_{i,j}^t$ représente la vitesse de la particule i dans la dimension j à l'instant t .
- $R1$ et $R2$ sont des nombres aléatoires compris dans l'intervalle $[0, 1]$
- w est une constante, appelée coefficient d'inertie.
- C_1 et C_2 sont des constantes, appelées coefficients d'accélération.

Le déplacement de la particule est influencé par les trois composantes citées ci-dessous :

1. La composante d'inertie $wv_{i,j}^t$ introduite par Shi et Eberhart [SO98], commande l'impact de la vitesse précédente (tendance à suivre la même direction) sur le déplacement futur, une grande valeur de w favorise l'exploration, tandis qu'une petite valeur favorise l'exploitation.
2. La composante cognitive $C_1R1_{i,j}^t[Pbest_{i,j}^t - x_{i,j}^t]$ qui correspond à l'attraction linéaire vers la meilleure position déjà trouvée par la particule i .
3. La composante sociale $C_2R2_{i,j}^t[gbest_{i,j}^t - x_{i,j}^t]$ qui correspond à l'attraction linéaire vers la meilleure position trouvée par le voisinage de la particule.

Algorithme 1 : PSO

```

début
  initialiser aléatoirement P particules : positions et vitesses;
  tant que (critère d'arrêt n'est pas satisfait) faire
    pour  $i \leftarrow 1$  à P faire
      Evaluer la fitness  $f(\vec{x}_i)$ ;
      Mettre à jour  $\vec{Pbest}_i$  et  $\vec{Gbest}_i$  selon (Eq.2.1) et (Eq.2.2);
      Déplacer les particules selon (Eq.2.3) et (Eq.2.4);
      Limiter les vitesses dans l'intervalle  $[-v_{max}, +v_{max}]$ ;
    fin
  fin
fin

```

Paramétrés de l'algorithme

Nombre de particules P Le nombre de particules affectées à la solution d'un problème d'optimisation dépend principalement de la taille de l'espace de recherche. Il n'y a pas de règles pour déterminer ce paramètre, faire beaucoup de tests vous donnera la valeur correcte. En général, l'augmentation du nombre de particules augmente la diversité et réduit ainsi la probabilité de se coincer dans les minima locaux [Omro4]. Selon Shi et Eberhart [SO98], la valeur de P ne doit pas être trop petite.

Étape d'initialisation Les positions des particules X sont initialisées aléatoirement, mais couvrent tout l'espace de recherche. Notez que si la position de la particule n'est pas initialisée dans tout l'espace de recherche, si la position de la particule est en dehors de la région initiale, l'algorithme PSO a du mal à trouver la valeur optimale.

Facteur d'inertie (w) Le facteur d'inertie (**w**) introduit par Shi et Eberhart [SO98] contrôle l'effet de l'orientation actuelle de la particule sur le déplacement

futur. Ce paramètre est introduit pour réaliser un compromis entre la recherche locale (exploit) et la recherche globale (explore). Des valeurs plus grandes du facteur d'inertie entraînent une grande exploration de l'espace de recherche, tandis que des valeurs plus petites concentrent localement la recherche sur un petit espace [Dor12]. Une étude menée dans [ESoo] a montré que la convergence pour $w \in [0.8, 1.2]$ est meilleure. Au-delà de la valeur 1.2, l'algorithme tend à converger avec quelques difficultés. Les auteurs proposent également un facteur d'inertie dynamique décroissant avec le temps, qui commence à 0.9 et décroît linéairement jusqu'à 0.4 selon la formule suivante :

$$w(t) = \frac{iter_{max} - t}{iter_{max}}(w_{max} - w_{min}) + w_{min} \quad (2.5)$$

où

- t est l'itération courante,
- $iter_{max}$ est le nombre maximal d'itérations,
- w_{max} et w_{min} désignent respectivement les valeurs maximales et minimales du facteur w .

Notez que la stratégie d'utilisation de w dynamique améliore considérablement les performances de PSO pour plusieurs problèmes d'optimisation [Dor12].

Confinement des particules : lors du déplacement de particules, celles-ci peuvent quitter l'espace de recherche. Pour éviter cette "explosion", une méthode appelée contrainte (velocity Clamping) [RDRE05] est utilisée pour contraindre les vitesses dans l'intervalle $[-vmax, +vmax]$ et permettre de ramener les particules sorties de l'espace de recherche à l'intérieur de celles-ci, permettant un compromis efficace entre exploitation et exploration. La valeur de $Vmax$ dépend généralement de la taille du problème.

Coefficient C_1 , C_2 : la combinaison des paramètres w , C_1 et C_2 vous permet d'ajuster l'équilibre entre les recherches locales et globales. Dans [CKo2], les auteurs

ont choisi les valeurs expérimentales suivantes : $w = 0.7298$, $C1 = C2 = 1.496$. Cependant, d'autres valeurs ont été utilisées dans la littérature.

Critères d'arrêt Plusieurs critères d'arrêt peuvent être considérés pour terminer l'algorithme PSO :

- **Nombre maximal d'itérations** : Si le nombre maximal d'itérations est petit, l'algorithme PSO générera une solution non optimale.
- **Convergence vers la solution recherchée** : On utilise souvent un seuil c , si la valeur de c est trop grande, l'algorithme se terminera par une mauvaise solution, si elle est trop petite, la recherche se terminera lentement.
- **Aucune amélioration ou Nombre d'itérations** : Il existe plusieurs façons de mesurer l'amélioration de la recherche, comme un changement de vitesse proche de zéro ou un changement trop faible de la position des particules.

Avantages et inconvénients de PSO Les algorithmes PSO sont similaires aux algorithmes évolutionnaires en ce que les deux méthodes sont basées sur la population et pour chaque itération, les individus se déplacent dans l'espace de recherche pour améliorer leur fonction objective. Dans les algorithmes génétiques, le mécanisme de sélection prend en charge la survie du plus apte, et le chromosome avec une meilleure fonction objective existe toujours dans la génération suivante, ce qui est le concept central de tous les algorithmes évolutionnaires [ES98]. Cependant, dans PSO, il n'y a plus de mécanisme de sélection, et une particule actuellement médiocre peut devenir la meilleure dans le futur. Les particules recherchent des optima globaux dans la coopération, pas dans la compétition. L'algorithme PSO est influencé par le comportement social plutôt que par la survie du plus apte [Omro4]. Une autre différence importante est que dans PSO, chacun profite de son histoire, un mécanisme qui n'existe pas dans les algorithmes évolutionnaires.

De par sa rapidité et sa robustesse, l'algorithme PSO a été appliqué à un large éventail de problèmes d'optimisation (traitement d'images, industrie, etc.). Son avantage est qu'il est simple à mettre en œuvre et nécessite peu de paramètres à régler.

Son adaptation aux traitements parallèles et simultanés a été notée [Ste14]. Cependant, PSO présente également certains inconvénients : le problème de la convergence prématurée conduit à des optima locaux ; le problème de la sélection des paramètres sur les performances de l'algorithme [Omro4]. Dans la littérature, des travaux ont été proposés pour résoudre ces problèmes et améliorer les performances des PSO. Une grande partie de ces travaux est basée sur l'hybridation avec d'autres méthodes. Pour une revue plus complète de ces travaux, le lecteur est référé à [SM09].

2.11 Conclusion

Les algorithmes méta-heuristiques bio-inspirés sont une nouvelle révolution en informatique. Le champ de ce domaine est large car les problèmes de l'informatique ne sont qu'un sous-ensemble par rapport à la nature. Malheureusement, Il faut se rappeler que l'imitation des mécanismes biologiques ne confère pas nécessairement des avantages, soit parce que la technologie ne peut pas correspondre à la biologie, soit parce que la cible visée peut différer du mécanisme de la biologie. La bonne pratique des simulations biologiques oblige les scientifiques à s'efforcer d'extraire les principes de l'intelligence biologique des données et des théories fournies par les biologistes afin de traduire ces principes en algorithmes et techniques fonctionnelles.

Chapitre 3

L'approche proposée et Conceptions

3.1 Introduction

Ce chapitre présente la conception de notre proposition et discute des choix d'implémentation. Il commence par donner un aperçu abstrait du système. Ensuite, il passe en revue l'architecture proprement dite, dans laquelle, tout d'abord, les éléments fondamentaux que sont le véhicule et la demande sont définis, puis les principaux défis rencontrés, tels que les problèmes de routage et de relocalisation, sont présentés et la façon dont le modèle les traite est expliquée.

3.2 Exigences

Les villes modernes évoluent rapidement, se développent jour après jour et transforment leurs réseaux internes. Une solution capable de faire face à ces changements et de répondre aux besoins d'une ville est nécessaire. Le système a été conçu pour s'adapter aux changements de la ville. Pour y parvenir, on utilise un système de coordination abstrait qui n'est pas lié à un modèle.

Au niveau de la flotte, on suppose que chaque voiture mis en œuvre est capable de fournir un service continu. Ainsi, les véhicules sont sans défaillance et ne peuvent pas avoir d'accident. Pour que la solution soit conforme à la propriété citée, une approche centralisée est recommandée. De cette manière, le coordinateur central serait nécessaire. Pour assurer cette dynamique, deux sous-exigences sont nécessaires :

- **Représentation abstraite du véhicule**

Représenter chaque véhicule de la même manière : Par exemple, deux véhicules ayant un nombre de sièges différent doivent être représentés de la même manière.

- **Pas de partage d'informations**

Chaque véhicule doit être indépendant et ne pas connaître l'état des autres véhicules qui agissent en même temps que lui. Ainsi, un véhicule pour agir dans la zone, n'a pas besoin d'information sur la taille de la flotte ou la position

des autres véhicules qui agissent dans l'environnement au même moment.

3.3 Conception du système

Cette section donne un aperçu de la conception abstraite du système. Pour commencer, nous présentons la conception du véhicule avec les informations clés sur les actions [MMA22].

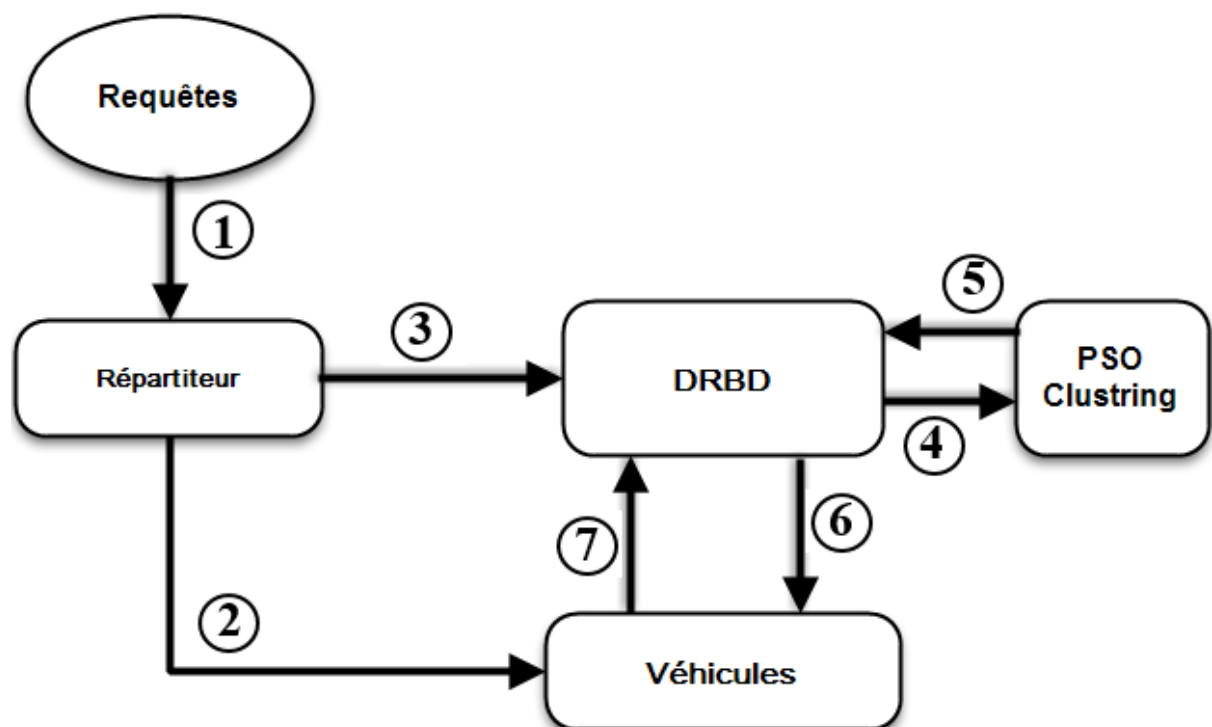


Figure 3.1 – Schéma d'architecture de notre proposition.

- ① L'arrivée des demandes ou bien les passagers qui veulent se déplacer,
- ② Faire la correspondance entre les demandes et voitures dans le cas où il existe une correspondance,
- ③ dans le cas où pas de correspondance, la demande reste en instance dans le module DRBD,

- ④ envoyer l'ensemble des demandes en instance et qui sont valide au module PSO clustering.
- ⑤ retourne au module DRBD les clusters et leurs centroids ainsi que les probabilité,
- ⑥ affecter les voitures vides vers les centroids des clusters
- ⑦ demander la relocalisation en cas de la voiture est libre.

3.3.1 Conception des véhicules

Le seul attribut qui compte pour le trajet est le nombre de sièges libres disponibles pour transporter des passagers, mais tous les véhicules n'ont pas le même nombre de sièges. Une stratégie permettant de traiter tous les véhicules de la même manière, indépendamment du nombre de passagers qu'ils peuvent transporter, a été définie. Cette abstraction peut sembler triviale mais elle fait la différence lors de la conception.

La conception d'un véhicule nécessite de définir des états et les transactions de l'un à l'autre. L'état est défini en fonction de l'état interne du véhicule et de sa perception. Les actions permettent aux véhicules de passer d'un état à un autre. Ci-dessous sont décrits ces trois points clés.

— État interne

L'état interne exprime la position géospatiale réelle du véhicule, sa destination et le nombre de places libres disponibles à un moment donné.

