



HAL
open science

Alliance entre les SMA et l'Optimisation au service du Covoiturage Dynamique

Corwin Fevre

► **To cite this version:**

Corwin Fevre. Alliance entre les SMA et l'Optimisation au service du Covoiturage Dynamique. Informatique. Centrale Lille Institut, 2022. Français. NNT : 2022CLIL0022 . tel-04067847

HAL Id: tel-04067847

<https://theses.hal.science/tel-04067847v1>

Submitted on 13 Apr 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

CENTRALE LILLE

THÈSE

présentée en vue d'obtenir le grade de

DOCTEUR

en

Spécialité : Informatique

par

Corwin FÈVRE

DOCTORAT DÉLIVRÉ PAR CENTRALE LILLE

Titre de la thèse :

Alliance entre les SMA et l'Optimisation au service du Covoiturage Dynamique

Soutenue le 01 décembre 2022 devant le jury d'examen :

Présidente	Mme. Mariagrazia DOTOLI	Professeure, Politecnico di Bari
Rapporteur	M. Flavien BALBO	Professeur, Mines Saint-Etienne
Rapporteuse	Mme. Dominique QUADRI	Professeure, Université Paris-Saclay
Examinatrice	Mme. Mariagrazia DOTOLI	Professeure, Politecnico di Bari
Examinatrice	Mme. Zahia GUESSOUM	Maître de conférences, Université de Reims
Examinatrice	Mme. Hayfa ZGAYA-BIAU	Professeure, Université de Lille
Directeur	M. Slim HAMMADI	Professeur, Centrale Lille Institut
Directeur	M. Philippe MATHIEU	Professeur, Université de Lille

Remerciements

Merci... ça n'est pas si simple de dire merci. On a toujours peur d'en faire trop, ou, au contraire, de ne pas témoigner assez de gratitude. On a peur d'oublier certaines personnes ou de ne pas trouver les mots. J'essaierai ainsi, dans ces quelques lignes,
5 de faire au mieux.

Merci, donc, en premier lieu, à mes encadrants, Philippe et Slim. Merci pour la bonne humeur manifeste qui a régné à chacune de nos réunions. Merci de m'avoir poussé et de m'avoir permis de réaliser ce travail de recherche dans des conditions idéales. Merci aussi à Hayfa, qui a participé à part entière dans mon encadrement,
10 notamment dans l'élaboration de nos diverses publications scientifiques.

Merci aux rapporteurs et examinateurs de mon jury, composé de M.Balbo, Mme Dotoli, Mme Quadri et Mme Guessoum, d'avoir accepté d'entreprendre la conséquente tâche que peut être de lire et d'étudier ma thèse. Merci d'y avoir apporté toutes ces critiques constructives m'ayant permis de prendre du recul afin de parfaire mon
15 travail. Enfin, merci d'avoir accepté de parcourir plusieurs centaines de kilomètres afin d'assister à ma soutenance.

Merci à ma famille pour son soutien indéfectible et le réconfort qu'elle a pu m'apporter tout au long de mes études. Merci à mon père, Olivier, pour ces nuits de discussions autour du feu et pour ta remarquable implication dans la compréhension
20 de mon travail. Merci à ma mère, Valérie, pour ton aide précieuse, ta présence et ton écoute attentive. Merci à ma soeur, Roxane, pour ta gentillesse et ton sens du style.

Merci à tous mes amis, pour avoir essayé de comprendre mon sujet de thèse et m'avoir changé les idées avec tous ces merveilleux moments passés ensemble. Merci de m'avoir fait rire quand les conditions ne s'y prêtaient pas.

25 Enfin, merci à Kathleen, l'amour de ma vie, pour toutes ces fois où je t'ai tirée de ton sommeil pour m'aider tant sur le plan professionnel que personnel. Merci pour ton travail remarquable sur les figures et animations qui illustrent cette thèse, et ce, à de nombreuses reprises.

Résumé

30 La crise écologique et énergétique actuelle mène à la nécessité de trouver des alternatives à l'autosolisme (conduite solitaire). Le covoiturage représente une solution prometteuse, notamment sous sa forme dynamique. Un système de covoiturage dynamique s'adapte en temps réel aux changements d'état de l'environnement afin de satisfaire les utilisateurs, à savoir les passagers et les conducteurs. Pour respecter
35 ce caractère temps réel, il est nécessaire, d'une part, de maîtriser la complexité liée à l'optimisation des trajets, et d'autre part, de veiller à éviter la concentration de données sur un même système. En outre, il est important de prendre en compte les préférences de chaque utilisateur afin d'optimiser de façon personnalisée des objectifs tels que le temps d'attente, le temps de trajet ou encore le nombre de transferts.

40 Dans cette thèse, nous nous intéressons aux divers mécanismes et variantes de cette forme de covoiturage. Nous défendons une approche individu-centrée, afin de mieux respecter le caractère hétérogène des utilisateurs, et décentralisée, pour permettre une meilleure répartition du flux de données, et donc un temps de réponse plus rapide. Nous proposons d'allier les systèmes multi-agents - dont la nature est intrinsèquement
45 individu-centrée et décentralisée - à différents algorithmes d'optimisation dans le but de répondre aux problématiques énoncées. Les différents résultats issus de nos simulations attestent de la performance et de la robustesse de notre approche, notamment dans des situations complexes telles que la pénurie d'offres de covoiturage. Le système proposé dans cette thèse permet de traiter dynamiquement des requêtes complexes
50 de passagers tout en minimisant l'impact du partage de trajet pour les conducteurs, et ce, pour un large spectre de préférences et de comportements.

Mots-clés : Covoiturage dynamique, Systèmes multi-agents, Optimisation, Simulation, Mobilité, Transport

Abstract

55 The current ecological and energy crisis leads to the necessity to find alternatives to autosolism (solitary driving). Ridesharing represents a promising solution, especially in its dynamic form. A dynamic ridesharing system adapts in real time to the changing state of the environment in order to satisfy the users, namely the riders and the drivers. To respect this real-time nature, it is necessary, on the one hand, 60 to control the complexity related to the optimization of the trips, and on the other hand, to ensure that the concentration of data on a single system is avoided. Furthermore, it is important to consider the preferences of each user in order to optimize in a personalized way objectives such as waiting time, travel time or number of transfers. In this thesis, we focus on the various mechanisms and variants of this form of 65 ridesharing. We advocate an individual-centric approach, in order to better respect the heterogeneous nature of the users, and a decentralized approach, to allow a better distribution of the data flow and thus a faster response time. We propose to combine multi-agent systems - whose nature is intrinsically individual-centric and decentralized - with different optimization algorithms in order to address the stated problems. 70 The various results of our simulations attest to the performance and robustness of our approach, especially in complex situations such as the scarcity of ridesharing offers. The system proposed in this thesis allows to dynamically handle complex requests from riders while minimizing the impact of ride sharing for drivers, and this, for a wide spectrum of preferences and behaviors.

75 **Keywords** : Dynamic Ridesharing, Multi-agent systems, Optimization systems, Simulation, Mobility, Transport

Table des matières

	Introduction générale	1
	1 Les systèmes de transport : passé, présent et futur	4
80	1.1 Les systèmes de transport statiques	4
	1.1.1 Les transports en commun	5
	1.1.2 Le covoiturage statique	5
	1.2 Les systèmes de transport dynamiques	7
	1.2.1 Le transport à la demande	8
85	1.2.2 Le covoiturage dynamique	9
	1.2.3 Le transport en libre-service	10
	1.3 Les systèmes d'information et d'aide à la décision de transport	12
	1.3.1 Transports en commun : le Système d'Aide à l'Exploitation et à l'Information Voyageurs (SAEIV)	12
90	1.3.2 Transport sur réseau routier	13
	1.3.3 Les systèmes de transport intelligents : vers l'aide à la décision	17
	1.3.4 Les systèmes d'aide à la décision	20
	1.4 Les milieux ruraux : grands oubliés du schéma de transport	22
	1.4.1 La fracture du transport	22
95	1.4.2 Les actions du gouvernement	24
	1.5 Problématique de la thèse : De la multimodalité à la comodalité	28

	2 État de l’art de la recherche sur le covoiturage dynamique et la multimodalité des transports	30
	2.1 Le problème du covoiturage : les variantes et leurs caractéristiques . .	30
100	2.1.1 Les requêtes de covoiturage : différées ou à la demande	31
	2.1.2 La réservation : décision irrévocable ou révocable	33
	2.1.3 Les détours	35
	2.1.4 Le paiement	37
	2.1.5 Le transfert de passager : simple saut et multi-saut	37
105	2.1.6 Les modalités	39
	2.1.7 Le sujet d’étude : le problème du covoiturage dynamique multi-saut	40
	2.2 La modélisation du réseau de transport	40
110	2.2.1 La provenance du graphe de transport : données réelles ou artificielles	41
	2.2.2 Les niveaux de granularité d’un réseau de transport	46
	2.3 La modélisation des utilisateurs : les systèmes multi-agents et approches individu-centrées	51
	2.3.1 La notion d’agent	52
115	2.3.2 La notion d’environnement	52
	2.3.3 La notion de système multi-agents (SMA)	54
	2.3.4 Théorie du bien-être social	55
	2.3.5 Exemples d’approches SMA	56
120	2.4 La classe de complexité du problème de covoiturage : comparaison avec des problèmes d’optimisation de l’état de l’art	60
	2.4.1 Le problème du voyageur de commerce (TSP)	61
	2.4.2 Le problème du chauffeur à la demande (<i>dial-a-ride problem</i>) .	61
	2.4.3 Le problème du sac à dos (KP)	62
	2.4.4 Le problème de couverture par ensembles (<i>set cover problem</i>) .	62
125	2.4.5 Le <i>Maximum Weighted Bipartite Matching</i> (MWBM)	63

	2.4.6	Le problème du plus proche voisin d'un groupe (GNN/ANN)	64
	2.5	Positionnement face à l'état de l'art	65
	3	Approche basée des graphes artificiels : le problème de l'autostop	67
	3.1	Formulation du problème	68
130	3.1.1	Définition d'une instance du problème de l'auto-stop	68
	3.1.2	Formulations	70
	3.2	L'architecture multi-agents proposée	72
	3.2.1	Le comportement des agents passagers et conducteurs	73
	3.2.2	L'agent service de transport <i>t</i> _{sa} : responsable du tableau noir	75
135	3.3	L'approche ReTAG : Un graphe des covoiturages disponibles en temps réel	76
	3.3.1	Définition du graphe ReTAG	76
	3.3.2	Génération du graphe ReTAG	77
	3.4	Expérimentations et résultats	81
140	3.4.1	Protocole expérimental	81
	3.4.2	Une simulation avec une génération continue de conducteurs	82
	3.4.3	Une simulation avec une génération unique de conducteurs	85
	3.4.4	Discussions et Perspectives	88
	3.5	Conclusion du chapitre	89
145	4	Le covoiturage dynamique multi-saut	90
	4.1	Formulation du problème	91
	4.1.1	Spécification du réseau routier et des agents du système	92
	4.1.2	Contraintes et Formulations	96
	4.2	Le système multi-agents proposé	98
150	4.2.1	La mise à jour de la perception des agents	99

	4.2.2	L'algorithme d'optimisation des correspondances par force brute (BFMO)	105
	4.3	Expérimentations et résultats	109
	4.3.1	Protocole expérimental	110
155	4.3.2	Résultats	111
	4.4	Conclusion et perspectives	114
	5	Le covoiturage dynamique multi-saut avec gestion optimisée des nœuds de transfert	117
	5.1	Formulation du problème	118
160	5.1.1	Spécification du réseau routier et des agents du système	118
	5.1.2	Contraintes et Formulations	120
	5.2	Le système multi-agents proposé	122
	5.2.1	L'algorithme d'optimisation préférentielle du covoiturage (PMO)	123
165	5.2.2	L'algorithme de restriction euclidienne incrémentale sous contraintes (CIER)	124
	5.2.3	L'aspect multi-objectif de la solution proposée	129
	5.3	Expérimentations et résultats	130
	5.3.1	Création des profils utilisateurs	131
	5.3.2	Protocole expérimental	132
170	5.3.3	Résultats	132
	5.4	Conclusion et perspectives	137
		Conclusion générale	139
		Bibliographie	142

Table des figures

175	1.1	Nombre de véhicules nécessaires pour acheminer 45 personnes	6
	1.2	Un exemple de voie HOV	6
	1.3	Le libre-service avec borne	11
	1.4	Le libre-service sans borne	11
	1.5	Un pupitre SAEIV Ineo dans un tramway	13
180	1.6	Un centre de commandement et de régulation de la RATP	13
	1.7	Écran d'information en temps réel dans un bus du réseau Ilévia ²¹	13
	1.8	Recherche d'itinéraire multimodale d'Ilévia	14
	1.9	Un panneau à messages variables autoroutier	15
185	1.10	Une camionnette de la sécurité routière avec un panneau à messages variables	15
	1.11	Capture de l'application Waze	16
	1.12	Les différents signalements possibles pour les utilisateurs de Waze	16
	1.13	Axes de recherche des Systèmes de Transport Intelligents	18
	1.14	Recherche d'itinéraire sur Google Maps	20
190	1.15	Les différents modes de déplacement domicile-travail	23
	1.16	Dépenses de transport selon la taille de l'aire d'attraction des villes	24
	1.17	Vignettes Crit'Air	26
	2.1	Une fenêtre de temps de trajet	31
	2.2	Optimisation du covoiturage par horizon mobile	34

195	2.3	Les différents patterns de covoiturage à saut unique	36
	2.4	Un exemple de covoiturage multi-saut	38
	2.5	Un exemple d'itinéraire comodal	39
	2.6	Un exemple de graphe de réseau routier	41
	2.7	Un exemple de requête sur l'API OpenStreetMap	42
200	2.8	Un exemple de résultat sous forme XML	42
	2.9	Un exemple de résultat sous forme graphique	43
	2.10	Un exemple de graphe obtenu avec la librairie OSMNX	44
	2.11	Le graphe artificiel de Sioux Falls	45
	2.12	Un graphe partitionné et modélisé à partir de l'île de Manhattan	45
205	2.13	Un exemple de découpe spatiale R-Arbre et de l'arbre résultant	48
	2.14	Exemple de requête de chevauchement d'un point dans l'espace	49
	2.15	Exemple de requête de chevauchement d'une zone dans l'espace	49
	2.16	Un exemple de nœuds en commun dans plusieurs couches de transport	50
	2.17	Représentation du réseau multiplexé de Manhattan	50
210	2.18	Exemple d'un graphe de transfert	51
	2.19	Exemple d'un graphe co-modal	51
	2.20	Diverses possibilités de modélisation des obstacles	54
	2.21	Un agent en interaction avec son environnement et les autres agents	55
	2.22	Les différentes perceptions d'un agent automobiliste	57
215	2.23	Capture d'écran d'un modèle de trafic coopératif	58
	2.24	Illustration du processus de prise de décision des agents voyageurs	60
	2.25	Problème du covoiturage sous la forme d'un <i>Maximum weighted bipartite matching</i>	63
	2.26	Exemple d'un problème ANN dans un réseau routier	64
220	3.1	Diagramme illustrant une instance de problème de l'autostop	69
	3.2	Le diagramme séquentiel des agents du système	74

	3.3	Diagramme du processus d'identification de l'ensemble des positions atteignables	76
	3.4	Exemple d'un graphe ReTAG	78
225	3.5	Exemples de graphes d'infrastructures routières générés et étudiés dans nos simulations	82
	3.6	Nombre de pas de simulation nécessaire pour achever une instance . .	84
	3.7	Temps de trajet moyen des agents passagers	84
	3.8	Temps d'attente moyen des passagers	85
230	3.9	Proportion de demandes de passager satisfaites	86
	3.10	Temps de trajet moyen des passagers	87
	3.11	Temps d'attente moyen des passagers	87
	4.1	Exemple d'un covoiturage multi-saut	92
	4.2	Le diagramme séquentiel des agents du système	97
235	4.3	Exemple de rectangle englobant minimal (MBR)	101
	4.4	Exemple de calcul d'une perception	102
	4.5	Indexation R-Arbre	104
	4.6	Structure R-Arbre	104
	4.7	Candidats au covoiturage et zone de transfert	108
240	4.8	Perceptions multiples d'un conducteur	109
	4.9	Le graphe étudié : une section du côté nord de la ville de San Francisco	110
	4.10	La distance moyenne de trajet des conducteurs	112
	4.11	Le taux de service	113
	4.12	La distance totale de trajet économisée	114
245	4.13	Le temps d'attente des passagers	115
	5.1	Le diagramme séquentiel des agents du système	121
	5.2	Exemple d'un MINDIST	128
	5.3	Distance additionnelle moyenne de détour des conducteurs	133

	5.4	Nombre moyen de véhicules impliqués dans le covoiturage	134
250	5.5	Temps d'attente moyen d'un passager	135
	5.6	Taux de service du système	136

Liste des tableaux

	1.1	Récapitulatifs des différentes formes de transport évoquées	28
255	2.1	Comparaison du TSP avec le problème de transport co-modal provenant de l'article [Dotoli <i>et al.</i> , 2017]	61
	5.1	Tableau des différents profils des agents passagers	131

Liste des variables usuelles

- $G = \langle V, E \rangle$: graphe du réseau routier
- V : ensemble des nœuds du graphe de réseau routier
- 260 — $v = (x, y)$: un nœud du graphe de réseau routier référencé spatialement avec des coordonnées longitudinales x et latitudinales y
- E : ensemble d'arêtes du graphe de réseau routier
- $e = (v_i, v_j)$: une arête du graphe routier reliant le nœud v_i au nœud v_j .
- $SP(G, v_i, v_j, w)$: ensemble des nœuds du chemin le plus court entre v_i et v_j
- 265 — dans un graphe G en fonction du poids w
- $SPd(G, v_i, v_j, w)$: la distance du chemin le plus court
- $SPT(G, v_i, v_j, w)$: la durée du chemin le plus court
- $time$: le pas de simulation courant
- $U = D \cup R$: ensemble des agents utilisateurs du système, c'est-à-dire les
- 270 — conducteurs et les passagers
- u : un utilisateur du système
- $u.v_s$: le nœud de départ d'un utilisateur
- $u.v_l$: le nœud courant d'un utilisateur
- $u.v_e$: le nœud de destination d'un utilisateur
- 275 — $u.det_{max}$: le facteur de détour d'un utilisateur
- $u.t_s$: l'heure d'apparition d'un utilisateur
- $u.t_e$: la dernière heure d'arrivée d'un utilisateur
- $u.p$: la ou les perceptions d'un utilisateur
- D : l'ensemble des conducteurs du système
- 280 — d : un conducteur du système
- $d.trip$: l'itinéraire d'un conducteur, une liste de nœuds contigus
- $d.c_{max}$: la capacité maximum du véhicule d'un conducteur
- $d.c$: la capacité restante d'un conducteur, le nombre de sièges libre actuellement
- 285 — $d.stops$: la liste des prochains arrêts obligatoire du conducteur, une liste de nœuds
- $d.arr$: la liste des heures d'arrivée, au plus tôt, pour chaque arrêt programmé
- $d.ddl$: la liste des heures d'arrivée, au plus tard, pour chaque arrêt programmé
- $d.slk$: la liste des marges de temps avant chaque arrêt programmé
- 290 — R : l'ensemble des passagers du système
- r : un passager du système
- $r.wt_{max}$: le temps d'attente maximum pour tout le trajet du passager

- $r.t_{dep}$: l'heure de départ au plus tard du passager
- $r.pref$: les préférences du passager, un vecteur de valeur visant à pondérer la fonction objectif
- 295 — $r.M$: l'ensemble des meilleures associations de covoiturage
- $m \in M$: une association ne violant pas les contraintes et optimisant au moins un objectif
- $m.d_1, m.d_2$: les conducteurs responsables du covoiturage
- 300 — $m.total_wt$: le temps d'attente total induit par la solution de covoiturage m
- $m.nb_tsf$: le nombre de transferts induits par la solution de covoiturage m
- $m.arr_time$: l'heure d'arrivée du passager avec la solution de covoiturage m
- 305 — $m.add_dist$: la distance additionnelle totale de détour parcourue par les conducteurs avec la solution de covoiturage m
- tsa : un agent service de transport effectuant l'interface entre les conducteurs et les passagers (tableau noir)
- RT_p : un R-Arbre indexant les perceptions des agents conducteurs
- 310 — Q_1, Q_2, Q_3 : les requêtes spatiales effectuées par un passager pour déterminer les conducteurs candidats au covoiturage
- $candidates_s$: l'ensemble des conducteurs candidats au covoiturage au départ du passager
- d_s : un conducteur candidat au covoiturage au départ du passager
- 315 — $candidates_e$: l'ensemble des conducteurs candidats au covoiturage vers l'arrivée du passager
- d_e : un conducteur candidat au covoiturage vers l'arrivée du passager
- τ : l'ensemble des nœuds de la zone de transfert pour un covoiturage multi-sauts
- 320 — T : l'ensemble des nœuds de transfert considérés dans la résolution du problème ANN
- N : l'ensemble des nœuds de départ et d'arrivée des utilisateurs considérés dans la résolution du problème ANN
- RT_tau : un R-Arbre indexant l'ensemble des nœuds de transfert considérés dans la résolution du problème ANN
- 325 — $dist_{agg}(o, N)$: distance réelle agrégée entre l'objet o (nœud ou boîte englobante) et l'ensemble des nœuds N .
- $dist_{agg}^E(o, N)$: distance euclidienne agrégée entre l'objet o (nœud ou boîte englobante) et l'ensemble des nœuds N .
- 330 — gnn_dist : distances variant en fonction de la sélection du nœud de transfert
- $init_dist$: les distances pré-calculées initialement prévues par les conducteurs pour rejoindre leur destination sans covoiturage
- non_gnn_dist : distances précalculées liées au détour des conducteurs mais indépendantes du nœud de transfert
- 335 — add_dist^E : la distance euclidienne additionnelle totale de détour parcourue par les conducteurs dans le problème ANN

Introduction générale

Sur nos routes, la majorité des voitures particulières (VP) sont occupées par un seul utilisateur, c'est ce que l'on appelle l'autosolisme. Ce phénomène induit une pollution importante, une plus grande occupation des routes donnant lieu à des embouteillages et un pouvoir d'achat réduit pour la population du fait de l'augmentation constante des coûts de l'énergie. Le covoiturage apparait comme une solution efficace pour réduire ce phénomène, mais force est de constater qu'il n'est que peu pratiqué sur de courts et moyens trajets. Ce constat peut être expliqué par de nombreux facteurs : le manque d'automatisation et d'optimisation du covoiturage, la faible considération des préférences des utilisateurs ou encore l'absence de prise en compte du trafic en temps réel. En bref, les services de covoiturage existant aujourd'hui ne sont pas adaptés au contexte actuel et à la population.

C'est pourquoi, depuis de nombreuses années, la recherche s'articule autour de la compréhension et de la résolution des problématiques liées au covoiturage. De nombreux thèmes sont abordés : la réservation qui lie par un contrat un passager et un conducteur, le détour lié à l'acheminement d'un passager ou encore la possibilité, pour un passager, d'effectuer plusieurs covoitages pour atteindre sa destination, et donc d'effectuer des transferts. Chacun de ces thèmes apporte une complexité importante, tant dans la modélisation du problème que dans sa résolution.

Dans cette thèse, nous défendons une approche individu-centrée, afin de mieux respecter le caractère hétérogène des utilisateurs, et décentralisée, pour permettre une meilleure répartition du flux de données et donc un temps de réponse plus rapide. Nous proposons d'allier les systèmes multi-agents - dont la nature est intrinsèquement individu-centrée et décentralisée - à différents algorithmes d'optimisation dans le but de répondre aux problématiques énoncées.

Une telle approche nécessite la prise en compte des préférences de chaque utili-

sateur afin d'optimiser, de façon personnalisée et conjointe, les objectifs tels que le temps d'attente, le temps de trajet ou encore le nombre de transferts. De plus, un tel système de covoiturage doit susciter un intérêt autant pour les conducteurs que pour les usagers afin de maintenir une offre et une demande stable et ainsi assurer un bénéfice mutuel. La maîtrise de cette composante d'équilibre est inhérente à la performance et à la robustesse de systèmes d'aide à la décision tels que ceux présentés dans nos travaux.

Au travers de diverses simulations, nous décrivons des agents dont le comportement est proche de la réalité et proposons différentes techniques d'optimisation, notamment afin de réduire le temps de calcul et d'établir des itinéraires de covoiturage optimaux. Nous étudions en premier lieu l'autostop - une forme de covoiturage purement dynamique, générique et multi-saut - sur des graphes artificiels. Nous proposons un algorithme d'exploration et d'optimisation récursif pour le résoudre de façon optimale. En second lieu, nous réalisons un modèle s'appuyant sur des données réelles afin d'optimiser le trajet de passagers en temps réel et en fonction de leurs préférences. Nous proposons des méthodes d'indexation spatiale afin d'établir une perception dynamique pour les agents conducteurs et passagers. Cette perception leur permet d'identifier rapidement les covoiturations réalisables et les lieux de transferts optimaux.

Plan de thèse

Nous expliquons ces concepts et démontrons l'intérêt de nos travaux au travers de 5 étapes :

1. Le premier chapitre est consacré à un état des lieux des différentes solutions de transport existant aujourd'hui. Nous détaillons leurs caractéristiques, atouts et inconvénients afin de mettre en contexte cette thèse. Enfin, nous présentons notre problématique principale de recherche et nos objectifs.
2. Nous présentons dans ce deuxième chapitre les différentes méthodes de modélisation et d'optimisation associées au problème de covoiturage. Nous mettons notamment en évidence l'importance d'une approche individu-centrée, basée sur un système multi-agents, afin de modéliser un large spectre de comportements d'utilisateurs dans un système de covoiturage.
3. Ce troisième chapitre présente notre première proposition de modèle basée sur des graphes artificiels qui nous a permis d'étudier une forme de covoiturage

purement dynamique, générique et multi-saut. Nous y décrivons le problème de l'autostop et nous proposons un algorithme d'exploration et d'optimisation récursif.

400 4. Dans le quatrième chapitre, nous présentons un modèle s'appuyant sur des données réelles afin d'optimiser le trajet de passagers en temps réel et en fonction de leurs préférences. Nous présentons un modèle de perception dynamique de l'environnement permettant d'identifier rapidement les covoiturages réalisables.

405 5. Enfin, dans le dernier chapitre, nous développons les préférences de nos agents passagers. Nous détaillons une fonction multi-objectifs, pondérée par leurs préférences, permettant de simuler un large éventail de comportements réalistes. De plus, nous proposons une heuristique visant à réduire la complexité liée à l'identification du nœud de transfert optimal, l'un des problèmes les plus complexes du covoiturage multi-saut.

410 Chapitre 1

Les systèmes de transport : passé, présent et futur

Durant des années, l'Homme a cherché à réduire son temps de trajet. Peu à peu, il a construit des véhicules ainsi que des infrastructures lui permettant de circuler le plus confortablement possible. Il a ensuite naturellement cherché à optimiser ces véhicules et infrastructures afin qu'ils soient plus sûrs, plus rapides et moins chers. Dans le contexte écologique, politique et social actuel, les priorités de la population ont changé et l'industrie du transport a évolué. Cette industrie cherche maintenant à concevoir des véhicules et services plus propres et à vocation de partage. Sont alors apparus des véhicules électriques légers ou lourds en libre-service, des services de covoiturage et d'autopartage et bien d'autres systèmes de mobilité intelligente et innovante. L'information de ces systèmes de mobilité est facilement accessible, la population peut facilement interagir avec ces services grâce aux smartphones ou aux ordinateurs, en temps réel ou différé. Dans ces systèmes de transport, on distingue deux types : les systèmes de transport statiques et les systèmes de transport dynamiques.

1.1 Les systèmes de transport statiques

On définit un système de transport statique comme étant un système de transport avec des lieux et horaires de passage définis à l'avance. Il n'y a que peu d'interactions avec, ou entre, les utilisateurs et l'environnement. Ces systèmes de transport ne sont que très peu adaptatifs et c'est au passager de s'adapter à l'offre de transport. Nous

présentons deux exemples emblématiques de ce type de système de transport : les transports en commun et le covoiturage statique.

1.1.1 Les transports en commun

Les transports en commun comme les métros, tramways, bus ou trains disposent
435 d'itinéraires et d'horaires de passage fixes. Les horaires de ces moyens de transport
sont généralement consultables à chaque station ainsi que via des applications sur
smartphone ou sur internet. Ces moyens de transport n'effectuent pas de détours de
façon automatique ou sont même incapables de le faire.

Les transports en commun sur rails ont l'avantage de ne pas subir les aléas du
440 trafic routier, mais ont pour inconvénient une immobilisation du trafic sur l'intégralité
ou une partie du tronçon concerné en cas de panne. Ils sont dépendants de leur
infrastructure et de leur itinéraire pour circuler. Une panne est donc critique et peut
engendrer de nombreuses répercussions néfastes sur le reste du schéma de transport
en commun.

445 Concernant les bus, ils peuvent facilement effectuer des détours au détriment d'une
éventuelle suppression de station. Ce processus de détour fonctionne correctement
quand l'information est traitée en amont et donc en différé. L'information est alors
transmise aux conducteurs des bus avec un nouvel itinéraire ainsi qu'aux utilisateurs
du réseau afin qu'ils puissent le prendre en compte pour se diriger dans le réseau de
450 transport. Cependant, dans le cas d'un événement ponctuel, et donc dans un contexte
changeant, l'information d'une route congestionnée ou impraticable a souvent des dif-
ficultés à être remontée rapidement dans le système. La généralisation d'une déviation
de l'itinéraire aux autres véhicules du système de transport ou aux utilisateurs n'est
pas automatique et peut prendre un certain temps. Les bus disposent donc d'une
455 modularité supérieure aux moyens de transport sur rails, mais ils ne peuvent pas être
considérés comme un moyen de transport dynamique.

1.1.2 Le covoiturage statique

Le covoiturage est l'action qui consiste à partager son véhicule avec un ou plusieurs
passagers. Dans un contexte statique, ce partage d'itinéraire est prévu quelques jours
460 ou heures à l'avance. Le conducteur saisit son itinéraire sur une plateforme de mise en

relation ainsi qu'un lieu et un horaire précis de départ. Les passagers peuvent ensuite chercher un ou une suite de covoiturages leur permettant d'atteindre leur destination. Dans ce cas, les passagers doivent se conformer aux préférences et à l'itinéraire du conducteur. Bien qu'une négociation soit possible entre les utilisateurs de la plateforme via un système de communication, aucune optimisation ou négociation du trajet n'est automatique. L'intérêt du covoiturage est de pouvoir partager les coûts associés au trajet ainsi que de réduire son impact sur l'environnement et la congestion du trafic (voir Fig. 1.1).

Pour motiver la population à covoiturer, les gouvernements ont mis en place un certain nombre de mesures. Dans de nombreux pays du monde, les voies réservées aux véhicules à occupation multiple (en anglais, *high-occupancy vehicle lane* ou *HOV Lane*) sont des voies réservées aux véhicules dont le nombre de passagers dépasse un certain seuil (voir Fig. 1.2). Ces voies permettent alors aux véhicules les empruntant d'éviter les bouchons des autres voies. Il en résulte un cercle vertueux entre une fluidification du trafic et un plus grand nombre de covoitureurs et de passagers dans les bus.

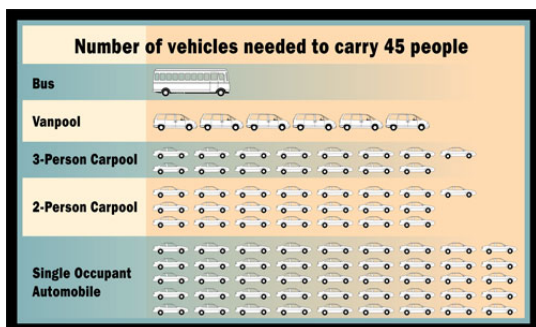


FIGURE 1.1 – Nombre de véhicules nécessaires pour acheminer 45 personnes



FIGURE 1.2 – Un exemple de voie HOV

Une autre mesure est celle des aires de covoiturage. Le principe est alors de proposer des infrastructures afin de faciliter la rencontre des covoitureurs comme des parkings gratuits sur les axes stratégiques, par exemple. Il existe plus de 2000 aires de covoiturage en France, réparties en aire de covoiturage domicile-travail et en aire de covoiturage longue durée, afin de couvrir les différents besoins de la population.

Il existe de nombreuses plateformes de mise en relation pour le covoiturage statique

comme BlaBlaCar¹ (2006) et Mobicoop² (2018) en France et en Europe, ainsi que
485 CarpoolWorld³ (2000) dans le monde. Le covoiturage statique est essentiellement
utilisé pour des trajets de longue et moyenne distance. Concernant les courts trajets
tels que les trajets domicile-travail, BlaBlaCar a récemment développé une offre de
covoiturage nommé BlaBlacar Daily⁴ (anciennement BlaBlaLines, 2017).

Bien que ces systèmes de transport statiques se complètent et permettent de circu-
490 liser correctement sur le territoire français, ils ne permettent pas d’atteindre la flexibilité
d’un véhicule personnel. En effet, ces transports statiques nécessitent de rejoindre des
emplacements précis à des horaires définis et n’autorisent donc pas un voyage de
porte-à-porte. De plus, il est nécessaire de prévoir son trajet en amont en optimisant
soit même son itinéraire.

495 Une autre forme de système de transport permet d’atteindre ces objectifs de flexi-
bilité dans l’espace et le temps, on les appelle alors des systèmes de transport dyna-
miques.

1.2 Les systèmes de transport dynamiques

On définit un système de transport dynamique comme un système de transport à
500 la demande prenant en compte l’environnement immédiat d’un utilisateur pour établir
un itinéraire. Un tel système de transport a la capacité d’évoluer en temps réel, et
ce, tout au long de son itinéraire. Il existe alors de nombreuses interactions entre les
utilisateurs et le système de transport. Un système de transport dynamique s’adapte
autour des utilisateurs, à savoir, les conducteurs et les passagers. Une infrastructure
505 fiable et efficace est alors nécessaire afin de gérer les informations relatives à ces
interactions et à l’environnement englobant les trajets. Nous présentons trois exemples
emblématiques de ces systèmes de transport : le transport à la demande, le covoiturage
dynamique et les transports libres-services.