La destination peut être nulle si le véhicule ne sert aucun passager. En revanche, si le véhicule fait du covoiturage, sa destination correspond à la destination de la demande la plus rapide qui peut être servie. La position est mise à jour chaque fois que le véhicule atteint un nouveau client ou se rend à sa destination. La position et la destination sont conformes au système de coordination des cartes bidimensionnelles.

— Perception

La perception P est composée de plusieurs demandes. C'est un ensemble et ses dimensions dépendent de la puissance des véhicules. Plus ils sont puissants, plus ils peuvent percevoir de requêtes. L'ensemble de perception standard est limité à trois demandes, les trois plus proches du véhicule. Pour chaque demande, les informations prises en compte sont l'origine, la destination et le nombre de passagers. Ci-dessous est présentée la définition générale de la perception du véhicule.

$$P : \{S_{r_1}, S_{r_2}, S_{r_i}, \dots, S_{r_n}\}$$

Ou,

$$S_{r_i} : [r_{Pos}^i, r_{Dest}^i, r_{passagers}^i]$$
(3.1)

tel que,

- P indique l'ensemble des perceptions pour un véhicule.
- S_{r_i} indique l'état de la i -ème requête perçue par le véhicule.

— Actions

Par le biais de ces actions, le véhicule peut prendre des passagers pour faire du covoiturage, s'il est déjà en train de répondre à une ou plusieurs demandes entre-temps, ou peut se diriger vers sa destination pour répondre à une demande en cours de traitement suivant l'algorithme 2. En outre, le véhicule peut choisir de rester stationné pendant une période prédéfinie. Pendant la phase de prise en charge, le véhicule évalue le temps nécessaire pour atteindre l'origine d'une demande. Ce temps doit être conforme au temps d'attente maximum défini pour les demandes, sinon il n'est pas servi, car le décalage entre la génération de la demande et le moment où les clients seraient pris en charge n'est pas souhaitable. Les actions sont définies en détail dans la section 3.4.

Algorithme 2 : contrôler véhicules**Paramètres :** v véhicule et R les requêtes**Résultat :** Meilleure action pour v **Fonction Action (v)**

```

si ( $v$  est vide)      // Aucune requête  $r$  proche de  $v$  à servir
alors
| Relocaliser ( $v$ );
sinon
|
|           //  $v$  se dirige vers une destination
| si (Existe ( $r$ ) correspondant à ( $v$ .Action) et existe des sièges suffisants) alors
| | V.servir( $r$ ) ;           // Faire le covoiturage
| fin
fin

```

Fin Fonction**3.3.2 Les demandes ou les requêtes**

Demande ou requête est la structure qui contient les informations relatives à une demande de transport. Chaque instance est caractérisée par deux positions, origine et destination, son heure de départ et par le nombre de passagers (voir la table 3.1).

N	Ptime	Olng	Olat	Dlng	Dlat	Npass
1	2015/07/07 – 07 : 00 : 01	-74.015488	40.715603	-74.010475	40.721542	1
2	2015/07/07 – 07 : 00 : 05	-73.985352	40.722023	-73.999344	40.733822	1
3	2015/07/07 – 07 : 00 : 11	-73.996910	40.725388	-74.011169	40.709332	2

Table 3.1 – La structure de la base de données

où **Ptime** : c'est le temps d'arrivée d'un passager, **Olng** : longitude du point de départ (origine), **Olat** : latitude du point de départ (origine), **Dlng** : longitude du point d'arrivée (Destination), **Dlat** : latitude du point d'arrivée (Destination), **Npass** : le nombre des passagers par demande.

Chaque demande peut être assignée au maximum à un véhicule. Une fois qu'il prend en charge une demande, les informations sur le véhicule et l'heure de

prise en charge sont stockées. Enfin, lorsqu'elle est satisfaite, l'heure de dépôt associée est indiquée dans la demande. Les requêtes doivent être prises en charge dans un délai prédéfini après l'heure de dépôt de la demande, sinon elles sont rejetées et considérées comme non servies. Comme le montre la figure 3.2, seules les demandes valides sont prises en compte, pas les précédentes servies ou des demandes rejetées (estimées ou prévues). Ce contrôle est effectué par le répartiteur de demandes. Il peut être considéré comme une unité centralisée qui gère les demandes et son rôle principal est d'organiser les demandes, en les laissant disponibles quand c'est approprié ou en les classant comme rejetées quand il n'est plus possible de les satisfaire.

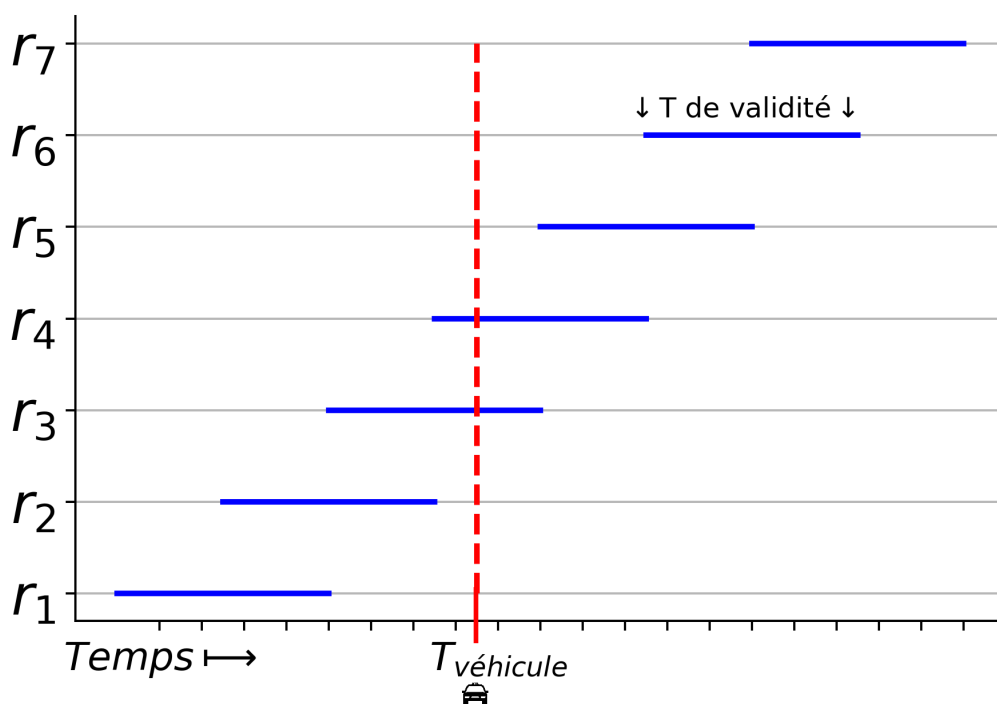


Figure 3.2 – Requêtes actives à l'instant t_v , seules (r_3 ; r_4) sont prises en compte lors du rééquilibrage.

3.4 Définition des actions

Les actions disponibles sont représentées par l'ensemble A . Sa cardinalité dépend de la taille de la perception du véhicule, du nombre de demandes que chaque

véhicule implique dans le processus d'évaluation pour la prise en charge d'une nouvelle demande.

$$A : \{0, 1, 2, \dots, |A| - 1\} \quad (3.2)$$

Les comportements associés à chaque action de A sont présentés ci-dessous,

— **0**

Il correspond à rester stationné. Avec cette action le véhicule ne fera rien pendant une période de temps prédéfinie.

— **1**

Si le véhicule a des demandes en cours de traitement, cette action correspond à l'accomplissement de la dernière demande dans sa file d'attente.

— **2 ou plus**

Signifie que le véhicule doit se diriger vers l'origine de l'une des demandes disponibles dans sa perception pour la prendre en charge. Étant donné l'action a , avec $a > 2$, cela signifie que le véhicule doit charger les passagers de $(a - 1)$ requête à partir de sa perception.

Algorithme 3 : État et la perception de véhicule

début

Résultats : vecteur représente l'état interne et la perception de V

Fonction DonnerEtat(V)

$V.info \leftarrow [v.position, v.destination, v.siegesVide()];$

$RequeteInfo \leftarrow [];$

RequêtesProches= TrouverRequetesProches(); // Algorithme 4

pour chaque ($requete\ r \in RequetesProches$) **faire**

| $RequeteInfo.ajouter(r.origine, r.destination, r.NbrePassagers);$

fin

Fin Fonction

fin

L'identification des demandes les plus proches pour un véhicule est présentée dans l'algorithme 3 tandis que la définition de la proximité, pour une demande et un véhicule, est fournie ci-dessous. Une requête r est proche d'un véhicule v si elle est

conforme à :

1. Le véhicule v doit avoir suffisamment de sièges libres pour charger les passagers de r .
2. Le temps de génération de r doit être inférieur ou égal au temps du véhicule actuel. En effet, il est impossible pour v de connaître ou de servir une demande future qui n'est pas encore disponible.
3. Le temps d'attente total, défini comme le délai entre la génération de la demande et la prise en charge du client, doit être inférieur au maximum autorisé.

Algorithme 4 : Requêtes Proches

début

Résultats : vecteur avec les requêtes proches de V

TrouverRequetesProches(V)

si (v n'a pas de place libre) **alors**
 | retourner

sinon
 | ;

fin

$R\text{TempsNcessaire} \leftarrow []$;

pour chaque ($requete\ r \in RequetesDisponibles$) **faire**

| **si** ($v.NbreSiegesLibre() \geq r.NbrePassagers$) ET ($v.clock \geq$

| ($r.TempsDebut$) **alors**

| | $TempsPourServir \leftarrow TempsNcessaire(v.position, r.origine)$

| | $DiffTemp \leftarrow v.clock - r.TempsDebut$;

| | **si** ($(DiffTemp + TempsPourServir) \leq TempsMaxAttendre$) **alors**

| | | $R\text{TempsNcessaire.ajouter}(r, TempsNcessaire)$;

| | **fin**

| **fin**

fin

$R\text{TempsNcessaire.Trier}()$; // Trier par ordre de temps nécessaires pour la
 prise en charge

Retourner $R\text{TempsNcessaire}$

fin

3.5 Routage

L'un des principaux sous-problèmes du covoiturage est le routage. Il affecte le résultat de la solution, pour plusieurs raisons. Tout d'abord, pour assurer une bonne performance du routage, un véhicule, qu'il fasse du covoiturage ou non, doit avoir des informations sur l'état des routes qu'il rencontre sur son chemin. Sans aucune donnée sur le système de transport, il est impossible de prévoir si un véhicule sera coincé dans un embouteillage ou non. Le trafic est influencé par de nombreuses variables et des changements fréquents et soudains se produisent, ce qui le rend très difficile à prévoir. La façon optimale de simuler le routage entre deux points est d'utiliser les données de trafic en temps réel, mais cela est assez lourd en raison de l'indisponibilité de ces informations et de l'efficacité potentielle pour les récupérer.

Faire face à un problème de routage en simulation implique principalement deux aspects lors de l'évaluation du modèle à utiliser. La qualité des données et le temps nécessaire à la récupération de ces informations. Trouver le bon compromis entre ces deux aspects est un point clé. D'une part, en tant que simulation, nous cherchons à le faire en un temps raisonnable, et d'autre part, nous voulons reproduire *in silico*¹ un scénario aussi réel que possible.

Plusieurs approches potentielles ont été prises en considération pour ce travail. La plus simple était basée sur la distance exprimée comme la différence entre les coordonnées de latitude et de longitude. Cette approche est très facile à mettre en œuvre, mais elle est trop triviale pour notre contexte, car nous n'évaluons pas le système routier physique.

Une autre stratégie pourrait être d'évaluer la longueur en termes de distance routière (mètres ou miles), mais cette solution manque également d'informations. Elle n'estime pas un point crucial pour le routage, l'état du trafic.

Pour surmonter les problèmes décrits ci-dessus, cette solution s'appuie sur un service appelé Open Source Routing Machine (OSRM) [LV11]. Grâce à ce service, il

1. **in silico** : Une recherche ou un essai effectué au moyen de calculs complexes informatisés ou de modèles informatiques.

est possible d'obtenir une vue instantanée d'une carte pour une zone géographique limitée, puis cet outil fournit des informations sur les conditions de circulation pendant une journée complète pour plusieurs modes de transport. L'OSRM réplique une instance de la carte sur un serveur local et il est donc possible de l'interroger par le biais de l'API. La réponse contient des informations sur la distance et le temps nécessaires pour se rendre de deux points aux conditions de circulation actuelles.

Le principal avantage de l'OSRM est la réalité des données. Le temps de déplacement d'un point à un autre est affecté par les conditions de circulation. Le principal inconvénient est l'effort nécessaire pour assurer la reproductibilité d'un épisode ou d'un trajet unique en raison des changements soudains de la circulation sur le réseau routier.

3.6 Clustering

3.6.1 Introduction

L'analyse de cluster appartient à la famille de l'apprentissage non supervisé. Il s'agit de diviser la population ou les points de données en groupes, appelés clusters, de sorte que les points de données appartenant au même groupe soient plus similaires aux autres points de données dans d'autres clusters. Une particularité importante du clustering est le critère choisi pour regrouper les points de données. En fait, avec le même ensemble de données, il est possible d'obtenir des résultats différents en fonction du critère de base utilisé. La figure 3.3 rapporte quelques exemples de regroupements différents sur le même ensemble de données animales.

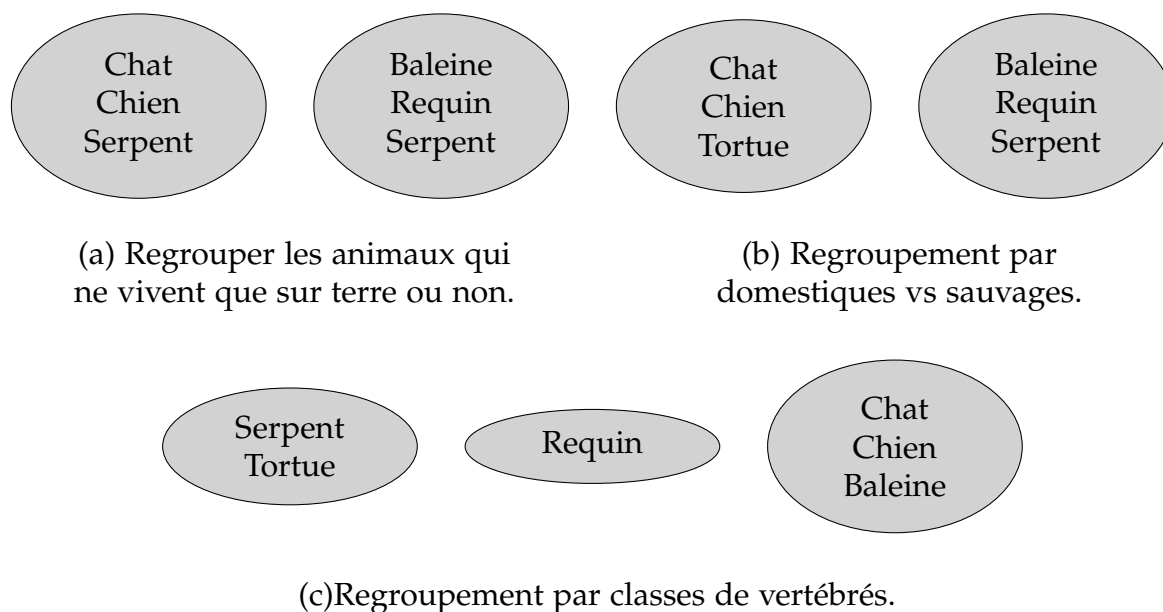


Figure 3.3 – Critères de regroupement différent pour le même jeu de données

Le regroupement de données est le processus de regroupement de vecteurs de données multidimensionnels similaires en un certain nombre de grappes ou de bacs. Les algorithmes de clustering ont été appliqués à un large éventail de problèmes, notamment l'analyse exploratoire de données, l'exploration de données [Eva+01], la segmentation d'images [LK79] et la programmation mathématique [Rao71].

Les techniques de clustering ont été utilisées avec succès pour résoudre le problème d'évolutivité de la machine learning et d'exploration de données, où avant et pendant la formation, les données de formation sont regroupées, et des échantillons de ces clusters sont sélectionnés pour la formation, réduisant ainsi la complexité de calcul du processus de formation, et améliorant même les performances de généralisation [Fis87], [Qui14], [Poto6], [Eng99].

Les algorithmes de clustering fait partie de l'apprentissage non supervisé, un expert n'existe pas et les vecteurs de données sont regroupés en fonction de la distance les uns des autres. Ce travail se concentre sur le clustering non supervisé. De nombreux algorithmes de clustering non supervisés ont été développés. La plupart de ces algorithmes regroupent les données en clusters indépendamment de la topologie de l'espace d'entrée. Ces algorithmes incluent, entre autres, K-means

[FOR65],[HW79] ISODATA [BH67] et les quantificateurs vectoriels d'apprentissage (LVQ) [Fau94]. La carte de caractéristiques auto-organisatrices (SOM) [Koh98], d'autre part, effectue un regroupement topologique, où la topologie de l'espace d'entrée d'origine est conservée. Alors que les algorithmes de clustering sont généralement non supervisés, ont été développés qui effectuent l'apprentissage non supervisé LVQ [Fau94].

3.6.2 K-Means Clustering

L'un des composants les plus importants d'un algorithme de clustering est la mesure de similarité utilisée pour déterminer à quel point deux modèles sont proches l'un de l'autre. Le clustering K-means regroupe les vecteurs de données dans un nombre prédéfini de clusters, en fonction de la distance euclidienne comme mesure de similarité. Les vecteurs de données au sein d'un cluster ont de petites distances euclidiennes les uns des autres et sont associés à un vecteur centroïde, qui représente le "point médian" de ce cluster. Le vecteur centroïde est la moyenne des vecteurs de données appartenant au cluster correspondant. Pour plus de détails, définissons les symboles suivants :

- N_d : désigne la dimension d'entrée, c'est-à-dire le nombre de paramètres de chaque vecteur de données.
- N_o : indique le nombre de vecteurs de données à regrouper.
- N_c désigne le nombre de centroïdes de cluster (tel que fourni par l'utilisateur), c'est-à-dire le nombre de clusters à former.
- Z_p : désigne le p^{ime} vecteur de données.
- m_j : désigne le vecteur centroïde du cluster j .
- n_j : est le nombre de vecteurs de données dans le cluster j .
- c_j : est le sous-ensemble de vecteurs de données qui forment le cluster j .

En utilisant la notation ci-dessus, l'algorithme standard K-means se résume comme suit :

1. Initialiser au hasard les vecteurs centroïdes du cluster N_c

2. Répéter

- (a) Pour chaque vecteur de données, assigner le vecteur en classe avec le vecteur centroïde le plus proche, où la distance du centroïde est déterminée en utilisant

$$d(z_p, m_j) = \sqrt{\sum_{k=1}^{N_d} (z_{pk} - m_{jk})^2} \quad (3.3)$$

- (b) Recalculer les vecteurs centroïdes du cluster, en utilisant

$$m_j = \frac{1}{n_j} \sum_{\forall z_p \in C_j} z_p \quad (3.4)$$

jusqu'à ce qu'un critère d'arrêt soit satisfait.

3.6.3 PSO Clustering

Dans l'algorithme de clustering basé sur l'algorithme Particle Swarm Optimization algorithm [PHHo8], chaque particule $Y_i = (y_1, y_2, \dots, y_K)$ représente les centres des K classes, tandis que $y_j (j = 1, 2, \dots, K)$ représente le vecteur des coordonnées du point central de la j_{me} classe dans la i_{me} particule (la dimension de y_j est décidée en fonction de la situation réelle). L'essaim de particules constitue de nombreux plans classés candidats. On sait qu'il s'agit d'une clé de clustering qui utilise un algorithme d'optimisation pour évaluer la qualité du plan de classification, les auteurs de [PHHo8] proposent donc une fonction d'adaptabilité f comme suit :

$$f(y_i) = \frac{\max(\bar{d}_1(y_i))}{\max(d_2(y_i))}. \quad (3.5)$$

$\max(\bar{d}_1(y_i))$ est la valeur maximale des valeurs moyennes des distances au sein des mêmes classes dans le plan de classement, qui est exprimée par la particule Y_i , tandis que $\max(d_2(y_i))$ est la valeur minimale des distances entre les classes dans le plan de classement, qui est exprimée par particule Y_i .

$$\max(\bar{d}_1(y_i)) = \max_{j=1,2,\dots,k} \left(\sum_{\forall x_i \in y_i} \frac{d(x_i, y_i)}{|y_j|} \right).$$

- $|y_j|$ est le numéro de l'élément dans la classe j^{me} .
- $\min(d_2(y_i)) = \min_{\forall i, j, i \neq j} (d(y_i, y_j))$;
- $i, j = 1, 2, \dots, k$

Si la valeur minimale de la fonction d'adaptabilité (3.5) satisfait simultanément une petite distance au sein d'une même classe et une distance considérable au sein des classes, la stratégie de classification est plus forte. L'algorithme de clustering basé sur l'algorithme Particle Swarm Optimization comprend les étapes suivantes :

1. Dans l'espace à n dimensions, nous définissons la taille de la population m , les coefficients d'accélération $c1$ et $c2$, l'hypothèse de nombre d'itération num , le nombre de clustering K et un ensemble de points donné avec N points, etc. Définir la meilleure position historique de chaque particule p_{best} égale à la position initiale et fixer la meilleure position globale de l'essaim de particules p_{best} égale au meilleur de tous p_{best} dans une population de particules avec des positions et des vitesses aléatoires (les vecteurs de position et de vitesse sont constitués de K vecteurs d'espace à n dimensions).
2. Pour chaque particule Y_i , recalculer les distances entre l'ensemble $\{x_1, x_2, \dots, x_n\}$ et les centres K et diviser l'ensemble $\{x_1, x_2, \dots, x_n\}$ selon la régulation de distance de l'algorithme K-means.
3. Pour chaque particule Y_i , calculer l'évaluation de la fitness selon l'expression $f(Y_i)$ selon l'équation (3.5).
4. Comparer et réinitialiser la meilleure position historique p_{best} et la meilleure évaluation de fitness de chaque particule, et, comparer et réinitialiser la meilleure position globale g_{best} et la meilleure évaluation de fitness de l'essaim de particules.
5. Modifier la vitesse et la position des particules selon les équations (12.3) et (3.7) et les limiter selon les équations (3.6) et (3.8).

$$\begin{cases} v_{id} = v_{max} & Si v_{id} > v_{max} \\ v_{id} = -v_{max} & Si v_{id} < -v_{max} \end{cases} \quad (3.6)$$

$$x_{id} = v_{id} \quad (3.7)$$

$$\begin{cases} x_{id} = x_{max} & Si x_{id} > x_{max} \\ x_{id} = -x_{max} & Si x_{id} < -x_{max} \end{cases} \quad (3.8)$$

Dans l'expression (3.8), on sélectionne le maximum valeur de chaque dimension en tous les points comme x_{max} .

6. Inspecter la condition de terminaison (l'algorithme a atteint les temps d'itération les plus longs de l'hypothèse); si elle est satisfaite, l'algorithme doit être terminé; sinon, retourner à l'étape (2).
7. Résultat de la classification obtenus.

3.7 Relocalisation

Nous présentons Dynamic Rebalancing Based on Demand (DRBD), une nouvelle stratégie de relocalisation. L'objectif de la DRBD est de répartir les véhicules de manière efficace et dynamique en fonction de la tendance actuelle de la demande. Grâce à DRBD, il est possible d'éviter des problèmes tels que le rassemblement de nombreux véhicules dans une zone restreinte.

La relocalisation est évidemment utile lorsqu'un véhicule se trouve dans une zone éloignée et qu'il n'a pas de demandes environnantes à servir. En outre, la relocalisation est utile pour éviter le rassemblement de plusieurs véhicules dans une zone restreinte. Un exemple est illustré dans la figure 3.4 En bas de la carte, il y a beaucoup de véhicules et peu de demandes, tandis qu'en haut, il n'y a pas assez de véhicules pour satisfaire la demande. Enfin, la relocalisation peut réduire la concur-

rence entre les véhicules en déplaçant un ou plusieurs véhicules dans une zone qui leur fait défaut. En relocalisant, il est possible d'améliorer la performance locale de chaque véhicule et la performance globale de la flotte, puisque la charge de travail est répartie de manière équitable.

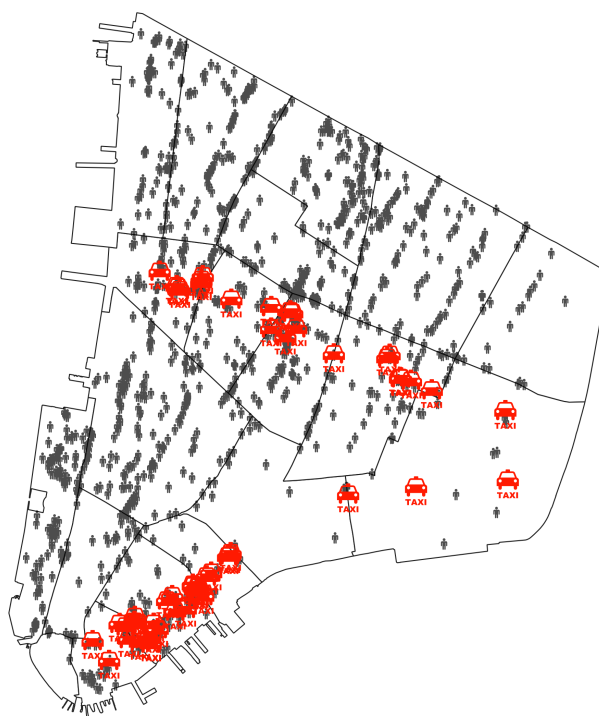


Figure 3.4 – Flotte déséquilibrée.

La relocalisation efficace, avec le routage, est un point clé pour augmenter la performance d'un système de covoiturage. La politique de relocalisation peut évaluer plusieurs variables complexes. Par exemple, le comportement des clients qui peut changer d'un jour à l'autre. Il est facile d'observer des comportements différents dans une grande ville, surtout si l'on compare les trajets effectués les jours ouvrables avec ceux effectués les jours fériés, mais aussi le même jour en fonction de l'heure. Le matin, on s'attend à ce que les gens se déplacent vers leur lieu de travail, en partant de différents endroits et en convergeant vers quelques zones. En revanche, le soir, la tendance devrait s'inverser, les gens se dispersant de quelques endroits vers d'autres, car ils rentrent chez eux ou vont ailleurs.

Les schémas qui décrivent les différents comportements des usagers ne sont pas aussi marqués dans les ensembles utilisés pour la simulation, car ils sont liés aux déplacements en taxi. Les taxis sont principalement utilisés par les touristes plutôt que comme moyen de transport pour les navetteurs. La prise en compte de toutes les variables qui peuvent se produire nécessite une solution sophistiquée et peut être assez lourde à calculer. Cependant, l'une des principales caractéristiques de la DRBD est sa capacité à fournir une assistance en temps réel.

L'analyse ne doit être effectuée que sur un ensemble restreint de données. DRBD maintient un calcul simple qui n'implique que quelques éléments au cours du processus d'évaluation à chaque fois puisqu'il n'utilise que les demandes en temps réel qui n'ont pas encore été servies. L'objectif de la DRBD est de prédire les zones, avec les probabilités associées, où les véhicules doivent être déplacés afin d'anticiper la demande future. Pour déterminer les zones, DRBD s'appuie sur un algorithme de clustering. Le découpage change dynamiquement en fonction de la distribution de la demande. Chaque zone correspond à un cluster et leur position évolue dans le temps. Comme présenté précédemment, le regroupement est effectué sur l'ensemble des demandes, et pour maximiser l'efficacité du calcul, seules les dernières demandes qui n'ont pas encore été attribuées à un véhicule sont évaluées.