1. www.blablacar.fr
2. www.mobicoop.fr
3. www.carpoolworld.com
4. www.blablacardaily.com

1.2.1 Le transport à la demande

510 Un système de transport à la demande est un système proposant d’acheminer un passager de son point de départ à son point d’arrivée, à sa demande. Un passager effectue une demande via un service de téléphonie ou une application et un chauffeur se déplace afin de prendre en charge le passager de façon rapide et personnalisée. Ces services de transport se composent d’une flotte de véhicule idéalement répartie par
515 secteur d’affluence et intervenant essentiellement pour de courts trajets. Il y a quelques années, le secteur du transport à la demande était essentiellement occupé par des services de taxis. La gestion de la flotte se faisait alors à l’aide d’opérateurs humains qui répartissaient les taxis en fonction des demandes des clients. L’optimisation de cette répartition en amont ou en aval des requêtes n’était donc pas automatisée et
520 sous-optimale dans sa globalité. De plus, les clients n’avaient aucune information en temps réel sur l’arrivée du chauffeur.

Par la suite, les réseaux de VTC (Voiture de Transport avec Chauffeur) tel qu’Uber⁵ (2009) et Lyft⁶ (2012) ont proposé un réseau de transport à la demande innovant. Ces sociétés ont proposé un service semblable à celui des taxis, tout en abaissant les
525 coûts associés au trajet en automatisant la chaîne de mise en relation et d’optimisation de prise en charge. La totalité du processus de transport se déroule alors via une application sur smartphone : la requête du passager, le suivi en temps réel du conducteur ainsi que le paiement. Ces applications proposent notamment d’optimiser le trajet non seulement pour le passager, mais aussi pour le conducteur, permettant
530 des voyages plus rapides, et donc augmentant le taux de prise en charge des passagers. Bien qu’il y ait eu de nombreuses controverses⁷ sur les conditions de travail des conducteurs ou sur la concurrence déloyale envers les taxis, ces nouveaux services de transports ont radicalement changé le paysage du transport en agglomération. En effet, ils ont largement été adoptés par une grande partie de la population résidant
535 en aire urbaine.

5. www.uber.com

6. www.lyft.com

7. www.lemonde.fr/societe/article/2021/09/10/uber-france-condamnee-a-verser-plus-de-180-000-euros-a-910-chauffeurs-de-taxi_6094189_3224.html

1.2.2 Le covoiturage dynamique

Le covoiturage dynamique est un covoiturage adaptatif en temps réel et à la demande. La différence avec le covoiturage statique réside dans le fait que la requête de covoiturage est cette fois spontanée et à vocation immédiate. Dans cette forme de covoiturage, les conducteurs et passagers spécifient une destination ou un itinéraire et une fenêtre de temps de trajet. L'objectif est alors d'associer au mieux les passagers et conducteurs en fonction de ces contraintes et d'assurer une optimisation tout au long du trajet. Le caractère temps réel du covoiturage dynamique nécessite une infrastructure efficace de gestion des données au vu de la quantité d'informations transitant dans le système. L'optimisation des trajets est, elle aussi, bien plus complexe à mettre en place du fait du court laps de temps disponible et de l'absence totale de prévision d'offre et de demande.

On distingue alors deux formes de covoiturage dynamique. Le covoiturage dynamique privé permet de partager la course d'un autre passager dans un système de transport à la demande (cf. section 1.2.1). Ce service permet de réduire les coûts des passagers, tout en augmentant la rémunération du chauffeur privé, au détriment du temps de trajet dû aux détours nécessaires. À titre d'exemple, Uber a implémenté ce service dans sa plateforme sous le nom UberPool (2014), ensuite renommé UberX Share à la suite de la suspension de ce service due à la pandémie de la COVID-19.

Le covoiturage dynamique entre particuliers permet quant à lui à des particuliers de partager les coûts associés à leurs déplacements. On parle alors d'une participation aux frais pour les passagers et d'une indemnisation pour le conducteur. En France, cette indemnité est indexée dans la limite du barème fiscal kilométrique. Il n'existe actuellement pas de solution de covoiturage dynamique générique, c'est-à-dire mise en place à grande échelle ou pour le grand public. En effet, plusieurs tentatives ont vu le jour lors des précédentes années, mais leur activité a dû être stoppée pour des manquements à la loi, c'est le cas d'UberPop (2014-2015) par exemple, ou souffre d'un manque de fréquentation, comme PassPassCovoiturage⁸ à Lille. Il existe tout de même des solutions de covoiturage dynamique entre particuliers d'une même agglomération ou d'une même entreprise, ces solutions sont souvent conçues par des entreprises spécialisées dans ce secteur et adoptées suite à un appel d'offres des communes ou

8. www.passpasscovoiturage.fr

société dans le besoin, comme Karos⁹ ou Klaxit¹⁰ en France. Le service de mise en relation GoCarma¹¹ propose quant à lui une solution de covoiturage dynamique entre particuliers pour la réduction des frais liés aux péages et l'utilisation des voies HOV
570 aux États-Unis.

Paradoxalement, le covoiturage dynamique est la première forme de covoiturage ayant été utilisée par la population, et ce, depuis les années 30. En effet, l'autostop permet de partager un véhicule sans réservation, c'est-à-dire sans lieu ni heure de rencontre définis. Il est historiquement non connecté et générique. Bien que cette
575 forme de covoiturage ait perdu en popularité avec le temps en Europe de l'Ouest et aux États-Unis, elle est restée populaire dans de nombreux pays du monde. On constate une reprise de son activité en occident, favorisée par la transition écologique et le coût grandissant de l'usage d'un véhicule. Une pratique plus moderne et citadine de l'autostop s'appelle le "**slugging**". Un conducteur prend alors en charge des passagers
580 non payants à des endroits clés de la ville, afin de lui permettre d'utiliser une voie HOV ou de bénéficier d'une réduction de péage. Le "slugging" est particulièrement utilisé à Washington où la congestion est un réel problème.

1.2.3 Le transport en libre-service

Le transport en libre-service permet aux utilisateurs de louer un véhicule pour, en
585 général, un trajet relativement court. Cette location est alors rapide et simplifiée, avec des dispositifs tels que des bornes automatiques ou des applications smartphones, et est souvent facturé à la minute ou via un abonnement. Il y a deux types de transport en libre-service : avec ou sans borne.

Les services de transport en libre-service avec borne permettent à un utilisateur de
590 louer tous types de véhicules en les prenant et en les déposant à des lieux précis (voir figure 1.3). La disponibilité des bornes ainsi que leur lieu de stationnement est souvent renseigné sur des applications smartphones. Des emplacements de stationnement sont réservés et permettent aux utilisateurs de se stationner plus facilement ainsi que de recharger ces véhicules souvent électriques. Le réseau coopératif Citiz¹² permet
595 de l'autopartage en libre-service dans de nombreuses villes de France. L'entreprise

9. www.karos.fr

10. www.klaxit.com

11. www.gocarma.com

12. <https://citiz.coop/>

GetAround¹³ propose, en plus de leur flotte d'autopartage, de partager son véhicule entre particuliers.

Les services de transport en libre-service sans borne, dit en "*free-floating*", s'affranchissent des bornes pour permettre aux utilisateurs de prendre et de déposer le véhicule loué à tout moment et tout endroit (voir 1.4). Les sociétés de services emploient alors des opérateurs afin de collecter les véhicules et d'effectuer les entretiens nécessaires (rechargement, réparation). C'est le cas d'entreprises telles que Lime¹⁴, Dott¹⁵ et Tier¹⁶.



FIGURE 1.3 – Des V'lille, vélos en libre-service avec borne à Lille



FIGURE 1.4 – Des véhicules en "*free-floating*", c'est-à-dire en libre-service sans borne, de l'opérateur Lime¹⁴

Bien que des services d'autopartage existent, la majorité des services de transport en libre-service concerne les micro-mobilités. La micro-mobilité désigne l'ensemble des véhicules de transport légers, électriques ou non, destinés à de la courte distance : les vélos électriques, les vélos pliables, les hoverboards, les trottinettes électriques, les gyroroues, les skateboards électriques... Reconnus comme une nouvelle catégorie de véhicules depuis 2019¹⁷, les engins de déplacement motorisés (EDPM) se présentent comme une alternative efficace en centre-ville. Il n'y a pas réellement de lois encadrant ces véhicules, si ce n'est celle du code de la route.

Les atouts sont nombreux :

— l'entretien, l'assurance et l'éventuel rechargement sont gérés par la société de

13. <https://fr.getaround.com/>

14. www.li.me/fr-fr/

15. <https://ridedott.com/fr>

16. <https://www.tier.app/fr/>

17. www.interieur.gouv.fr/Actualites/Infos-pratiques/Trottinette-electrique-et-EDPM-prudence-sur-la-route

service ;

- 615 — un véhicule peut être emprunté et déposé à des endroits différents, c'est-à-dire sans obligation de devoir le rapatrier à l'origine ;
- l'utilisateur n'a pas à conserver et stocker le véhicule.

Ces services de transport proposent une offre diversifiée quant à la tarification, l'échelle du trajet et la dynamique de réservation. Pour articuler ces services de transport, une gestion des données efficace est nécessaire, autant pour les exploitants que 620 pour les utilisateurs. Cette gestion des données est organisée par des systèmes d'information de transport.

1.3 Les systèmes d'information et d'aide à la décision de transport

625 Un système d'information (SI) est défini par l'ensemble des éléments relatifs à la gestion, le stockage, le traitement, le transport et la diffusion de l'information dans une structure ou entreprise. Dans le cadre du transport, il permet d'organiser l'offre de transport et d'informer les usagers d'un système de transport en temps réel ou différé.

630 1.3.1 Transports en commun : le Système d'Aide à l'Exploitation et à l'Information Voyageurs (SAEIV)

Concernant le cas spécifique des transports en commun, on parle de Système d'Aide à l'Exploitation et à l'Information Voyageurs (SAEIV). Le SAEIV est chargé du maintien de l'exploitation, de la gestion et de la régularité des véhicules en circulation. Ce système est séparé en 2 parties distinctes : Le Système d'Aide à l'Exploitation 635 (SAE) et le Système d'Information Voyageurs (SIV).

Le Système d'Aide à l'Exploitation (SAE)

Le Système d'Aide à l'Exploitation (SAE) est chargé de l'acquisition et la gestion des itinéraires de l'offre de transport en temps réel. Ainsi, il permet de réguler le trafic

640 via des échanges d'information entre les opérateurs et les conducteurs des véhicules, de façon manuelle ou automatique. Il combine plusieurs outils (principalement le GPS et l'odomètre) pour localiser le véhicule, et envoie ensuite ces informations au conducteur ou à l'exploitant via des interfaces comme celles des figures 1.5 et 1.6.



FIGURE 1.5 – Un pupitre SAEIV Ineo dans un tramway

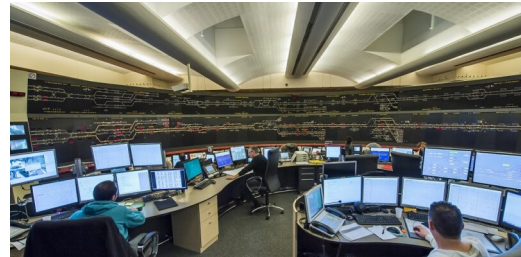


FIGURE 1.6 – Un centre de commandement et de régulation de la RATP

Le Système d'Information Voyageurs (SIV)

645 Le Système d'Information Voyageurs (SIV) gère la transmission et l'interfaçage des informations pour les voyageurs du système. Aussi appelé système d'information dynamique des voyageurs, il informe les clients de la destination des transports, de la prochaine station et de l'info trafic sur la ligne via différents outils tels que des panneaux numériques aux stations ou dans les transports, des applications sur
650 smartphone ou encore une page web (voir figures 1.7 et 1.8).



FIGURE 1.7 – Écran d'information en temps réel dans un bus du réseau Ilévia²¹

1.3.2 Transport sur réseau routier

Dans un réseau routier, le conducteur peut être professionnel ou particulier. Dans les deux cas, il est nécessaire de transmettre au conducteur des informations sur

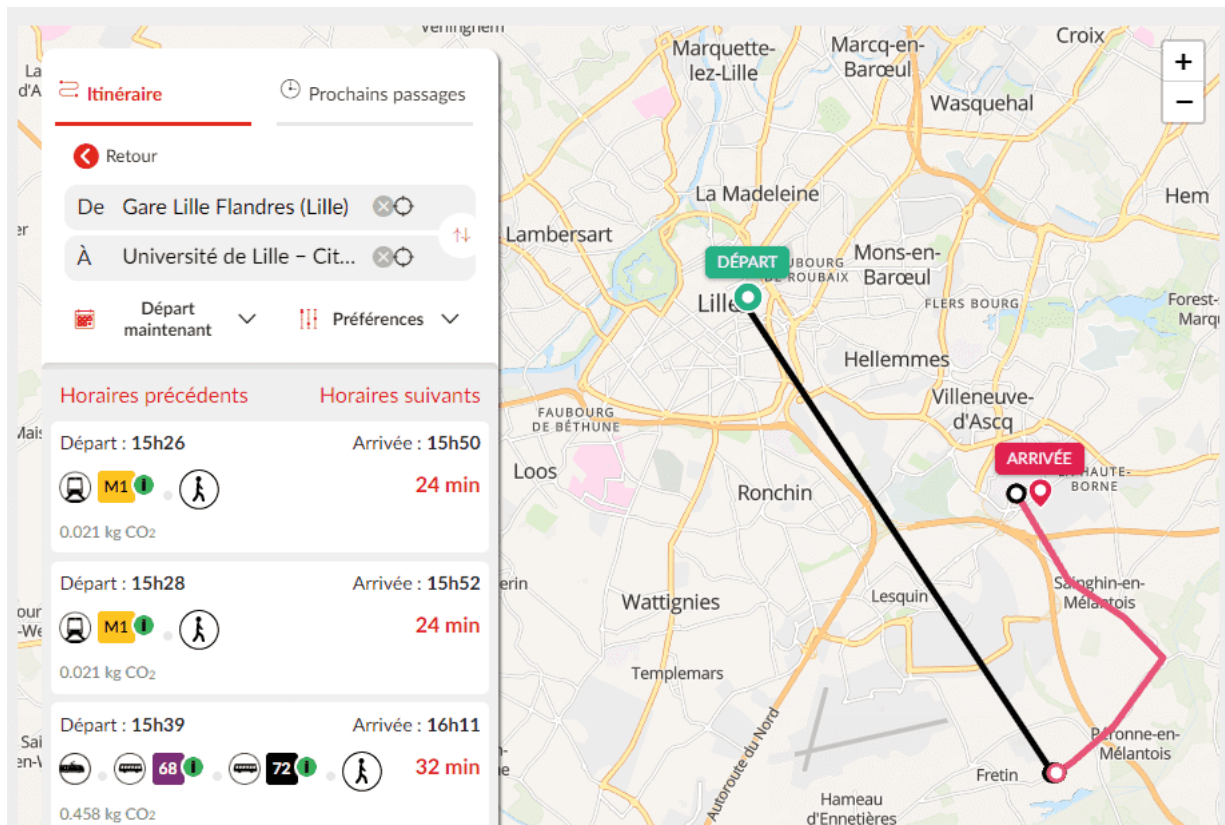


FIGURE 1.8 – Site web d’Ilévia²¹ avec l’outil de recherche d’itinéraire en temps réel proposant plusieurs solutions pour une même paire origine-destination

l’état de l’environnement routier dans lequel il évolue. On distingue alors deux sous-
 655 catégories d’information routière : la dynamique de l’information et la provenance de l’information.

La gestion temporelle de l’information routière

Il y a deux politiques de gestion temporelle de l’information routière : les informations différées ou en temps réel.

660 Les informations différées sont l’ensemble des informations n’évoluant pas dans le temps, elles témoignent des conditions normales de circulation. Il s’agit majoritairement des différents panneaux de signalisation permettant d’annoncer une interdiction, une obligation, un danger ou encore une direction. Ces informations sont le plus souvent relatives à l’organisation du trafic sur l’infrastructure routière afin d’éviter des
 665 accidents en clarifiant les conditions de circulation.



FIGURE 1.9 – Un panneau à messages variables autoroutier



FIGURE 1.10 – Une camionnette de la sécurité routière avec un panneau à messages variables

Les informations en temps réel témoignent d'un événement perturbant les conditions normales de circulation : un accident, des travaux, un objet/animal/humain sur la route, etc. Ces informations permettent de prévenir les usagers en temps réel afin qu'ils puissent anticiper un danger et éviter un sur-accident, par exemple. Des mécanismes tels que des panneaux autoroutiers, des radios-traffic ou encore des dispositifs mobiles de sécurité annoncent des messages de prévention et d'avertissement (voir figures 1.9 et 1.10). Ces dispositifs sont des équipements dynamiques de la route (ÉQUIDYN, 1995). ÉQUIDYN est l'architecture standardisée française organisant en réseau les équipements dynamiques routiers.

675 La provenance de l'information routière

Les informations routières sont collectées via différentes sources et de façons automatique et manuelle.

Concernant l'état, la source d'information principale repose sur le Système Informatisé de REcueil de DONnées routières (SIREDO¹⁸, 1991). Ce système est composé d'environ 3000 stations de recueil de mesures SOL2 permettant de rassembler des données tant pour des analyses statistiques, en temps différé, qu'à des fins de gestion du trafic et d'information de l'utilisateur en temps réel. Associés aux nombreuses caméras présentes sur le réseau routier, ces dispositifs permettent une acquisition et transmis-

18. www.trafic-routier.data.cerema.fr/siredo-systeme-national-de-recueil-de-donnees-de-a37.html



FIGURE 1.11 – Une capture de l’application Waze¹⁹ proposant un itinéraire alternatif (en mauve) afin d’éviter un bouchon (en rouge)



FIGURE 1.12 – Les différents signalements possibles pour les utilisateurs de Waze

sion rapide et efficace de l’information routière. Le traitement de ces informations est effectué par des centres de contrôle et de régulation, semblables à celui présenté en figure 1.6, qui diffusent ensuite l’information sur les supports évoqués en section 1.3.2. Dans le cas présent, l’information est recueillie manuellement et automatiquement.

Depuis une quinzaine d’années, avec l’apparition du réseau 4G et la démocratisation des smartphones, de nombreuses applications telles que Waze et Google Maps permettent aux usagers d’alerter et d’être alertés en temps réel sur les différents événements survenant sur leur trajet. Pour maintenir le caractère temps réel de l’information, cette dernière est collectée de deux façons : automatiquement ou manuellement.

Les données collectées de façon automatique sont récoltées selon le même schéma que le SAEIV : les véhicules disposent de GPS par l’intermédiaire de leur smartphone et leur position est transmise à intervalles réguliers au système d’information. Le système peut alors identifier des ralentissements, des bouchons ainsi que des routes fermées en analysant les données en temps réel.

Les données collectées de façon manuelle proviennent de l’action d’un utilisateur. En effet, un utilisateur peut à tout moment informer le système d’une route barrée, d’un bouchon, d’une anomalie sur la route (nid de poule, objet, véhicule en panne...) ou encore d’un contrôle de police, par exemple (voir Fig.1.12). C’est typiquement ce

qu'a démocratisé l'application Waze¹⁹ (2010) : une utilisation communautaire de la route visant à prévenir les usagers — de façon dynamique — des événements qui y surviennent (voir figure 1.11).

705 Ces applications fonctionnent donc sur un principe d'échange de données. Un utilisateur connecté à l'application accepte de fournir un ensemble de données tel que sa vitesse, son itinéraire et sa position et, en échange, l'application lui propose un itinéraire optimisé à partir des informations des autres usagers. L'application se comporte alors comme un système d'aide à la décision.

710 1.3.3 Les systèmes de transport intelligents : vers l'aide à la décision

De nombreux acteurs du transport ont développé des systèmes d'aide à la décision de transport afin de renforcer le caractère intelligent de leur système de transport. Les systèmes de transport intelligents (STI, étudiés notamment par le Centre
715 d'études et d'expertise sur les risques, l'environnement, la mobilité et l'aménagement : CEREMA²⁰) intègrent les nouvelles technologies de l'information et de la communication pour automatiser et optimiser le triptyque : Infrastructure-Véhicule-Conducteur (voir figure 1.13). Dans cette thèse, nous nous intéressons principalement à cet aspect, à savoir l'optimisation et l'automatisation des systèmes d'information et d'aide à la
720 décision.

Au sein d'un SI, un système d'aide à la décision permet de rationaliser, de simuler et d'optimiser l'architecture des informations, dans le but de permettre aux utilisateurs de prendre la meilleure décision possible. Dans le cas d'un système de transport, un système d'aide à la décision effectue un filtrage des solutions pertinentes en fonction des préférences d'un utilisateur et de l'état du réseau de transport en temps
725 réel.

19. www.waze.com

20. www.cerema.fr/fr/innovation-recherche/recherche/equipes/sti-systemes-transport-intelligents-infrastructure-demain

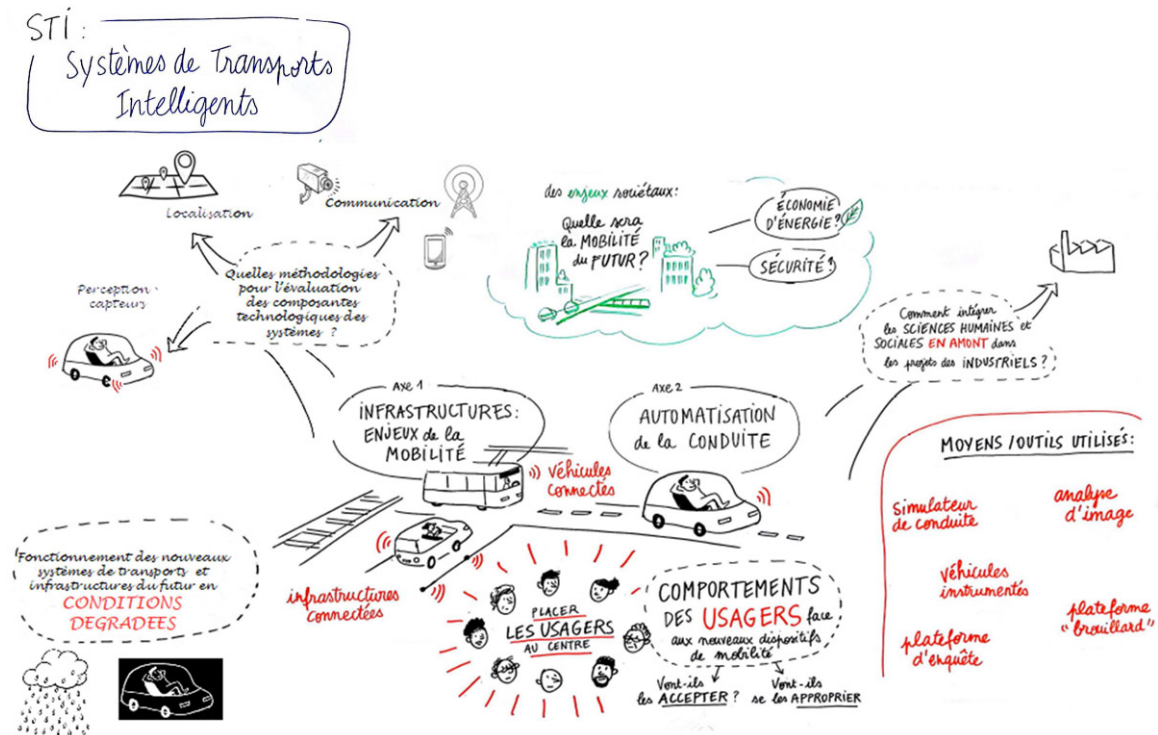


FIGURE 1.13 – Illustration proposée par le Centre d'études et d'expertise sur les risques, l'environnement, la mobilité et l'aménagement (CEREMA²⁰) pour définir les axes de recherche sur les Systèmes de Transport Intelligents (STI)

Les préférences des utilisateurs

Les préférences des utilisateurs d'un ou plusieurs réseaux de transport sont diverses. Certains cherchent à arriver le plus vite possible à leur destination quand d'autres, pas exemple, préfèrent favoriser le trajet ayant l'impact carbone le plus faible. L'hétérogénéité des préférences et profils des utilisateurs d'un système de transport entraîne la nécessité, pour un système d'aide à la décision, de proposer des solutions optimisées individuellement. En conséquence, si deux individus partent du même endroit pour rejoindre la même destination à la même heure, ils ne se verront pas forcément proposer la même solution.

D'autre part, la pondération et le nombre de critères de préférence dans un système d'aide à la décision sont cruciaux. On comprend aisément que des préférences sur la nature du transport ou le temps de trajet sont bien plus importantes que sur la couleur d'un véhicule ou un siège côté fenêtre. Il existe alors une pondération de base de chaque critère en fonction de l'importance qu'il peut avoir "en général" dans un trajet. Ce système permet de garantir un équilibre dans les solutions proposées

par le système. À cette pondération d'équilibre s'ajoute une pondération individuelle en fonction des préférences d'un individu. On obtient ainsi une pondération finale permettant d'orienter l'utilisateur vers la meilleure décision à prendre.

745 Les préférences des utilisateurs sont dynamiques par essence. Elles dépendent en effet de nombreuses variables telles que l'état du système de transport, la météo, l'importance du trajet, voire l'état psychologique de l'individu. Ainsi, un individu cherchant habituellement à minimiser son impact carbone peut décider de favoriser un trajet lui permettant de s'abriter de la pluie. De même qu'un individu habitué à
750 prendre son véhicule se verra conseiller de prendre le train à cause d'embouteillages immobilisant le trafic.

L'intégration et la modélisation de ces préférences sont donc un défi majeur dans un système d'aide à la décision. Un exemple de ces préférences est illustré dans le cadre (2) de la figure 1.14 avec l'outil Google Maps.

755 **L'intermodalité et la multimodalité des transports**

Un itinéraire peut être composé de différents modes de transports. On distingue alors deux notions complémentaires : la multimodalité et l'intermodalité [[Jonuschat et al., 2015](#)].

La multimodalité renvoie à l'existence de plusieurs modes de transport entre
760 deux lieux. Ainsi, comme on peut le voir sur la figure 1.14, la liaison Gare Lille Flandres - Cité Scientifique est multimodale, car il existe plusieurs itinéraires possibles composés de différents moyens de transport (bus, métro, TER).

L'intermodalité désigne quant à elle la combinaison de plusieurs modes de transport pour un même trajet. C'est le cas du troisième trajet proposé par Google Maps
765 dans cette même figure 1.14.

La planification d'itinéraires intermodaux et multimodaux optimisés peut être complexe. L'utilisateur doit identifier une combinaison optimale de transport en prenant en compte les différentes et nombreuses caractéristiques de chaque transport : horaires, lieu de prise en charge, titres de transport... Les systèmes d'aide à la décision
770 de transport proposent d'automatiser cette tâche.

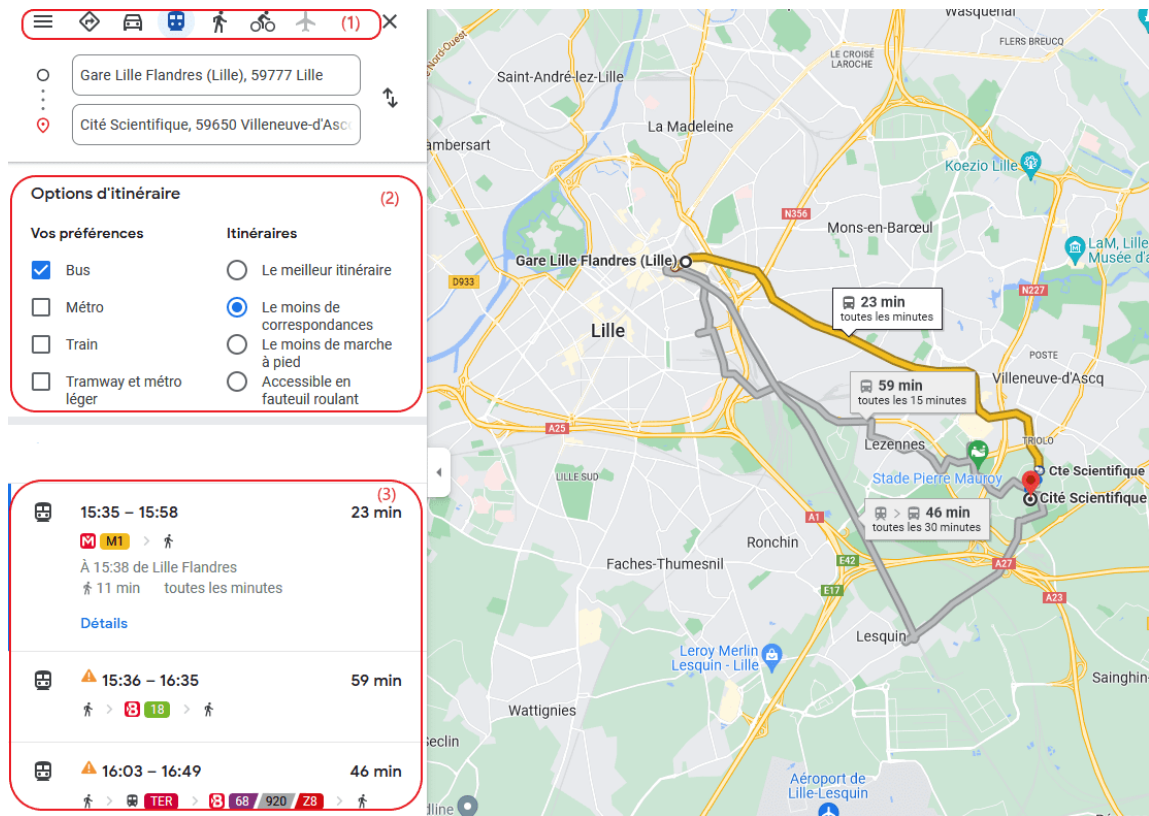


FIGURE 1.14 – Capture d’une recherche d’itinéraire sur Google Maps. (1) : Liste des différents moyens de transport disponibles; (2) : Liste des préférences/options d’itinéraires; (3) : Différents itinéraires disponibles, mono et multimodaux.

1.3.4 Les systèmes d’aide à la décision

Les systèmes d’aide à la décision centralisent les informations sur les différents modes de transport disponibles pour un même trajet à l’aide des différentes API des réseaux de transports. Ces API sont disponibles, la plupart du temps, librement sur internet et permettent un accès temps réel aux informations de transport de nombreux systèmes de transport. De nombreuses API permettant la planification d’itinéraires distribués existent comme DELFI, EU-Spirit, WSM, Journey-Web ou APII-SIM et ce sujet a notamment été récemment (2020) étudié dans la thèse [Shorten, 2020].

Dans la majorité des systèmes de transport, un chauffeur professionnel conduit un véhicule en suivant un itinéraire défini par les exploitants du service. L’aide à la décision de transport est donc essentiellement tournée vers le voyageur en lui proposant un itinéraire optimisé selon ses préférences.

À l’occasion d’un covoiturage dynamique entre particuliers, le conducteur n’est pas

professionnel et effectue le trajet dans un but personnel. Il est donc lui aussi utilisateur
785 d'un service de transport avec ses propres contraintes et préférences. Ainsi, dans les
systèmes d'aide à la décision de transport dynamique impliquant un véhicule, il faut
d'une part orienter le voyageur comme dans les autres systèmes de transport, mais
aussi d'autre part orienter le conducteur du véhicule en temps réel. En effet, l'efficacité
790 des itinéraires des conducteurs en amont et pendant la prise en charge du voyageur est
directement liée à l'efficacité du transport du ou des voyageurs, notamment pour les
détours occasionnés par le covoiturage. L'aide à la décision intervient donc pour les
deux parties qui forment un même ensemble d'utilisateurs du système de transport.

Ainsi, il y a des systèmes d'aide à la décision multimodale créés spécifiquement par
et pour un réseau de transport. C'est notamment ce que propose Ilévia²¹, le réseau
795 de transport en commun de Lille, via son outil de recherche présenté en Figure 1.8.
L'outil proposé par Ilévia propose des itinéraires composés uniquement des différentes
modalités de son réseau de transport (bus, métros, trams, vélos en libre-service...).

D'autres outils, tels que Google Maps²² et CityMapper²³, proposent d'ajouter aux
itinéraires possibles via les transports publics les différentes alternatives disponibles
800 (voir le cadre (1) de la figure 1.14). Des itinéraires mono et multimodaux peuvent
alors être proposés de façon générique en fonction de la disponibilité des services
de transports dans la zone cible (voir le cadre (3) de la figure 1.14). Ces itinéraires
peuvent alors être composés de différentes modalités telles que la voiture, les trans-
ports à la demande, le covoiturage statique, les véhicules en libre-service ainsi que le
805 vélo et la marche.

Malheureusement, les itinéraires en voiture optimisés en temps réel proposés par
GoogleMaps ou encore Waze sont à vocation monomodale et ne proposent pas de
covoiturage dynamique. De plus, à l'occasion d'un covoiturage, ils ne proposent pas
la possibilité de changer de véhicule ou de mode de transport en cours de route. On
810 appellera par la suite ce type d'itinéraire "multi-saut" ("*multi-hop*" en anglais). Enfin,
ces systèmes d'aide à la décision ne profitent qu'à une certaine partie de la population.
En effet, l'absence d'articulation entre les transports publics et la voiture, à l'occasion
d'un covoiturage ou non, oblige les habitants de milieux ruraux à utiliser leur véhicule
de façon monomodale.

21. www.ilevia.fr

22. www.google.fr/maps

23. <https://citymapper.com>

815 Pour conclure, en terme général, moins la concentration de population dans une
aire d'habitation est élevée, plus les habitants ont des difficultés à covoiturer ou à
utiliser les transports en commun. Nous démontrons cette fracture du transport dans
la prochaine section.

1.4 Les milieux ruraux : grands oubliés du schéma 820 de transport

Les services de transport et systèmes d'aide à la décision cités précédemment pro-
fitent essentiellement aux citadins des grandes agglomérations, laissant les habitants
de milieux ruraux dans une situation de précarité d'accès au transport. De ce fait,
ces habitants n'ont d'autre solution que de prendre leur véhicule pour aller travailler
825 ou plus généralement, pour se déplacer dans les zones urbaines. Ce phénomène de
fracture du transport est étudié dans ce rapport INSEE de 2019 ²⁴ et est illustré par
la figure 1.15 issue de ce même rapport.

1.4.1 La fracture du transport

On constate alors que, d'après cette étude de l'INSEE, seulement 10% des Parisiens
830 utilisent leur véhicule pour leurs déplacements pendulaires, tandis que ce taux atteint
rapidement 42% pour les habitants de banlieue, et même plus de 70% pour ceux de
la couronne d'aire urbaine. Ce même phénomène est présent dans les autres grandes
agglomérations françaises, avec une utilisation d'un véhicule personnel de 50% pour
les citadins, 75% en banlieue et enfin plus de 85% pour le reste.

835 Bien que certains métiers, comme la livraison à domicile ou les métiers de l'arti-
sanat, nécessitent l'usage d'un véhicule, cette étude permet de mettre en évidence les
carences manifestes des systèmes de transport en commun en banlieue et en milieux
ruraux. Cette nécessité d'utiliser un véhicule entraîne un certain nombre de coûts
supplémentaires aux habitants des communes à faible densité ou attractivité, comme
840 illustré par la figure 1.16 d'une autre étude de l'INSEE, datée de 2021 ²⁵.

Ainsi, la part consacrée au transport dans le revenu disponible d'un ménage d'une

24. www.insee.fr/fr/statistiques/3714237

25. www.insee.fr/fr/statistiques/5358250

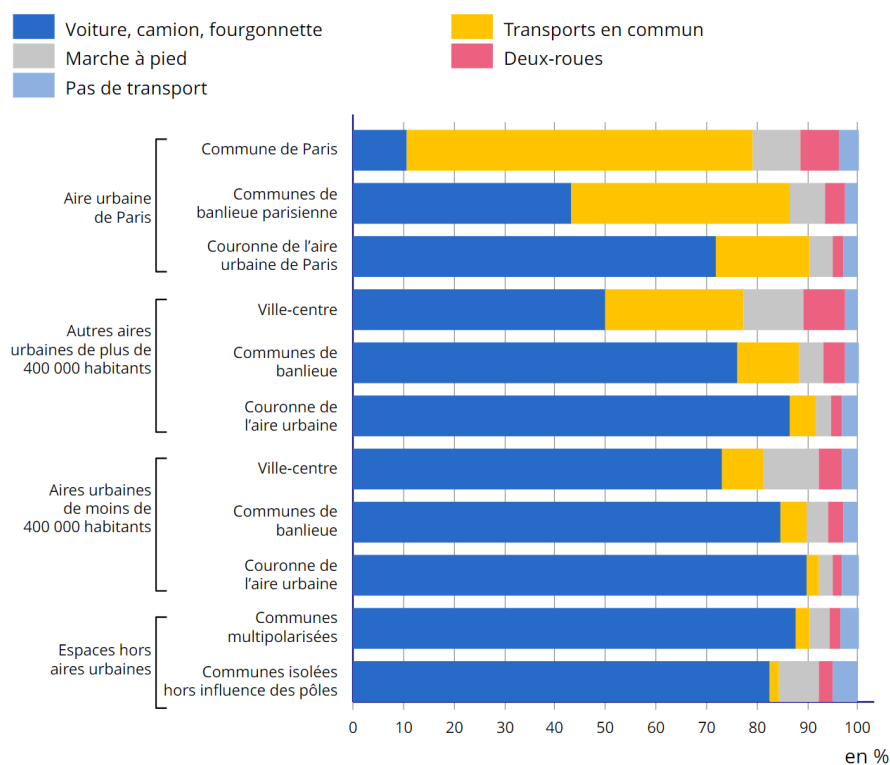


FIGURE 1.15 – Proportion des différents modes de déplacement principal domicile-travail des salariés français (source : INSEE, 2019²⁴)

commune hors de l'attraction des villes est de 4% supérieur à celle d'un ménage dans l'aire de Paris, par exemple (parts respectives de 16% et 12%). De fait, tandis qu'une grande partie des dépenses de transport d'un Parisien est consacrée aux transports collectifs, cette part n'est qu'anecdotique pour le reste de la population.

De plus, la majorité des véhicules en France sont sous-utilisés par rapport à leur capacité, si l'on se réfère à leur occupation moyenne selon le type de trajet :

- Courte distance : 1,4 passager ;
- Mixte France : 1,6 passager ;
- Longue distance : 2,2 passagers.

Ces statistiques sont issues de la cinquième enquête nationale transports et déplacements (ENTD - 2008)²⁶ et sont utilisées comme référence par l'agence de l'environnement et de la maîtrise de l'énergie (ADEME)²⁷. Un constat simple est alors qu'en augmentant cette occupation moyenne, ne serait-ce qu'à 2 ou 3 sur les courtes

26. www.statistiques.developpement-durable.gouv.fr/enquete-nationale-transports-et-deplacements-entd-2008

27. www.bilans-ges.ademe.fr/documentation/UPLoad_DOC_FR/index.htm?routier2.htm

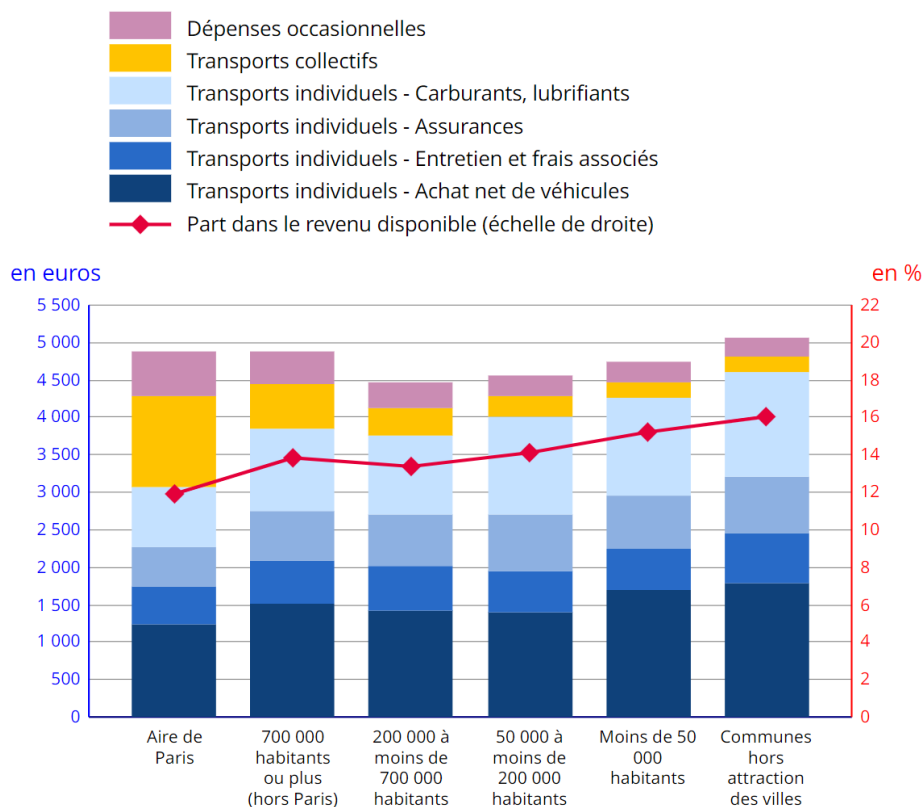


FIGURE 1.16 – Dépenses de transport et parts dans le revenu disponible des ménages selon la taille de l'aire d'attraction des villes en 2017 (source : INSEE, 2021²⁵)

855 distances, on réduit le nombre de véhicules en proportion, comme on peut le voir dans la figure 1.1.

1.4.2 Les actions du gouvernement

860 Pour sensibiliser, voire contraindre, la population à adopter une démarche plus responsable de l'utilisation de son véhicule, de nombreux États dans le monde ont mis en place des mesures restrictives, visant à avantager les citoyens utilisant des véhicules plus propres ou partageant leur véhicule.

La pollution en ville est un enjeu majeur d'écologie et de santé publique. D'après une étude de Santé publique France²⁸, la pollution de l'air est responsable de 9% de la mortalité en France, soit 40 000 personnes chaque année. L'État a donc mis en

28. www.santepubliquefrance.fr/presse/2021/pollution-de-l-air-ambient-nouvelles-estimations-de-son-impact-sur-la-sante-des-francais

865 place un certain nombre de mesures afin de réduire le nombre de véhicules polluants.
On EN distingue alors deux formes : les mesures incitatives et les mesures restrictives.

Mesures incitatives : des aides financières pour la transition écologique

Pour inciter la population à rouler dans des véhicules plus propres, l'État propose un système de bonus-malus écologique pour l'achat ou la location d'un véhicule neuf
870 ou d'occasion. Ce système a beaucoup évolué depuis la mise en place de ce qui fut baptisé la "Balladurette" en 1993, puis "Jupette" en 1995. Le principe est que la taxe appliquée aux voitures les plus polluantes (malus) finance l'aide au financement de véhicules plus propres (bonus) tels que des voitures ou des camionnettes électriques ou hybrides²⁹. Cette aide est destinée aussi bien aux particuliers qu'aux entreprises,
875 avec notamment une aide pour la transition écologique des PME³⁰.

Le bonus vélo fonctionne quant à lui de la même façon. L'état propose de financer une partie de l'achat d'un vélo à assistance électrique (VAE) ou d'une remorque électrique pour vélo³¹. De plus, de nombreuses collectivités proposent ce même genre de bonus sur les vélos classiques, les trottinettes électriques et, en général, sur la
880 majorité des mobilités douces.

Enfin, des collectivités proposent, via différentes applis de covoiturage courte distance (Klaxit, Mobicoop, Karos), de prendre en charge intégralement les coûts associés au covoiturage. Ainsi, le passager ne paie rien et le conducteur est indemnisé à hauteur de 6 centimes le kilomètre en moyenne.

885 L'objectif de ces mesures est alors multiple : inciter la population à changer de comportement, dynamiser la recherche et le développement de véhicules plus propres et enfin accélérer le remplacement écologique du parc de véhicule actuel.

Mesures restrictives : restriction de l'usage des véhicules

En juin 2015, l'État français a mis en place le certificat de qualité de l'air (aussi
890 appelé Crit'Air³²) classifiant les véhicules du moins au plus polluants sur une échelle

29. www.service-public.fr/particuliers/vosdroits/F34014

30. <https://agirpoulatransition.ademe.fr/entreprises/aides-financieres/2022/tremplin-transition-ecologique-pme>

31. www.service-public.fr/particuliers/vosdroits/F35475

32. <https://www.certificat-air.gouv.fr/>

de 0 à 5. Ce certificat, prenant la forme d'une vignette à apposer sur le véhicule (voir figure 1.17), est alors obligatoire en cas de pic de pollution et/ou dans les Zones à Faibles Émissions (ZFE).



FIGURE 1.17 – Vignettes Crit'Air

En fonction de la catégorie du certificat, le véhicule peut bénéficier de condition de stationnement et de circulation favorables. Des mesures semblables ont été mises en place dans de nombreuses villes du monde, comme des péages ou taxes de circulation (Londres, Oslo...), des zones de basses émissions (Anvers, Bruxelles...) ou encore des réductions de vitesses en centre-ville et sur les axes des périphéries. Certaines villes ont pour objectif, à terme, d'interdire l'accès de leur centre-ville aux voitures d'ici à quelques années.

Une solution peut alors être de laisser les véhicules à la sortie de la ville et de proposer aux voyageurs de terminer leur trajet via des transports en commun, ces derniers disposant d'un impact carbone bien moindre. Les parkings relais, situés en périphérie des grandes métropoles et proches des terminus des lignes de transport principales, proposent aux conducteurs de stationner leurs véhicules gratuitement à condition d'utiliser les transports en commun à disposition. Ces infrastructures proposent un double bénéfice : d'une part, elles permettent aux usagers d'éviter la chronophagie de la congestion et de la recherche de stationnement en centre-ville. Et d'autre part, elles contribuent globalement à réduire l'émission de carbone et de bruit en aire urbaine. Le parking relais est alors prometteur et se présente comme un élément important dans la logique des transports multimodaux.

La loi d'orientation des mobilités

Le ministère français chargé des transports a émis un rapport en 2021 encadrant et définissant une feuille de route pour le covoiturage en France³³. Dans ce rapport, le covoiturage est défini comme : *"l'utilisation en commun d'un véhicule terrestre à*

33. www.ecologie.gouv.fr/covoiturage-en-france-avantages-et-reglementation-en-vigueur

moteur par un conducteur et un ou plusieurs passagers, effectuée à titre non onéreux, excepté le partage des frais, dans le cadre d'un déplacement que le conducteur effectue pour son propre compte". Il y est ensuite légalement décrit les limites du covoiturage, notamment sur les échanges financiers (ex. un barème kilométrique (0,20€/km par passager non soumis à la TVA), avant d'évoquer les chiffres clés du covoiturage en France.

Dans ces chiffres clés, on retrouve notamment un objectif de 3 millions de trajets quotidiens en covoiturage, contre 900 000 aujourd'hui. Pour atteindre cette cible, le ministère propose un ensemble d'actions dans la loi d'orientation des mobilités (LOM) promulguée le 24 décembre 2019 :

- le versement d'une allocation aux conducteurs et aux passagers dans le cadre de déplacements en covoiturage ;
- l'inscription des questions de déplacements des travailleurs comme l'un des thèmes des négociations obligatoires à mener dans les entreprises de plus de 50 salariés sur un même site ;
- l'accélération de la création de voies réservées sur les axes routiers structurants et de stationnements dédiés au covoiturage ;
- la prise en charge des frais de déplacement des salariés sur leur trajet domicile-travail effectué à vélo, en covoiturage ou à l'aide d'autres services de mobilité partagée

Ce rapport se termine sur l'avenir du covoiturage en France, et notamment sur la nécessité de coupler les offres de covoiturage avec les abonnements de transports collectifs et de les intégrer aux systèmes d'information multimodaux. Le covoiturage courte-distance se présente alors comme une solution de mobilité à part entière parmi le bouquet de solutions offertes en agglomération, complémentaires aux réseaux existants.

Cette thèse s'inscrit notamment dans le développement de nouvelles méthodes et outils permettant d'atteindre les objectifs de ce rapport.

1.5 Problématique de la thèse : De la multimodalité à la comodalité

945

Dans les sections précédentes, nous avons mis en évidence les différentes problématiques liées au transport de la personne avec un focus particulier sur la France et sur les trajets domicile-travail. Le manque d'alternative au véhicule personnel pour les différents déplacements des habitants de villes en dehors des grandes agglomérations est l'un des facteurs principaux de congestion et de la fracture du transport. Le covoiturage peut être une solution, mais les solutions existantes ne sont pas assez optimisées, génériques et dynamiques pour entrer dans les habitudes de la population. Plus généralement, l'utilisation du véhicule personnel n'est pas inscrite dans le schéma de transport multimodal existant et les outils d'aide à la décision ne le considèrent pas. Une approche comodale peut alors permettre de résoudre ce problème en effectuant la complémentarité entre transports publics et partage des trajets en voiture.

950

955

La comodalité des transports

Les systèmes de transport présentés précédemment, et résumés dans le tableau 1.1, ont vocation à être utilisés conjointement afin d'accroître la couverture du territoire et l'efficacité des déplacements. On parle alors dans ce cas d'un système de transport comodal.

960

	Monomodale	Multimodale	Comodale
Dynamique	Uber Lyft Lime/Tier/Dott	GoCarma CityMapper	
Statique	BlaBlaCar MobiCoop CarPoolWorld	Transports en commun	Karos Klaxit

TABLE 1.1 – Récapitulatifs des différentes formes de transport évoquées

La comodalité est une notion introduite en 2006 par la Commission européenne dans le domaine de la politique des transports qui désigne « le recours efficace à

965 différents modes de transport isolément ou en combinaison » afin d'atteindre une «
utilisation optimale et durable des ressources »³⁴. Elle englobe à la fois le transport
de personnes et de biens. Cette notion implique un changement de direction de la
politique de transport où l'on n'oppose plus le transport routier aux autres modes de
transport. On s'intéresse plutôt à trouver un optimum en tirant parti des espaces de
970 pertinence des différents modes et de leurs combinaisons.

Objectifs de la thèse

L'objectif de cette thèse est dès lors de créer un système d'aide à la décision de co-
voiturage pour des particuliers, leur permettant de saisir les contraintes et préférences
relatives à leurs trajets, et ainsi de se voir proposer un itinéraire dynamique et opti-
985 misé. Pour ce faire, il faut dans un premier temps développer un système similaire au
SAEIV pour le covoiturage. La couche d'exploitation (SAE) s'occupe alors des itiné-
raires des conducteurs et de leur géolocalisation. Et la couche d'information voyageur
(SIV) gère l'interfaçage des conducteurs et des partages de trajets disponibles pour
les passagers. Enfin, la couche d'aide à la décision permet aux conducteurs comme
990 aux passagers d'identifier les meilleures associations, en fonction de leur préférence et
de l'état du système, afin de permettre une optimisation du covoiturage dynamique.

Le système d'aide à la décision du covoiturage dynamique doit alors reposer sur
un système d'information de transport collectant conjointement les informations des
voyageurs et des conducteurs de façon individualisante. La quantité d'information à
985 traiter en temps réel est titanesque et nécessite de nombreuses techniques d'organi-
sation de la donnée (décentralisation, indexation...). Aussi, l'optimisation relative à
l'association des conducteurs et passagers pour le covoiturage dynamique nécessite
des algorithmes puissants avec une complexité contenue afin de pouvoir transmettre
un résultat rapide aux utilisateurs. Nous présentons un état de l'art des différentes
990 méthodes développées par la recherche pour lever ces verrous scientifiques dans le
prochain chapitre.

34. <https://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=COM:2006:0314:FIN:EN:PDF>

Chapitre 2

État de l’art de la recherche sur le covoiturage dynamique et la multimodalité des transports

995

1000

La recherche actuelle sur les mécanismes de covoiturage est riche et les problèmes étudiés sont nombreux. Nous allons dans un premier temps référencer les différentes variantes de ce problème afin de positionner le sujet d’étude de cette thèse. Nous critiquerons ensuite les différentes approches proposées dans la littérature pour clarifier notre démarche scientifique et son point de départ.

2.1 Le problème du covoiturage : les variantes et leurs caractéristiques

1005

Dans les sections [1.1.2](#) et [1.2.2](#), nous avons brièvement défini et listé différentes formes de covoiturage existant dans le monde. Cette section s’intéresse aux différentes variantes de ce problème. Nous effectuons une analyse exhaustive des différents paramètres ayant une influence majeure sur ce champ d’études. Nous verrons alors qu’il est difficile d’effectuer une taxonomie de ces problèmes tant leurs interprétations et formalisations sont différentes dans la littérature.

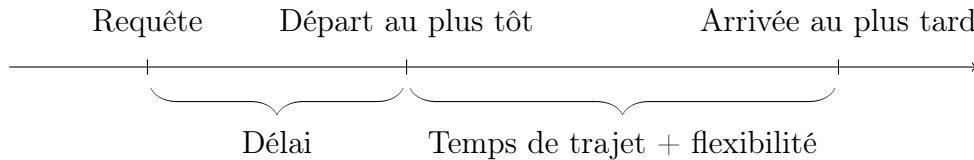


FIGURE 2.1 – Fenêtre de temps de trajet de l'article [Agatz *et al.*, 2011]

2.1.1 Les requêtes de covoiturage : différées ou à la demande

1010 Quand un passager potentiel entre dans un système de covoiturage, il émet une
 requête. Dans cette requête, il transmet au système les informations relatives à son
 itinéraire (sa position, sa destination), ses préférences et ses contraintes (heure de
 départ au plus tôt, heure d'arrivée au plus tard..). Ces informations permettent d'es-
 1015 timer la marge de manœuvre ou la fenêtre de temps du passager et donc d'effectuer
 une recherche pour l'associer à des conducteurs afin de covoiturer.

De nombreux chercheurs ont formalisé et utilisé ces fenêtres de temps (*time-*
windows) [Solomon, 1987, Agatz *et al.*, 2011, Agatz *et al.*, 2012, Herbawi et Weber,
 2012b] avec trois constantes (voir Figure 2.1) : l'heure de la requête, l'heure de dé-
 part au plus tôt et l'heure d'arrivée la plus tardive. On distingue alors un délai entre
 1020 l'heure de la requête et le départ au plus tôt de l'utilisateur, indiquant sa flexibilité
 quant au trajet et permettant d'identifier le meilleur covoiturage et donc une solution
 optimale au problème. Cette flexibilité permet éventuellement d'effectuer des détours
 (voir section 2.1.3) ou de patienter pour un covoiturage futur ou un transfert (voir
 section 2.1.5). Un covoiturage est possible si les fenêtres de temps d'un conducteur et
 1025 d'un passager coïncident. Le problème du covoiturage à fenêtre temporelle est sem-
 blable au problème de routage et d'ordonnancement de véhicules avec des contraintes
 de fenêtre temporelle qui a été démontré NP-Dur dans [Solomon, 1987].

Le problème du covoiturage a été étudié avec différentes tailles de fenêtre tempo-
 relle. La littérature fait d'abord mention de requêtes différées, c'est-à-dire effectuées
 1030 plusieurs jours ou heures à l'avance [Wolfer Calvo *et al.*, 2004, Baldacci *et al.*, 2004] et
 donc admettant une grande fenêtre temporelle. Dans ce cas, le système dispose d'une
 certaine marge de manœuvre pour optimiser globalement les requêtes de covoiturage,
 de conducteur et de passager. Le problème peut être considéré sous la forme d'un pro-
 blème d'allocation de ressource multicritère (voir 2.4). Par la suite, la recherche s'est
 1035 intéressée au problème de covoiturage à la demande. Dans cette forme du problème,
 la fenêtre de temps est très courte (de l'ordre de quelques minutes). Le problème est

alors bien différent : il n'est plus question d'optimiser globalement le système, mais de trouver rapidement une solution optimisée, approchée ou exacte, pour l'utilisateur. Nous détaillons les diverses approches pour chacun de ces problèmes plus tard, dans
1040 les prochaines sections de ce chapitre.

Il est tentant d'associer le problème du covoiturage à requêtes différées au covoiturage statique, et le problème du covoiturage à la demande au covoiturage dynamique. En réalité, de nombreuses subtilités existent et la définition du covoiturage dynamique et statique a beaucoup évolué ces 20 dernières années. Nous allons donc maintenant
1045 parcourir les différents articles mentionnant ces problèmes afin de formuler et d'argumenter notre définition de ces problèmes.

Un système de covoiturage statique est un système de covoiturage où l'offre de transport, c'est-à-dire l'itinéraire et la fenêtre de temps des conducteurs, est renseignée en amont sous la forme d'une liste d'offres. De la même façon, les passagers effectuent
1050 des requêtes avec leur itinéraire et leur fenêtre de temps, et le système tente d'associer chaque requête avec celles des conducteurs [Baldacci *et al.*, 2004]. Un système de covoiturage statique ne propose pas, en général, la possibilité de modifier les trajets des conducteurs (avec des détours par exemple). C'est donc au passager de s'adapter, à la manière d'un système de transport classique (bus-métro-train...). De plus, ce
1055 système ne considère pas l'environnement des utilisateurs en temps réel, les temps de trajets et itinéraires sont empiriques et donc prédits en fonction des conditions habituelles de circulation. Ce type de système est alors propice aux longs trajets où l'important est de rejoindre une ville à une autre et où la marge de manœuvre en termes de temps (retard) et de distance (point de rencontre) est négligeable.

Un système de covoiturage dynamique est un système de covoiturage prenant en compte l'environnement direct des utilisateurs pour effectuer des associations de covoiturage. Un tel système collecte en temps réel différents flux d'information dans l'environnement comme les itinéraires et temps de trajets des conducteurs en cours de route ainsi que les requêtes des passagers. Le caractère dynamique réside alors
1065 dans le flux continu de conducteurs et de passagers entrant, évoluant et sortant du système. Cette forme de covoiturage est majoritairement étudiée à l'échelle de grandes métropoles [Di Febbraro *et al.*, 2013, Alonso-Mora *et al.*, 2017, Tafreshian et Masoud, 2020] et dans le cadre de trajets courts et spontanés [Nourinejad et Roorda, 2016, Xu *et al.*, 2020]. En terme général, et à l'inverse du covoiturage statique, c'est aux
1070 conducteurs de s'adapter en effectuant de courts détours en temps réel afin de faciliter

l'acheminement des passagers [Furuhata *et al.*, 2013].

Dans ces deux systèmes de covoiturage, un élément capital entre en jeu : la réservation.

2.1.2 La réservation : décision irrévocable ou révocable

1075 Quand une réservation est effectuée auprès d'un conducteur, le passager s'assure
que le conducteur lui alloue une place dans son véhicule afin de l'acheminer d'un
endroit à un autre, dans un laps de temps défini. Une réservation est à double tran-
chant : d'un côté, le passager est rassuré, car il sait qu'un véhicule va le prendre en
charge, mais d'un autre côté, il doit honorer sa réservation, idem pour le conducteur.
1080 Dans ce cas, la décision de covoiturage est un contrat irrévocable : on considère que le
passager ne peut pas changer d'avis et une optimisation en cours de trajet est impos-
sible [Agatz *et al.*, 2012]. Un covoiturage dont la décision de réservation est révocable
est difficilement envisageable ou applicable avec des humains, du moins dans le cadre
d'un covoiturage avec détours (voir section 2.1.3). En effet, si un passager change
1085 d'avis alors qu'un conducteur est en train d'effectuer un détour pour le prendre en
charge, le conducteur est lésé et pourrait ne plus vouloir participer à la plateforme de
covoiturage. Cette révocation pourrait toutefois permettre d'optimiser en continu le
trajet des passagers et est applicable dans les cas où les conducteurs n'effectuent pas
de détours ou n'attendent pas les passagers. On note que la révocation d'une résér-
1090 vation pourrait tout à fait être envisageable dans un contexte de véhicule autonome
ou d'acheminement de biens.

Le type de réservation peut apporter des contraintes supplémentaires, par exemple
[Baldacci *et al.*, 2004, Wolfler Calvo *et al.*, 2004, Lloret-Batlle *et al.*, 2017] considèrent
des trajets quotidiens aller-retour, le retour devant être assuré avant de pouvoir partir.

1095 Dans un contexte de covoiturage statique [Wolfler Calvo *et al.*, 2004, Baldacci
et al., 2004], la réservation ne doit pas intervenir trop tôt, au risque d'être sous opti-
male. En effet, les requêtes étant inscrites dans le système plusieurs jours ou heures à
l'avance, il est bien possible que de nouvelles offres apparaissent entre temps et soient
plus intéressantes que celles actuellement présentes. D'autre part, la réservation ne
1100 doit pas intervenir trop tard au risque de rater une offre optimale, voire qu'il n'y en
ait plus du tout. La difficulté est donc de savoir quand faire intervenir la réservation,
cette dernière stoppant la possibilité d'une optimisation future. L'approche par **hori-**

1105 **zon mobile (rolling horizon)** décrite dans de nombreux articles tels que [Kleiner *et al.*, 2011, Agatz *et al.*, 2011, Nourinejad et Roorda, 2016] permet d'effectuer cette optimisation dans un contexte dynamique et statique (voir Figure 2.2).

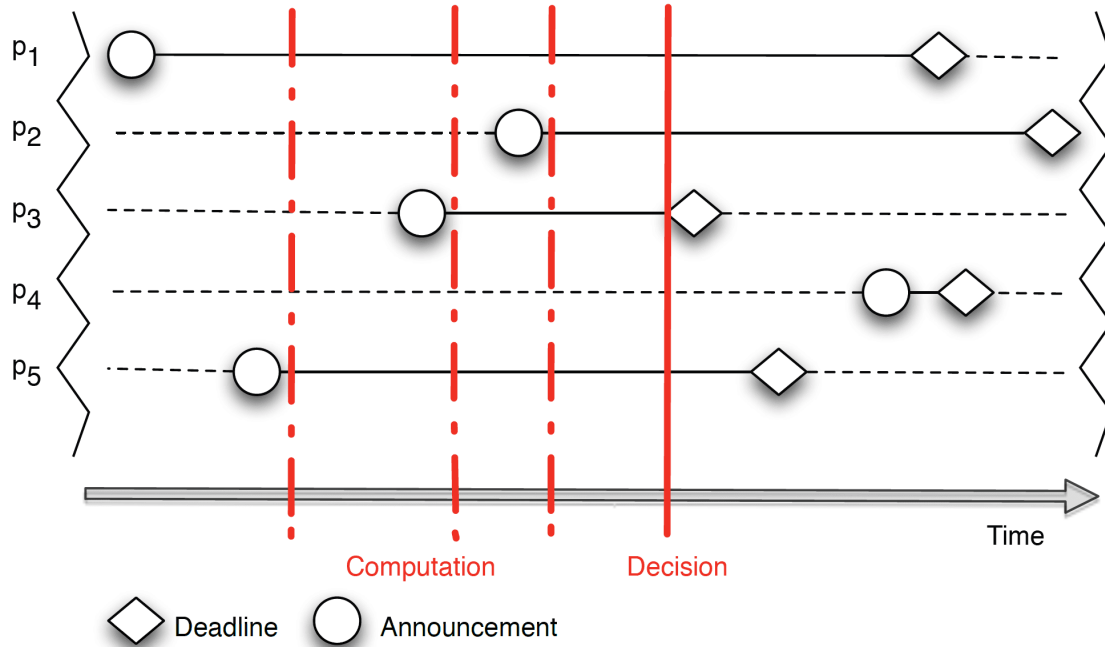


FIGURE 2.2 – Optimisation du covoiturage par horizon mobile, un conducteur reçoit les requêtes (cercles) et les deadlines (losange) des passagers p1 à p5. À chaque nouvelle requête, il recalcule l'ensemble des associations optimisant la fonction objectif définie (computation). Enfin, si la requête d'un passager atteint sa deadline, il doit prendre une décision [Kleiner *et al.*, 2011].

Dans le contexte d'un covoiturage dynamique, la marge de manœuvre pour l'optimisation est considérablement réduite, car elle est limitée par le temps d'attente des passagers (quelques minutes). On retrouve, là aussi, nos deux approches dans la littérature : avec réservation ou sans réservation. Le *slugging* et l'autostop (cf. section 1.2.2) sont des formes du covoiturage dynamique sans réservation et sans détours prévus [Ma et Wolfson, 2013, Lin *et al.*, 2016], c'est-à-dire que le passager peut à tout moment changer d'avis et descendre d'un véhicule pour en emprunter un plus intéressant. L'optimisation s'effectue alors tout au long du trajet et nécessite des mises à jour régulières de l'environnement de chaque individu. L'approche la plus étudiée est toutefois le covoiturage dynamique avec réservation. Dans ce cas, l'optimisation de l'itinéraire d'un passager est effectuée jusqu'à la réservation. Il y a donc matière à discussion sur la nature réellement dynamique d'un tel covoiturage. En effet, le covoiturage dynamique avec réservation est dynamique jusqu'à l'obtention d'un contrat de covoiturage, et donc jusqu'à la réservation. Il est d'ailleurs courant que le passager

1120 ou le conducteur disparaisse du système une fois son itinéraire réservé. Il n’y a donc plus de prise en compte de l’environnement direct du passager et il n’y a pas vraiment d’adaptation possible en cas d’événement fortuit (bouchons, accident...).

Un covoiturage dynamique avec réservation étalonnée pourrait être étudié. Prenons l’exemple d’un système de covoiturage multi-saut, c’est-à-dire impliquant plusieurs véhicules pour l’acheminement d’un passager (voir section 2.1.5). Dans la littérature, la totalité du voyage doit être réservée avant de pouvoir confirmer une prise en charge d’un passager. C’est-à-dire que si le contrat de covoiturage d’un passager s’effectue en le transférant via plusieurs conducteurs, il est nécessaire d’identifier les lieux de transfert et d’effectuer une réservation par conducteur dans le but d’assurer l’acheminement de ce passager. Cela implique d’exclure tout covoiturage ou suite de covoiturage n’acheminant pas jusqu’au bout le passager. Or, il est fort probable que d’ici à l’acheminement du passager vers sa destination, un conducteur apparaisse avec un itinéraire semblable au sien. Il existe alors une notion de prise de risque qui pourrait être évaluée en fonction de la fréquentation moyenne du secteur, par exemple.

1135 2.1.3 Les détours

L’avantage principal du covoiturage entre particuliers est la flexibilité induite par le caractère non professionnel du conducteur du véhicule [Emmerink et van Beek, 1997]. En effet, bien qu’il ait des contraintes et des préférences, le conducteur n’a pas d’obligation en termes d’horaire de passage ou d’itinéraire, contrairement à un chauffeur de bus par exemple. De plus, il n’est pas forcément déjà en train d’acheminer un passager comme peut le faire un chauffeur privé.

Ainsi, un conducteur particulier est en mesure de modifier son trajet, en temps réel ou différé, afin de récupérer ou de déposer un passager. Autrement dit, il peut effectuer des détours sur son trajet initial.

1145 Dans un système de covoiturage, ces détours doivent être limités dans l’espace et/ou le temps afin qu’ils respectent les contraintes et préférences des utilisateurs. C’est pourquoi de nombreux chercheurs ont formalisé les fenêtres de temps (*time-windows* [Agatz et al., 2011, Agatz et al., 2012, Herbawi et Weber, 2012b] afin de définir la marge de temps disponible pour effectuer ces détours. Cette fenêtre de temps est définie avec trois constantes (voir figure 2.1 : l’heure de la requête, l’heure de départ au plus tôt et l’heure d’arrivée la plus tardive [Agatz et al., 2011]).