La méthode de PSO clustering est utilisée comme méthode de clustering, l'algorithme 5 montre la procédure de clustering. Si nous évaluons toutes les demandes, plutôt que de nous concentrer uniquement sur un sous-ensemble restreint, cela pourrait conduire à un résultat non souhaité. Car, comme expliqué ci-dessus, la distribution de la demande change au cours de la journée. De plus, le fait d'impliquer de nombreuses demandes dans l'évaluation du clustering augmente à la fois le temps et la complexité nécessaires au calcul de la répartition.

L'évaluation de toutes les dernières demandes ne semble pas être une solution appropriée. Cette approche prendrait en considération les demandes qui ont déjà été servies ou celles qui sont servies entre-temps. Par conséquent, on pourrait rassembler plusieurs véhicules dans une seule zone dans laquelle il n'y a pas de demandes disponibles, ce qui est l'anti-but de notre relocalisation. L'objectif de DRBD

Algorithme 5 : Définition des zones pour la relocalisation

```

début
  Résultats :( Clusters, Probabilité)
  Fonction: TrouverClusters ( Requetes valides,K)
  Clusters,C = PSOClustering(Requetes valides,K) // K: le nombre des
    clusters
  pour chaque ( $i \in clusters$ ) faire
    Probabilité= $\frac{Taille(i)}{(Total\ requete\ valide)}$ ;
    ProbRelocalisation[i]=Probabilité;
  fin
  Retourner : C,ProbRelocalisation
  Fin Fonction
fin

```

est de prédire les zones dans lesquelles les demandes à venir pourraient ne pas être satisfaites et de répartir les véhicules sur le cluster injuste pour éviter les scénarios indésirables.

Une fois le clustering calculé, la politique d'étalement de la DRBD est naïve. Tout d'abord, elle extrapole la distribution de probabilité définie sur le nombre d'éléments appartenant à chaque cluster. Ensuite, en fonction de cette distribution, le répartiteur relocalise le véhicule. qui a une probabilité d'être réaffecté au cluster i selon l'algorithme 6.

Algorithme 6 : Relocalisation des véhicules vides

Paramètres : V les véhicules vides**Résultat** : Relocaliser vehicle V a une nouvelle zone**Procédure Relocaliser (V)** $(C, ProbRelocalisation) = \text{TrouverClusters}(\text{Requêtes valides}, K)$; // Algo 5

```
pour chaque ( $v \in V$ ) faire
  rnd = générer valeur aléatoire  $\in [0, 1]$ 
   $i = 1$ 
  tant que ( $i \leq \text{taille}(c)$ ) faire
    si ( $ProbRelocalisation[[i] \geq rnd$ ) alors
       $v.\text{destination} = C[i].\text{postion}$ 
       $v.\text{DeplacerVersDestination}$ 
    fin
     $i++$ 
  fin
fin
```

Fin Procédure

Chapitre 4

Expérimentation et Résultats

4.1 Étude de cas (ville de New York)

4.1.1 Introduction

Ce chapitre donne un aperçu de l'implémentation de l'approche avec des données réelles. Tout d'abord, sera présentée une analyse de données, montrant comment la demande est distribuée à travers la zone, en discriminant sur le nombre de passagers pour chaque demande. Ensuite, sera esquissé un scénario impliquant un véhicule montrant son état interne avec sa perception des demandes environnantes. Enfin, une instance du module DRBD est présentée.

Le moteur de cheminement a été configuré avec les données de Manhattan pour récupérer le temps de trajet et le chemin le plus court. Ceci est fait en fonction de l'état du trafic reproduit par le moteur, grâce aux données enregistrées obtenues par les services cartographiques.

4.1.2 Data sets utilisées

Plusieurs base de données, concernant les véhicules de transport, sont disponibles gratuitement sur le web mais très peu d'entre elles peuvent être utilisés pour les besoins de ce travail. Les données, pour être utilisées, doivent contenir des informations sur la position et la destination de chaque voyage, définies par le système de coordination géo-spatiale. Toutes ces données n'adoptent pas cette norme. Certaines d'entre elles, par exemple, divisent la région en zones et indiquent ensuite la destination et l'origine de chaque voyage en se référant à l'identifiant de la zone, plutôt qu'aux paires de latitude et de longitude. Les autres informations requises concernent le nombre de passagers et la date et l'heure liées à la génération de chaque voyage.

Les deux dernières caractéristiques, la date et le nombre de passagers, peuvent être estimées si le data set original ne les fournit pas. Cependant, cela peut affecter les

modèles dans les données. En effet, plus les données reflètent un scénario réel, plus la simulation sera précise et plus le système sera fiable. La plupart des ensembles de données disponibles proviennent de taxis ou d'autres sociétés de mobilité à la demande comme Uber ou Lyft. Les services de mobilité à la demande sont largement utilisés par les touristes, en particulier lorsque ceux-ci se trouvent dans des métropoles. Il serait plus facile de trouver des modèles si les données concernaient les passagers qui utilisent ces services pour rejoindre leur lieu de travail depuis leur domicile ou l'inverse plutôt que les touristes.

Pour émuler certains types de modèles, un ensemble de données prétraitées a été utilisé. En particulier, l'ensemble final est obtenu en fusionnant les données de 50 mardis consécutifs allant de juillet 2015 à juin 2016 sur les trajets de taxis collectés dans la région de Manhattan à New York. Ensuite, les requêtes fusionnées ont été réparties en 4 bandes horaires différentes représentant différents moments de la journée. Les jeux de données ont été aimablement fournis par les auteurs des articles référencés [GD18],[Cas+21].

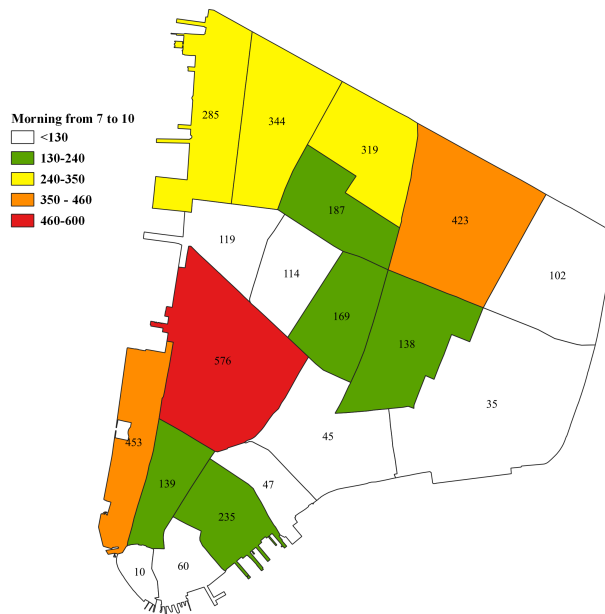
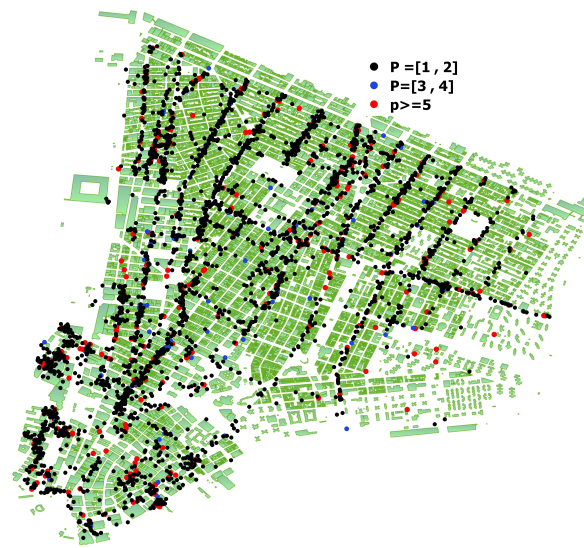
On a utilisé les données relatives à l'heure de pointe du matin entre 7 et 10 heures et la tranche du soir de 18 à 21 heures. Deux tranches horaires différentes ont été utilisées pour simuler une tendance différente entre la demande.

4.1.3 Exploration des données

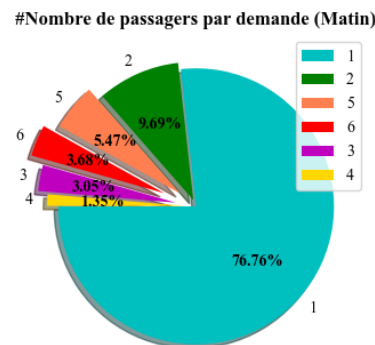
Les figures suivantes montrent les distributions des demandes. La figure 4.1 montre la distribution de la zone horaire du matin définie de 7 à 10 heures, tandis que l'autre figure 4.2 montre comment les demandes sont réparties pendant la zone horaire du soir définie de 18 à 21 heures.

Les demandes de transport sont affichées en fonction de leur position dans la zone, et sont regroupées par nombre de passagers. Un point noir signifie que le nombre de passagers pour une demande est inférieur à 3 tandis qu'un point est rouge lorsque les passagers sont plus de 4. Enfin, le point est bleu si les passagers

sont un groupe de 3 ou 4 personnes. A noter que le plus grand groupe de passagers rencontré est composé de 6 personnes.

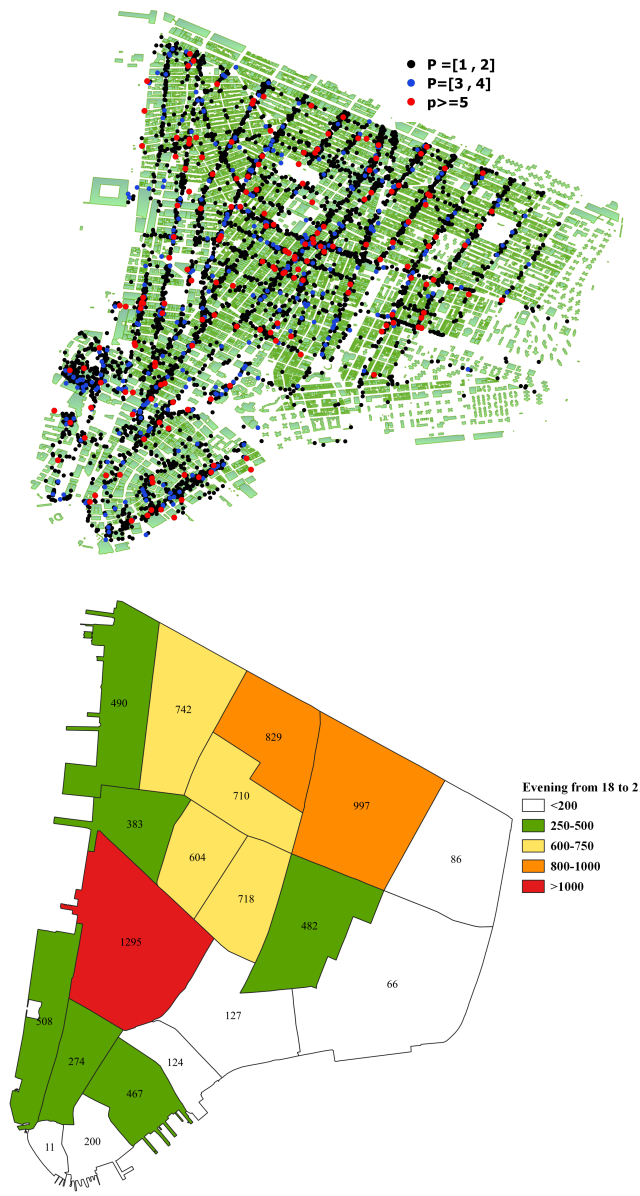


(a) Distribution des demandes dans la région de Manhattan

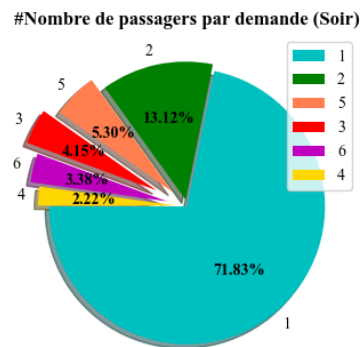


(b) Nombre des passagers pour les demandes

Figure 4.1 – Bande horaire d’exploration des données de 7 à 10 heures du matin.



(a) Distribution des demandes dans la région de Manhattan



(b) Nombre des passagers pour les demandes

Figure 4.2 – Bande horaire d’exploration des données de 18 à 21 heures.

4.1.4 Statut et perception du véhicule

Le tableau 4.1 ci-dessous et la figure 4.3 donnent un aperçu complet de l'état interne et de la perception du véhicule lors de l'évaluation de l'environnement, comme défini dans le chapitre précédent à la section 3.3 Les demandes dans le tableau sont présentées en fonction de la métrique la plus proche définie pour les véhicules.

	Position	Nbre de sièges libres	
Véhicule	(40,73686218,-74,0054245)	3	
Les demandes en attente			
	Origine	Nbre passagers	Destination
Première Demande	(40,73453522,-74,00609589)	1	(40,73210907,-73,99658203)
Deuxième Demande	(40,72830582,-74,00305939)	1	(40,72202301,-73,98535156)
Troisième Demande	(40,72538757,-73,9969101)	1	(40,72335052,-73,97943115)

Table 4.1 – État interne du véhicule avec sa perception. Version numérique de la représentation graphique de la figure 4.3

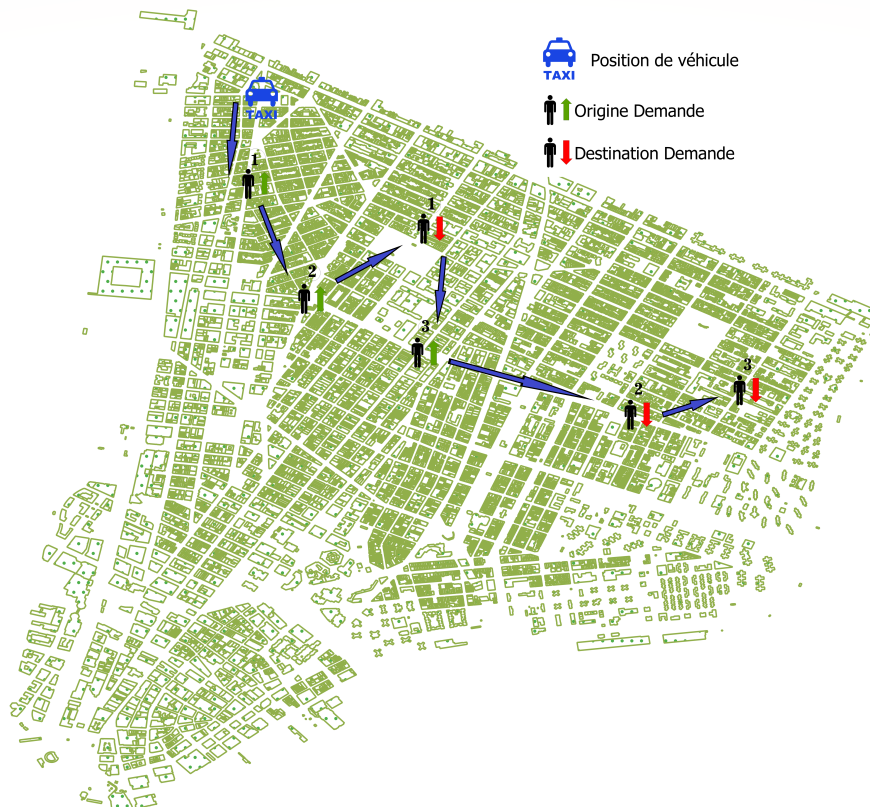


Figure 4.3 – Représentation de l'emplacement et de la destination du véhicule avec sa perception.