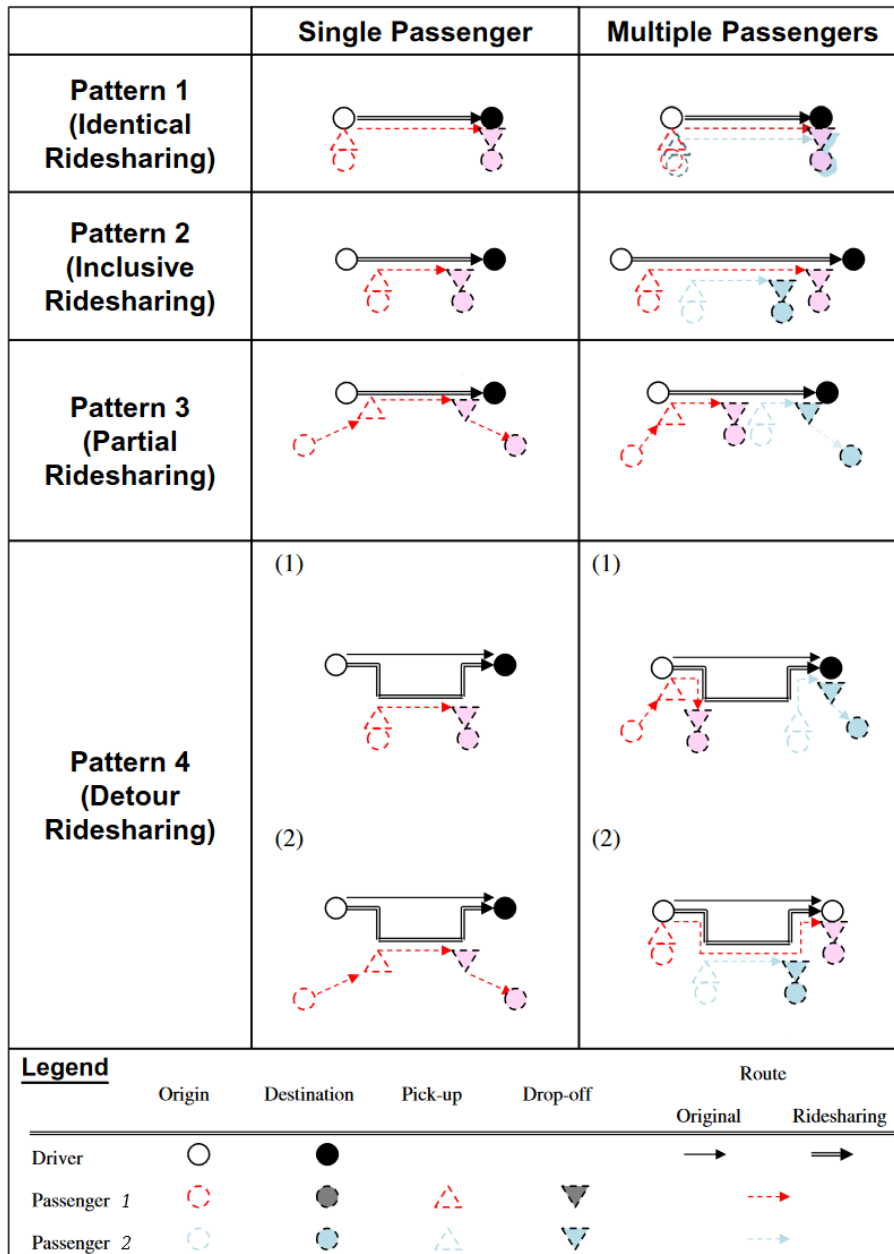


FIGURE 2.3 – Les différents patterns de covoiturage à saut unique, avec et sans détours, et pour un ou plusieurs passagers [Furuhata *et al.*, 2013]

On distingue le covoiturage avec détours du covoiturage sans détour tel que le slugging [Ma et Wolfson, 2013, Lin *et al.*, 2016] ou l'autostop. L'autostop et le slugging s'effectuent selon les patterns 1, 2 et 3 de la Figure 2.3 issue de l'article [Furuhata *et al.*, 2013], à savoir, des covoitages à l'identique, inclusifs ou partiels. Tandis que les covoitages avec détours peuvent s'effectuer selon tous les patterns de la Figure 2.3.

2.1.4 Le paiement

D'autres chercheurs se sont intéressés au paiement ou au défraiement du covoitu-
1160 rage, notamment sur des questions de rentabilité maximale, de pénétration de mar-
ché ou encore de juste prix. L'optimisation est alors dirigée vers la maximisation du
dédommagement pour le conducteur et/ou vers la minimisation des coûts pour les
passagers.

Une partie de la littérature s'oriente vers des mécanismes d'enchères VCG ([Vi-
1165 ckrey, 1961, Clarke, 1971, Groves, 1973]) pour fixer le prix. Ces mécanismes ont pour
but d'inciter des individus à révéler la vraie valeur qu'ils attribuent à un bien ou un
service. Le montant des enchères de chaque approche est alors calculé à partir de
plusieurs informations telles que la distance de trajet, le détour occasionné ou encore
en fonction des diverses préférences des individus du système.

Cette approche est NP-Dur, en effet, un passager en passe de remporter l'enchère
1170 d'un conducteur peut déjà avoir remporté une enchère plus avantageuse ailleurs. Il faut
alors considérer le second meilleur enchérisseur et de proche en proche, de nombreuses
combinaisons surgissent. De plus, certaines approches privilégient le taux de service et
donc le pourcentage de requêtes pourvues. Dans ce cas, le problème est d'identifier la
1175 combinaison optimale pour le bien commun (voir section 2.3.4). C'est dans ce contexte
hautement combinatoire que [Nisan et Ronen, 2007] propose une solution approchée
dans un contexte général et non réservé au covoiturage. Concernant le covoiturage,
[Kamar et Horvitz, 2009] propose d'effectuer des groupes de passagers de covoiturage
pour un même conducteur en évaluant un juste prix collectif. [Kleiner *et al.*, 2011]
1180 utilise un système d'enchères parallèles où les passagers sont ordonnés en fonction
de leur intérêt (financier, pratique...) pour un conducteur souhaitant covoiturer. Le
passager le plus utile remporte alors l'enchère. À l'inverse, [Nourinejad et Roorda,
2016] propose cette fois que ce soit aux conducteurs d'effectuer des enchères de Vickrey
sur chaque passager qu'il peut acheminer, le conducteur le moins cher remporte alors
1185 l'enchère d'un passager.

2.1.5 Le transfert de passager : simple saut et multi-saut

À l'occasion d'un covoiturage, un passager peut être acheminé par un seul et
unique conducteur, c'est ce que l'on nomme **le covoiturage simple saut** (ou *single-*

hop en anglais). Le problème du covoiturage simple-saut consiste à identifier le meilleur
 1190 conducteur pour acheminer un passager entre deux positions. Il peut alors s'agir d'un
 trajet direct du point de départ d'un passager à son point d'arrivée, comme on peut
 le voir dans les patterns 1,2 et 4.(2) de la Figure 2.3; d'un acheminement entre deux
 points de rencontres – à la manière de blablacar –, comme dans les patterns 3 et 4.(1);
 ou enfin une hybridation de ces patterns. C'est la première forme de covoiturage étu-
 1195 diée dans la littérature [Kamar et Horvitz, 2009, Agatz et al., 2011, Kleiner et al.,
 2011, Di Febbraro et al., 2013] de par ses similarités avec les problèmes d'allocation
 de ressources.

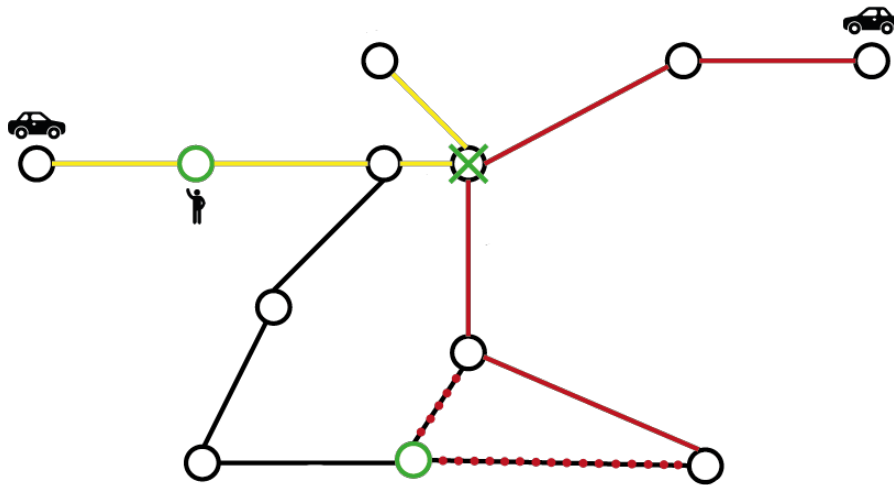


FIGURE 2.4 – Un exemple de covoiturage multi-saut (2-sauts) : le passager covoiture avec deux véhicules différents dont les itinéraires initiaux sont en jaune et rouge. Il effectue un transfert sur la croix verte. Un détour, en pointillé rouge, est nécessaire pour l'itinéraire rouge.

Il est aussi possible pour un passager d'utiliser plusieurs conducteurs, et donc
 d'effectuer des transferts afin d'arriver à sa destination (voir Figure 2.4). Cette va-
 1200 riante s'appelle **le covoiturage multi-saut** (ou *multi-hop* en anglais) ou encore le
 covoiturage avec transferts [Cortés et al., 2010]. Cette méthode permet notamment
 d'améliorer les résultats obtenus avec le covoiturage simple-saut en augmentant le
 taux de service grâce à l'élargissement des possibilités de covoiturage [Herbawi et
 1205 Weber, 2012a], en réduisant la quantité de détours nécessaire pour acheminer des
 passagers [Xu et al., 2020], ou encore en baissant les coûts généraux associés au co-
 voiturage [Coltin et Veloso, 2014]. Le covoiturage multi-saut implique une complexité
 accrue, car il faut cette fois identifier la meilleure combinaison de conducteurs ainsi
 que le lieu de transfert optimal pour tous les acteurs du covoiturage [Coltin et Veloso,

2014].

1210 2.1.6 Les modalités

Lorsque le covoiturage est considéré comme unique modalité disponible, le problème est dit unimodal et est résolu en prenant en compte l'ensemble des paramètres que nous avons énoncés dans cette section. Il peut aussi être considéré en combinaison avec d'autres modalités, transformant alors le problème en covoiturage multimodal, voire comodal (cf. section 1.5). Par exemple, dans la Figure 2.5, un utilisateur covoiture dans un premier temps sur la route marron, avant d'effectuer une correspondance pour rejoindre une ligne de transport en commun, en vert, pour rejoindre sa destination.



FIGURE 2.5 – Un exemple d'itinéraire comodal

Différentes modalités peuvent être considérées : [Lin *et al.*, 2016] considère la marche et le covoiturage de taxis pour le transit entre la ville de New York et son aéroport, [Chen *et al.*, 2019] s'intéresse à différents modes de transports tels que la conduite individuelle, en covoiturage et les bus. Enfin, [Jeribi *et al.*, 2011] apporte

une approche comodale dans laquelle sont considérés : un réseau de transport public complet (bus, métro, tram), des voitures en libre-service et du covoiturage.

1225 Ce type de covoiturage nécessite d'avoir accès aux informations de transport afin de pouvoir planifier des itinéraires multimodaux. Au vu de l'éventail des possibilités existantes, cela demande une grande puissance de calcul, des approches multi-échelles ou d'indexation spatiale (voir section [2.2.2](#))

2.1.7 Le sujet d'étude : le problème du covoiturage dynamique multi-saut

1230

Parmi toutes ces variantes de problème de covoiturage, nous avons choisi de nous intéresser au covoiturage dynamique multi-saut entre particuliers. À notre sens, ce covoiturage est le plus à même de s'inscrire dans le schéma de mobilité existant afin de réduire la fracture du transport. Cela permettrait sans doute d'atteindre l'objectif européen de la comodalité des transports. Dans la suite de ce chapitre, nous orientons donc notre état de l'art sur le covoiturage dynamique multi-saut afin de limiter le cadre de recherche, tant les approches de la littérature sont diverses et variées. Nous verrons alors qu'il existe de nombreuses manières de modéliser ce problème, notamment concernant les différents niveaux de granularité des utilisateurs et du réseau de transport.

1240

2.2 La modélisation du réseau de transport

Le réseau routier d'un problème de covoiturage dynamique est, dans la littérature, essentiellement représenté par un graphe connexe $G = \langle V, E \rangle$, où V est l'ensemble des nœuds et E l'ensemble des arêtes (voir Figure [2.6](#)). Les nœuds modélisent des lieux d'intérêts ou de rencontre, et les arêtes les routes permettant de circuler entre ces lieux. Les arêtes de ce graphe sont essentiellement pondérées soit par les distances, soit par les temps de trajets nécessaires pour rejoindre deux nœuds. Il est bien sûr possible que les arêtes soient pondérées par des vecteurs de valeurs, et donc par plusieurs informations.

1245

1250 L'intérêt d'utiliser un graphe dont les arêtes sont pondérées par des temps ou distance de trajet réside dans la capacité d'identification des plus rapides ou plus

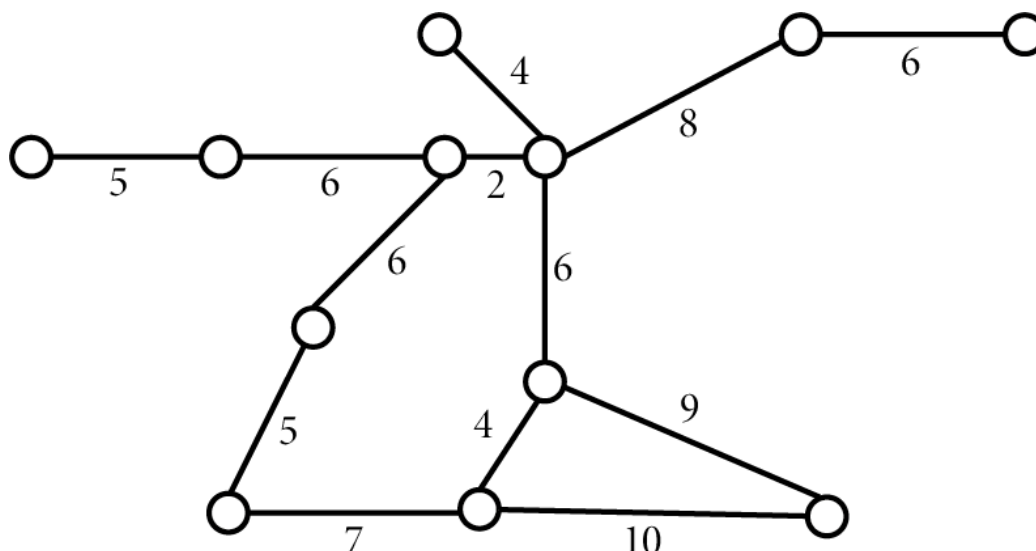


FIGURE 2.6 – Un exemple de graphe de réseau routier

courts chemins entre deux nœuds, positions ou zones, afin de déterminer des itinéraires optimisés pour les utilisateurs. Pour identifier ces itinéraires, la littérature utilise des algorithmes de complexité polynomiale basés sur l’algorithme de [Dijkstra, 1959] tel que A* [Hart *et al.*, 1968] ou basés sur l’algorithme de [Bellman, 1958, Ford, 1956, Moore, 1957].

2.2.1 La provenance du graphe de transport : données réelles ou artificielles

Un graphe de transport peut provenir de données réelles ou peut être fabriqué, donnant lieu à des données artificielles. Chacune de ces provenances possède ses avantages : la première est proche de la réalité, tandis que la seconde permet de créer des patterns spécifiques, et ce, à plusieurs échelles.

Les graphes de transport réels

L’un des principaux outils pour l’extraction d’information sur les infrastructures de transport du monde réel est OpenStreetMap¹. Cette carte du monde, collaborative et open source, référence de nombreuses informations sur :

1. <https://www.openstreetmap.org/>

- le réseau routier : rues, croisements, limitations de vitesse, direction, distance de trajet...
- le réseau de transport public : stations, lignes...
- 1270 — les pistes cyclables et piétonnes

Les données peuvent être extraites à l'aide d'une API² en effectuant une requête sur une boîte englobante (*bounding box* en anglais) définie par 4 coordonnées : la latitude minimale et maximale et la longitude minimale et maximale de la zone géographique à extraire. De nombreuses étiquettes (*tags* en anglais) peuvent être ajoutées afin de préciser la requête comme dans la Figure d'exemple 2.7. Les informations sont 1275 alors représentées sous la forme d'un fichier textuel avec une syntaxe XML (voir Figure 2.8) ou graphiquement (Figure 2.9).

```
[bbox:{{bbox}}];
way[highway~"^(motorway|trunk|primary|secondary|tertiary|
(motorway|trunk|primary|secondary)_link)$"]->.major;
way[highway~"^(unclassified|residential|living_street|service)$"]->.minor;
node(w.major)(w.minor);
out;
```

FIGURE 2.7 – Un exemple de requête sur l'API OpenStreetMap pour extraire les intersections des axes routiers majeurs et mineurs. Exécutable et visualisable via ce lien permanent : <https://overpass-turbo.eu/s/1j5h>

```
<?xml version="1.0" encoding="UTF-8"?>
<osm version="0.6" generator="Overpass API 0.7.58.2 2b5354b1">
<note>The data included in this document is from www.openstreetmap.org. The data is made available under ODbL.</note>
<meta osm_base="2022-06-06T23:03:12Z"/>

<bounds minlat="50.6030700" minlon="3.1258678" maxlat="50.6168367" maxlon="3.1563807"/>

<node id="254492282" lat="50.6058634" lon="3.1466448"/>
<node id="254985339" lat="50.6108283" lon="3.1477342"/>
<node id="256173569" lat="50.6161698" lon="3.1419614"/>
<node id="267375358" lat="50.6079221" lon="3.1343040"/>
<node id="267375363" lat="50.6104935" lon="3.1375126"/>
<node id="267375367" lat="50.6123275" lon="3.1392451"/>
<node id="267375369" lat="50.6126586" lon="3.1404418"/>
<node id="267375380" lat="50.6060284" lon="3.1328685"/>
<node id="267375381" lat="50.6063556" lon="3.1330398"/>
<node id="297974406" lat="50.6152179" lon="3.1410409"/>
<node id="301240905" lat="50.6079753" lon="3.1330858"/>
<node id="303366805" lat="50.6162033" lon="3.1361353">
  <tag k="highway" v="traffic_signals"/>
  <tag k="traffic_signals" v="signal"/>
  <tag k="traffic_signals:direction" v="forward"/>
</node>
```

FIGURE 2.8 – Un exemple de résultat sous forme XML de la requête de la Figure 2.7

Pour transformer ce résultat en graphe, de nombreux outils sont listés par [Orozco et al., 2021] tels qu'OSMNX³. La librairie OSMNX [Boeing, 2017] permet notamment

2. https://wiki.openstreetmap.org/wiki/Overpass_API
3. <https://geoffboeing.com/2016/11/osmnx-python-street-networks/>



FIGURE 2.9 – Un exemple de résultat sous forme graphique de la requête de la Figure 2.7. Les intersections identifiées sont sous la forme de cercles jaunes.

1280 d'effectuer des requêtes semblables à celle de la Figure 2.7 et de récupérer directement un graphe respectant l'échelle géographique et la forme des routes et contenant toutes les informations relatives à l'infrastructure de transport (voir Figure 2.10).

Concernant les horaires de passage des transports publics, [Orozco *et al.*, 2021] référence de nombreux outils, tel OpenMobilityData⁴ qui permet d'avoir accès à plus de
 1285 1300 jeux de données sur la mobilité dans le monde. Ces données sont sous le standard GTFS pour *General Transit Feed Specification* (traduction littérale : spécification générale pour les flux relatifs aux transports en commun). Ce format, initialement créé par Google et Trimet, est, selon Google⁵, *un format de fichier commun pour les horaires de transports en commun et les informations géographiques associées. Les "flux" GTFS permettent aux agences publiques de publier leurs informations de transports en commun et aux développeurs de créer des applications qui utilisent ces données de manière interopérable.* Les flux GTFS existent sous deux formes : le flux statique,
 1290

4. <https://database.mobilitydata.org/>

5. <https://developers.google.com/transit/gtfs?hl=fr>



FIGURE 2.10 – Un exemple de graphe obtenu avec la librairie OSMNX³

et donc non temps réel, et le flux dynamique, mis à jour en temps réel et qui sert notamment dans le calcul d’itinéraire de Google Maps.

1295 L’utilisation de données réelles permet de comparer des approches sur une même instance de problème ou encore de mettre en perspective une solution. C’est par exemple le cas de New York avec le quartier de Manhattan (voir Figure 2.12) utilisé par de nombreux articles tels que [Lin *et al.*, 2016, Alonso-Mora *et al.*, 2017, Mathieu et Nongaillard, 2018, Tafreshian et Masoud, 2020]. Les données réelles peuvent cepen-

1300 dant aussi mener à des solutions ad hoc pour une ville ou un pays, et peuvent donc souffrir d’un manque de généralité. Enfin, l’utilisation de données réelles nécessite impérativement un travail de filtrage, de transformation et de validation des données pouvant être fastidieux [Orozco *et al.*, 2020]. Par exemple, le découpage étant majoritairement effectué par une bounding box, il y a des routes qui ne mènent plus

1305 nulle part. Il est ainsi courant que des données soient incorrectes ou manquantes. Les simulations basées sur ces données ainsi que les résultats issus de ces simulations peuvent être faussés dans le cas où cette phase d’uniformisation des données n’est pas effectuée.

Les graphes de transport artificiels

1310 Pour toutes les raisons évoquées précédemment, des chercheurs se sont intéressés
au déploiement de leur solution sur des graphes artificiels afin d'apporter des solu-
tions moins sensibles aux biais existants dans les données réelles [Coltin et Veloso,
2014, Chen *et al.*, 2019, Nourinejad et Roorda, 2016]. Les graphes sont alors géné-
rés aléatoirement ou sont des graphes artificiels de la littérature comme Sioux Falls
1315 (voir Figure 2.11). Les expériences sont renouvelées de nombreuses fois sur différents
graphes, afin d'obtenir des résultats moyens de la performance d'une solution, puis
comparés à des algorithmes de la littérature, eux aussi testés sur ces mêmes graphes.

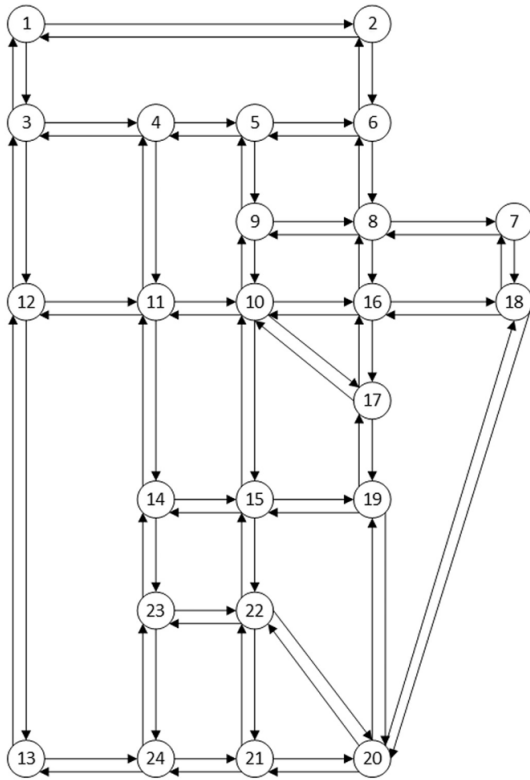


FIGURE 2.11 – Le graphe artificiel de Sioux Falls

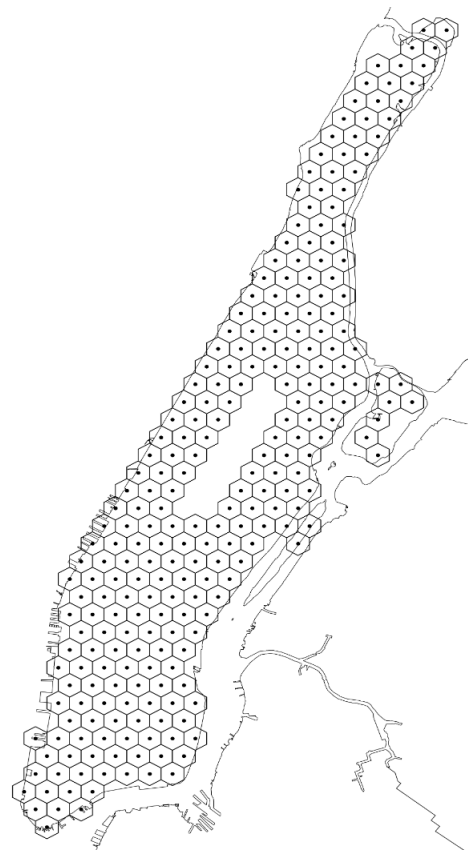


FIGURE 2.12 – Un graphe partitionné et modélisé à partir de l'île de Manhattan à New York, issu de l'article de [Tafreshian et Masoud, 2020]

L'utilisation de graphes de transport artificiels reste minoritaire dans la littérature, notamment dans le domaine de l'optimisation, du fait du manque d'instance de
1320 benchmark dans la littérature.

La nature du graphe, qu'il soit réel ou artificiel, n'a pas d'impact sur les algorithmes de covoiturage. Les graphes artificiels permettent néanmoins de générer automatiquement d'énormes bibliothèques d'infrastructures routières spécifiques.

2.2.2 Les niveaux de granularité d'un réseau de transport

1325 L'exploration spatiale d'un graphe de transport est complexe. En effet, si l'on considère par exemple un graphe représentant à l'exact ne serait-ce que le réseau routier de New York City, on atteint 166 296 nœuds (carrefours) et 405 460 arêtes (routes) [Xu *et al.*, 2020]. Chacun de ces nœuds peut être candidat au transfert d'un passager ou au détour d'un conducteur, ce qui rend le problème très complexe, notamment si l'on considère une grande population d'utilisateurs. La littérature a donc
1330 développé différentes approches visant à simplifier l'espace ou à optimiser l'exploration spatiale en décentralisant la charge computationnelle.

Le partitionnement de l'espace

Une approche classique de simplification d'espace consiste tout simplement à abstraire l'espace en le partitionnant en plusieurs zones. Cette division de l'espace peut
1335 alors être effectuée statiquement avec des zones de forme et taille égales [Tafreshian et Masoud, 2020, Shuo Ma *et al.*, 2013], comme dans la Figure 2.12, ou par quartier, par exemple [Nourinejad et Roorda, 2016]. Des stations de covoiturage (*pickups and delivery stations* en anglais) peuvent compléter ces zones, situées alors à leur centre
1340 troïde [Di Febbraro *et al.*, 2013], ou être placées à distance plus ou moins proches en fonction de la topologie du réseau routier réel [Nourinejad et Roorda, 2016]. Une dernière approche est celle du partitionnement de l'espace par hiérarchisation des routes en fonction de leur importance (les autoroutes, les routes principales et enfin les routes secondaires) [Pelzer *et al.*, 2015].

1345 Ces méthodes permettent de réduire le nombre de nœuds et d'arêtes à considérer, et donc d'approximer l'espace pour faciliter les différentes requêtes d'exploration spatiale. Cependant, ce type de covoiturage ne peut pas être considéré comme porte-à-porte ou exact, car il achemine les passagers d'une zone à une autre zone, et donc de façon approchée. De plus, ce partitionnement est statique en se basant uniquement
1350 sur la topologie d'un territoire plutôt que sur l'affluence en temps réel des requêtes.

Or, dans des conditions réelles, si l'on considère un serveur par zone, par exemple, les serveurs présents dans les zones de grande affluence seront bien plus surchargés que les autres serveurs. Des chercheurs se sont donc penchés sur l'indexation spatiale pour rendre dynamique le partitionnement de l'espace dans le but d'une gestion plus efficace du flux d'information.

L'indexation spatiale

Il est possible d'indexer spatialement un graphe de transport en attribuant à chaque nœud une paire de coordonnées (longitude et latitude) et en conservant la forme et l'échelle du réseau routier, comme on peut le voir dans la Figure 2.10. Cette indexation spatiale permet par exemple de déterminer les nœuds présents dans une zone de l'espace (zone délimitée par 4 coordonnées géographiques) sans nécessité de devoir les parcourir un par un.

La technique d'indexation spatiale R-Arbre (*R-Tree* en anglais), inventée par [Guttman, 1984] et enrichie par [Beckmann *et al.*, 1990, Manolopoulos *et al.*, 2010, Manolopoulos *et al.*, 2014], décrit des algorithmes performants de délimitation, d'insertion, de modification et de déletion de ces boîtes englobantes dans un arbre afin d'organiser l'espace de façon à rendre son exploration plus rapide.

Les objets présents dans l'espace sont alors indexés par des boîtes englobantes de différentes tailles et ensuite insérés dans un arbre par ordre décroissant de taille, de la plus grande boîte à la plus petite comme dans la Figure 2.13. Les nœuds de cet arbre sont alors appelés des pages et les feuilles sont les différents objets contenus dans l'espace (ex. les nœuds du graphe de route).

La capacité de ces pages peut être limitée en fonction d'une surface d'indexation maximale, ou d'un nombre d'objets maximal, comme dans l'exemple de la Figure 2.13 où le nombre maximal est fixé à 3. Cette capacité maximale fait alors varier la taille de l'arbre et a une influence directe sur la performance de l'exploration spatiale.

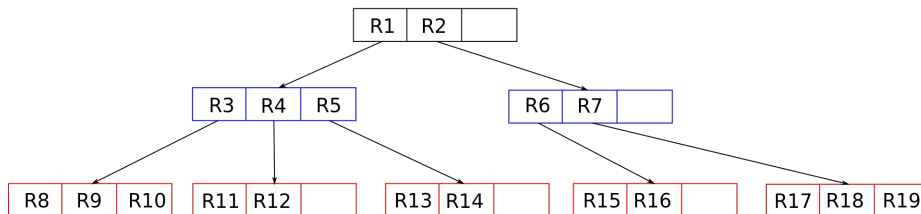
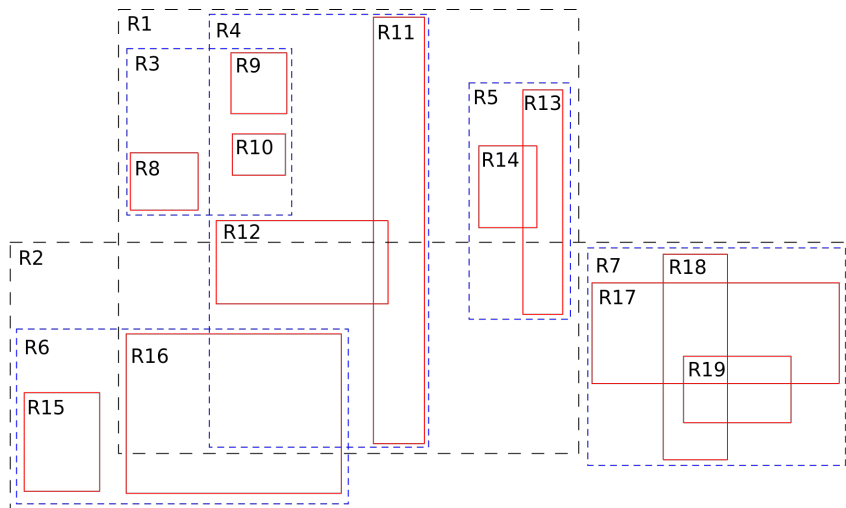


FIGURE 2.13 – Un exemple de découpe spatiale R-Arbre et de l'arbre résultant

On référence deux types de requêtes spatiales :

- la requête de chevauchement d'un point dans l'espace (*spatial point query*) : elle vise à collecter l'ensemble des objets chevauchant partiellement ou complètement le point cible (voir Figure 2.14)
- la requête de chevauchement d'une zone dans l'espace (*spatial range query*) : elle vise à collecter l'ensemble des objets chevauchant partiellement ou complètement la zone cible (voir Figure 2.15)

L'indexation spatiale R-Arbre permet de restreindre l'exploration spatiale uniquement aux zones concernées par la requête spatiale. Cette procédure de restriction et d'exploration est schématisée en vert dans les figures 2.14 et 2.15. À chaque itération, l'algorithme de parcours de l'arbre identifie les pages chevauchant la zone ou le point cible (arêtes en vert) et élimine les autres pages. À la fin, les objets contenus ou chevauchés par la zone ou le point cible sont retournés (objets encadrés en vert).

De nombreuses applications de ces R-Arbres sont référencées dans la littérature, notamment dans le cadre de la recherche du ou des plus proches voisins d'un groupe (*Group Nearest Neighbor (GNN)* ou *Aggregate Nearest Neighbor (ANN)*) [Papadias

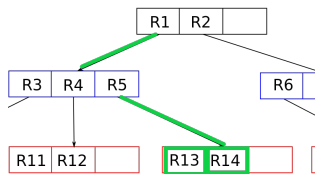
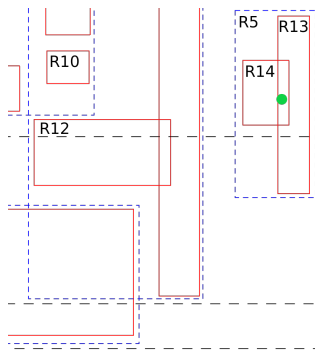


FIGURE 2.14 – Exemple de requête de chevauchement d'un point dans l'espace dans le R-Arbre de la Figure 2.13

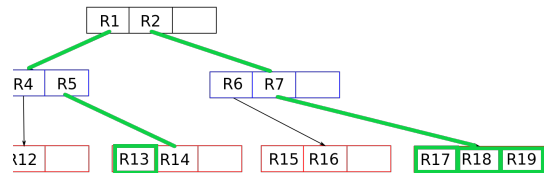
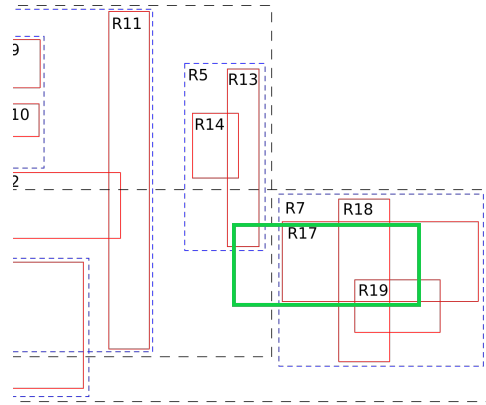


FIGURE 2.15 – Exemple de requête de chevauchement d'une zone dans l'espace dans le R-Arbre de la Figure 2.13

1395 *et al.*, 2005]. Nous verrons dans la suite de cet état de l'art (section 2.4.6) que ce problème est semblable au problème d'identification d'un lieu de transfert optimal [Yiu *et al.*, 2005].

L'abstraction de plusieurs couches de transport

1400 Un réseau de transport multimodal est composé de plusieurs réseaux de transport unimodaux définis par des propriétés qui leur sont propres (voies de circulation, nature de l'itinéraire...). Dans une même ville, ces réseaux de transport unimodaux peuvent se chevaucher sur un ou plusieurs mêmes lieux (voir Figure 2.16).

1405 Ils donnent alors lieu à la possibilité pour un passager d'effectuer un changement de mode, autrement dit, un transfert (c.f. section 2.1.5). Pour identifier un itinéraire multimodal et optimal dans ces réseaux de transport, il faut pouvoir lier ces graphes afin de pouvoir naviguer entre les différentes couches de transport. La littérature a donc développé plusieurs solutions pour atteindre ce but.

Le graphe de transport multimodal peut être considéré comme un réseau multiplexé [Boccaletti *et al.*, 2014, Kivela *et al.*, 2014, Orozco *et al.*, 2020] dont les nœuds

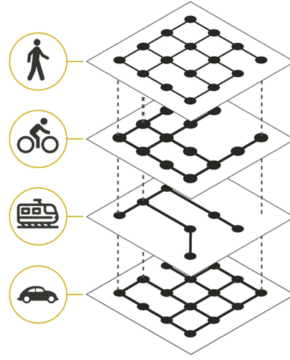


FIGURE 2.16 – Un exemple de nœuds en commun dans plusieurs couches de transport

peuvent être actifs dans une ou plusieurs couches du système. Dans la même logique que le graphe routier, les nœuds représentent les intersections (qui peuvent être présentes dans une ou plusieurs couches) ou les stations, et les liens représentent les rues, les pistes cyclables, les infrastructures ferroviaires ou les infrastructures piétonnes. Un seul et unique graphe centralise alors tous les réseaux de transport et il peut exister plusieurs arêtes entre deux nœuds contigus, ou une arête disposant d'un vecteur de valeur par mode de transport.

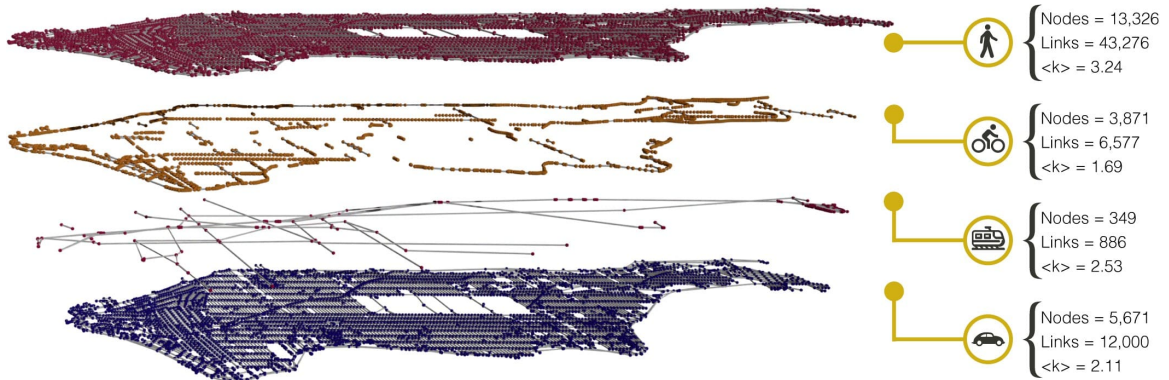


FIGURE 2.17 – (À gauche) Représentation du réseau multiplexé de Manhattan avec quatre couches d'infrastructures de transport (chemins piétonniers, pistes cyclables, lignes ferroviaires et réseau routier) issues des données d'OpenStreetMap. (À droite) Informations sur le réseau pour chaque couche, nombre de nœuds, liens et degré moyen (k). ([Orozco *et al.*, 2021])

Une autre approche est celle des graphes de transfert [Ayed *et al.*, 2011, Jeribi *et al.*, 2011, Dotoli *et al.*, 2017]. Dans cette approche, les graphes de transports ne sont pas fusionnés et la résolution est décentralisée. Quand un passager effectue une requête, cette dernière est transmise à chaque couche de transport qui identifie les combinaisons possibles, indépendamment des autres couches. Le système identifie ensuite les transferts possibles entre ces modalités (voir Figure 2.18). Le problème est

enfin résolu en combinant toutes les couches via la création du graphe de transfert comodal issue de la combinaison des nœuds de transfert (voir Figure 2.19).

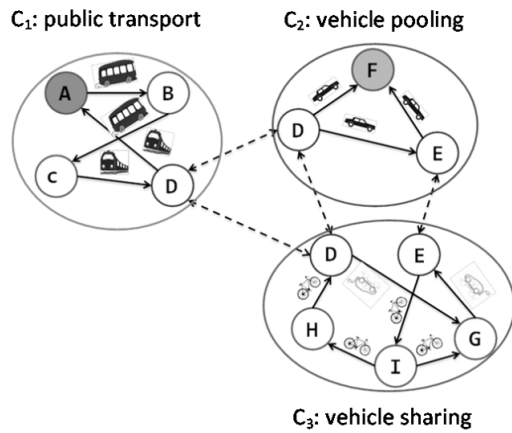


FIGURE 2.18 – Graphe de transfert avec 3 composants et 5 types de modalités, deux nœuds de transfert existent : D et E. [Dotoli et al., 2017]

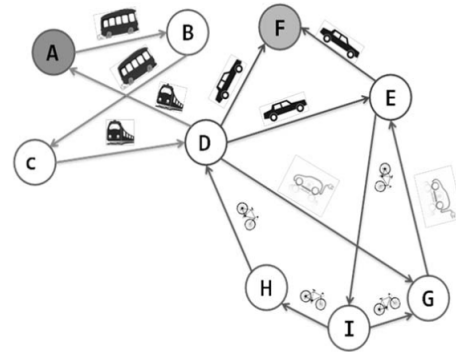


FIGURE 2.19 – Graphe co-modal construit par la combinaison des nœuds de transfert D et E du graphe de transfert de la Figure 2.18. [Dotoli et al., 2017]

2.3 La modélisation des utilisateurs : les systèmes multi-agents et approches individu-centrées

1425 D'après [Ferber, 1997], il est nécessaire de distribuer l'activité et l'intelligence pour
au moins l'une des raisons suivantes :

- "Les problèmes sont physiquement distribués" : un réseau de transport est, par nature, réparti sur un espace important ;
- "Les problèmes sont fonctionnellement très distribués et hétérogènes" : l'ensemble des problèmes est trop vaste pour un seul individu ;
- 1430 — "Les réseaux imposent une vision distribuée" : tout ordinateur fixe, portable ou mobile est connecté à l'ensemble des autres ordinateurs ;
- "La complexité des problèmes impose une vision locale" : quand les problèmes sont trop vastes pour être analysés et résolus globalement, les approches locales
- 1435 — "Les systèmes doivent pouvoir s'adapter à des modifications de structure ou d'environnement".

Les systèmes multi-agents sont des outils qui ont spécifiquement été créés pour répondre à ces problématiques.

1440 2.3.1 La notion d'agent

Le concept d'agent est né de la fusion entre les systèmes distribués et l'intelligence artificielle [Russell et Norvig, 1995, Moulin et Chaib-draa, 1996]. Ce concept a ensuite été développé par des chercheurs majeurs du domaine de l'intelligence artificielle tels que [Weiss, 1999] ou, plus récemment, par [Wooldridge, 2009] et dans les éditions
1445 suivantes du livre *Intelligence artificielle - Une approche moderne* [Russell et Norvig, 2010, Russell et Norvig, 2021].

La notion d'agent repose essentiellement sur trois principes : l'autonomie, la rationalité limitée et le comportement des agents. Dans son livre, [Ferber, 1997] propose une définition minimale commune d'un agent : Un agent est une entité physique
1450 ou virtuelle capable d'agir dans un environnement et de communiquer avec d'autres agents. Il dispose d'une rationalité limitée, c'est-à-dire d'une perception partielle de l'environnement, et est capable d'effectuer des actions de manière autonome. Enfin, il dispose d'un comportement tendant à satisfaire ses objectifs, en prenant en compte les ressources et compétences dont il dispose et en fonction de sa perception, de ses
1455 représentations et des communications qu'il reçoit (voir Figure 2.21).

Le comportement d'un agent est souvent modélisé par un cycle de vie perception-décision-action ou encore croyance-désir-intention (BDI, *Belief-Desire-Intention*) [Rao et Georgeff, 1995]. L'agent met régulièrement à jour les informations disponibles dans les limites de sa perception, il décide ensuite de l'action à entreprendre avant de
1460 réaliser cette action dans son ou ses environnements.

Un agent n'est pas forcément une personne ou une entité en mouvement. Un radar ou un feu tricolore pourraient être des agents d'un système multi-agents. Un agent peut aussi superviser différentes fonctions relatives au système tel que la génération de nouveaux agents ou la gestion des flux d'informations.

1465 2.3.2 La notion d'environnement

L'environnement d'un agent constitue tout ce qui l'entoure. On distingue deux types d'environnement pour un agent :

- **l'environnement physique** constitue l'ensemble des objets de l'espace que l'agent peut percevoir ou sur lesquels il peut agir. Si l'on considère un conducteur sous la forme d'un agent, il agit sur un véhicule afin de se déplacer et doit
1470

rester sur une infrastructure routière.

- **l'environnement social** est composé des autres agents avec lesquels l'agent communique. À l'occasion d'un covoiturage multi-saut, l'environnement social d'un passager est celui des conducteurs avec lesquels il va covoiturer ou des conducteurs dans sa perception.

1475

La dynamique et la nature d'un environnement peuvent être définies selon les termes suivants [Russell et Norvig, 1995] :

- **Environnement observable/partiellement observable** : si l'agent a accès à l'état complet de l'environnement, l'environnement est accessible ou observable, s'il a accès à une partie de l'environnement, il est partiellement observable.

1480

- **Environnement physique discret/continu** : dans un environnement physique discret, le nombre d'actions et les limites de perception d'un agent sont clairement définis. Par exemple, un jeu d'échecs est discret, il y a un nombre de cases fini et chaque pièce dispose d'une capacité de mouvement définie. À l'inverse, dans un environnement physique continu, les actions et perceptions sont infinies. Le monde réel est continu, car le nombre d'actions pouvant être réalisées par une personne est illimité. La gestion du temps dans l'environnement peut aussi être continue ou discrète, avec des pas de simulation ou des heures précises par exemple.

1485

1490

- **Environnement dynamique/statique** : un environnement qui ne change pas pendant que l'agent réfléchit est statique. Un environnement dynamique peut aussi être considéré comme tel s'il évolue d'une autre façon que par l'action des agents.

1495

Dans nos approches, nous considérons des environnements partiellement observables, discrets, déterministes et dynamiques ou statiques.

Dans leurs travaux, [Mathieu *et al.*, 2015, Mathieu *et al.*, 2016] défendent l'idée que les deux principales fonctions d'un environnement sont : (1) situer les agents les uns par rapport aux autres (distance) ; (2) porter de l'information (obstacle par exemple, voir Figure 2.20).

1500

Ils identifient quatre patterns d'environnements correspondant à des situations-types fréquentes, et précisent qu'un agent appartient généralement à plusieurs de ces environnements :

1. **AgentSet**, qui consiste à doter chaque agent d'une position, l'environnement se réduisant à une collection d'agents ;

1505

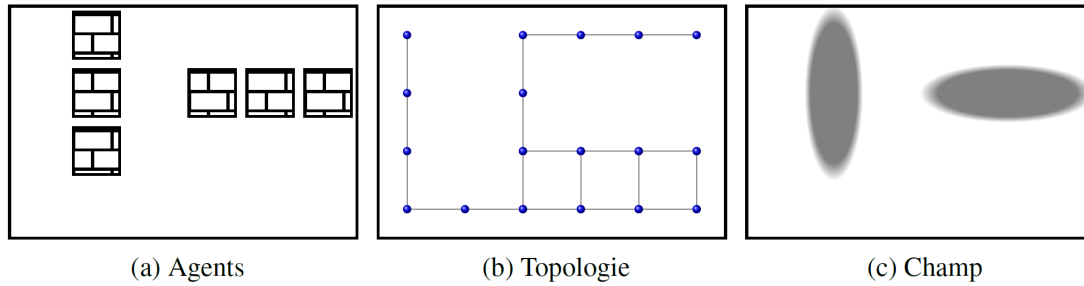


FIGURE 2.20 – Diverses possibilités de modélisation des obstacles de l'article [Mathieu *et al.*, 2016]. Ceux-ci peuvent être représentés soit par des agents (a), soit par une fonction de distance c.-à-d. la topologie de l'espace (b), soit par une fonction de champ (c)

2. **StandardGrid**, qui représente l'environnement sous la forme d'un espace discret organisé sous la forme d'une grille ;
3. **AggregateGrid**, qui permet de représenter des environnements continus en projetant les positions dans un espace discret ;
- 1510 4. **SocialNet**, qui représente l'environnement uniquement à travers les accointances des agents.

Dans nos travaux, nous présentons une approche basée sur un environnement StandardGrid dans le chapitre 3 et une approche basée sur un environnement AggregateGrid dans le chapitre 4.

1515 2.3.3 La notion de système multi-agents (SMA)

Un système multi-agents (SMA) est un système composé d'un ensemble d'agents évoluant dans un certain environnement et interagissant selon certaines règles. Un SMA est le chef d'orchestre des agents, il s'emploie à résoudre de nombreux problèmes autour de l'organisation et des interactions entre agents. Les questions issues de ces
 1520 problèmes sont classées en cinq catégories par [Ferber, 1997] :

- **la problématique de l'action** : comment plusieurs agents peuvent-ils agir simultanément et quelles sont les conséquences de leurs actions dans l'environnement ?
- **l'agent et sa relation au monde** : quelle est l'architecture interne d'un agent ? Quels sont les moyens qu'il met en œuvre afin d'assurer sa viabilité et
 1525 la satisfaction de ses objectifs ?
- **l'interaction** : quelles sont les différentes formes d'interaction que des agents peuvent pratiquer pour accomplir leurs tâches et satisfaire leurs buts ?

1530

- **l'adaptation** : quels sont les mécanismes comportementaux des agents à mettre en place afin d'assurer leur adaptation (apprentissage, évolution, régulation) ?
- **la réalisation et l'implémentation de SMA** : quels sont les langages d'implémentation, les architectures informatiques, les plateformes de développement et les aspects méthodologiques à utiliser dans la mise en œuvre de ces systèmes ?

1535

Ces questions permettent de mesurer la difficulté de la mise en place d'un système multi-agents. Chaque question nécessite une réponse précise et motivée par le domaine de recherche et le niveau d'abstraction considéré.

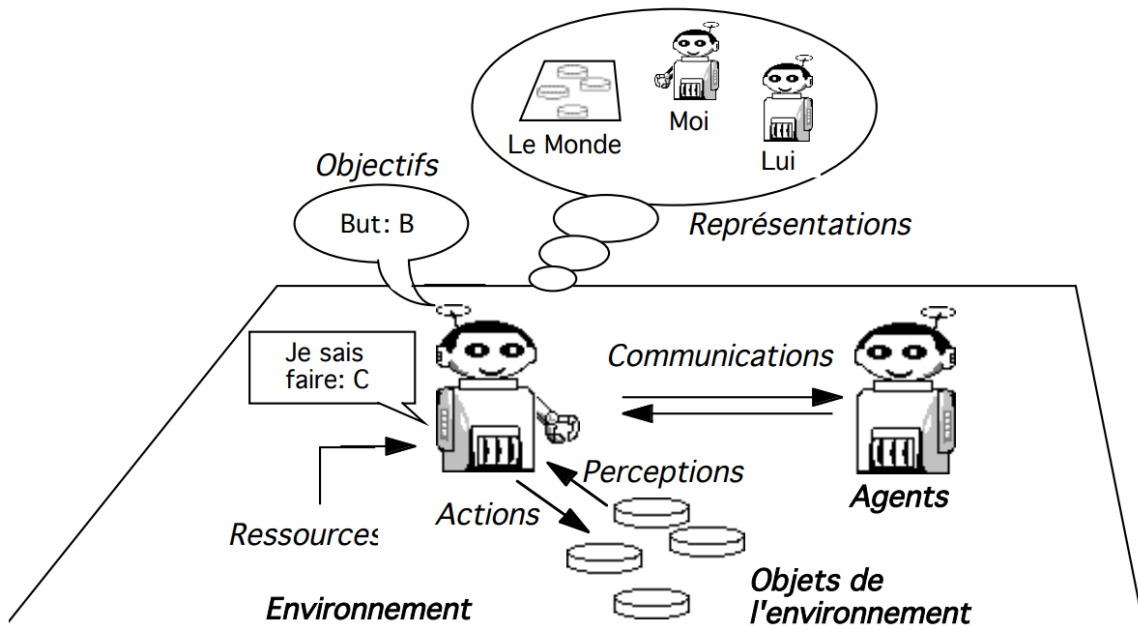


FIGURE 2.21 – Représentation imagée d'un agent en interaction avec son environnement et les autres agents ([Ferber, 1997])

2.3.4 Théorie du bien-être social

1540

Nous avons précédemment expliqué que chaque agent dispose de ses propres critères et préférences d'optimisation quant à la résolution d'une instance de problème. On peut alors s'intéresser à privilégier le bien-être individuel des individus du système – avec un modèle *premier-arrivé*, *premier-servi*, par exemple. Dans un système à ressources limitées, une telle approche à des chances de léser les agents intervenant

1545 plus tard dans la simulation. Mais on peut aussi y voir une forme de justice : après tout, l'agent arrivé en premier est prioritaire et il n'a pas d'obligation à partager les ressources qu'il convoite. Une autre vision est celle du bien-être collectif. L'objectif est cette fois d'optimiser une fonction sociale mesurant le bien-être collectif des agents. La question est alors de définir ce qui est le mieux pour la société. Pour répondre à cette
1550 question, la littérature s'est articulée autour de la théorie du choix social computationnel visant à décrire plusieurs mesures du bien-être collectif [Moulin, 1988, Moulin, 2004] :

- Utilitariste : maximise la somme des utilités de tous les individus de la société, visant ainsi l'efficacité globale du système (max-sum) ;
- 1555 — Égalitariste : maximise l'utilité minimale de tous les individus de la société, maximisant ainsi l'équité individuelle (max-min) ;
- Proportionnelle-équitable (*Proportional-fair (PF)*) : vise à produire un équilibre entre les politiques Utilitariste et Égalitariste [Kelly *et al.*, 1998, Nicosia *et al.*, 2017], basé sur l'Équilibre de Nash [Nash, 1950] (max du produit) ;
- 1560 — Élitiste : maximise l'utilité maximale des individus de la société, visant ainsi l'efficacité individuelle, autrement dit la loi du plus fort (max-max) ;
- Leximin : maximise le plus petit élément d'un vecteur de valeur, visant à éviter les valeurs trop petites dans une optimisation multicritère (classement des vecteurs de valeurs).

1565 Le choix d'une mesure peut alors se faire en les comparant sur un système identique, par exemple. Il dépend aussi du problème considéré et du but souhaité.

2.3.5 Exemples d'approches SMA

Dans cette section, nous présentons quelques approches notables basées sur les systèmes multi-agents pour simuler et résoudre des problèmes liés au transport.

1570 Exemples de modélisation de la perception des agents

La perception des agents peut être modélisée de différentes façons. Dans le cas d'une simulation de trafic en environnement continu, par exemple, [Mathieu et Non-gaillard, 2018] modélise des halos de perception et un champ de vision afin de simuler la limite des informations perceptibles par un automobiliste (voir Figure 2.22).

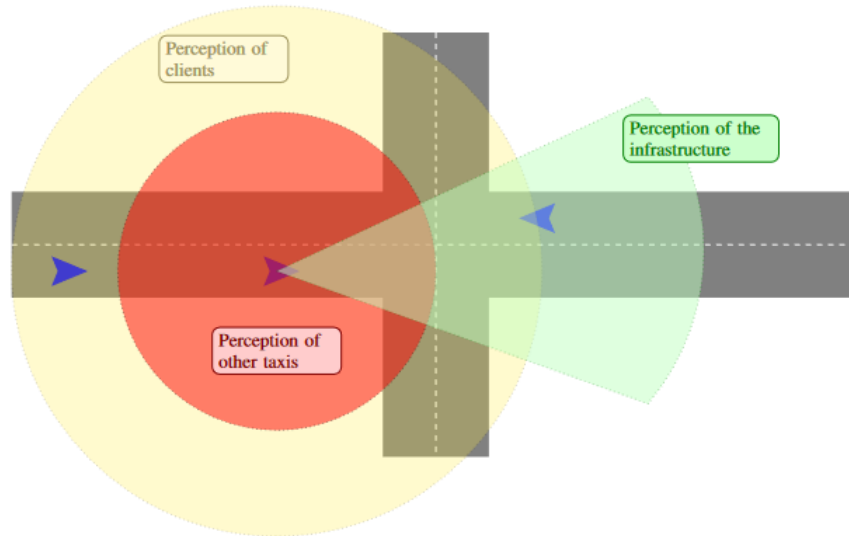


FIGURE 2.22 – Les différentes perceptions d’un agent automobiliste dans l’article [Mathieu et Nongaillard, 2018]

1575 Les limites de cette perception peuvent alors être établies en se basant sur les limites du champ de vision humain ou sur la portée des dispositifs de captation ou de communication intégrés aux véhicules autonomes.

[Daoud *et al.*, 2020] utilise un halo de perception semblable pour limiter la portée des communications inter-véhiculaires dans une flotte de véhicules autonomes. Les véhicules communiquent par diffusion via un réseau de communication ad-hoc de véhicule à véhicule (V2V) : chaque véhicule qui reçoit une nouvelle information la rediffuse et les véhicules sont ainsi connectés par transitivité dans la zone de communication. [Guériaux *et al.*, 2016] adopte une stratégie semblable avec une perception physique de l’environnement associée à une perception partagée et collaborative inter-véhiculaire (voir Figure 2.23). Ils introduisent notamment un concept de confiance améliorant la robustesse du modèle contre les perturbations introduites par le manque de fiabilité des capteurs.

Dans le contexte du covoiturage dynamique et dans un environnement discret, la perception d’un agent utilisateur est souvent limitée par sa fenêtre de temps (cf. section 2.1.1). En effet, une relation simple entre la marge de temps d’un conducteur ou d’un passager et la vitesse moyenne des véhicules du système permet d’obtenir une distance. Cette distance correspond alors au rayon d’action des agents du système, limitant les agents conducteurs susceptibles de covoiturer avec un passager ou les détours possibles par un conducteur. Il est bien sûr nécessaire de considérer d’autres

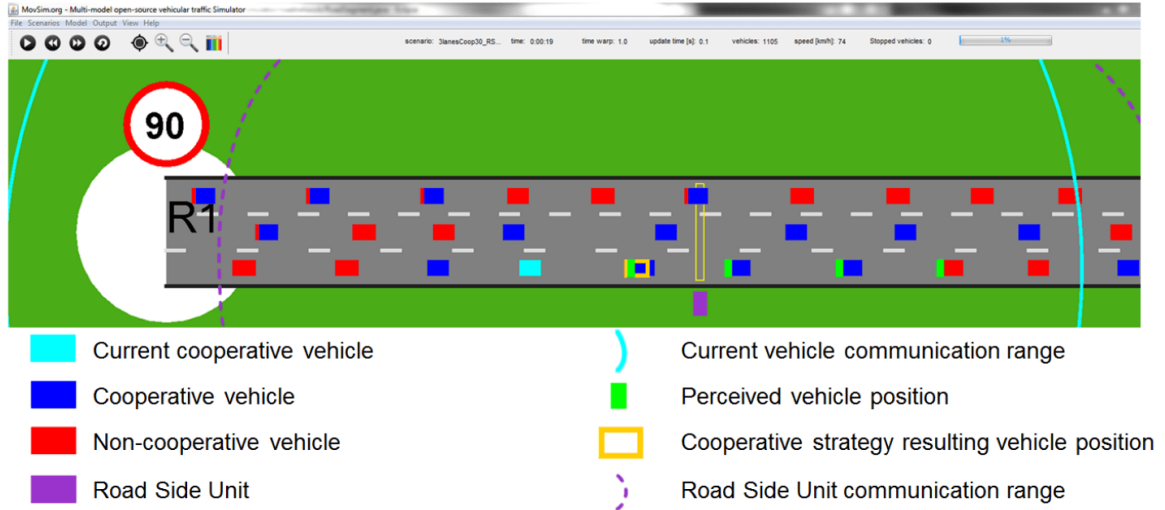


FIGURE 2.23 – Capture d’écran du modèle de trafic coopératif de [Guériau *et al.*, 2016] dans le simulateur de trafic MovSim.

1595 variables et contraintes que nous détaillons plus tard dans le chapitre 4.

Les agents d’un système doivent avoir accès, bien que partiellement, à la même information au même moment. Il doit donc y avoir une synchronisation de l’information. Cette tâche peut être remplie par le tableau noir (en anglais *blackboard*). Le tableau noir témoigne de l’état de l’environnement et peut être mis à jour selon plusieurs politiques. Dans un système dynamique, il est souvent mis à jour de façon événementielle [Dong *et al.*, 2005] : à chaque événement modifiant l’environnement (déplacement d’un véhicule par exemple), le tableau noir met à jour la représentation du système. Dans le problème du covoiturage et dans une approche individu-centrée, le tableau noir peut représenter la position des véhicules et des passagers sur le graphe de route. De cette façon, les agents du système sont capables d’identifier les autres agents présents près d’eux ou sur leur itinéraire, et ce, en temps réel.

Exemple de différentes utilisations des SMA

Dans la littérature spécialisée sur les thèmes du transport, on distingue des milliers d’applications des systèmes multi-agents. Une partie de la littérature utilise les SMA afin de décentraliser le processus de construction et d’optimisation d’itinéraire [Jeribi *et al.*, 2011, Dotoli *et al.*, 2017]. Dans ce cas, l’architecture agent repose sur des agents responsables d’effectuer des requêtes aux services de transport, de construire les différentes combinaisons possibles et enfin d’identifier le meilleur itinéraire possible. Pour

accélérer le processus d'identification du meilleur chemin, des techniques de coalition
1615 et de collaboration d'agents sont souvent utilisées [Bistaffa *et al.*, 2015, Cheikh *et al.*,
2015, Dotoli *et al.*, 2017, Feng *et al.*, 2017], de même que des algorithmes évolution-
naires [Cheikh et Hammadi, 2016, Lopes Silva *et al.*, 2018] ou de l'apprentissage [Li
et al., 2019].

Une autre approche est celle de l'utilisation des SMA afin d'apporter du réalisme
1620 dans la simulation en modélisant les utilisateurs du système de covoiturage et leurs
comportements. L'autonomie des agents réside alors dans la prise de décision en fon-
ction de leur perception et de leurs préférences. [Nourinejad et Roorda, 2016] propose
une négociation entre agents basée sur le prix du trajet, ainsi le passager proposant le
rapport prix/distance le plus intéressant remporte l'enchère. [Balbo et Pinson, 2005]
1625 propose un modèle pour évaluer une perturbation dans un réseau de transport et
déterminer ses conséquences sur le reste du réseau afin d'informer un poste de régu-
lation semblable à celui présenté en section 1.3.1. [Bessghaier *et al.*, 2012] s'intéresse
à réduire le temps de recherche d'une place de parking en aire urbaine à l'aide d'un
système de communication d'information coopératif entre agents. [Guériaux *et al.*,
1630 2016] propose une stratégie de communication coopérative entre véhicules afin d'ho-
mogénéiser le trafic routier et d'accroître la sécurité. [Lacroix *et al.*, 2009a, Lacroix
et al., 2009b, Lacroix *et al.*, 2013] se focalise sur des agents conducteurs dotés d'un
comportement normatif et d'un système permettant de violer les normes de manière
graduelle. Il peut ainsi mettre dans son simulateur différents types d'agents conduc-
1635 teurs qui respectent, par exemple, plus ou moins le code de la route, rendant ainsi ses
simulations plus réalistes. Enfin, [Chen *et al.*, 2019] modélise des profils d'utilisateurs
sur le type de transport à privilégier, voire à écarter. Il considère notamment qu'il
n'y a plus deux ensembles distincts de conducteurs et de passagers, mais un ensemble
unique d'utilisateurs nommé "voyageurs" (ou *Traveler* en anglais). Ces voyageurs ont
1640 à leur disposition une voiture ou non et disposent de préférences binaires quant à
l'utilisation des différents modes de transports. La question est alors de savoir s'ils
doivent prendre la route avec leur véhicule, covoiturer, utiliser les transports en com-
mun ou effectuer une combinaison de ces moyens de transport, comme on peut le voir
dans le schéma décisionnel de la Figure 2.24.

1645 Dans cette thèse, nous proposons d'effectuer une hybridation entre ces deux grandes
familles d'approches. Notre système multi-agents est ainsi composé d'agents dédiés à
la gestion du système et à l'optimisation des trajets, de même que d'agents utilisateurs
simulant des comportements humains.

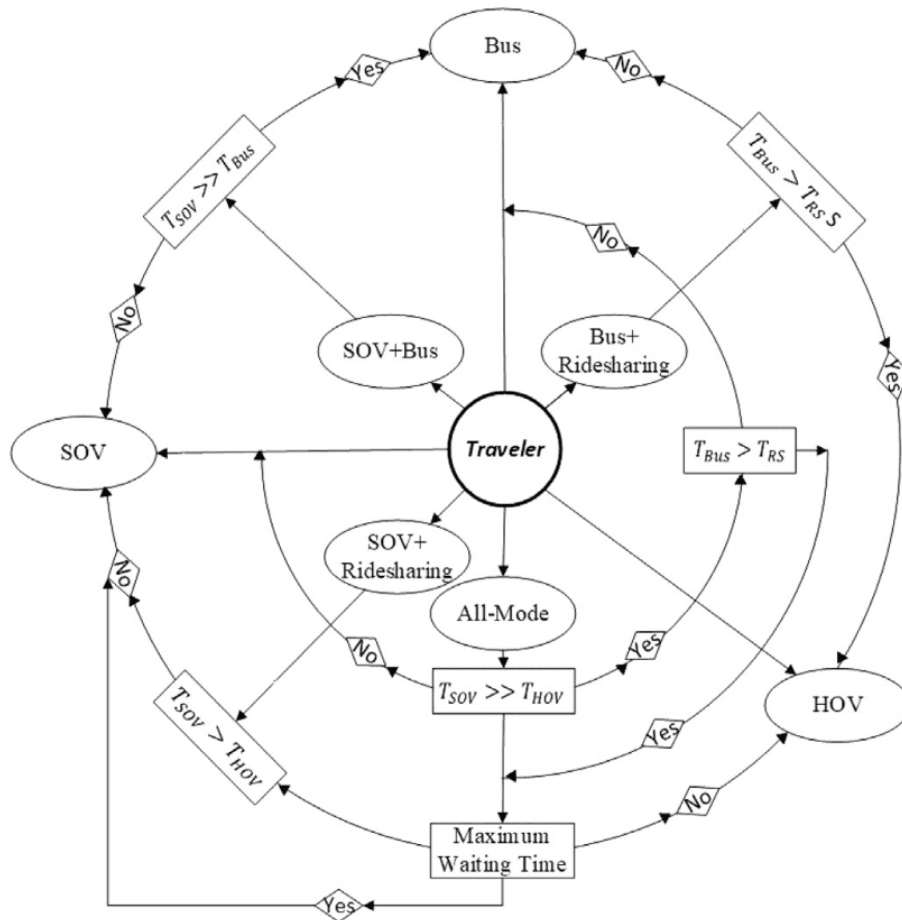


FIGURE 2.24 – Illustration du processus de prise de décision des agents voyageurs de l'article [Chen et al., 2019].

2.4 La classe de complexité du problème de covoiturage : comparaison avec des problèmes d'optimisation de l'état de l'art

1650

1655

Nous avons précédemment présenté des travaux portant sur la modélisation de l'environnement et des utilisateurs d'une instance de problème de covoiturage. Dans cette section, nous détaillons les différents problèmes d'optimisation de l'état de l'art se rapprochant de ce type de problème, ainsi que diverses approches portant sur la résolution d'instances. Le problème du covoiturage a été associé à de nombreux problèmes NP-Dur et NP-Complet de la littérature.

2.4.1 Le problème du voyageur de commerce (TSP)

Dans [Dotoli *et al.*, 2017], les chercheurs effectuent le rapprochement entre le problème du covoiturage et le problème du voyageur de commerce (*TSP : traveling salesman problem* en anglais). Le problème du voyageur de commerce consiste à identifier la tournée optimale pour une liste de ville pouvant être parcourue au maximum une fois. Les chercheurs proposent une étude comparative résumée dans le tableau 2.1 afin de mettre en exergue les ressemblances entre les deux problèmes.

Problème	Voyageur de Commerce (TSP)	Problème de Transport Co-Modal
Paramètre	Villes de passage du voyageur	Nœuds de correspondance
Contrainte	Chaque ville doit être visitée exactement une fois par un seul voyageur	Chaque véhicule, dans son itinéraire, ne doit visiter qu'une seule fois chaque nœud de correspondance, en respectant les délais.
Fonction Objectif	Trouver le circuit optimisé : le chemin le plus court reliant toutes les villes.	Satisfaction de l'utilisateur : fournir un itinéraire co-modal optimisé en fonction des préférences et des exigences de l'utilisateur.