Le véhicule est déjà en train de servir un client, ce qui laisse 3 places pour d'autres clients. Sur le trajet du véhicule, de sa position actuelle à sa destination, le véhicule peut faire monter d'autres passagers en covoiturage. Notez que, avec les données disponibles, nous ne pouvons pas savoir si cela peut se produire avec certitude. Les données rapportées manquent d'informations concernant le retard que les clients ont déjà pris.

Par conséquent, nous ne pouvons pas conclure si les passagers réels sont prêts à partager leur trajet avec l'une des demandes environnantes. L'état du véhicule ne donne aucune indication sur le nombre de demandes qu'il est en train de traiter, donc nous ne savons pas si les passagers qu'il est déjà en train de traiter appartiennent à la même demande ou non.

4.1.5 Rééquilibrage dynamique sur la base des demandes (DRBD)

Dans cette section est présentée une instance de la tâche DRBD. Elle commence par prédire les zones dans lesquelles se trouvent les demandes qui potentiellement ne pourraient pas être servies. Nous rappelons que les zones sont prédites sur les demandes qui n'ont pas encore été servies dans les 15 dernières minutes et le résultat obtenu est montré dans la figure ci-dessous.

4.2 Évaluation

4.2.1 Introduction

L'objectif de cette recherche est de définir une approche générique pour les villes, capable de s'adapter aux changements de la ville, pour un système de covoiturage urbain. La solution repose sur une architecture orchestrée par une flotte de véhicules pour répondre aux demandes de covoiturage. Nous présentons une nouvelle stratégie de relocalisation qui repose sur un regroupement dynamique

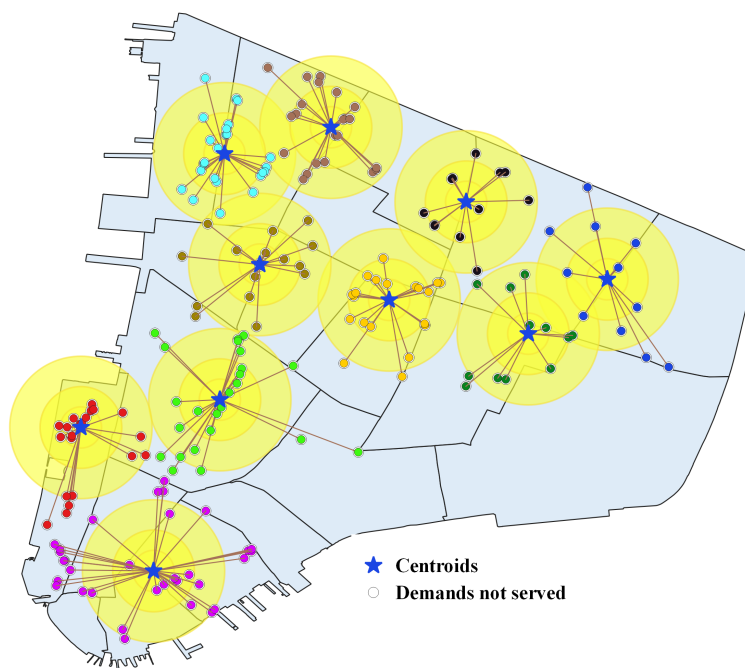


Figure 4.4 – Clusters obtenus via DRBD sur les dernières requêtes pas encore servies. .

effectué sur les demandes disponibles en temps réel.

Cette recherche ne prend pas en compte les impacts de flotte sur l'environnement, ni d'un point de vue financier, ni d'un point de vue plus éthique, à savoir la pollution de l'air. Enfin, dans cette recherche, nous n'évaluons pas comment la navigation est influencé par notre modèle.

4.2.2 Mesures d'évaluation

Le système mis en œuvre a été conçu pour être orienté vers le client. Ainsi, la satisfaction des clients est la priorité. Par exemple, l'algorithme a été conçu pour maintenir le temps d'attente des clients à un niveau faible. Lorsqu'une demande est générée, un véhicule doit venir servir les clients dans les quinze minutes. Si aucun véhicule ne peut le faire dans le temps défini, la demande est rejetée. Il est à noter

que ce délai comprend le temps nécessaire à un véhicule pour atteindre le point de départ de la demande pour servir.

Outre la limitation du temps d'attente, une contrainte codée en dur, a été introduite. Cette contrainte limite le temps qu'un client peut passer en faisant du covoiturage. Ceci est fait pour donner aux clients une meilleure expérience.

La première mesure pour évaluer la qualité de notre proposition est la couverture de la demande par la flotte des voitures pour satisfaire la majorité des clients. Ensuite, la performance de l'approche a été évaluée en considérant d'abord l'expérience du client à travers plusieurs aspects, y compris le temps d'attente et le temps pour satisfaire une demande. Enfin, d'autres caractéristiques locales ont été observées pour décrire la performance des véhicules. Ces analyses visent à décrire comment la charge de travail est répartie entre les véhicules et comment ils se comportent.

4.2.3 Scénarios

Dans l'approche proposée par cette recherche, les véhicules sont autorisés à répondre à des demandes de transport en faisant du covoiturage. De plus, dans certaines circonstances, un véhicule peut demander à être déplacé vers une autre zone.

Un véhicule peut être relocalisé s'il ne dessert aucune demande et ne perçoit aucune demande dans la zone voisine. Le DRBD est chargé du processus de relocalisation. Sa première tâche consiste à calculer les zones potentielles. Cela se fait par le biais d'un regroupement dynamique par l'algorithme d'optimisation de l'essaim de particules (PSO) effectué sur les demandes non servies en temps réel. Une fois que le DRBD a calculé les zones, un véhicule est relocalisé vers le centre de l'une de ces zones dans laquelle son utilité pourrait être améliorée. Enfin, la politique de répartition est définie par le nombre d'éléments appartenant à chaque cluster. En fonction de cela, une distribution de probabilité est définie. Dorénavant, certains acronymes seront utilisés pour indiquer ce que le véhicule est autorisé à faire.

- **RQB** signifie relocalisation. Le module DRBD est actif. Ainsi, si un véhicule n'est pas en service et ne perçoit aucune demande, il peut demander à trouver une nouvelle zone dans laquelle s'installer.
- **NRQB** Signifie pas de rééquilibrage. C'est le contraire du précédent, et donc quel que soit le statut du véhicule, il ne peut jamais demander à être relocalisé. Dans ce cas, une fois qu'un véhicule est vide, la seule action légale autorisée est d'attendre qu'une demande, suffisamment proche pour être servie, soit générée.
- **COV** Signifie covoiturage. Cela signifie qu'un véhicule peut servir plusieurs demandes simultanément si elles sont éligibles pour cela. Par exemple, étant donné deux demandes et en supposant que le véhicule est actuellement en train de servir l'une d'entre elles et que le véhicule perçoit l'autre, alors la nouvelle demande doit être conforme à la contrainte définie au point 4.3 pour être servie en même temps.
- **NCOV** Signifie pas de covoiturage. Pour chaque paire de demandes, éligibles pour être servies ensemble, leurs passagers ne peuvent pas se chevaucher en même temps dans le même véhicule. Du point de vue du véhicule, cela signifie qu'une fois qu'une demande est prise en charge, le véhicule doit rouler vers sa destination avant qu'une autre demande puisse lui être attribuée.

Pour évaluer les avantages de notre nouvelle proposition, différents scénarios ont été comparés. Dans la suite, les scénarios de base sont également appelés scénarios de référence ou benchmark.

4.2.4 Configuration expérimentale

Tous les résultats présentés dans ce chapitre ont été obtenus sur les mêmes données et dans les mêmes conditions afin de garantir l'équité. Précisément, ont été utilisées des données relatives à l'événement avec une flotte composée de 200 véhicules. Les demandes disponibles sont au nombre de 10000 et la distribution de la demande à travers Manhattan est représentée par la figure 4.1. Les demandes sont générées dans un intervalle de temps de 5 heures. Le temps de travail d'un véhicule

est au maximum de 5 heures plus le temps nécessaire pour satisfaire les dernières demandes dans sa file d'attente.

Le taux de satisfaction de la demande est inférieur à 100%. Comme on peut le déduire du graphique de distribution des données, quelques demandes ont 5 et 6 passagers et nécessitent donc des véhicules avec 5 ou 6 sièges disponibles ensemble. Tous les véhicules utilisés pendant la simulation ont été limités à 4 sièges. Cette contrainte a été introduite pour garantir des conditions préalables équitables pour chaque simulation. Leurs positions initiales seraient définies de manière aléatoire, et donc un ou plusieurs scénarios plus chanceux pourraient résulter comme privilégiés juste pour l'allocation aléatoire.

4.2.5 Évaluation et analyse

Cette section est organisée comme suit : d'abord, une analyse spécifique de chaque critère de jugement à part pour évaluer la qualité de notre proposition de covoiturage. Une comparaison a été faite entre les résultats de notre solution et les résultats des scénarios de base, donnant une vue d'ensemble des scénarios et, enfin, sont présentés les avantages et les inconvénients d'une approche de covoiturage. Successivement, nous nous concentrerons sur l'impact du DRBD sur la performance en relocalisant les véhicules.

En ce qui concerne les mesures d'évaluation utilisées, nous devrions nous attendre à ce que le temps d'attente et la distance parcourue nécessaires pour répondre à une demande soient moindres lorsque la fonction COV est activée pour les véhicules. D'autre part, dans le cadre du covoiturage, le temps moyen nécessaire pour répondre à une demande devrait augmenter par rapport à celui obtenu sans activation du COV. Le temps d'attente est calculé sur l'ensemble des demandes tandis que la distance parcourue, ainsi que le nombre de passagers transportés, sont rapportés à un seul véhicule.

Temps d'attente

La figure 4.5 montre les résultats obtenus à partir de scénario de base et de la proposition. En haut à gauche, La figure 4.5(a), le covoiturage n'est pas activé, ni le rééquilibrage. La figure 4.5(b) décrit le cas dans lequel le covoiturage est activé mais les véhicules ne peuvent pas être déplacés. En revanche, dans le coin inférieur gauche, la figure 4.5(c), sont esquissés les résultats de la dernière ligne de base restante : le cas dans lequel le covoiturage est désactivé mais le véhicule peut être déplacé vide.

Enfin, la figure 4.5(d) montre les résultats obtenus par notre proposition. Ainsi, les véhicules peuvent servir plus de demandes en même temps, en faisant du covoiturage, et ils peuvent aussi être relocalisés si leurs conditions préalables le permettent.

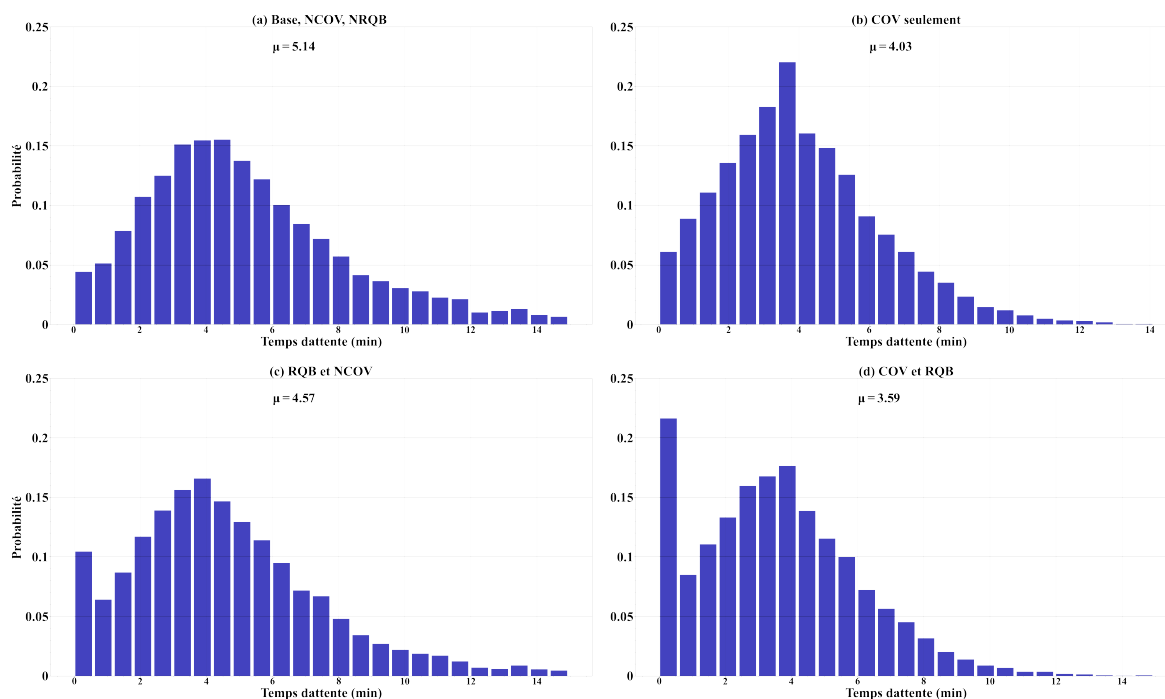


Figure 4.5 – Comparaison des temps d'attente

Les attentes concernant le temps d'attente ont été confirmées. Le covoiturage est un moyen impressionnant de réduire le temps d'attente moyen. En outre, la relocalisation seule diminue également le temps d'attente. Notre proposition a réduit le tiers de temps d'attente d'un passager avant d'être servi par rapport au scénario de la figure 4.5(a). Notre approche permet de prendre en charge plus de la moitié des

demandes en moins de 4 minutes.