TABLE 2.1 – Comparaison du TSP avec le problème de transport co-modal provenant de l'article [Dotoli *et al.*, 2017]

Par analogie, ils estiment alors que pour n nœuds, il existe $(n - 1)!/2$ itinéraires possibles dans un graphe comodal formé par l'union des informations de transport des différents modes pris en compte (voir 2.19). Ils expliquent ensuite que si l'on considère que la complexité de l'évaluation du temps de parcours d'un itinéraire est en $O(n)$, la complexité d'une instance de problème de transport comodal (pour un voyageur et au temps t) est de $O(n!)$. Pour 6 nœuds, on a alors 720 itinéraires possibles et pour 10 nœuds, 3 628 800 possibilités. Dans leur article, ils considèrent de nombreuses requêtes de passager au temps t . Le problème de transport comodal, semblable au problème de covoiturage dynamique multi-saut, est alors un problème d'optimisation hautement combinatoire.

2.4.2 Le problème du chauffeur à la demande (*dial-a-ride problem*)

Pour définir la classe de complexité du problème de covoiturage, d'autres tels que [Shuo Ma *et al.*, 2013, Ma et Wolfson, 2013] effectuent une analogie avec le problème

du chauffeur à la demande (*dial-a-ride* en anglais) [Sayarshad et Chow, 2015]. Le
1680 problème du chauffeur à la demande consiste à répondre aux différentes requêtes de
passager à l'aide d'une flotte de véhicule de chauffeurs professionnels (taxis, VTC...).
De nombreuses approches existent alors : par exemple, répartir de façon optimale la
flotte en prévision de nouvelles requêtes, ou encore optimiser la prise en charge des
1685 clients sur des objectifs tels que la distance parcourue, le temps d'attente des passagers
et le chiffre d'affaires réalisé. La différence avec le problème du covoiturage est que le
conducteur est un professionnel et n'a donc aucune contrainte de détour ou d'heure
d'arrivée. Le problème du covoiturage apporte donc des contraintes supplémentaires
au problème de dial-a-ride prouvé NP-Dur [Cordeau et Laporte, 2007].

2.4.3 Le problème du sac à dos (KP)

1690 Dans [Ma et Wolfson, 2013], les chercheurs effectuent le parallèle entre le problème
du covoiturage, la capacité des véhicules étant limitée, et le problème du sac à dos.
Dans ce dernier, il faut en effet maximiser la valeur totale d'un ensemble d'objets,
sans dépasser la capacité maximum du sac à dos. De nombreuses variantes existent
en mono ou multicritères : la valeur peut correspondre à un prix ou à une utilité, et
1695 la capacité peut correspondre à un poids ou à une taille, par exemple. Le problème
du sac à dos fait partie des 21 problèmes NP-complets de [Karp, 1972]. L'analogie
est effectuée, pour la capacité, sur les places disponibles dans un véhicule et, pour la
valeur, sur la maximisation d'objectifs tels que le temps de trajet des passagers ou la
distance totale parcourue par les véhicules du système.

1700 2.4.4 Le problème de couverture par ensembles (*set cover problem*)

Dans [Kamar et Horvitz, 2009], les auteurs transforment le problème de covoitu-
rage en problème de couverture par ensembles faisant, lui aussi, partie des 21 pro-
blèmes NP-complets de [Karp, 1972]. Ainsi, étant donné un ensemble d'agents A dans
1705 le système de covoiturage, l'objectif est de trouver l'ensemble des sous-ensembles de
 A qui couvre tous les agents de A en offrant la valeur cumulative la plus élevée. Un
ensemble couvrant est appelé plan de covoiturage et sa valeur est calculée en effec-
tuant la somme des valeurs de chaque groupe de covoiturage. La valeur d'un groupe

de covoiturage est quant à elle calculée, dans les grandes lignes, en effectuant la dif-
1710 férence entre les coûts de déplacement dans le cadre d'un déplacement individuel et
les coûts de déplacement dans le cadre d'un covoiturage, permettant ainsi de mettre
en évidence les bénéfices du partage d'un trajet.

2.4.5 Le *Maximum Weighted Bipartite Matching* (MWBM)

De nombreuses approches centrées-groupes, telles que [Agatz *et al.*, 2011, Tafre-
1715 shian et Masoud, 2020], modélisent le problème en *Maximum Weighted Bipartite Mat-
ching* (MWBM). Le problème de MWBM peut être défini comme suit : soit deux
ensembles distincts, l'objectif est de trouver les correspondances entre les deux en-
sembles maximisant la somme des valeurs des arêtes issues de ces correspondances.

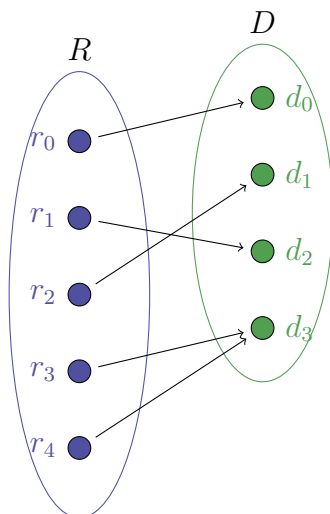


FIGURE 2.25 – Problème du covoiturage avec l'ensemble des passagers R et l'ensemble des conduc-
1720 teurs D sous la forme d'un *Maximum weighted bipartite matching* (MWBM)

Les deux ensembles correspondent alors aux conducteurs et aux passagers, et les
1720 poids des arêtes peuvent par exemple être les économies réalisées via le covoiturage
comme dans la Figure 2.25.

Le MWBM est majoritairement utilisé pour modéliser des problèmes d'allocation
de ressources. Il convient au covoiturage dans un contexte statique et est souvent
associé à la méthode de l'horizon mobile (cf. section 2.1.2 et Figure 2.2).

1725 **2.4.6 Le problème du plus proche voisin d'un groupe (GNN/ANN)**

Le problème du ou des plus proches voisins d'un groupe (GNN pour *Group Nearest Neighbor*, aussi appelé plus récemment ANN pour *Aggregate Nearest Neighbor*) consiste à identifier, pour des ensembles distincts de points Q et P dans l'espace, le ou les points de P optimisant une fonction objectif agrégative [Papadias *et al.*, 2004, Papadias *et al.*, 2005]. La transposition est faite avec l'ensemble Q correspondant au passager et aux conducteurs d'un covoiturage, et avec l'ensemble P correspondant aux points de transferts possibles. Le problème consiste dans ce cas à identifier le nœud de transfert optimal pour les trois parties. Le choix de la fonction objectif agrégative peut se baser sur la théorie du bien-être social (cf. section 2.3.4) avec un point de vue utilitariste, visant une minimisation de la distance totale parcourue par les utilisateurs, ou avec un point de vue égalitariste visant à éviter de léser quelqu'un.

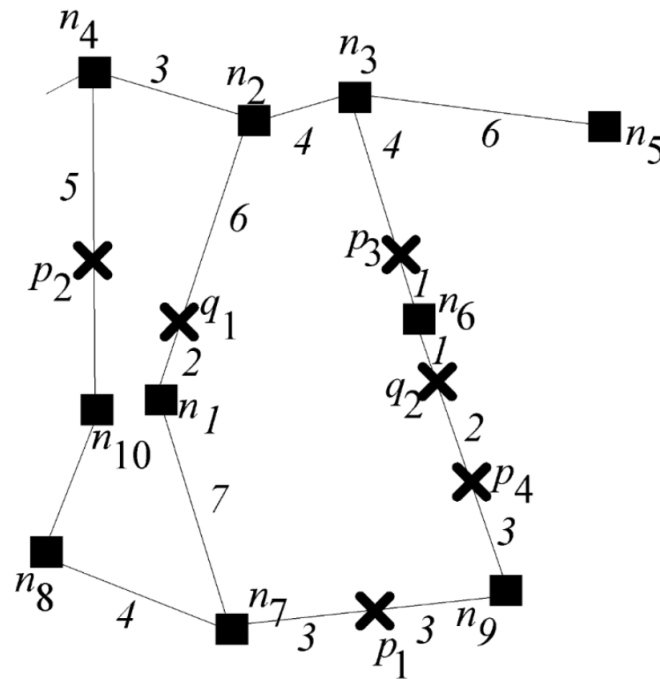


FIGURE 2.26 – Exemple d'un problème ANN dans un réseau routier, $q_1, q_2 \in Q$ représentent le groupe à l'origine de la requête ANN et $p_1, p_2, p_3, p_4 \in P$ représentent les positions candidates

D'abord étudié avec des distances euclidiennes ainsi qu'avec l'indexation spatiale R-Arbre (cf. section 2.2.2), ce problème est ensuite considéré dans un réseau routier (voir la Figure 2.26) par [Yiu *et al.*, 2005, Chen *et al.*, 2020], ainsi que dans le cadre du problème du covoiturage dynamique multi-saut [Xu *et al.*, 2020], afin d'identifier le nœud de transfert optimal. Nous détaillons les approches de l'état de l'art et étendons

le champ applicatif de ce problème dans le chapitre 4 de cette thèse.

2.5 Positionnement face à l'état de l'art

Dans cette thèse, nous nous positionnons à la croisée des domaines de la simulation
1745 multi-agents et de l'optimisation afin de contribuer de façon originale et pertinente
à la recherche sur le covoiturage dynamique entre particuliers. L'intégralité de nos
approches sont individu-centrées et prennent la forme d'outils d'aide à la décision de
voyage pour les passagers et les conducteurs. Nos contributions majeures sont orien-
tées, d'une part, vers la modélisation d'une perception dynamique de l'environnement
1750 des agents utilisateurs, c'est-à-dire qui change au fur et à mesure de la simulation en
fonction des décisions des agents et de leur environnement. Et d'autre part sur l'éta-
blissement de systèmes d'aide à la décision prenant en compte des profils réalistes
d'utilisateurs.

Dans un premier temps, nous nous attachons à la création d'un modèle de graphe
1755 artificiel permettant de créer un environnement dont la complexité est maîtrisable.
Ce modèle de graphe, générique et facilement reproductible, nous permet d'étudier
des comportements réalistes de passager modélisés dans un système multi-agents avec
un minimum de bruit dans les résultats. Dans cette simulation, les déplacements des
agents sont discrets, ces derniers se déplaçant de nœud en nœud. Nous proposons
1760 d'étudier l'autostop, un covoiturage minimal : sans détours et sans réservation. Cette
forme de covoiturage regagne en popularité ces dernières années, mais reste peu traitée
dans la littérature. Cette étude est réalisée dans le chapitre 3.

Ensuite, dans le chapitre 4, nous élargissons notre étude en intégrant la réservation
et les détours dans notre modèle. Nous proposons une méthode basée sur l'indexation
1765 spatiale R-Arbre afin de modéliser des perceptions dynamiques pour les utilisateurs.
Nous effectuons nos simulations sur des graphes réels et simulons des déplacements
continus des utilisateurs dans un environnement physique discret. Ces simulations
nous montrent les limites de notre approche, à savoir, une trop grande complexité liée
à l'utilisation d'une recherche exhaustive pour l'identification du nœud de transfert
1770 optimal ainsi qu'une fonction objective ne permettant pas d'assurer une hétérogénéité
semblable au monde réel dans le profil des passagers.

Enfin, convaincus de l'importance des préférences dans l'optimisation de trajet de

covoiturage, nous développons tout particulièrement ce concept dans le chapitre 5. Nous soulevons alors des questions sur la modélisation du comportement des utilisateurs, notamment sur l'impact de leurs préférences, que ce soit du point de vue de l'utilisateur comme du point de vue du système, et donc, à terme, de la population. Nous effectuons le rapprochement entre le problème du nœud de transfert optimal et le problème du plus proche voisin d'un groupe (GNN/ANN) et étendons le champ applicatif de l'algorithme de référence dans la littérature pour la résolution du problème susnommé.

Chapitre 3

Approche basée des graphes artificiels : le problème de l'autostop

1785 Au commencement de cette thèse et au fur et à mesure de la construction de
notre état de l'art, nous avons constaté que la littérature décrit de nombreux modèles
différents de création d'instances de covoiturage. Nous décrivons ces modèles dans
la section 2.2. Le problème est alors qu'une solution implémentée et comparée sur
une instance peut être inefficace sur une instance différente. Dans ce contexte, il peut
être complexe d'étudier et de comparer l'efficacité de stratégies comportementales ou
1790 d'heuristiques d'optimisation. Notre idée a alors été de créer un modèle générique
et reproductible, dont la complexité est maîtrisable, avec un nombre limité de pa-
ramètres. Nous proposons ainsi de nous intéresser au covoiturage par autostop, un
covoiturage dynamique par nature, sans réservation et sans détours. Nous décrivons,
dans un premier temps, ce modèle de graphe, ainsi que le problème de covoiturage
1795 par autostop.

Dans un second temps, nous proposons un algorithme récursif nommé ReTAG
pour résoudre et optimiser le problème de l'auto-stop. Nous proposons une simulation
individu-centrée utilisant une architecture multi-agents pour l'étudier. Une analyse
comparative de deux comportements d'agent passager est réalisée sur des instances de
1800 problèmes de complexité variable. Le premier simulant le comportement d'un auto-
stoppeur réel, baptisé l'auto-stoppeur naïf, et le second simulant un auto-stoppeur
connecté à un système d'information, baptisé l'auto-stoppeur ReTAG. Les résultats
de cette étude impliquent de meilleures performances pour l'auto-stoppeur ReTAG,

et ce, notamment lorsque la situation est complexe.

1805 3.1 Formulation du problème

Dans cette section, nous définissons le problème de covoiturage par autostop dont une instance est composée d'un réseau routier, d'un ensemble de conducteurs et d'un ensemble de passagers. Nous détaillons les contraintes liées au bon déroulement d'un covoiturage, ainsi que les fonctions nécessaires à l'optimisation du problème et à
1810 l'évaluation d'une solution.

3.1.1 Définition d'une instance du problème de l'auto-stop

Nous considérons une instance d'un problème de l'auto-stop comme suit :

1. **une infrastructure routière** représentée par un graphe $G = \langle V, E \rangle$, les routes sont modélisées par des arêtes $E = \{e_1, e_2, \dots, e_n\}$ et les intersections de routes par des nœuds $V = \{v_1, v_2, \dots, v_n\}$.
1815
2. **un ensemble de passagers** R , avec un passager $r_i \in R$; $r_i : (id, v_s, v_l, v_e, wt, p)$ défini respectivement par un identifiant unique, un nœud de départ, un nœud courant, un nœud de destination, le temps d'attente total cumulé et une fonction de perception. Les nœuds v_s, v_l, v_e appartiennent tous au graphe de route V . Le temps d'attente wt correspond au temps d'attente actuel du passager dans la simulation. Il s'incrémente à chaque pas de simulation si le passager ne bouge pas, c'est-à-dire s'il n'est pas en train de covoiturer. La perception p d'un passager correspond à l'ensemble des nœuds du graphe de routes accessibles par un covoiturage avec les conducteurs, nous détaillons ce concept plus tard dans la section 3.3. Cette perception est initialisée avec son nœud de départ.
1820
1825
3. **un ensemble de conducteurs** D , avec un conducteur $d_i \in D$; $d_i : (id, v_s, v_l, v_e, trip, c)$ défini respectivement par un identifiant unique, une position de départ, une position courante, une position de destination, un itinéraire et une capacité. À l'instar du passager, les nœuds v_s, v_l, v_e appartiennent tous au graphe de route V . Le trajet du conducteur est constitué de l'ensemble des nœuds contigus qui composent le plus court chemin entre son nœud de départ et son nœud d'arrivée. Un conducteur ne s'arrête pas pour attendre un passager. La
1830

capacité d'un conducteur correspond au nombre de places disponibles dans son véhicule.

1835

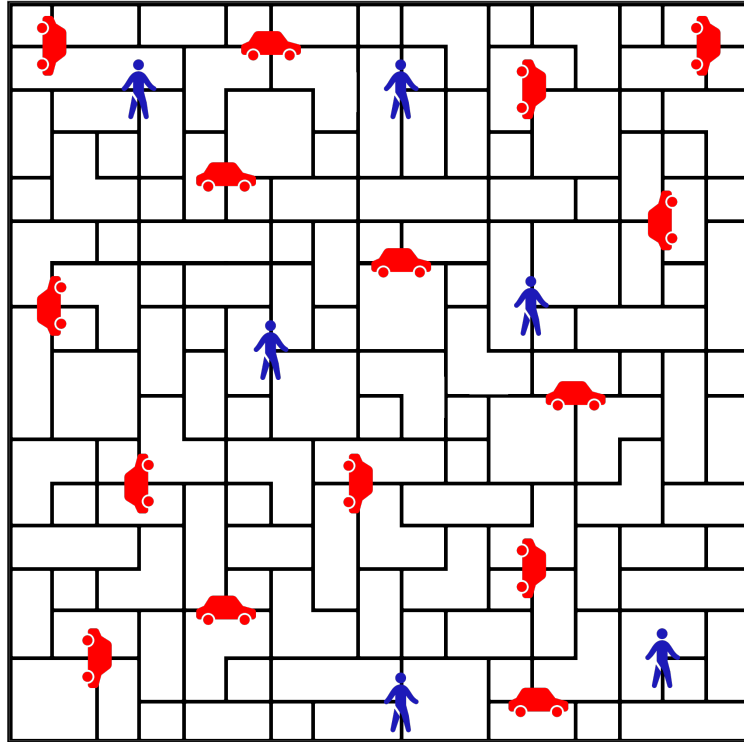


FIGURE 3.1 – Diagramme illustrant une instance de problème de l'autostop. Le graphe de route est un graphe en grille avec des perturbations. Les passagers sont représentés par les bonshommes bleus et les voitures par les voitures rouges.

Nous utilisons une simulation à temps et espace discrets. Le temps évolue à intervalles constants et le déplacement d'un nœud à un nœud adjacent nécessite une unité de temps. Chaque arc dispose donc d'une valeur de distance de 1. Par conséquent, dans notre simulation, le temps et la distance sont fusionnés et nous désignons la distance-temps entre deux nœuds v_i et $v_j \in V$: $dt(v_i, v_j) = SPL(v_i, v_j)$, SPL étant la longueur du plus court chemin entre les deux nœuds. Chaque passager et conducteur peut se déplacer sur le graphe de nœud en nœud adjacent. Plusieurs passagers et conducteurs peuvent être présents sur un même nœud. À chaque pas de temps, tous les passagers et conducteurs présents dans le système peuvent effectuer une action en fonction de leur comportement. Nous détaillons plus précisément ces comportements dans la suite de ce chapitre.

1840

1845

3.1.2 Formulations

Pour identifier si un partage de trajet est possible entre un passager et un conducteur, nous utilisons plusieurs contraintes.

1850 **La contrainte de partageabilité** : pour qu'un conducteur d_j soit candidat au covoiturage avec un passager r_i , il doit passer par un des nœuds présents dans la perception $r_i.p$ de ce passager : $\exists v_k$ tel que $(v_k \in r_i.p) \wedge (v_k \in d_j.trip)$.

1855 **La contrainte de transférabilité** : pour qu'un passager r_i en covoiturage avec un conducteur d_j puisse être transféré avec un autre conducteur d_k au nœud de transfert v_t , le premier conducteur d_j doit arriver avant ou en même temps que le second conducteur d_k au nœud de transfert v_t .

$$dt(d_j.v_l, v_t) \leq dt(d_k.v_l, v_t) \quad (3.1)$$

$$dt(d_k.v_l, v_t) - dt(d_j.v_l, v_t) \geq 0 \quad (3.2)$$

Si le covoiturage est possible, le résultat du côté gauche de l'Eq.3.2 est appelé le délai *delay* :

$$delay(d_j, d_k, v_t) = dt(d_k.v_l, v_t) - dt(d_j.v_l, v_t) \quad (3.3)$$

1860 **Le délai *delay*** représente alors le temps d'attente du passager sur le nœud de transfert entre l'arrivée du premier conducteur et le départ du second conducteur. Nous précisons que si le passager n'est pas dans une voiture - et donc en état stationnaire - le délai représente le temps nécessaire à un conducteur d_j pour rejoindre le nœud courant du passager r_i . Il se simplifie alors en la distance temporelle entre le nœud actuel du conducteur $d_j.v_l$ et le nœud actuel du passager $r_i.v_l$ comme on peut
1865 le voir dans les équations 3.4 et 3.5 :

$$delay(r_i, d_j, r_i.v_l) = dt(d_j.v_l, r_i.v_l) - dt(r_i.v_l, r_i.v_l) \quad (3.4)$$

$$delay(r_i, d_j, r_i.v_l) = dt(d_j.v_l, r_i.v_l) \quad (3.5)$$

Nous avons montré comment déterminer les conducteurs candidats au covoiturage avec un passager. Il s'agit maintenant d'identifier si le partage de trajet avec l'un de ces conducteurs est bénéfique au passager. En effet, on peut imaginer qu'un
 1870 conducteur qui se dirige dans la direction opposée, bien que satisfaisant les contraintes énoncées, n'est pas forcément pertinent. Pour évaluer le bénéfice du covoiturage avec un conducteur, nous utilisons deux variables : le délai, que nous venons de décrire, et la contribution.

La contribution d'un déplacement vers un nœud $v_k \in d_j.trip$ pour le trajet
 1875 d'un passager est décrite comme la différence entre :

1. la longueur du plus court chemin (SPL) entre le nœud actuel du passager $r_i.v_l$ et son nœud d'arrivée $r_i.v_e$
2. la longueur du plus court chemin entre le nœud cible v_j et le nœud d'arrivée du passager $r_i.v_e$

$$contrib(r_i.v_l, r_i.v_e, v_k) = SPL(r_i.v_l, r_i.v_e) - SPL(v_k, r_i.v_e) \quad (3.6)$$

1880 De cette manière, il est possible de comparer l'état actuel du passager avec son état futur et d'estimer si une progression est possible vers son nœud de destination. Par exemple, un passager séparé de 10 nœuds de sa destination et s'intéressant à un covoiturage lui permettant de se rapprocher à une distance cette fois de 7 nœuds octroie une contribution de $10 - 7 = 3$.

1885 Nous considérons comme pertinent un covoiturage entre un passager r_i et un conducteur d_j à destination d'un nœud v_k si le délai induit ne dépasse pas la contribution du déplacement.

$$contrib(r_i.v_l, r_i.v_e, v_k) > delay(r_i, d_j, v_k) \quad (3.7)$$

$$contrib(r_i.v_l, r_i.v_e, v_k) - delay(r_i, d_j, v_k) > 0 \quad (3.8)$$

Le profit d'un covoiturage correspond au résultat du côté gauche de l'équation
 1890 3.8, c'est-à-dire de la différence entre la contribution et le délai d'un covoiturage vers un nœud cible.

$$profit(r_i, d_j, v_k) = contrib(r_i.v_l, r_i.v_e, v_k) - delay(r_i, d_j, v_k) \quad (3.9)$$

Ainsi, un profit négatif ou nul présente peu d'intérêt, tandis qu'un profit positif permet à l'auto-stoppeur de progresser efficacement vers sa destination. En revanche, un conducteur dont le profit est nul ou négatif peut conduire un passager vers un autre conducteur beaucoup plus intéressant. Il faut donc être capable d'évaluer non seulement la contribution directe d'un conducteur, mais aussi les opportunités de transfert qu'il offre avec son itinéraire.

Pour conclure, si un conducteur permet à un passager d'avancer, même sans lui permettre d'atteindre sa destination, ce passager peut décider de covoiturer pour se rapprocher de sa destination. Aussi, si le conducteur ne lui permet pas d'avancer vers sa destination, il peut quand même covoiturer s'il existe un transfert profitable dans le futur. Un passager doit s'intéresser à l'ensemble des possibilités pour chaque conducteur accessible et maximiser le profit agrégé de l'ensemble des déplacements.

Par conséquent, nous définissons la fonction objectif d'un passager comme la maximisation de la somme des profits des conducteurs.

$$\max \sum_{j=0}^{card(D)} \sum_{k=0}^{card(d_j.trip)} profit(r_i, d_j, v_k) \quad (3.10)$$

3.2 L'architecture multi-agents proposée

Dans cette section, nous détaillons notre approche basée sur une simulation individualisée et un graphe de transfert.

Les conducteurs et les passagers présents dans notre système d'auto-stop sont autonomes. Ils prennent leurs propres décisions et communiquent pour partager des informations sur leur voyage. Comme précisé en section 2.3, un système multi-agents (SMA) permet d'organiser une telle population : il est composé d'agents autonomes qui ont un comportement et des caractéristiques propres et qui interagissent avec leur environnement. Comme précisé dans la section 2.3.1, dans un SMA, les agents ne sont pas systématiquement des individus en mouvement, ils peuvent aussi être responsables de tâches liées au système, comme la collecte et le partage d'informations sur les agents utilisateurs.

3.2.1 Le comportement des agents passagers et conducteurs

Notre système multi-agents est composé de divers agents ayant des fonctions et des objectifs spécifiques résumés dans le diagramme séquentiel de la Figure 3.2. Les agents conducteurs D ont pour objectif d'arriver à leur destination le plus rapidement possible et sans détours. Les agents passagers R veulent atteindre leur destination en faisant du covoiturage avec les agents conducteurs. Pour ce faire, ils doivent décider – à chaque étape de la simulation – de l'action à réaliser pour se rapprocher de leur objectif. Nous détaillons deux comportements d'agents passagers.

Le comportement "naïf" de l'agent passager

Le premier comportement d'auto-stoppeur est nommé **naïf**, il reproduit le comportement d'un véritable auto-stoppeur. Un auto-stoppeur prend généralement le premier véhicule qui lui permet d'avancer sur son trajet, il n'est pas connecté et ne planifie pas son voyage. Selon cette idée, dans notre système, un agent passager naïf ne considère que son nœud actuel, c'est-à-dire le nœud de sa position courante, et il ne parcourt pas le reste de l'environnement. Si un ou plusieurs conducteurs apparaissent sur son nœud actuel, il évalue la contribution de leur prochain mouvement. S'il s'avère que le prochain mouvement d'un des conducteurs le fait avancer vers son objectif, alors il monte dans le véhicule, sinon il ne bouge pas et incrémente son temps d'attente.

Le comportement "ReTAG" de l'agent passager

Le second comportement d'auto-stoppeur est appelé **ReTAG** (*Real-time Trip Availability Graph*). Il correspond à un auto-stoppeur connecté à un service de transport lui permettant d'avoir des informations sur les conducteurs de l'environnement (une application sur smartphone par exemple). Il peut alors planifier un itinéraire dynamique et optimisé. Si un ou plusieurs conducteurs apparaissent sur son nœud courant, le passager évalue leur profit ainsi que le profit des conducteurs accessibles sur le chemin via un transfert. De plus, il prend en compte les conducteurs qui passeront par son nœud actuel dans le futur. En effet, il peut être préférable d'attendre quelques pas de simulation pour qu'un autre conducteur fournisse une meilleure solution.

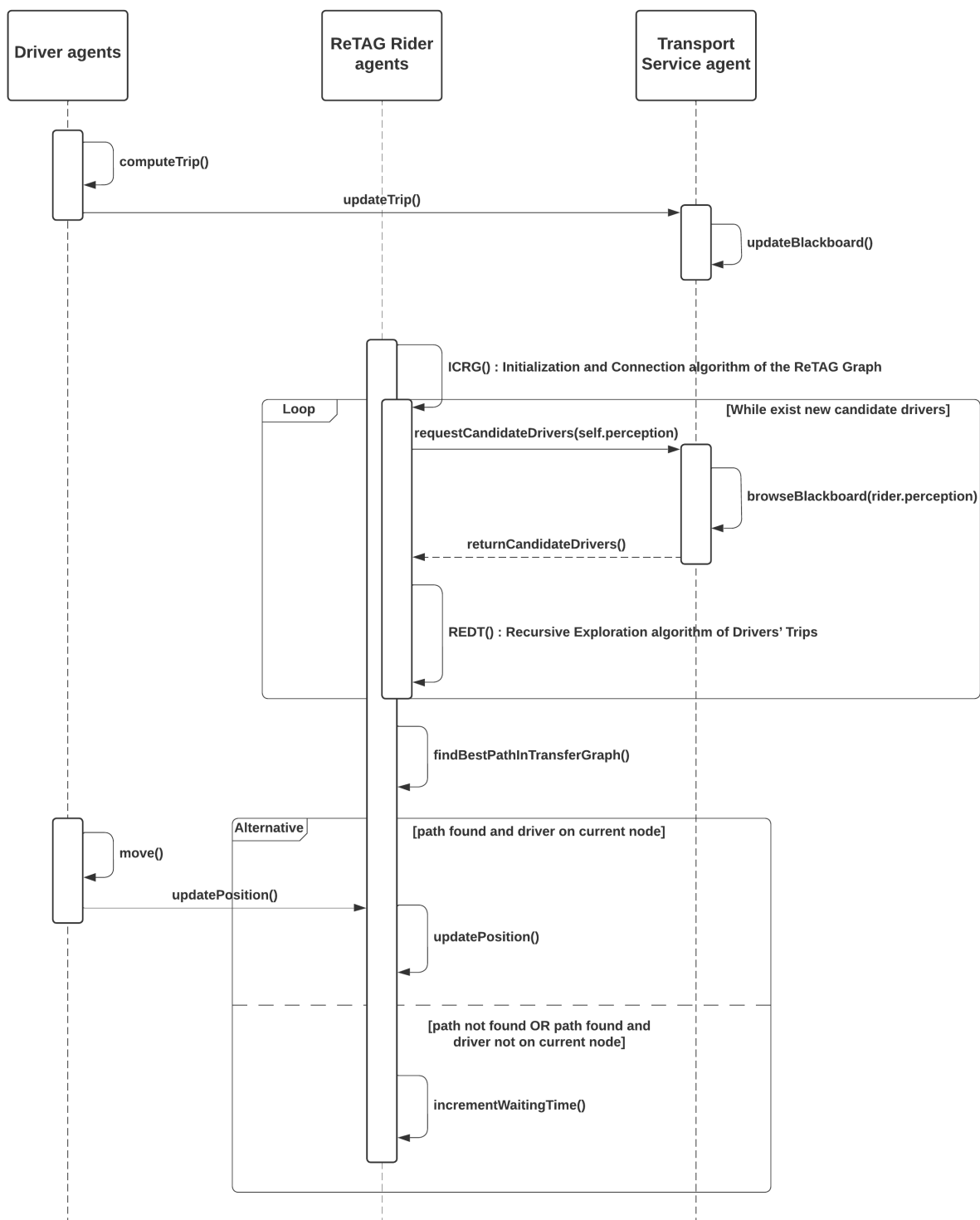


FIGURE 3.2 – Le diagramme séquentiel des agents du système. Les agents sont représentés par des lignes de vie. Les différentes actions et communications entre agents sont ordonnées chronologiquement de haut en bas. Les phases d'activation des agents sont représentées par les plus petits rectangles et les flèches représentent les appels de fonction et les communications inter-agents.

Un agent passager ReTAG doit disposer d'informations sur les itinéraires des conducteurs du système afin de pouvoir construire son itinéraire de covoiturage. Une solution peut alors être d'interroger tous les conducteurs sur leurs itinéraires et temps de trajet à chaque étape de simulation. Cette solution implique que chaque passager interroge tous les conducteurs du système à chaque pas de simulation. Or, étant donné que les conducteurs n'effectuent pas de détour dans la simulation de ce chapitre, l'itinéraire retourné par un conducteur suite à l'appel d'un passager est strictement le même, indépendamment du passager. Afin d'optimiser l'accès à l'information conducteur, nous avons choisi de centraliser les informations sur les itinéraires des conducteurs dans un tableau noir (*blackboard* en anglais).

3.2.2 L'agent service de transport *t_{sa}* : responsable du tableau noir

La gestion du tableau noir est assurée par l'agent de service de transport *t_{sa}* (*transport service agent* en anglais). Cet agent stocke les positions actuelles et futures des conducteurs et leur associe le délai associé, c'est-à-dire le nombre d'étapes de simulation nécessaires avant d'atteindre la position concernée. Ainsi, lorsqu'un nouvel agent conducteur apparaît dans le système, il transmet son itinéraire à l'agent *t_{sa}*. L'agent *t_{sa}* met à jour à chaque pas de simulation le tableau noir en décrémentant les délais et en supprimant ou ajoutant des conducteurs sur chaque nœud de manière dynamique.

Pour établir sa perception et prendre une décision, un agent passager ReTAG envoie une requête sur sa perception actuelle, c'est-à-dire l'ensemble des nœuds accessibles par les conducteurs candidats, à l'agent *t_{sa}*, qui renvoie l'ensemble des agents conducteurs admissibles, leurs trajets et leurs délais. L'agent passager applique l'approche ReTAG, dont les algorithmes sont détaillés plus loin dans ce chapitre, et met à jour son graphe de transfert et sa perception. Il effectue cette opération tant que de nouveaux conducteurs candidats sont renvoyés à l'agent *t_{sa}* à la suite de l'établissement de sa perception. Enfin, il applique la fonction objectif détaillée dans la section 3.1.2 et identifie s'il doit mettre à jour sa position suite au déplacement du conducteur optimal ou rester au même endroit et incrémenter son temps d'attente. L'identification de cet ensemble de nœuds et de conducteurs associés est représentée de manière schématique sur la Figure 3.3.

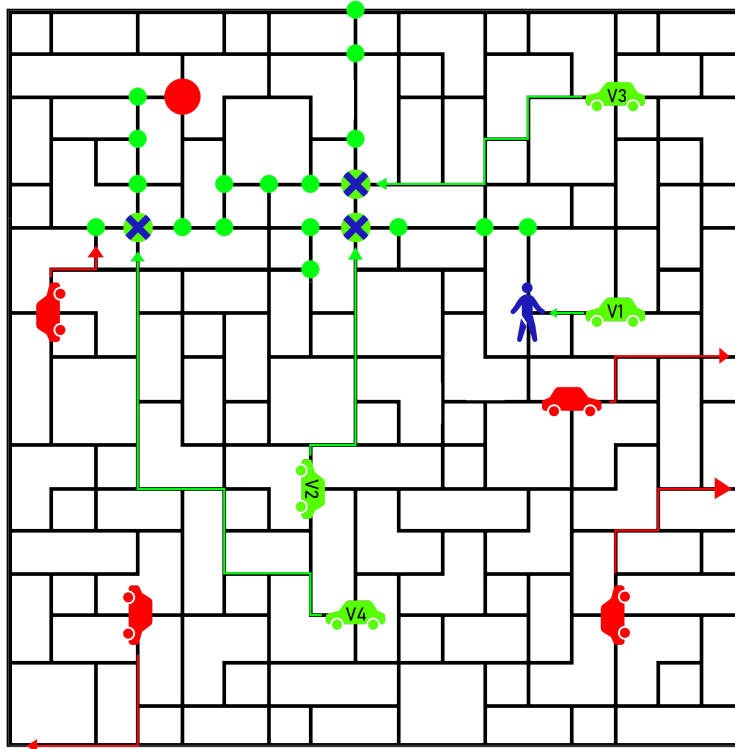


FIGURE 3.3 – Diagramme du processus d’identification de l’ensemble des positions atteignibles (nœuds verts) par le biais des conducteurs respectant les contraintes (voitures vertes) du passager (homme bleu). Les transferts possibles sont représentés par les nœuds verts avec une croix bleue et le nœud d’arrivée est en rouge. Les conducteurs violant les contraintes sont représentés par les voitures rouges. Il existe un chemin entre la position actuelle du passager et son arrivée.

1980 3.3 L’approche ReTAG : Un graphe des covoiturages disponibles en temps réel

Une fois l’ensemble des agents conducteurs admissibles renvoyé par l’agent de service de transport *tsa*, l’agent passager ReTAG construit le graphe ReTAG. Ce graphe s’inspire du graphe de transfert de [Coltin et Veloso, 2014, Jeribi *et al.*, 2011] et du graphe des positions pertinentes (*relevant position graph*) de [Ayed *et al.*, 2011].

1985 3.3.1 Définition du graphe ReTAG

Le graphe ReTAG doit contenir suffisamment d’informations pour permettre à l’agent passager d’identifier la solution optimale à chaque étape de simulation, mais il doit aussi être suffisamment léger pour être parcouru et mis à jour régulièrement et de manière peu coûteuse. Il est donc exclu de créer un graphe incluant tous les

nœuds de la réponse à la requête. Nous définissons donc le graphe ReTAG $G^{RTG} = \langle V^{RTG}, E^{RTG} \rangle$ permettant de surmonter ce problème.

Nous définissons deux types de nœuds dans ce graphe. Le premier type est le nœud de transfert, qui est ajouté au graphe dans le cas d'un transfert possible entre plusieurs conducteurs si les opportunités de déplacement associées à ce transfert entraînent un profit positif. Le deuxième type est le nœud de profit. Lorsqu'un agent passager cherche le meilleur itinéraire vers sa destination, il se peut qu'il n'y ait pas encore de séquence de covoiturage qui lui permette de remplir son objectif. Dans ce cas, son objectif est d'emprunter le chemin qui le rapproche le plus possible de sa destination. Il est alors nécessaire de restreindre l'intérêt d'un covoiturage à la position limitant le profit, cette borne est représentée par le nœud profitable. Chacun des nœuds mentionnés ci-dessus possède un attribut : le temps d'arrivée *arrTime* représentant le nombre de pas de simulation nécessaires à un passager pour atteindre le nœud concerné. Il est constitué de la somme entre les retards précédents et actuels des conducteurs et les distances précédentes et actuelles entre chaque nœud du graphe. Nous notons enfin que, puisque l'ensemble des nœuds du graphe ReTAG est au-dessus de tous les nœuds du graphe d'itinéraires, cet ensemble est un sous-ensemble des nœuds du graphe de route $V^{RTG} \subseteq V$.

Les arêtes du graphe ReTAG contiennent des informations sur le trajet entre deux nœuds. Ces arêtes sont orientées en fonction de la direction de l'agent conducteur responsable du trajet. Elles contiennent l'identifiant de l'agent conducteur, la contribution du voyage et le délai, c'est-à-dire le temps d'attente avant d'être pris en charge. Il est important de mentionner que ce modèle est générique et que nous pourrions ajouter d'autres attributs à ces arêtes. Un exemple des informations contenues dans ces arêtes est présenté en marge de celles-ci dans la Figure 3.4.

3.3.2 Génération du graphe ReTAG

Un graphe ReTAG est construit récursivement en connectant, étape par étape, les sous-graphes résultant de l'exploration du trajet des conducteurs. Pour chaque nouveau conducteur compatible avec la situation du passager (au nœud de départ ou par un transfert et respectant les contraintes), un sous-graphe d'exploration du trajet de ce conducteur est créé. L'objectif est d'identifier récursivement si un nouveau chemin rentable existe dans l'espace des solutions. Ainsi, au cours de la récursion, si

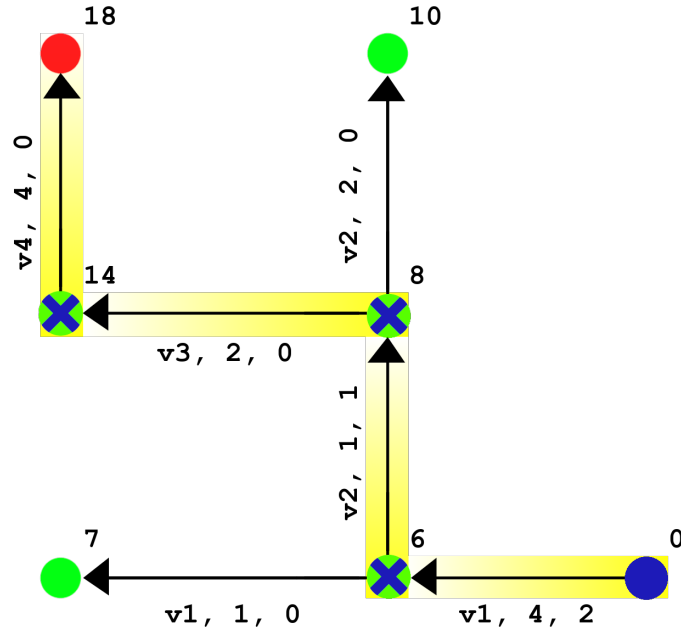


FIGURE 3.4 – Exemple d’un graphe ReTAG construit à partir du scénario de la Figure 3.3. Le nœud bleu et le nœud rouge sont respectivement le nœud de départ et d’arrivée de l’agent passager. Les nœuds verts et les nœuds verts avec une croix bleue sont respectivement les nœuds de profit et les nœuds de transfert. Les différents arcs représentent les trajets possibles entre les nœuds. Le meilleur chemin (à cette étape de la simulation) vers le nœud d’arrivée est surligné en jaune. Les valeurs sur les arcs représentent respectivement : le conducteur, la contribution et le délai.

un nœud est identifié comme profitable, le sous-graphe résultant de l’exploration est connecté au reste du graphe ReTAG, et ce, jusqu’à ce que l’ensemble des solutions
 2025 possibles soit couvert. Nous notons qu’avec cette méthode, chaque agent conducteur ou nœud du graphe routier G est visité et étudié une seule fois.

La génération du graphe ReTAG est basée sur deux algorithmes. L’algorithme 1 est nommé ICRG pour *Initialization and Connection algorithm of the ReTAG Graph*. Il vise à initialiser le graphe ReTAG avec les conducteurs passant par le nœud de
 2030 départ du passager (ICRG, L1-2). Il exécute l’algorithme 2 sur chacun de ces conducteurs candidats et connecte les sous-graphes résultant de l’exploration des transferts possibles dans leur trajet si une solution profitable est trouvée (ICRG, L3-8). L’algorithme ICRG renvoie le graphe ReTAG à la fin du processus (ICRG, L9).

L’algorithme 2 est appelé REDT pour *Recursive Exploration of Drivers’ Trips algorithm*. Cet algorithme est chargé d’explorer récursivement les trajets des conduc-
 2035 teurs à la recherche de nœuds profitables ou de transfert. Il considère comme paramètres un conducteur candidat au covoiturage $curD$, le nœud étudié lors de la récursion précédente $prevNode$ et le temps accumulé lors de la récursion précédente

Algorithm 1 ICRG - Initialization and Connection algorithm of the ReTAG Graph

Input : G = road infrastructure graph, r_i = a rider agent
Output : G^{RTG} = the rider ReTAG graph

- 1: instantiate G^{RTG} with rider current node ($r_i.v_l$, $arrTime = 0$)
- 2: AD = query tsa for available drivers passing by rider current node $r_i.v_l$
- 3: **for all** $d_j \in AD$ **do**
- 4: $node = REDT(d_j, r_i.v_l, delay(r_i, d_j, r_i.v_l))$
- 5: **if** $node$ not $None$ **then**
- 6: $G^{RTG}.addEdge(r_i.v_l, node, d_j.id, contrib(r_i.v_l, r_i.v_e, node), delay(r_i, d_j, r_i.v_l))$
- 7: **end if**
- 8: **end for**
- 9: **return** G^{RTG}

$prevArrTime$ (temps de trajet du conducteur et temps d'attente du conducteur).

2040 Ces variables sont essentielles pour vérifier si les contraintes de covoiturage ne sont pas violées et pour maintenir la cohérence temporelle dans chaque exploration. Ainsi, $prevNode$ permet de limiter l'ensemble des nœuds à considérer dans le trajet du conducteur $curD$. En effet, tous les nœuds antérieurs à $prevNode$ représentent l'itinéraire passé du conducteur au pas de simulation du covoiturage. En d'autres termes, 2045 les nœuds précédant $prevNode$ sont des nœuds qui seront déjà parcourus par le conducteur une fois que ce dernier aura atteint le passager, ils ne doivent donc pas être considérés. Le sous-ensemble $curD.subTrip$ est alors créé et comprend tous les nœuds du trajet du conducteur à partir de $prevNode$ exclu (REDT, L1). La variable $prevArrTime$ initialise la référence temporelle dans la récursion (REDT, L2).

2050 Nous initialisons ensuite deux variables : $firstNode$ et $lastNode$ (REDT, L3). Ces variables sont utilisées pour trouver le premier et le dernier nœud ajouté au sous-graphe des nœuds rentables et de transfert. En effet, une fois que le chemin du conducteur courant $curD$ a été exploré, il est nécessaire de déterminer dans un premier temps si un chemin profitable a été trouvé à partir de ce conducteur. Si c'est 2055 le cas, il est nécessaire de déterminer quel nœud connecter au graphe de la récursion précédente. Cette fonction est remplie par la variable $firstNode$. Elle est initialisée à $None$ de sorte que, si à la fin de la récursion, il n'y a toujours pas de nœud associé à cette variable, alors il n'y a pas d'intérêt, pour le moment, à covoiturer avec ce conducteur (ICRG, L5).

2060 Concernant le $lastNode$, il stocke le dernier nœud ajouté au sous-graphe courant lors de la traversée des nœuds du conducteur courant $curD$. Nous allons détailler son utilité en analysant l'algorithme 2. L'algorithme REDT parcourt chaque nœud $node$ du sous-trajet du conducteur courant $curD.subTrip$. Le temps de référence $arrTime$ est alors incrémenté de 1 : un déplacement d'un nœud à un autre nœud adjacent 2065 nécessitant 1 pas de simulation. L'agent passager fait une requête à l'agent de service

Algorithm 2 REDT : Recursive Exploration algorithm of Drivers' Trips.

Input : $curD$: the current studied driver, $prevNode$ the previous studied node, $prevArrTime$: the time it takes the rider to reach the previous node with $curD$

Output : $firstNode$: the first node of the driver's trip added to the ReTAG graph

```
1:  $curD.subTrip = curD.trip$  nodes from  $prevNode$  to the end of the trip
2:  $arrTime = prevArrTime$ 
3:  $firstNode = None, lastNode = None$ 
4: for all  $node \in curD.subTrip$  do
5:    $arrTime+ = 1$ 
6:    $AD = \text{query } tsa \text{ for available drivers passing by } node$ 
7:   for all  $d_j \in AD$  do
8:     if  $delay(curD, d_j, node) \geq 0$  then
9:       # possible transfer on  $node$  between  $d_j$  and  $curD$ 
10:       $newNode = REDT(d_j, node, delay(curD, d_j, node) + arrTime)$ 
11:      if  $newNode \neq None$  then
12:        # profitable node found through transfers
13:        if  $node \notin G^{RTG}$  then
14:           $G^{RTG}.addNode(node, arrTime)$ 
15:        end if
16:        if  $firstNode == None$  then
17:           $firstNode = node$ 
18:        else
19:           $G^{RTG}.addEdge(lastNode, node, curD.id, contrib(lastNode, r_i.v_e, node), delay = 0)$ 
20:        end if
21:         $lastNode = node$ 
22:         $G^{RTG}.addEdge(node, newNode, d_j.id, contrib(node, r_i.v_e, newNode), delay(curD, d_j, node))$ 
23:      end if
24:    end if
25:  end for
26: end for
27: if  $firstNode == None$  then
28:   # no transfer found or no profitable node found through transfers
29:   search for the most profitable node in  $curD.trip$ 
30:   if exists a profitable node then
31:      $firstNode = \text{most profitable node}$ 
32:   end if
33: end if
34: return  $firstNode$ 
```

de transport tsa sur $node$ pour obtenir les conducteurs passant par ce nœud. Le résultat de cette requête, à savoir l'ensemble des conducteurs disponibles, est stocké dans un nouvel ensemble AD (REDT, L4-6). Pour chacun de ces conducteurs, on vérifie si la contrainte de transférabilité définie dans la section 3.1.2 est respectée.

2070 Si oui, un transfert est possible et l'algorithme effectue une nouvelle récursion sur le conducteur d_j et le nœud $node$ (REDT, L7-9). Si cette nouvelle récursion retourne un nœud, il s'agit du premier nœud du sous-graphe de cette nouvelle récursion. On obtient ainsi l'information que le conducteur actuellement étudié $curD$ permet au passager d'atteindre un nœud profitable au moyen d'un transfert sur le nœud $node$.
2075 Le nœud $node$ est alors un nœud d'intérêt et est ajouté au sous-graphe de la récursion courante (REDT, L11-15).

Enfin vient la phase de liaison entre le sous-graphe résultant du transfert avec le conducteur d_j et le sous-graphe du conducteur actuel $curD$. Tout d'abord, le nœud de transfert $node$ est ajouté dans le graphe ReTAG. S'il s'agit du premier nœud ajouté

2080 dans le graphe pour le conducteur actuel $curD$, on associe sa valeur à $firstNode$.
Sinon, il doit être lié par une arête au dernier nœud ajouté lors de la récursion en
cours. Ce nœud n'est pas nécessairement le nœud précédant $node$ dans le sous-chemin
du conducteur courant. En effet, nous avons précisé précédemment que seuls les nœuds
rentables ou de transfert sont inclus dans le graphe ReTAG. Certains nœuds du trajet
2085 du conducteur sont omis, car ils ne présentent pas d'intérêt pour le passager. Il est
donc essentiel de disposer d'une variable telle que $lastNode$ et de la mettre à jour à
chaque fois qu'un nœud est ajouté (REDT, L16-21). Enfin, l'algorithme relie par une
arête le sous-graphe résultant du transfert avec le conducteur d_j avec le sous-graphe
du conducteur actuel $curD$ sur le nœud de transfert $node$ (REDT, L22).

2090 Il peut arriver qu'après avoir visité tous les nœuds du trajet du conducteur actuel
 $curD$, l'algorithme n'ait identifié aucun transfert intéressant pour le passager. Dans
ce cas, l'algorithme recherche le nœud le plus rentable du conducteur et l'associe au
 $firstNode$ (REDT, L27-33). Si un tel nœud n'existe pas, le conducteur et son chemin
sont naturellement écartés par les deux algorithmes (ICRG, L5 et REDT, L11).

2095 Une fois qu'un agent passager a généré son graphe ReTAG, il recherche le chemin
maximisant son profit en appliquant la fonction objectif décrite dans la section 3.1.2.
La résolution est simplifiée, car la perception du passager est limitée au graphe ReTAG
et il doit seulement identifier le chemin, c'est-à-dire la séquence d'arêtes, maximisant
le profit : la somme des différences des délais sur la contribution.

2100 3.4 Expérimentations et résultats

Cette section détaille le cadre de travail de nos simulations et les résultats qui
en découlent. Nous définissons les différents paramètres utilisés pour la création
d'instances du problème d'auto-stop et comparons l'auto-stoppeur naïf avec l'auto-
stoppeur utilisant l'approche ReTAG.

2105 3.4.1 Protocole expérimental

Chacun des deux comportements de passager (naïf et ReTAG) est testé sur 50 ins-
tances différentes et les résultats sont moyennés en conservant l'écart-type des valeurs.
Une instance est composée d'un graphe de routes, d'un ensemble d'agents conducteurs

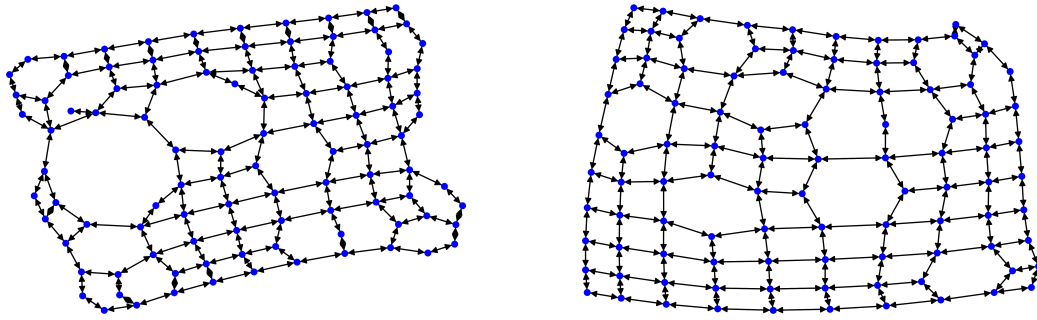


FIGURE 3.5 – Exemples de graphes d’infrastructures routières générés et étudiés dans nos simulations

et d’un ensemble d’agents passagers. Les nœuds de départ et d’arrivée des agents pas-
 2110 sagers et conducteurs sont tirés aléatoirement et uniformément dans chaque instance.
 Les stratégies des passagers sont toutefois comparées sur les mêmes instances afin
 d’être dans les mêmes conditions d’offre et de demande. Pour la génération de notre
 graphe d’itinéraire, nous avons choisi d’étudier un graphe en grille, car de nombreux
 travaux utilisent la ville de Manhattan pour leurs expériences comme [Alonso-Mora
 2115 *et al.*, 2017] ou [Tafreshian et Masoud, 2020]. Les rues de cette ville sont organi-
 sées sous forme de grille régulière avec quelques perturbations, comme des parcs par
 exemple. Les graphes en grille régulière ont aussi été utilisés par [Coltin et Veloso,
 2014] afin d’étudier les transferts de passagers. Nous avons choisi d’appliquer une
 perturbation à cette grille régulière afin de pouvoir produire différents graphes de
 2120 routes et de reproduire des espaces sans routes comme les parcs de Manhattan. Cette
 perturbation consiste à supprimer aléatoirement une proportion des arêtes du graphe,
 tout en vérifiant que le graphe reste connecté. Des exemples des graphes étudiés sont
 présents dans la Figure 3.5.

Nous proposons deux types de simulation : une simulation avec une génération
 2125 continue de conducteurs et une simulation avec une génération unique d’un nombre
 de conducteurs fixé à l’avance. Dans les deux cas, tous les passagers sont générés à
 l’initialisation de l’instance et leur nombre est fixé à l’avance.

3.4.2 Une simulation avec une génération continue de conduc- teurs

2130 La simulation avec une génération continue de conducteurs nous permet de repro-
 duire une situation réelle où il y a un flux entrant et sortant de conducteurs dans le

système. En variant la densité de ce flux, nous pouvons simuler tout un panel de trafic routier et évaluer les comportements en situation de rareté ou d'abondance de l'offre de covoiturage. La résolution de l'instance se termine lorsque tous les agents passagers ont atteint leur destination. Nous nous intéressons principalement à la métrique du nombre de pas de simulation nécessaire à la résolution d'une instance. En effet, bien que d'autres métriques soient nécessaires afin de mettre en perspective les résultats, nous cherchons dans cette expérimentation à mettre en évidence la performance d'une stratégie face à une autre.

Dans cette simulation, nous générons un graphe routier de 100 nœuds (10x10) perturbé sur 30% des arêtes pour obtenir un graphe routier convaincant. Le nombre d'agents passagers généré est de 100 et la densité du flux de conducteurs varie entre 25 et 225 afin de générer des situations de rareté et d'abondance de solutions de covoiturage pour les passagers. En d'autres termes, lors d'une étape de simulation, il ne peut y avoir plus de conducteurs que la densité de flux définie. Chaque fois qu'un agent conducteur arrive à sa destination et meurt, un nouvel agent conducteur est généré.

Les résultats de cette simulation sont présentés dans les Figures 3.6, 3.7 et 3.8. Dans ces figures, la variation du nombre maximal de conducteurs présents dans la simulation pour un pas de simulation, c'est-à-dire le flux maximal, est représentée sur l'axe des abscisses. Les différentes métriques étudiées, telles que le temps d'attente moyen, le temps de parcours moyen et le nombre d'étapes de simulation, sont représentées sur l'axe des ordonnées. Enfin, les courbes et les surfaces représentent respectivement les valeurs moyennes et l'écart-type des données résultant de l'application du comportement du passager ReTAG (orange) et naïf (bleu) sur 50 instances différentes.

Ces résultats montrent l'efficacité du comportement du passager ReTAG par rapport au comportement du passager naïf. En effet, le nombre moyen d'étapes de simulation nécessaires pour que tous les agents passagers arrivent à leur destination est considérablement réduit comme le montre la Figure 3.6. Cette réduction est plus prononcée lorsque le flux de conducteurs est en situation de rareté, c'est-à-dire pour un flux maximal de conducteurs compris entre 25 et 75 environ.

C'est également dans cet intervalle que nous observons un temps de trajet plus court pour les passagers ReTAG que pour les passagers naïfs dans la Figure 3.7. Une fois qu'un flux maximal de conducteurs de 150 est atteint, la courbe du passager

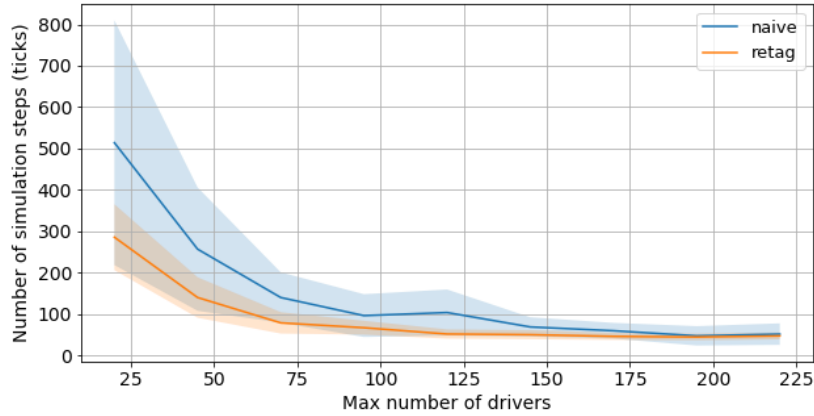


FIGURE 3.6 – Nombre de pas de simulation nécessaire pour achever une instance en fonction du nombre maximum de conducteurs et des comportements des conducteurs : naïf en bleu, ReTAG en orange. Plus les valeurs sont faibles, meilleur est le résultat, l’objectif étant de minimiser le nombre de pas de simulation.

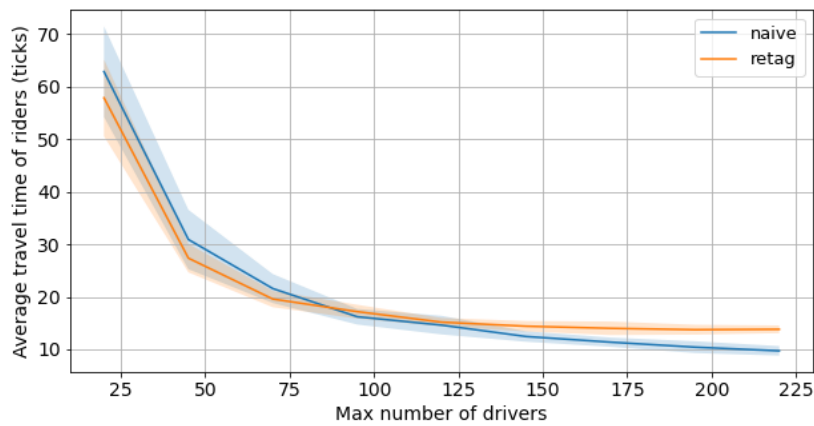


FIGURE 3.7 – Temps de trajet moyen des agents passagers en fonction du nombre maximal de conducteurs et des comportements des passagers : naïf en bleu, ReTAG en orange

ReTAG se stabilise, tandis que celle du passager naïf continue de diminuer et s’en sort donc mieux en termes de temps de parcours. Ce phénomène s’explique par le fait qu’un passager au comportement ReTAG ne considère pas seulement l’apport d’un covoiturage, mais aussi le temps d’attente associé à chaque transfert. Il privilégie donc le mouvement à l’attente, c’est-à-dire qu’il préfère faire des détours qui allongent son trajet plutôt que de rester à un endroit et d’attendre.

Ce phénomène est émergent et très intéressant, car une fois une situation d’abondance de solutions atteinte, les passagers ReTAG mettent en balance leur temps de déplacement et leur temps d’attente, c’est-à-dire les composantes du profit.

La Figure 3.8 tend à confirmer cette justification avec un temps d’attente moyen des passagers ReTAG inférieur à celui des passagers naïfs, quel que soit le flux maximal

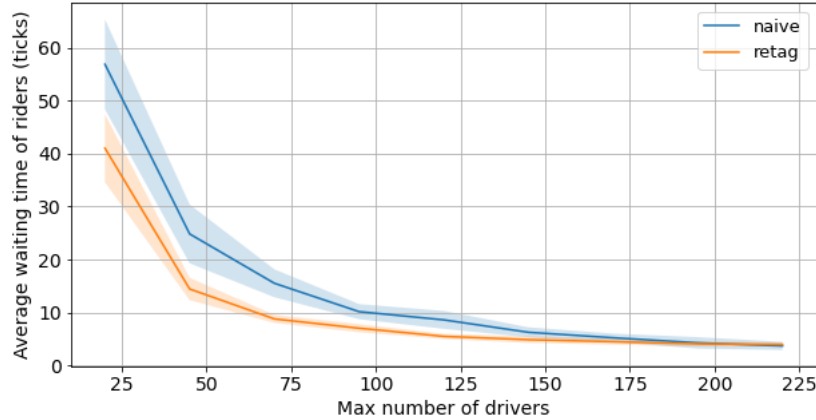


FIGURE 3.8 – Temps d’attente moyen des passagers en fonction du nombre maximum de conducteurs et des comportements des passagers : naïf en bleu, ReTAG en orange.

de conducteurs, avec encore une fois une différence plus importante quand l’offre de covoiturage est faible.

3.4.3 Une simulation avec une génération unique de conducteurs

2180

La simulation avec une génération unique de conducteurs permet d’étudier les deux comportements d’agent passager dans une situation où l’offre de covoiturage est de moins en moins importante. En effet, dans cette simulation, tous les agents conducteurs sont générés à l’instanciation, de la même manière que les agents passagers. Une fois qu’ils ont atteint leur destination, les agents conducteurs disparaissent du système et aucun autre agent conducteur n’est généré. La simulation s’arrête lorsqu’il n’y a plus d’agents conducteurs dans le système. Nous simulons une sorte de course contre-la-montre afin d’étudier le taux de service, c’est-à-dire la proportion de passagers atteignant leur destination avant la fin de la simulation.

2185

2190

Dans cette simulation, nous générons un graphe routier de 400 nœuds (20x20) perturbé sur 30% des arêtes pour obtenir un graphe routier convaincant. Le nombre d’agents passagers générés est de 100 et le nombre d’agents conducteurs générés varie entre 5 et 500 afin de générer des situations de rareté et d’abondance de solutions de covoiturage pour les passagers. Les résultats de cette simulation sont présentés dans les Figures 3.9, 3.11 et 3.10. Dans ces figures, la variation du nombre total de conducteurs générés est représentée sur l’axe des abscisses. Les différentes métriques étudiées, telles que le temps d’attente moyen, le temps de trajet moyen et la proportion de demandes

2195

de passagers satisfaites, sont représentées sur l'axe des ordonnées. Enfin, les courbes et les surfaces représentent respectivement les valeurs moyennes et l'écart-type des données résultant de l'application du comportement du passager ReTAG (orange) et naïf (bleu) sur 50 instances différentes.

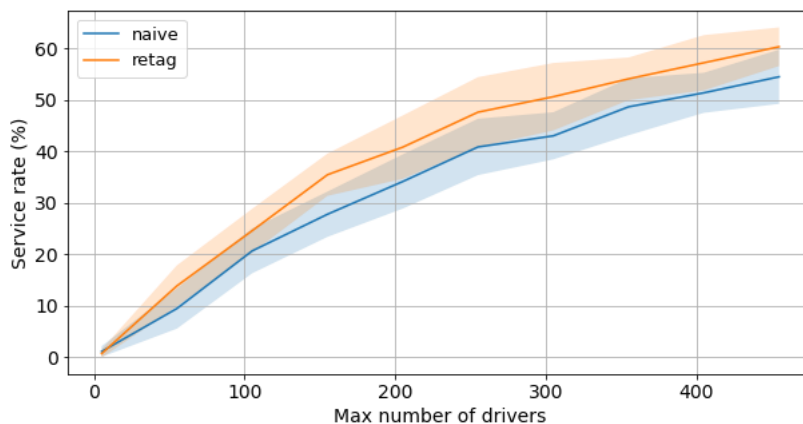


FIGURE 3.9 – Proportion de demandes de passager satisfaites en fonction du nombre maximum de conducteurs et des comportements de passager : naïf en bleu, ReTAG en orange. Plus les valeurs sont élevées, meilleur est le résultat, puisque l'objectif est de maximiser cette proportion.

Ces résultats montrent une fois de plus l'efficacité du comportement du passager ReTAG par rapport au comportement du passager naïf. En effet, le comportement ReTAG permet à un plus grand nombre de passagers d'atteindre leur destination que le comportement naïf. Bien que seule une différence moyenne de 5 à 10% du taux de service soit constatée entre les deux comportements sur la Figure 3.9, cette différence atteint plus de 20% quand l'on considère les écarts-types maximaux et minimaux de chaque comportement. De plus, on constate que dans le pire des cas, les passagers ReTAG font au moins mieux que la moyenne des passagers naïfs. La capacité des passagers ReTAG à construire des itinéraires complexes et optimisés leur permet d'atteindre leur destination plus souvent que les passagers naïfs. En effet, par rapport à la simulation précédente, nous avons multiplié par 4 l'espace pour renforcer le caractère complexe de la simulation. Il est donc possible que certains passagers se trouvent à une très grande distance de leur nœud d'arrivée. Ceci explique la relativement faible proportion de demandes satisfaites, quel que soit le comportement des passagers.

En ce qui concerne le temps de trajet, nous pouvons voir sur la Figure 3.10 que les passagers avec le comportement ReTAG voyagent en moyenne plus longtemps que les passagers naïfs. Ces résultats peuvent être expliqués par le fait qu'une partie des passagers qui n'ont pas réussi à atteindre leur destination avec le comportement naïf ont réussi avec le comportement ReTAG. Les passagers ReTAG éloignés de leur

2225 destination ou dans des situations défavorables ont ainsi dû construire un itinéraire complexe et multi-saut pour rejoindre leur arrivée. Ces itinéraires conduisent à des temps de voyage plus longs en raison des détours effectués par les passagers pour qu'ils s'assurent d'atteindre leur destination. En considérant dans nos résultats uniquement les passagers qui ont réussi à atteindre leur destination, il n'est pas étonnant de voir un temps de trajet plus important avec une stratégie multi-saut par rapport à une stratégie simple-saut.

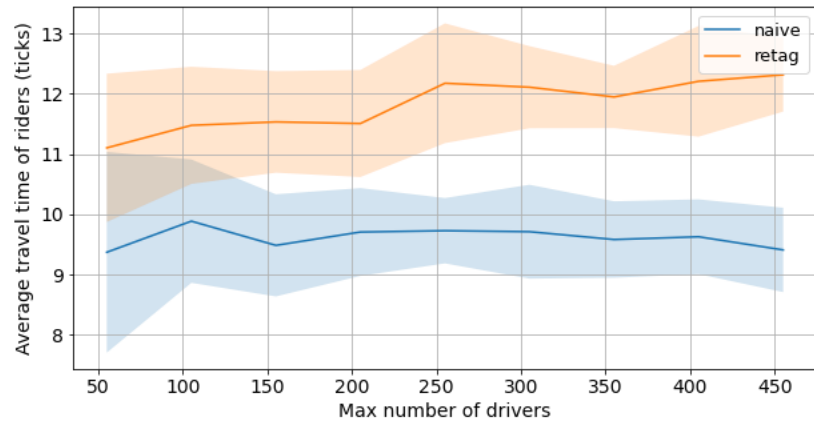


FIGURE 3.10 – Temps de trajet moyen des passagers en fonction du nombre maximal de conducteurs et des comportements des passagers : naïf en bleu, ReTAG en orange.

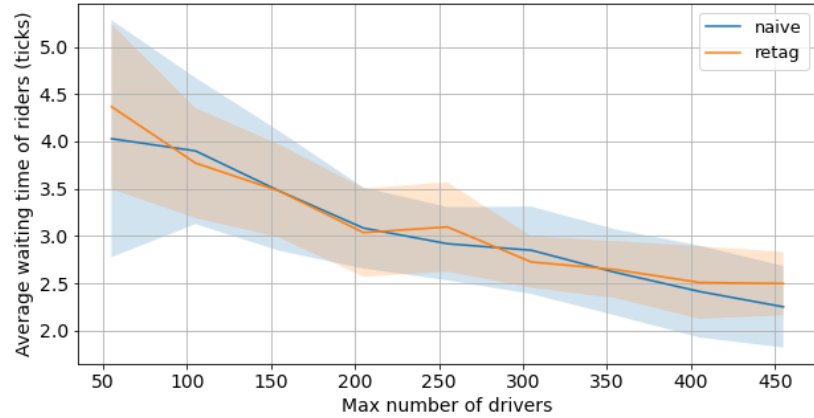


FIGURE 3.11 – Temps d'attente moyen des passagers en fonction du nombre maximal de conducteurs et des comportements des passagers : naïf en bleu, ReTAG en orange.