Les clients servis

La figure 4.6 illustre la distribution sur le nombre de passagers que chaque véhicule a servis. La première ligne concerne les scénarios dans lesquels les véhicules ne peuvent pas être déplacés. Dans la deuxième ligne, le véhicule peut exploiter le DRBD. Dans la première colonne, le covoiturage est désactivé, alors qu'il est activé dans la seconde.

La figure 4.6 montre les différences entre le scénario de base et les propositions, en montrant comment les passagers sont répartis sur la flotte. Comme le montre la figure 4.6 (a), la plupart des véhicules sont paresseux. Ils ne servent que quelques passagers, laissant toute la charge de travail aux autres, qui servent environ deux fois plus que la moyenne globale. A l'appui de cela, la variance de la distribution est très élevée.

Cette tendance à la discontinuité pourrait être atténuée en permettant aux véhicules d'être déplacés ou de faire du COV. Une cause possible de cette discontinuité est qu'un véhicule peut charger des passagers qui nécessitent un effort de conduite important pour atteindre la destination et qu'une fois atteint, le véhicule se trouve dans une zone éloignée sans aucune demande dans le voisinage.

Le fait de permettre aux véhicules d'être déplacés, tout en pratiquant le covoiturage, empêche l'utilisation inefficace des véhicules. De plus, le nombre de passagers pour chaque véhicule est réparti de manière équitable sur la flotte. Cela se remarque à la variance qui est très faible par rapport aux autres résultats.

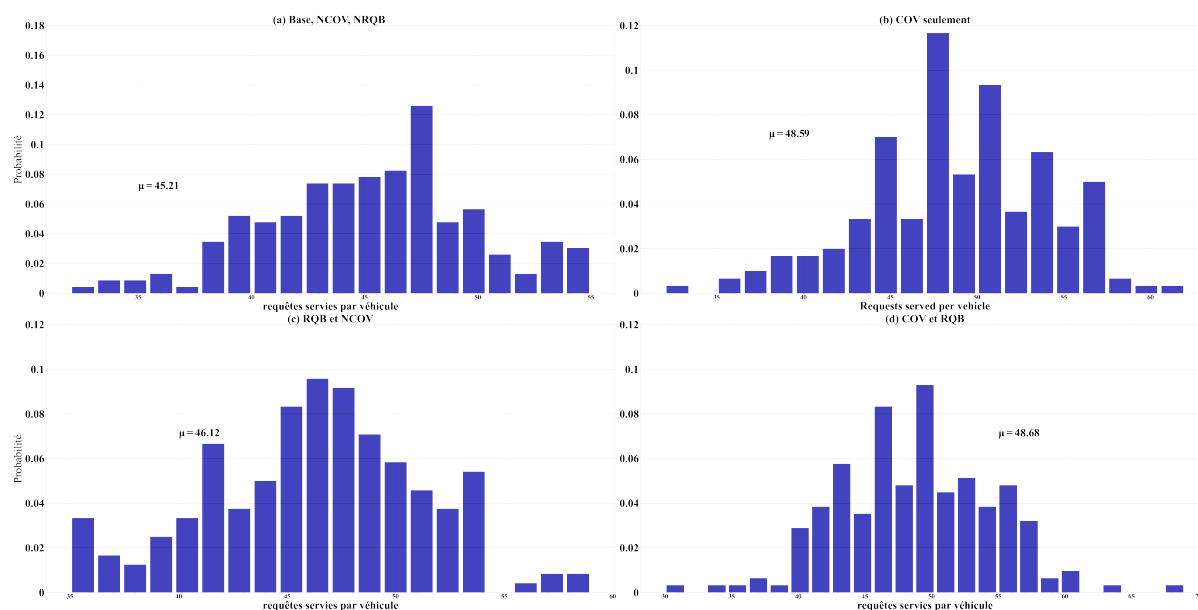


Figure 4.6 – Comparaison des nombres des clients servis

Distance parcourue

La figure 4.7 illustre la distribution de la distance parcourue par un véhicule en service. La ligne du haut concerne les scénarios dans lesquels le véhicule ne peut pas être déplacé, tandis que dans la deuxième ligne, le véhicule peut compter sur le DRBD. Dans la première colonne, le covoiturage est désactivé alors qu'il est activé dans la seconde.

La distance parcourue par chaque véhicule, pour répondre à ses demandes, a été analysée. Nous nous attendons à ce que cette caractéristique ait une corrélation avec le nombre de demandes servies. En outre, nous nous attendons à ce que, selon notre proposition, chaque véhicule ait parcouru presque le même nombre de kilomètres.

La figure 4.7 montre la distance parcourue, en rapportant la distribution de probabilité pour chaque scénario analysé. Si l'on analyse le scénario dans lequel le COV et le RQB sont tous deux désactivés, et si l'on compare les graphiques précédents aux résultats obtenus (figure. 4.7 (a)), il est très probable que les véhicules qui ont parcouru peu de kilomètres sont ceux qui ont le plus de chances d'atteindre les

objectifs fixés.

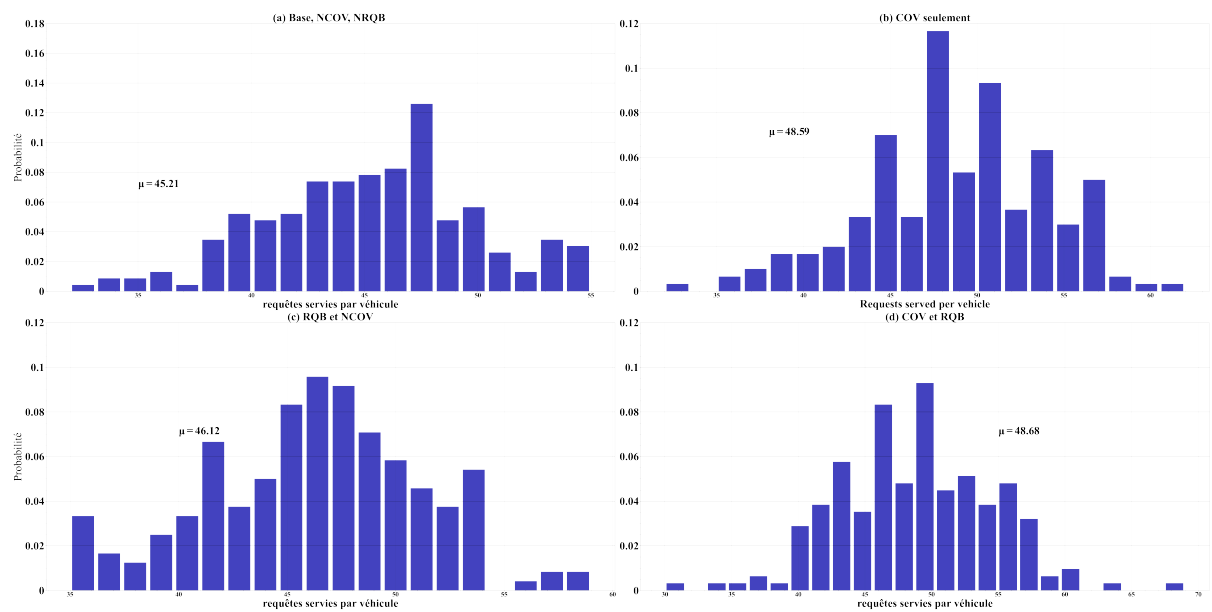


Figure 4.7 – Comparaison des distances parcourues

Comparaison avec scénario de base

Jusqu'à présent, seules des comparaisons d'efficacité sur les métriques définies ont été proposées. Ici, nous nous concentrons sur les avantages et les inconvénients de notre proposition par rapport au scénario de base. Tout d'abord, notre approche peut satisfaire environ 90.25% de la demande. Les demandes qui ne sont pas servies sont un cas extrême et on s'attendait à ce qu'elles ne soient pas servies. Comme indiqué précédemment, des véhicules avec plus de sièges sont nécessaires pour répondre à ces demandes.

L'activation du COV, quelle que soit la stratégie DRBD, est suffisante pour atteindre le plafond des demandes servies (99.33%). Cependant, en activant le DRBD, le temps d'attente est légèrement réduit. En outre, le comportement de chaque véhicule converge vers une valeur moyenne, ce qui entraîne une répartition équitable de la charge de travail entre les véhicules.

Le principal inconvénient du rééquilibrage est l'impact potentiel que les vé-

hicules peuvent avoir sur le trafic. En effet, les véhicules se déplacent sans aucun passager, ils encombrent donc le réseau routier sans rendre de service utile. Dans les scénarios analysés, la distance parcourue par les véhicules pour se déplacer est inférieure à 5 km par heure. Donc, si la zone analysée est assez petite, ces données ne sont pas alarmantes.

Les mesures analysées donnent un aperçu de la performance globale atteinte par notre approche par rapport au scénario de base. Malgré cela, ces mesures ne donnent aucune information sur la façon dont DRBD influence le processus de relocalisation d'un véhicule.

4.2.6 Conclusion

Notre rééquilibrage est une approche tangible qui peut être appliquée dans un scénario du monde réel. DRBD utilise le statut momentané de la demande pour distribuer les véhicules vides sur la zone, empêchant les véhicules d'être rassemblés dans quelques zones. D'autre part, le répartiteur de base a besoin d'une connaissance préalable de la tendance de la demande pour calculer les zones dans lesquelles les véhicules seront déplacés ultérieurement. Il n'est pas encore possible de récupérer ces informations. Ce problème est lié à l'importance de disposer d'un cadre efficace de prédiction de la demande. Alors que le DRBD s'appuie uniquement sur des données en temps réel, la prédiction de la demande n'est pas requise par notre stratégie de rééquilibrage. Dans cette recherche, nous ne nous concentrons pas sur l'impact que les véhicules ont sur le réseau routier de la ville. Cependant, avant d'appliquer cet algorithme dans le monde réel, des études supplémentaires sur l'impact des véhicules sur le trafic doivent être menées.

Résumé

Dans ce chapitre, nous avons présenté l'évaluation de notre proposition. Un rééquilibrage innovant en temps réel et dynamique. DRBD s'appuie sur un regroupement dynamique effectué sur un ensemble restreint des demandes disponibles (non servies). Ensuite, DRBD calcule la probabilité pour un véhicule d'être relocalisé dans les zones trouvées. Le tableau suivant montre un résumé des résultats obtenus :

	Requettes servies	Requettes servies (%)	Temps d'attente (min)	Distance parcourue(Km)
Base (NCOV,NRQB)	8388	83.88	5.14	124.22
COV seulement	9010	90.10	4.03	135.26
DRBD (COV et RQB)	9025	90.25	3.59	136.38

Table 4.2 – Tableau récapitulatif des résultats obtenus

Toutes les valeurs du tableau 4.2 se réfèrent à 10000 demandes et la flotte est composée de 200 véhicules. La distribution des demandes est illustrée par la figure 4.1 et 4.2.

- **RQB** Indique si les véhicules sont autorisés à être déplacés dans une autre zone ou non (NRQB).
- **COV** Indique si le covoiturage est activé ou non (NCOV). S'il est désactivé, toutes les demandes sont servies de manière isolée. En d'autres termes, les demandes servies par un véhicule sont exécutées de manière séquentielle. Par conséquent, les nouveaux passagers ne peuvent être pris en charge que si le véhicule ne transporte pas d'autres clients.
- **Distance parcourue** Exprimé en kilomètres. Il s'agit du kilométrage que le véhicule a parcouru tout au long de la simulation. Cette distance est exprimée en moyenne. Elle comprend la distance parcourue pour servir les passagers et la distance pour se déplacer en cas de besoin.

Il suffit de permettre aux véhicules de servir plusieurs demandes simultanément pour atteindre le pic maximal de demandes servies. Cependant, lorsque le DRBD est activé, les véhicules semblent, en moyenne, plus efficaces. Le taux de partage des trajets diminue sans compromettre les autres mesures de performance. On

observe plutôt une légère amélioration du temps nécessaire à un véhicule pour répondre à ses demandes, ainsi que du temps d'attente de chaque passager avant d'être pris en charge.

Conclusion et perspectives

Conclusion et perspectives

Enfin, c'est le chapitre conclusif de cette thèse. On y résume la contribution de ce travail de recherche, en mentionnant les problèmes que nous abordons et ceux qui ne le sont pas. En outre, seront présentés les nouveaux défis que ce travail a ouvert. Enfin, quelques indications pour les travaux futurs sont données.

Contribution de la thèse

Dans cette thèse, nous avons étudié une approche pour un système futuriste de mobilité urbaine, tel que Uber ou Lyft. Nous avons cherché à présenter un modèle entièrement décentralisé qui peut être mis en œuvre dans un scénario réel pour répondre aux demandes de transport.

La nouvelle stratégie proposée dans cette thèse surmonte certaines limitations présentes dans les modèles multi-agents, pour les demandes de transport, qui sont disponibles dans la littérature. Le point fort de notre approche est sa capacité à faire face aux changements de ville et sa reproductibilité à la structure d'autres villes.

En outre, notre proposition permet de faire évoluer la taille de la flotte. Si nécessaire, un nouveau véhicule peut rejoindre la flotte, sans l'annoncer aux autres véhicules. La nouveauté de notre proposition ne se limite pas à l'algorithme de covoiturage. En plus de celui-ci, nous avons introduit un nouvel algorithme pour rééquilibrer les véhicules vides de la flotte : Dynamic Rebalancing Based on Demand (DRBD).

L'objectif du DRBD est de définir les zones dans lesquelles les véhicules peuvent être relocalisés. Une fois le clustering effectué, il calcule la distribution de probabilité pour la relocalisation des véhicules. Notre rééquilibreur s'appuie sur un algorithme de clustering dynamique qui est effectué sur un ensemble de requêtes disponibles en temps réel par le PSO clustering. Précisément, le DRBD analyse les demandes qui n'ont pas encore été servies [MMA22].

L'évolutivité de la flotte, en termes de taille, est obtenue grâce à la représentation standard des véhicules et à la conception du modèle. Le modèle est reproductible à d'autres villes. Car, lors du processus d'évaluation effectué par un véhicule, ne sont pas impliquées des données corrélées à la structure de la ville. En effet, une position est représentée par une coordonnée géospatiale à deux dimensions. Cette élasticité n'est pas garantie par les modèles disponibles dans la littérature. La plupart d'entre eux nécessitent un prétraitement des données pour diviser la zone de travail en plusieurs zones.

Questions ouvertes

En fonction de l'état de l'art, ce travail aborde un domaine restreint de questions. Des recherches supplémentaires sont nécessaires pour étudier : premièrement, comment les véhicules ont un impact sur le réseau routier ? ; deuxièmement, comment le comportement des véhicules est affecté dans un environnement dans lequel ils contribuent eux-mêmes à la congestion ?.

A notre connaissance, l'approche proposée dans cette recherche est l'une des premières à aborder le problème du covoiturage dans une perspective à grande échelle. Nous suggérons notre stratégie comme un futur standard qui, grâce à son indépendance vis-à-vis de la ville et sa capacité à faire face aux changements, peut être largement appliqué dans toutes les zones urbaines. Bien sûr, elle présente quelques faiblesses, introduisant de nouvelles questions à étudier.

Nous avons présenté la DRBD comme un répartiteur capable de fonctionner en tant qu'unité centralisée. Le DRBD travaille sur les demandes non servies en temps réel pour prédire les zones dans lesquelles il pourrait y avoir des demandes non satisfaites dans un futur proche. Sa tâche consiste à prédire les zones et à relocaliser les véhicules dans une zone où leur présence pourrait être plus rentable. Si le DRBD est utilisé comme unité centralisée, il pourrait être possible de l'exploiter comme agent externe pour recommander le nombre de véhicules nécessaires pour faire face à la demande entrante. Cet agent devrait être capable d'agir en temps réel, en contrôlant

dynamiquement la flotte. Libérer les agents lorsqu'ils ne sont pas nécessaires ou allouer de nouvelles ressources si nécessaire.

Perspectives

Avant tout, nous avons prévu de réaliser d'autres expériences visant à tester le principal avantage de l'approche proposée. Sa capacité à être répliquée dans une autre ville sans avoir à modifier le modèle sous-jacent. La prochaine étape consiste donc à trouver un jeu de données valable, concernant les demandes de transport, pour évaluer notre proposition.

L'amélioration que nous suggérons d'étudier en termes d'apprentissage, les véhicules pourraient être activés à l'aide d'un modèle d'apprentissage par renforcement pour affiner leurs comportements en réponse aux nouveaux modèles de demande qui émergent. Le rééquilibrage pourrait être encore amélioré en tenant compte de la congestion du trafic en temps réel lors du choix du cluster à déplacer. Une faiblesse de notre proposition concerne la politique de répartition appliquée par la DRBD pour relocaliser les véhicules. La stratégie actuelle est naïve, un véhicule est relocalisé dans une zone en fonction de la probabilité de distribution définie par le rééquilibrage.

La version actuelle de la DRBD ne prend pas en compte la position du véhicule qui doit être déplacé. Par conséquent, plutôt que de déplacer chaque véhicule vers son cluster le plus proche, il pourrait se produire l'inverse. Un scénario indésirable qui peut se produire et qui est illustré par la figure 4.8, dans laquelle deux véhicules V_1 , V_2 , demandent à être relocalisés et la DRBD a trouvé deux zones C_1, C_2 , dans lesquelles les véhicules peuvent être relocalisés. Par exemple, V_1 est proche de C_1 et V_2 est proche de C_2 . Maintenant, chaque véhicule a la même probabilité $Prob$ d'être relocalisé dans l'un de ces clusters, a tangible. La conclusion est que chaque véhicule est affecté à la zone la plus éloignée.

Si une politique plus sophistiquée, permettant de réexpédier les véhicules vides, était utilisée, il serait possible de diminuer la distance parcourue sans service,

ce qui permettrait d'économiser des véhicules et de réduire l'impact de ces derniers sur le réseau routier.



Figure 4.8 – Deux véhicules qui demandent à être relocalisés avec deux zones trouvées par la DRBD.

Enfin, nous suggérons d'étudier la granularité dont disposent les véhicules pendant la tâche de conduite. Actuellement, un véhicule peut décider d'atteindre sa destination ou de servir une nouvelle demande. Une fois qu'une décision a été prise, les véhicules ne peuvent pas la changer, c'est-à-dire que si un véhicule décide d'atteindre sa destination, sa prochaine étape d'évaluation sera effectuée une fois arrivé à celle-ci.

Si, pendant la conduite, une nouvelle demande, qui pourrait être servie par le même véhicule, a été générée sur son chemin, l'implémentation actuelle ne permet pas de la servir. Une implémentation future pourrait évaluer ce type de situation en permettant à chaque véhicule de se conduire pas à pas tout en atteignant sa destination. Une façon possible de le faire est de diviser chaque trajet unique en N points, puis le véhicule conduit pas à pas d'un point à un autre jusqu'à ce que la destina-

tion finale soit atteinte. À chaque étape, le véhicule peut potentiellement évaluer de nouvelles demandes à servir. Une fois que l'approche aura été correctement affinée, il faudra procéder à des évaluations de l'impact de notre flotte sur le réseau routier.

Mes contributions scientifiques

Ce travail fait l'objet de :

- **Une publication internationale** dans le journal "Computación y Sistemas", ISSN :2007-9737 dont l'intitulé est : "Ride sharing using dynamic rebalancing with PSO clustering A Case Study of NYC".
- **Une communication internationale** dans le 11th International Conference on Information Systems and Advanced Technologies "ICISAT'2021", lintitulé est : "Using dynamic rebalancing Taxi with PSO clustering A Case Study of NYC".
- **Une communication nationale** dans le 1st National Conference on Applied Computing and Smart Technologies ACST'21 July 10, 2021, Ecole Supérieure en Informatique, Sidi Bel Abbès, Algeria, l'intitulé est : "Using dynamic rebalancing Taxi with PSO clustering A Case Study of NYC".

Références

- [AB14] Vincent ARMANT et Kenneth N. BROWN. “Minimizing the Driving Distance in Ride Sharing Systems”. In : *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI 2014-Decem* (déc. 2014), p. 568-575. ISSN : 10823409. DOI : [10.1109/ICTAI.2014.91](https://doi.org/10.1109/ICTAI.2014.91). URL : https://www.researchgate.net/publication/274695043_Minimizing_the_Driving_Distance_in_Ride_Sharing_Systems.
- [AGA19] Abubakr ALABBASI, Arnob GHOSH et Vaneet AGGARWAL. *DeepPool : Distributed Model-free Algorithm for Ride-sharing using Deep Reinforcement Learning*. Nov. 2019.
- [AMB16] Vincent ARMANT, Nahid MAHBUB et Kenneth N. BROWN. “Maximising the number of participants in a ride-sharing scheme : MIP versus CP formulations”. In : t. 2016-Janua. 2016. DOI : [10.1109/ICTAI.2015.123](https://doi.org/10.1109/ICTAI.2015.123).
- [AO04] Samad AHMADI et Ibrahim H. OSMAN. “Density based problem space search for the capacitated clustering p-median problem”. In : *Annals of Operations Research* 131 (1-4 2004). ISSN : 02545330. DOI : [10.1023/B:ANOR.0000039511.61195.21](https://doi.org/10.1023/B:ANOR.0000039511.61195.21).
- [ASN12] Chandramouli ANANDARAMAN, Arun Vikram Madurai SANKAR et Ramaraj NATARAJAN. “A New Evolutionary Algorithm Based on Bacterial Evolution and Its Application for Scheduling A Flexible Manufacturing System”. In : *Jurnal Teknik Industri* 14 (1 2012). ISSN : 1411-2485. DOI : [10.9744/jti.14.1.1-12](https://doi.org/10.9744/jti.14.1.1-12).

-
- [Bat+18] Kanika BATHLA et al. "Real-Time Distributed Taxi Ride Sharing". In : t. 2018-Novem. 2018. DOI : [10.1109/ITSC.2018.8569315](https://doi.org/10.1109/ITSC.2018.8569315).
- [BCT03] Christine BOURJOT, Vincent CHEVRIER et Vincent THOMAS. "A new swarm mechanism based on social spiders colonies : From web weaving to region detection". In : *Web Intelligence and Agent Systems* 1 (1 2003). ISSN : 15701263.
- [BH67] G. H. BALL et D. J. HALL. "A clustering technique for summarizing multivariate data." In : *Behavioral science* 12 (2 1967). ISSN : 00057940. DOI : [10.1002/bs.3830120210](https://doi.org/10.1002/bs.3830120210).
- [BHS97] Thomas BACK, Ulrich HAMMEL et Hans Paul SCHWEFEL. "Evolutionary computation : Comments on the history and current state". In : *IEEE Transactions on Evolutionary Computation* 1 (1 1997). ISSN : 1089778X. DOI : [10.1109/4235.585888](https://doi.org/10.1109/4235.585888).
- [Bon+99] Eric BONABEAU et al. *Swarm intelligence : from natural to artificial systems*. Oxford university press, 1999.
- [Bou17] Mohamed Amine BOUDIA. "Optimisation, intégration des données et découverte de connaissances à partir des données du web, methaheuristique". Université de Dr Taher Moulay, Saida, Algérie, 2017.
- [BOU17a] Hadj Ahmed BOUARARA. "A survey of computational intelligence algorithms and their applications". In : *Handbook of Research on Soft Computing and Nature-Inspired Algorithms* (2017). DOI : [10.4018/978-1-5225-2128-0.ch005](https://doi.org/10.4018/978-1-5225-2128-0.ch005).
- [BOU17b] Hadj Ahmed BOUARARA. "Les méta-heuristiques et les techniques bio-inspirées dans la recherche d'information dans le web social". Université de SAIDA, 2017.
- [BR03] Christian BLUM et Andrea ROLI. "Metaheuristics in Combinatorial Optimization : Overview and Conceptual Comparison". In : *ACM Computing Surveys* 35 (3 2003). ISSN : 03600300. DOI : [10.1145/937503.937505](https://doi.org/10.1145/937503.937505).
- [BS11] Janine M BENYUS et Céline SEFRAOUI. *Biomimétisme : quand la nature inspire des innovations durables*. 2011.
-

-
- [Cas+21] Alberto CASTAGNA et al. "Demand-responsive rebalancing zone generation for reinforcement learning-based on-demand mobility". In : *AI Communications* 34 (1 2021). ISSN : 09217126. DOI : [10.3233/AIC-201575](https://doi.org/10.3233/AIC-201575).
- [CDM+92] Alberto COLORNI, Marco DORIGO, Vittorio MANIEZZO et al. "An Investigation of some Properties of an" Ant Algorithm"." In : t. 92. 1992.
- [CK02] Maurice CLERC et James KENNEDY. "The particle swarm-explosion, stability, and convergence in a multidimensional complex space". In : *IEEE Transactions on Evolutionary Computation* 6 (1 2002). ISSN : 1089778X. DOI : [10.1109/4235.985692](https://doi.org/10.1109/4235.985692).
- [Dat+08] T. DATTA et al. "Improved adaptive bacteria foraging algorithm in optimization of antenna array for faster convergence". In : *Progress In Electromagnetics Research C* 1 (2008). ISSN : 15309681. DOI : [10.2528/PIERC08011705](https://doi.org/10.2528/PIERC08011705).
- [DG97] Marco DORIGO et Luca Maria GAMBARDELLA. "Ant colony system : A cooperative learning approach to the traveling salesman problem". In : *IEEE Transactions on Evolutionary Computation* 1 (1 1997). ISSN : 1089778X. DOI : [10.1109/4235.585892](https://doi.org/10.1109/4235.585892).
- [Dor12] Abbas El DOR. "Improvement of particle swarm optimization algorithms (applications in image segmentation and electronics)". PARIS-EST-Université (770839901), 2012. URL : <http://www.sudoc.fr/167495402>.
- [DOR92] Marco DORIGO. "Optimization, learning and natural algorithms". In : (1992).
- [EK95] Russell EBERHART et James KENNEDY. "New optimizer using particle swarm theory". In : 1995. DOI : [10.1109/mhs.1995.494215](https://doi.org/10.1109/mhs.1995.494215).
- [Eng99] Andries Petrus ENGELBRECHT. "Sensitivity analysis of multilayer neural networks". Stellenbosch : Stellenbosch University, 1999.
- [ES00] R. C. EBERHART et Y. SHI. "Comparing inertia weights and constriction factors in particle swarm optimization". In : t. 1. 2000. DOI : [10.1109/CEC.2000.870279](https://doi.org/10.1109/CEC.2000.870279).
-

-
- [ES98] Russell C. EBERHART et Yuhui SHI. “Comparison between genetic algorithms and particle swarm optimization”. In : t. 1447. 1998. DOI : [10.1007/bfb0040812](https://doi.org/10.1007/bfb0040812).
- [Eva+01] Iordanis E EVANGELOU et al. “Data mining and knowledge discovery in complex image data using artificial neural networks”. In : 2001.
- [Fau94] Laurene FAUSETT. *Fundamentals of Neural Network : Architectures, Algorithms, and Applications*. T. C-18. 1994.
- [Fis87] Douglas H. FISHER. “Knowledge acquisition via incremental conceptual clustering”. In : *Machine Learning* 2 (2 1987). ISSN : 0885-6125. DOI : [10.1007/bf00114265](https://doi.org/10.1007/bf00114265).
- [FKB15] Daniel J. FAGNANT, Kara M. KOCKELMAN et Prateek BANSAL. “Operations of shared autonomous vehicle fleet for Austin, Texas, market”. In : *Transportation Research Record* 2536 (2015). ISSN : 03611981. DOI : [10.3141/2536-12](https://doi.org/10.3141/2536-12).
- [FLG09] Xiang FENG, Francis C.M. LAU et Daqi GAO. “A new bio-inspired approach to the traveling salesman problem”. In : t. 5 LNICST. 2009. DOI : [10.1007/978-3-642-02469-6_12](https://doi.org/10.1007/978-3-642-02469-6_12).
- [Fog62] Robert William FOGEL. “A quantitative approach to the study of railroads in American economic growth : a report of some preliminary findings”. In : *The Journal of Economic History* 22 (2 1962), p. 163-197.
- [FOR65] E. FORGY. “Cluster analysis of multivariate data : efficiency versus interpretability of classifications”. In : *Biometrics* 21 (1965).
- [Gar+08] Oscar GARCÍA et al. “Structural assessment under uncertain parameters via interval analysis”. In : *Journal of Computational and Applied Mathematics* 218 (1 2008). ISSN : 03770427. DOI : [10.1016/j.cam.2007.04.047](https://doi.org/10.1016/j.cam.2007.04.047).
- [GD18] Maxime GUERIAU et Ivana DUSPARIC. “SAMoD : Shared Autonomous Mobility-on-Demand using Decentralized Reinforcement Learning”. In : t. 2018-November. 2018. DOI : [10.1109/ITSC.2018.8569608](https://doi.org/10.1109/ITSC.2018.8569608).
-

-
- [Gol89] D E GOLDBERG. "Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Reading, MA, 1989". In : *NN Schraudolph and J.* 3 (1989).
- [GYA13] Amir Hossein GANDOMI, Xin She YANG et Amir Hossein ALAVI. "Cuckoo search algorithm : A metaheuristic approach to solve structural optimization problems". In : *Engineering with Computers* 29 (1 2013). ISSN : 14355663. DOI : [10.1007/s00366-011-0241-y](https://doi.org/10.1007/s00366-011-0241-y).
- [Hac13] Hanaa HACHIMI. "Hybridations d'algorithmes métaheuristiques en optimisation globale et leurs applications". In : (juin 2013). URL : <https://tel.archives-ouvertes.fr/tel-00905604https://tel.archives-ouvertes.fr/tel-00905604/document>.
- [Han+17] Pierre HANSEN et al. "Variable neighborhood search : basics and variants". In : *EURO Journal on Computational Optimization* 5 (3 2017). ISSN : 21924414. DOI : [10.1007/s13675-016-0075-x](https://doi.org/10.1007/s13675-016-0075-x).
- [Hol73] John H. HOLLAND. "Genetic Algorithms and the Optimal Allocation of Trials". In : *SIAM Journal on Computing* 2 (2 1973). ISSN : 0097-5397. DOI : [10.1137/0202009](https://doi.org/10.1137/0202009).
- [HW79] J. A. HARTIGAN et M. A. WONG. "Algorithm AS 136 : A K-Means Clustering Algorithm". In : *Applied Statistics* 28 (1 1979). ISSN : 00359254. DOI : [10.2307/2346830](https://doi.org/10.2307/2346830).
- [Kar05] D KARABOGA. "An idea based on Honey Bee Swarm for Numerical Optimization". In : *Technical Report TR06, Erciyes University (TR06 2005)*. ISSN : 01406736.
- [KE95a] James KENNEDY et Russell EBERHART. "Particle swarm optimization". In : t. 4. 1995, p. 1942-1948.
- [KE95b] James KENNEDY et Russell EBERHART. "Particle Swarm Optimization, Proceedings of IEEE International Conference on Neural Networks Vol. IV : 1942-1948." In : *Neural Networks* 4 (1995). ISSN : 19353812.
- [Koh98] Teuvo KOHONEN. "The self-organizing map". In : *Neurocomputing* 21 (1-3 1998). ISSN : 09252312. DOI : [10.1016/S0925-2312\(98\)00030-7](https://doi.org/10.1016/S0925-2312(98)00030-7).
-

-
- [Koz+92] John R KOZA et al. "Evolution of subsumption using genetic programming". In : 1992, p. 110-119.
- [LK79] T. M. LILLESAND et R. W. KIEFER. "Remote sensing and image interpretation." In : *Remote sensing and image interpretation*. (1979). ISSN : 00167398. DOI : [10.2307/634969](https://doi.org/10.2307/634969).
- [LT08] I. E. LAGARIS et I. G. TSOULOS. "Stopping rules for box-constrained stochastic global optimization". In : *Applied Mathematics and Computation* 197 (2 2008). ISSN : 00963003. DOI : [10.1016/j.amc.2007.08.001](https://doi.org/10.1016/j.amc.2007.08.001).
- [LV11] Dennis LUXEN et Christian VETTER. "Real-time routing with OpenStreet-Map data". In : 2011, p. 513-516.
- [MMA22] Moustafa MAASKRI, Hamou MOHAMED et Tomouh ADIL. "Ride Sharing Using Dynamic Rebalancing with PSO Clustering A Case Study of NYC". In : *Computación y Sistemas* 26 (oct. 2022). DOI : [10.13053/cys-26-2-3942](https://doi.org/10.13053/cys-26-2-3942).
- [MML14] Seyedali MIRJALILI, Seyed Mohammad MIRJALILI et Andrew LEWIS. "Grey Wolf Optimizer". In : *Advances in Engineering Software* 69 (2014). ISSN : 09659978. DOI : [10.1016/j.advengsoft.2013.12.007](https://doi.org/10.1016/j.advengsoft.2013.12.007).
- [NW12] Ben NIU et Hong WANG. "Bacterial colony optimization". In : *Discrete Dynamics in Nature and Society* 2012 (2012). ISSN : 10260226. DOI : [10.1155/2012/698057](https://doi.org/10.1155/2012/698057).
- [OL96] Ibrahim H. OSMAN et Gilbert LAPORTE. "Metaheuristics : A bibliography". In : *Annals of Operations Research* 63 (1996). ISSN : 15729338. DOI : [10.1007/bf02125421](https://doi.org/10.1007/bf02125421).
- [Omro4] Mahamed G H OMRAN. "Particle Swarm Optimization Methods for Pattern Recognition and Image Processing by". In : *Doctor* (November 2004).
- [Pan12] Wen Tsao PAN. "A new Fruit Fly Optimization Algorithm : Taking the financial distress model as an example". In : *Knowledge-Based Systems* 26 (2012). ISSN : 09507051. DOI : [10.1016/j.knosys.2011.07.001](https://doi.org/10.1016/j.knosys.2011.07.001).
- [Pas02] Kevin M. PASSINO. "Biomimicry of Bacterial Foraging for Distributed Optimization and Control". In : *IEEE Control Systems* 22 (3 2002). ISSN : 1066033X. DOI : [10.1109/MCS.2002.1004010](https://doi.org/10.1109/MCS.2002.1004010).
-

-
- [Pha+07] D T PHAM et al. "Application of the Bees Algorithm to the Selection Features for Manufacturing Data". In : 2007.
- [PHHo8] Zhenkui PEI, Xia HUA et Jinfeng HAN. "The Clustering Algorithm Based on Particle Swarm Optimization Algorithm". In : *Intelligent Computation Technology and Automation, International Conference on* 1 (nov. 2008), p. 148-151. DOI : [10.1109/ICICTA.2008.421](https://doi.org/10.1109/ICICTA.2008.421).
- [Poto6] Gavin POTGIETER. "Mining continuous classes using evolutionary computing". University of Pretoria, 2006.
- [PSSo9] Upeka PREMARATNE, Jagath SAMARABANDU et Tarlochan SIDHU. "A new biologically inspired optimization algorithm". In : 2009, p. 279-284.
- [Qui14] J ROSS QUINLAN. *C4. 5 : programs for machine learning*. Elsevier, 2014.
- [Rao71] M. R. RAO. "Cluster analysis and mathematical programming". In : *Journal of the American Statistical Association* 66 (335 1971). ISSN : 1537274X. DOI : [10.1080/01621459.1971.10482319](https://doi.org/10.1080/01621459.1971.10482319).
- [RDREo5] Pat Simpson ROY DOBBINS RUSS EBERHART. "Computational Intelligence Pc Tools [Books in Brief]". In : *IEEE Transactions on Neural Networks* 8 (3 2005), p. 817-817. ISSN : 1045-9227. DOI : [10.1109/tnn.1997.572124](https://doi.org/10.1109/tnn.1997.572124).
- [SMo9] Davoud SEDIGHIZADEH et Ellips MASEHIAN. "Particle Swarm Optimization Methods, Taxonomy and Applications". In : *International Journal of Computer Theory and Engineering* (2009). ISSN : 17938201. DOI : [10.7763/ijcte.2009.v1.80](https://doi.org/10.7763/ijcte.2009.v1.80).
- [SO98] Yuhui SHI et B.Gireesha OBAIAHNAHATTI. "A Modified Particle Swarm Optimizer". In : t. 6. Nov. 1998, p. 69 -73. ISBN : 0-7803-4869-9. DOI : [10.1109/ICEC.1998.699146](https://doi.org/10.1109/ICEC.1998.699146).
- [SP97] Rainer STORN et Kenneth PRICE. "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces". In : *Journal of global optimization* 11 (4 1997), p. 341-359.
- [Ste14] Dan STEFANOIU. *Optimisation inspirée par la nature*. 2014.
-

-
- [SVA05] João Carlos SETUBAL et Sergio VERJOVSKI-ALMEIDA. “Advances in Bioinformatics and Computational Biology - Brazilian Symposium on Bioinformatics, BSB 2005, Proceedings : Preface”. In : *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 3594 LNBI (2005). ISSN : 03029743.
- [Tso08] Ioannis G. TSOULOS. “Modifications of real code genetic algorithm for global optimization”. In : *Applied Mathematics and Computation* 203 (2 2008). ISSN : 00963003. DOI : [10.1016/j.amc.2008.05.005](https://doi.org/10.1016/j.amc.2008.05.005).
- [TZ10] Ying TAN et Yuanchun ZHU. *LNCS 6145 - Advances in Swarm Intelligence*. 2010.
- [Yan10] Xin She YANG. “A new metaheuristic Bat-inspired Algorithm”. In : t. 284. 2010. DOI : [10.1007/978-3-642-12538-6_6](https://doi.org/10.1007/978-3-642-12538-6_6).
- [YHL08] Quan YUAN, Zhiqing HE et Huinan LENG. “A hybrid genetic algorithm for a class of global optimization problems with box constraints”. In : *Applied Mathematics and Computation* 197 (2 2008). ISSN : 00963003. DOI : [10.1016/j.amc.2007.08.081](https://doi.org/10.1016/j.amc.2007.08.081).
- [YTC07] Chenguang YANG, Xuyan TU et Jie CHEN. “Algorithm of marriage in honey bees optimization based on the wolf pack search”. In : 2007. DOI : [10.1109/IPC.2007.104](https://doi.org/10.1109/IPC.2007.104).

Résumé

Le covoiturage peut améliorer l'efficacité de la mobilité urbaine en réduisant le nombre de voitures, les embouteillages et la demande de stationnement. Il existe de nombreuses recherches scientifiques dans ce domaine et dans plusieurs axes différents, notre intérêt dans cette étude étant la relocalisation des voitures vides afin qu'elles puissent être exploitées de manière optimale. Les recherches actuelles sur le rééquilibrage divisent la zone de couverture du système en zones géographiques spécifiques, mais cela est invariablement réalisé au moment de la conception du système, limitant ainsi la capacité du système à évoluer. Dans la présente étude, une méthode est proposée pour relocaliser les véhicules inoccupés en temps réel tout en tenant compte des demandes de mobilité, en utilisant une méthode bio-inspirée connue sous le nom de Particle Swarm Optimization Clustering (PSO-Clustering). L'efficacité de cette solution a été examinée à l'aide de données sur l'utilisation des taxis à New York, en examinant d'abord le système traditionnel (pas de partage de passagers, pas de rééquilibrage), puis en utilisant le système d'autopartage, et enfin les deux systèmes ensemble, le covoiturage et de rééquilibrage.

ملخص

يمكن لمشاركة السيارات تحسين كفاءة التنقل الحضري من خلال تقليل السيارات والإزدحام و كذا الطلب على مواقف السيارات. يوجد الكثير من الأبحاث العلمية في هذا المجال و بعدة محاور مختلفة ينصب اهتمامنا في هذه الدراسة على اعادة توزيع السيارات الفارغة حتي يتسنى استغلالها الأمثل. تقسم أبحاث إعادة التوازن الحالية منطقة تغطية النظام إلى مناطق جغرافية محددة ، ولكن يتم تحقيق ذلك بشكل ثابت في وقت تصميم النظام ، مما يحد من قدرة النظام على التطور. في الدراسة الحالية ، تم اقتراح طريقة لإعادة توزيع المركبات الغير المشغولة في الوقت الفعلي أثناء النظر في طلبات التنقل ، باستخدام طريقة مستوحاة من الحيوية تُعرف باسم التقسيم بواسطة خوارزمية تحسين سرب الجسيمات (PSO-Clustering) . تم فحص مدى نجاعة هذا الحل باستخدام بيانات حول استخدام سيارات الأجرة في مدينة نيويورك ، حيث تم أولاً النظر في النظام التقليدي (عدم مشاركة الركاب ، وعدم إعادة التوازن) ، ثم استخدام نظام مشاركة السيارة، وأخيراً النظامين معا نظام مشاركة الركوب وإعادة التوازن.

Abstract

The shared vehicle can improve the efficiency of urban mobility by reducing car ownership and parking demand. Existing rebalancing research divides the system coverage area into defined geographical zones, but this is achieved statically at system design time, limiting the system's adaptability to evolve. In the current study, a method has been proposed for rebalancing unoccupied vehicles in real-time while considering travel requests, using a bio-inspired method known as Particle Swarm Optimization clustering (PSO-Clustering). The solution was examined using data on taxi usage in New York City, first looking at the traditional system (no ridesharing, no rebalancing), then carpooling, and finally of both ridesharing and rebalancing.