2230 Enfin, on peut constater dans la Figure 3.11 que le temps d'attente des passagers est sensiblement le même, peu importe le comportement. On en conclut alors que le fait que plus de passagers aient atteint leur destination avec le comportement ReTAG n'a pas eu d'impact notable sur le temps d'attente.

3.4.4 Discussions et Perspectives

Les deux types de simulations réalisées dans cette section nous permettent d'avoir deux perspectives distinctes sur la performance et la robustesse du comportement ReTAG par rapport au comportement naïf. Un premier point à considérer est l'efficacité dans des situations complexes. Ainsi, lorsqu'il y a peu de conducteurs ou lorsque le passager se trouve dans une situation défavorable (éloigné de son nœud d'arrivée ou dans une zone avec peu de conducteurs, par exemple), le comportement ReTAG cible un chemin lui assurant d'atteindre sa destination, si un tel chemin existe. En revanche, dans le cas d'une offre de covoiturage plus abondante, il est capable d'optimiser son temps d'attente en restant mobile, quitte à faire des détours et à allonger son temps de trajet. En conclusion, le comportement ReTAG permet à un passager d'établir un raisonnement complexe et adaptable à de nombreuses situations.

Cependant, on constate dans les différents résultats que l'autostop classique reste une solution performante dans notre système quand l'offre est abondante. Il faut toutefois mettre en perspective le fait que nous considérons dans cette étude que tous les conducteurs sont d'accord pour covoiturer, indifféremment avec des auto-stoppeurs naïfs ou connectés, ce qui n'est substantiellement pas le cas dans le monde réel. Dans la réalité, on peut supposer que les conducteurs seront beaucoup plus enclins à prendre un passager connecté, avec éventuellement une réputation, qu'un passager inconnu et non référencé. La perspective d'un taux de pénétration de marché pourrait aussi être intéressante à étudier. La pénétration de marché consiste à sélectionner la proportion de conducteurs ayant la volonté de covoiturer et est étudiée dans de nombreux articles sur le covoiturage tel que part [Nourinejad et Roorda, 2016, Chen *et al.*, 2019].

De plus, il serait intéressant d'approfondir le modèle en dépassant l'abstraction du temps et de l'espace. Chaque arête aurait alors une distance et une limite de vitesse permettant de calculer un temps de parcours réaliste pour chaque trajet. C'est notamment ce que nous proposons dans notre prochain chapitre 4.

Enfin, la comparaison avec le problème du *slugging* [Ma et Wolfson, 2013], c'est-à-dire l'auto-stop avec des lieux de rencontre prédéfinis, peut être très intéressante pour estimer l'impact de la concentration de l'offre et de la demande comme moyen de connexion et de transfert entre les passagers et les conducteurs.

3.5 Conclusion du chapitre

Cette étude a été conçue comme un cadre permettant d'aborder le problème du
2265 covoiturage sous une forme purement dynamique et minimale, à savoir, le problème
de l'auto-stop. Ce dernier nous a permis d'une part de découvrir la complexité et les
enjeux du covoiturage dynamique multi-saut à la demande. Et de contribuer d'autre
part à la littérature avec une forme de covoiturage qui n'a été que très peu étudiée
par le passé et la création d'un modèle de graphe artificiel reproductible et générique.
2270 Nous avons réalisé un article et une présentation de ces travaux à l'occasion de la 14e
conférence ICAART¹ (*International Conference on Agents and Artificial Intelligence*)
de 2022 : [Fèvre *et al.*, 2022].

1. <https://icaart.scitevents.org/>

Chapitre 4

Le covoiturage dynamique multi-saut

2275 Dans ce chapitre, nous proposons d'étendre le travail effectué précédemment en apportant la notion de détour et de réservation, tout en faisant évoluer nos agents sur des graphes issus de données réelles. Comme dans les travaux du chapitre 3, les agents conducteurs et passagers apparaissent dans le système de manière dynamique et essaient de partager leur trajet en fonction de leurs préférences et de leurs contraintes.

2280 Chaque covoiturage doit dorénavant faire l'objet d'une réservation (cf. section 2.1.2), et donc d'un contrat entre un conducteur et un passager. Ce contrat est, d'une part, l'assurance d'être pris en charge pour un passager : un conducteur viendra le chercher et le déposera à une heure et un lieu précis, son siège est réservé dans le véhicule, il s'agit d'une garantie sans surprise. D'autre part, le conducteur est assuré
2285 que le passager ne se désistara pas, et donc qu'un éventuel détour ne sera pas effectué en vain. Ce contrat implique aussi qu'il n'est plus possible pour un passager, ou un conducteur, de changer d'avis en cours de route, même si une meilleure solution apparaît dans le système.

De plus, les conducteurs peuvent dorénavant choisir d'effectuer des détours (cf.
2290 section 2.1.3) pour prendre ou déposer des passagers et ainsi élargir l'offre de transport du système de covoiturage. Ce détour prend la forme d'une modification d'itinéraire en temps réel limitée par l'heure d'arrivée souhaitée du conducteur ainsi que par les passagers avec lesquels il est sous contrat.

Aussi, nous effectuons une discrétisation d'une carte du monde réel en la transformant en graphe de route. Les distances sont réelles et parcourues par des véhicules
2295

disposant d'une vitesse. Les conducteurs peuvent parfois être entre deux nœuds, et donc sur une arête en quelque sorte. La topologie du graphe est respectée et toute entité présente dans l'environnement peut être située par un couple de coordonnées latitudinales et longitudinales. Ces coordonnées nous permettent d'utiliser une structure de données sous forme d'arbre, nommée R-arbre (*R-Arbre* en anglais), permettant une indexation dynamique et une exploration spatiale de données multidimensionnelles. Dans ce chapitre, nous utilisons cet outil pour optimiser le coût computationnel lié à l'identification des conducteurs candidats au covoiturage destinée aux passagers de notre système.

La perception de ces agents est définie dynamiquement et est limitée par leurs préférences spatio-temporelles. Nous utilisons la technique d'indexation spatiale R-Arbre pour stocker efficacement ces données et effectuer des requêtes superposées sur des points et des zones de l'espace (voir section 4.2.1). Nous détaillons deux formes de covoiturage : le covoiturage simple-saut, entre un passager et un seul conducteur, et le covoiturage multi-saut, où un passager effectue des transferts entre plusieurs conducteurs.

4.1 Formulation du problème

La flexibilité d'un système de covoiturage peut être modélisée et quantifiée par deux opérations. D'une part, les conducteurs peuvent choisir de faire des détours pour prendre ou déposer des passagers, et ainsi élargir l'offre de transport du système de covoiturage. D'autre part, les passagers peuvent choisir de passer d'un véhicule à l'autre, et donc d'effectuer un transfert, afin d'arriver à leur destination en plusieurs étapes, si cela est plus pratique pour eux (temps de trajet réduit, moins d'attente, etc.). Un exemple de ces opérations est illustré dans la Figure 4.1.

Le défi d'un tel cadre est d'équilibrer l'offre et la demande. Cet équilibre implique de nombreux paramètres tels que le ratio conducteur-passager, le détour acceptable par les utilisateurs en termes de temps et de distance ou encore le temps d'attente admissible par les passagers. Le problème est alors d'associer au mieux les passagers et les conducteurs en exploitant leurs marges de manœuvre afin de satisfaire chaque partie.

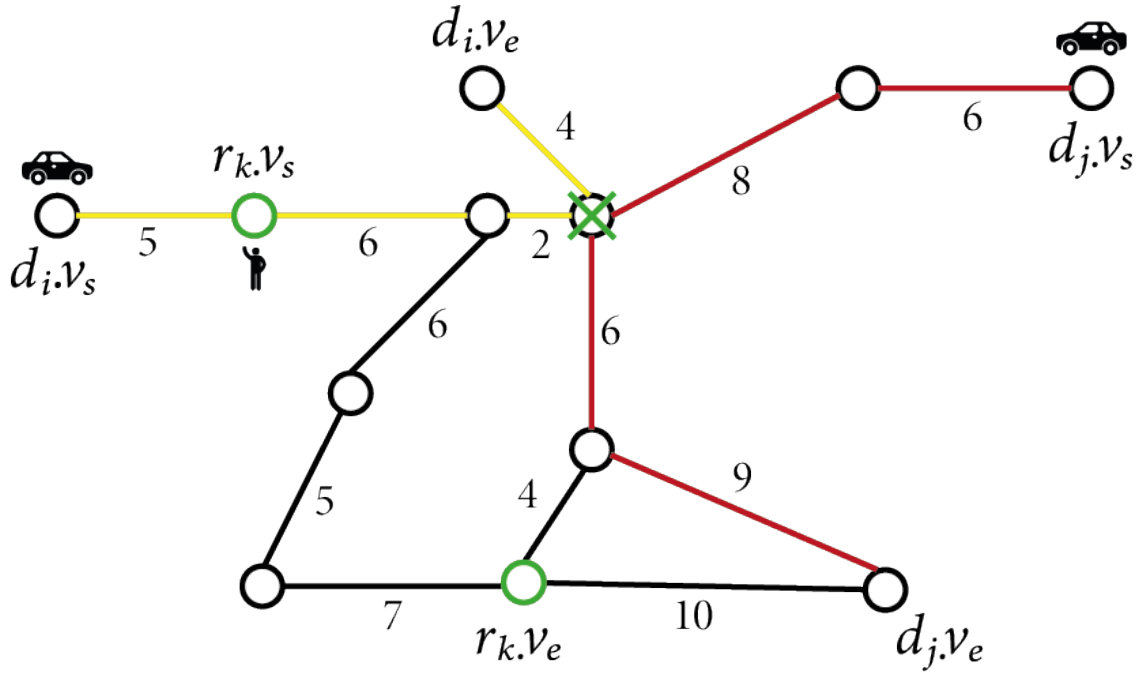


FIGURE 4.1 – Exemple de covoiturage à multi-saut Deux conducteurs d_i et d_j suivent leur itinéraire, respectivement en jaune et rouge, reliant leur nœud de départ v_s et leur nœud d'arrivée v_e . Un passager r_k attend sur son nœud de départ $r_k.v_s$ et veut rejoindre son nœud d'arrivée $r_k.v_e$, tous deux entourés de vert. Les trajectoires des conducteurs se croisent sur le nœud avec une croix verte. Un transfert est alors réalisable, d_i peut partager son trajet avec r_k puis, une fois arrivé au nœud de transfert, le passager peut effectuer un transfert avec le conducteur d_j . Le conducteur d_j doit cependant faire un détour pour déposer le passager, ce qui rajoute 5 unités de temps à son temps de trajet initial.

4.1.1 Spécification du réseau routier et des agents du système

Le réseau routier

Le réseau routier de notre système est modélisé par un graphe orienté et connexe $G = \langle V, E \rangle$ dans lequel V est l'ensemble des nœuds du graphe et E est l'ensemble des arêtes liant ces nœuds. Différents poids sont attribués aux arêtes désignant le coût du trajet entre deux nœuds. Pour une arête $e(v_i, v_j)$, on note la distance de parcours $d(v_i, v_j)$, la vitesse maximale autorisée sur ce tronçon $s(v_i, v_j)$ et enfin le temps de parcours $t(v_i, v_j)$, qui peut être calculé à partir de ces deux premières données. Un nœud $v_i = (x, y)$ représente une intersection de routes ou une impasse et est référencé spatialement avec un couple de coordonnées : une longitude x et une latitude y . Cette indexation spatiale nous permet d'effectuer des requêtes sur les nœuds présents dans une zone délimitée par un rectangle englobant ("*bounding box*" en anglais). Un

rectangle englobant est décrit par ses latitudes et longitudes minimales et maximales $BBOX(min_lat, min_lon, max_lat, max_lon)$. Nous désignons par le plus court chemin par $SP(G, v_i, v_j, w)$, défini comme le chemin entre deux nœuds $v_i, v_j \in V$ minimisant la somme des poids w , où w peut être le temps, la distance ou toute autre variable préalablement ajoutée aux arêtes du graphe. Nous désignons la longueur du plus court chemin par $SPd(G, v_i, v_j, w)$ et la durée du plus court chemin par $SPt(G, v_i, v_j, w)$.

2345 Nous utilisons une simulation en temps discret, le temps évolue à intervalles constants et le pas de simulation courant est dénoté par la variable *time*. À chaque pas de simulation, tous les passagers et conducteurs présents dans le système peuvent effectuer une action en fonction de leur comportement.

Les agents utilisateurs

2350 Un agent utilisateur du système $u_i = \langle v_s, v_l, v_e, det_{max} \rangle$, avec $u_i \in U$ représentant un passager ou un conducteur dans le système de covoiturage, est initialisé par quatre éléments : le nœud de départ v_s , le nœud de localisation actuel v_l , le nœud d'arrivée v_e et le facteur de détour det_{max} . Le facteur de détour quantifie la volonté de détour d'un agent utilisateur. Ainsi, s'il vaut 0, l'utilisateur ne souhaite pas faire de détour, et s'il
2355 vaut 1, l'utilisateur est prêt à doubler son trajet initial pour effectuer des détours. Nous faisons varier ce facteur entre 0 et 1 dans nos expériences. La première action d'un agent utilisateur dans le système est de calculer son heure d'arrivée la plus tardive t_e . L'heure d'apparition d'un agent utilisateur t_s correspond au pas de simulation de son entrée dans le système de covoiturage. Il s'agit alors, pour un conducteur, de l'heure
2360 du début de son voyage et, pour un passager, de l'heure de sa première requête de covoiturage. L'heure d'arrivée la plus tardive t_e est calculée en ajoutant au temps d'apparition de l'agent t_s , le temps nécessaire pour parcourir le plus court chemin entre ses nœuds de départ et d'arrivée augmenté de son temps de détour maximal acceptable, comme suit : $t_e = t_s + SPt(G, v_s, v_e, w) * (1 + det_{max})$. Cet élément permet
2365 de définir les limites d'action et de perception d'un agent. L'ensemble des agents utilisateurs est noté $U = \{u_1, u_2, \dots, u_n\}$ et cet ensemble représente la totalité des agents conducteurs D et passagers R du système : $U = D \cup R$.

Les agents conducteurs

Un agent conducteur $d_i = \langle v_s, v_l, v_e, det_{max}, c_{max} \rangle$, avec $d_i \in D$, est initialisé par
2370 cinq éléments. Un conducteur étant un utilisateur, il hérite des quatre premiers paramètres définis précédemment. Le dernier paramètre c_{max} définit le nombre maximal de sièges disponibles dans le véhicule du conducteur. Dans notre système, un conducteur a pour objectif d'atteindre sa destination avant son heure d'arrivée la plus tardive. Il suit donc son trajet de nœud en nœud, en prenant systématiquement le chemin
2375 le plus court. S'il doit faire un détour pour prendre et déposer un passager, il doit être capable de déterminer si cette action est possible vis-à-vis de son heure d'arrivée la plus tardive. Pour ce faire, il maintient un ordonnanceur de voyage composé de différentes tables stockant des informations spatio-temporelles. Dans ce contexte, nous étendons l'architecture de planification des déplacements présentée dans [Xu
2380 *et al.*, 2019, Xu *et al.*, 2020, Tong *et al.*, 2018] en ajoutant plusieurs nouvelles listes. L'architecture des travaux cités décrit principalement 3 listes : la liste **arr** stocke le temps le plus court nécessaire pour atteindre chaque arrêt ; la liste **ddl** stocke la dernière heure d'arrivée possible (*ddl* pour *deadline* en anglais) pour chaque arrêt afin de respecter les contraintes du conducteur (dernière heure d'arrivée) ou des pas-
2385 sagers transportés (temps d'attente et dernière heure d'arrivée). Enfin, la liste **slk** stocke le temps de marge de détours maximal disponible avant chaque arrêt (*slk* pour *slack time* en anglais). Notre contribution à cette architecture de référence est la suivante : la liste **stops** stocke les nœuds des arrêts programmés dans l'ordonnanceur de voyage du conducteur. Le premier élément de la liste **stops** est le prochain nœud à
2390 atteindre par le conducteur, c'est-à-dire celui qu'il atteindra à la fin de son transit sur l'arête actuelle. Les nœuds suivants, à l'exception du dernier, sont les lieux de prise en charge et de dépose des passagers, le cas échéant. Le dernier nœud correspond au nœud d'arrivée du conducteur. Le nombre de places disponibles à chaque arrêt est stocké par la liste **c**. Ce nombre de places est mis à jour à la réservation d'un
2395 passager entre son nœud de départ inclus et son nœud d'arrivée exclu (le passager quittant le véhicule, la place est disponible). Les boîtes englobantes associées aux perceptions de l'agent entre deux arrêts sont stockées dans la liste **p**. Dans les travaux précédents associés à cette architecture, les chercheurs étudient le covoiturage profes-
2400 sionnel. Dans ce cas, les conducteurs n'ont pas de limite de temps ou de distance à l'instanciation et la perception est donc infinie. Dans ces travaux, nous étudions le covoiturage entre particuliers, nous ajoutons donc à cette initialisation les contraintes spatio-temporelles du conducteur telles que l'heure d'arrivée la plus tardive et le nœud

d'arrivée. Ainsi, pour un agent conducteur d_i , et pour $x[y]$ l'élément à l'index y de la liste x , $stops[1] = d_i.v_e$, $c[1] = d_i.c_{max}$, $arr[1] = d_i.t_s + SPt(G, d_i.v_s, d_i.v_e, w)$,
 2405 $ddl[1] = d_i.t_e$ et $slk[1] = ddl[1] - arr[1]$.

L'ensemble des agents du conducteur est noté $D = \{d_1, d_2, \dots, d_n\}$ et, en admettant qu'il y ait toujours plusieurs conducteurs et plusieurs passagers dans le système, cet ensemble est un sous-ensemble propre de l'ensemble des agents de l'utilisateur U , donc $D \subset U$.

2410 Les agents passagers

Un agent passager $r_i = \langle v_s, v_l, v_e, det_{max}, wt_{max}, pref \rangle$, avec $r_i \in R$, est initialisé par six éléments. Un passager étant un utilisateur du système, il hérite des quatre premiers paramètres définis précédemment. Le paramètre wt_{max} correspond au temps d'attente acceptable au maximum par le passager pour tout le trajet. En effet, il
 2415 est clair qu'un passager n'attendra pas éternellement pour être pris en charge. Le temps d'attente maximal d'un passager dépend de ses préférences, qui sont elles-mêmes dues à l'état de son environnement et à la nature du trajet (pluie, urgence...). Cette limite d'attente est ajoutée au calcul de son heure d'arrivée la plus tardive :
 $t_e = t_s + SPt(G, v_s, v_e, w) * (1 + det_{max}) + wt_{max}$. On utilise également cette limite
 2420 pour calculer l'heure de départ la plus tardive en l'additionnant à l'heure d'apparition du passager $t_{dep} = t_s + wt_{max}$. La perception d'un agent passager est représentée et stockée dans la variable p . Cette perception permet au passager de ne retenir que les conducteurs accessibles, en excluant les conducteurs trop éloignés pour le prendre en charge.

2425 L'ensemble des agents passagers est noté $R = \{r_1, r_2, \dots, r_n\}$ et, pour les mêmes raisons et circonstances que les conducteurs, cet ensemble est un sous-ensemble propre de l'ensemble des agents des utilisateurs U , donc $R \subset U$.

L'agent service de transport

Un agent service de transport $tsa_i = \langle RT_p \rangle$ est responsable de la connexion
 2430 des agents utilisateurs. Il remplit deux fonctions. La première consiste à stocker et à mettre à jour toutes les perceptions des agents conducteurs dans un R-Arbre RT_p . Les agents conducteurs envoient des requêtes à l'agent service de transport lorsqu'un

changement de leur statut entraîne une modification de leurs perceptions, par exemple à l'occasion d'un nouveau covoiturage avec un passager. L'agent service de transport ajoute, supprime ou modifie alors ces perceptions. La seconde fonction est de répondre aux requêtes spatiales des agents passagers du système. De la même manière, les agents passagers envoient des requêtes spatiales (sous-section 4.2.1) pour identifier les candidats au covoiturage. L'agent service de transport parcourt alors le R-Arbre RT_p pour identifier ces agents conducteurs et renvoie le résultat aux agents passagers. L'agent service de transport est donc implémenté comme un "tableau noir" afin de mettre à jour et de transmettre les informations des agents du système lorsque cela est demandé.

L'ensemble des interactions entre ces agents est résumé dans la Figure 4.2. Comme on peut le voir, le comportement d'un agent passager est basé sur deux algorithmes : BFMO (section 4.2.2) et CIER (section 5.2.2). L'algorithme BFMO gère le processus d'optimisation en maintenant les meilleures associations et en identifiant le meilleur nœud de transfert dans le contexte d'un covoiturage multi-saut.

4.1.2 Contraintes et Formulations

Nous limitons notre étude au covoiturage à un et deux sauts, c'est-à-dire au covoiturage limité à deux conducteurs pour un transfert. En effet, plusieurs articles comme [Masoud et Jayakrishnan, 2017] ont montré que l'augmentation du nombre de transferts possibles améliore à peine les performances d'un système de covoiturage. Mais cela n'exclut pas l'intérêt d'étendre cette étude dans de futurs travaux au covoiturage à n sauts, pour n supérieur à 2.

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

- **Contrainte temporelle** : Les conducteurs et les passagers doivent arriver à leur destination avant leur dernière heure d'arrivée t_e . Les conducteurs doivent arriver avant l'heure limite ddl pour chaque nœud contenu dans $stops$. Un passager doit être pris en charge avant que son temps d'attente ne soit atteint, c'est-à-dire avant son heure de départ la plus tardive t_{dep} .
- **Contrainte de capacité** : La capacité courante d'un conducteur ne doit jamais dépasser la capacité maximale c_{max} du véhicule.
- **Contrainte de l'ordre des arrêts** : Un conducteur partageant son itinéraire

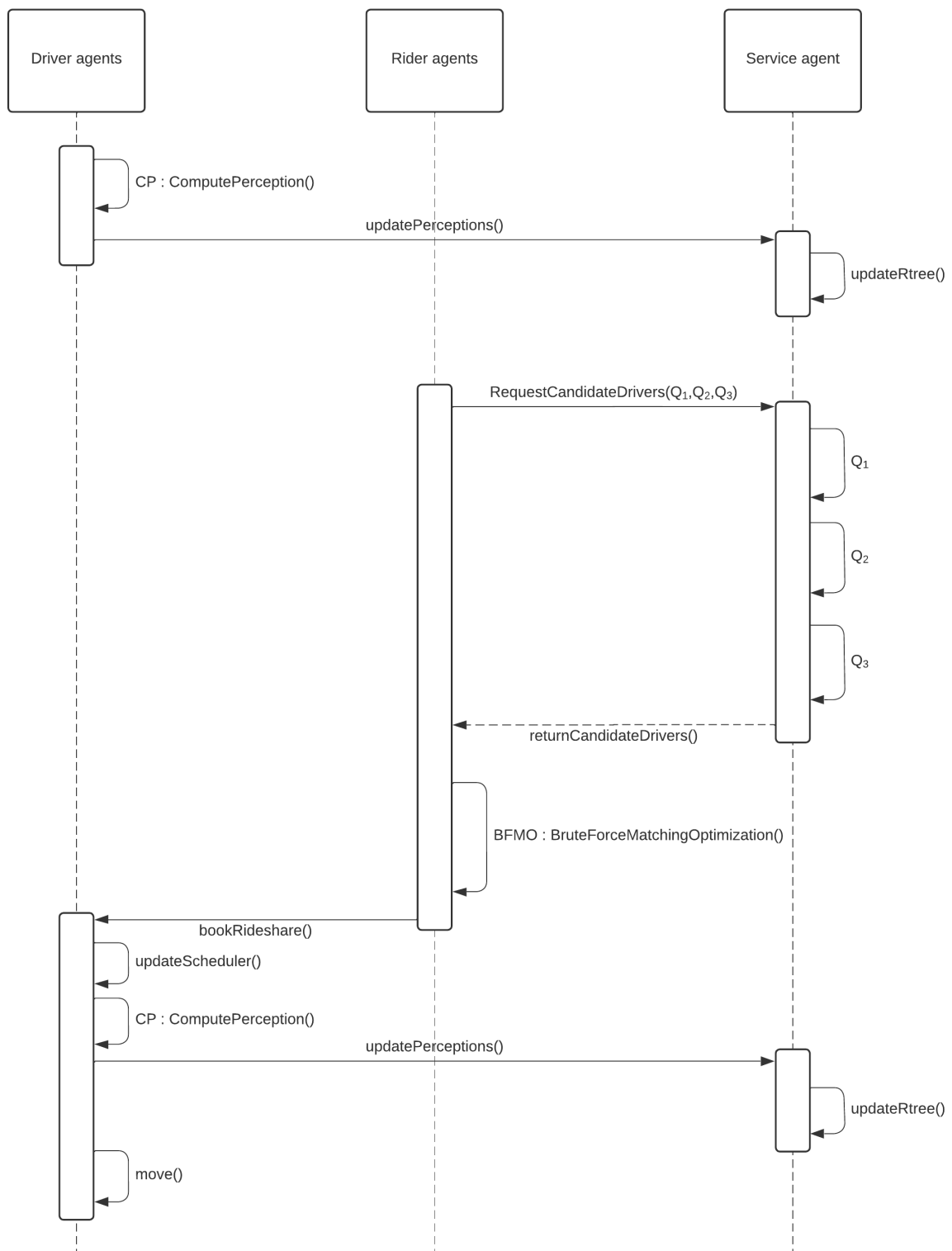


FIGURE 4.2 – Le diagramme séquentiel des agents du système. Les agents sont représentés par des lignes de vie. Les différentes actions et communications entre agents sont ordonnées chronologiquement de haut en bas. Les phases d’activation des agents sont représentées par les plus petits rectangles et les flèches représentent les appels de fonction et les communications interagents.

2465 avec un passager doit d'abord passer par le nœud de prise en charge du passager
avant de visiter le nœud de dépôt de ce dernier.

Une association de covoiturage ("*a match*" en anglais) $m \in r_i.M$ est définie par plusieurs variables : un ou deux conducteurs $d_j, d_k \in D$ (simple-saut : $j = k$, multi-saut : $j \neq k$), un nœud de transfert v_tsf le cas échéant, un nombre de transferts
2470 nb_tsf (0 ou 1 dans cette étude) et une heure d'arrivée arr_time . Nous choisissons de définir la fonction objectif des passagers comme la minimisation de leur heure d'arrivée arr_time , puis celle du nombre de transferts nb_tsf . Ainsi, à la même heure d'arrivée, les agents passagers préfèrent un trajet impliquant un seul conducteur (simple-saut) plutôt que deux (multi-saut). Nous justifions ce choix, d'une part, car
2475 l'heure d'arrivée est directement dérivée du temps d'attente et du temps de trajet, deux éléments que nous souhaitons minimiser. Et, d'autre part, car nous admettons que, dans le monde réel, un passager favorise un trajet sans transfert pour une question de confort.

Nous définissons ces deux objectifs comme :

- 2480 — $arr_time = t_s + total_wt + travel_time$: l'heure d'arrivée de l'agent passager. Il correspond à la somme de son heure d'apparition t_s , de son temps d'attente $total_wt$ et de son temps de trajet $travel_time$.
- $nb_tsf = nb_d - 1$: le nombre total de transferts d'un agent passager pour atteindre sa destination. Il correspond au nombre de conducteurs impliqués dans
2485 le covoiturage nb_d , soustrait de 1. En effet, un covoiturage avec 1 seul conducteur n'implique pas de transfert (0) alors qu'un covoiturage avec 2 conducteurs implique 1 transfert.

Notre simulation et notre optimisation étant individu-centrées, nous n'optimisons pas ces objectifs de manière globale. Les variables observées du point de vue de
2490 la population sont alors considérées comme des indicateurs de performance et nous permettent d'évaluer l'impact d'un comportement sur la résolution d'une instance de covoiturage.

4.2 Le système multi-agents proposé

Dans cette section, nous nous concentrons sur l'aspect comportemental du système
2495 multi-agents décrit dans la section précédente à travers la figure 4.2. Un agent est une entité intelligente et autonome réalisant des actions en fonction de sa propre

perception de l'environnement et de ses préférences. L'intérêt d'un tel système est de garantir l'hétérogénéité de la population étudiée afin de pouvoir construire une large gamme de comportements. Il permet également de limiter l'information disponible
2500 pour chaque agent avec une perception locale basée sur ses préférences et son état.

Nous notons que nous détaillons essentiellement dans cette section le comportement des agents représentant les utilisateurs de notre système de covoiturage, à savoir les conducteurs et les passagers. L'agent service de transport est défini comme un agent cognitif "module" ou réflexif [Ferber et Weiss, 1999]. Ce dernier est un agent
2505 auxiliaire capable de répondre à des questions ou d'accomplir des tâches qui lui sont ordonnées par d'autres agents, sans que ces demandes soient explicitement énoncées comme des objectifs de cet agent auxiliaire.

À chaque étape de la simulation, le tour d'un agent utilisateur est divisé en deux phases : la phase de mise à jour de sa perception de l'environnement et la phase de
2510 prise de décision.

4.2.1 La mise à jour de la perception des agents

La perception d'un agent utilisateur représente ce qu'il est capable d'utiliser dans l'environnement pour atteindre son objectif. Dans nos travaux, cette perception est dynamique et doit être mise à jour régulièrement. Un conducteur utilise essentielle-
2515 ment l'espace, c'est-à-dire le graphe routier. Son objectif est de trouver le meilleur itinéraire à suivre pour transporter les passagers et atteindre sa destination. Un passager utilise l'espace et les agents conducteurs. Il a besoin des informations des conducteurs présents dans sa perception pour choisir la meilleure association. Dans cette sous-section, nous définissons ces deux perceptions.

2520 La perception de l'espace

Dans cette phase, chaque agent utilisateur du système met à jour sa perception de l'espace pour lui-même et pour les autres agents du système. Cette perception est définie à partir des attributs et des préférences des agents et représente l'ensemble des nœuds qu'il peut utiliser pour atteindre sa destination avant l'heure limite d'arrivée.
2525 Cette perception dynamique a fait l'objet d'un article à l'occasion de la *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* [Fevre et al.,

2021].

Déterminer la perception exacte d'un agent utilisateur u_i est une tâche très lourde en termes de calcul. Il faudrait vérifier, pour chaque nœud v_j du graphe, si la somme du
2530 temps nécessaire pour aller de la position actuelle de l'agent au nœud v_j et du temps
nécessaire pour aller de ce nœud v_j à la destination de l'agent est inférieure ou égale
à l'heure d'arrivée la plus tardive. Soit, $v_j \in u_i.p$ seulement si $SPt(G, u_i.v_l, v_j, u_i.w) +$
 $SPt(G, v_j, u_i.v_e, u_i.w) \leq u_i.t_e$. Nous devrions donc effectuer une recherche en largeur
(*breadth-first* en anglais) et de nombreux calculs de plus courts chemins pour réaliser
2535 une telle tâche. Il est difficilement concevable d'appliquer cette méthode sur une
instance de problème de taille importante. Nous avons donc choisi d'approximer dans
un premier temps la perception d'un agent en utilisant l'algorithme 3 nommé CP
pour *Compute Perception*.

Algorithm 3 Compute Perception BBOX from slack time (CP)

Input : G = road infrastructure graph, $orig$ = origin node, $dest$ = destination node, slk = slack time
Output : $BBOX$ = perception of the agent

```
1:  $sp = SP(G, orig, dest, w)$ 
2:  $min\_lat, min\_lon, max\_lat, max\_lon = None$ 
3: for all  $node \in sp$  do
4:    $min\_lat = \min(min\_lat, node.min\_lat)$ 
5:    $max\_lat = \max(max\_lat, node.max\_lat)$ 
6:    $min\_lon = \min(min\_lon, node.min\_lon)$ 
7:    $max\_lon = \max(min\_lon, node.min\_lon)$ 
8: end for
9: if  $slk > 0$  then
10:   $speed = system\_speed/3.6$  #from km/h to m/s
11:   $meters = speed * slk$ 
12:   $earth\_radius = 6378.137$  #km
13:   $m = (1/((2 * pi/360) * earth\_radius))/1000$  #1 meter in degree
14:   $slk\_min\_lat = min\_lat - (meters * m)$ 
15:   $slk\_max\_lat = max\_lat + (meters * m)$ 
16:   $slk\_min\_lon = min\_lon - (meters * m)/\cos(slk\_min\_lat + pi/180)$ 
17:   $slk\_max\_lon = max\_lon + (meters * m)/\cos(slk\_max\_lat + pi/180)$ 
18: end if
19:  $slk\_min\_lat = \max(min\_lat, G.min\_lat)$ 
20:  $slk\_max\_lat = \min(max\_lat, G.max\_lat)$ 
21:  $slk\_min\_lon = \max(min\_lon, G.min\_lon)$ 
22:  $slk\_max\_lon = \min(max\_lon, G.max\_lon)$ 
23:  $BBOX = (slk\_min\_lat, slk\_min\_lon, slk\_max\_lat, slk\_max\_lon)$ 
24: return  $BBOX$ 
```

Pour calculer la perception $p_{j,k}$ d'un agent u_i entre deux nœuds v_j et v_k du sys-
2540 tème, on détermine d'abord le plus court chemin entre ces deux nœuds (L1). On
parcourt les nœuds de ce plus court chemin en cherchant les longitudes et latitudes
minimales et maximales (L3-8). Ces limites nous permettent de créer le rectangle
englobant minimal *MBR* (pour "*minimum bounding rectangle*", en anglais) du plus
court chemin de l'agent comme on peut le voir sur la Figure 4.3.

2545 Nous effectuons ensuite une extension du *MBR* avec le temps de marge de trajet

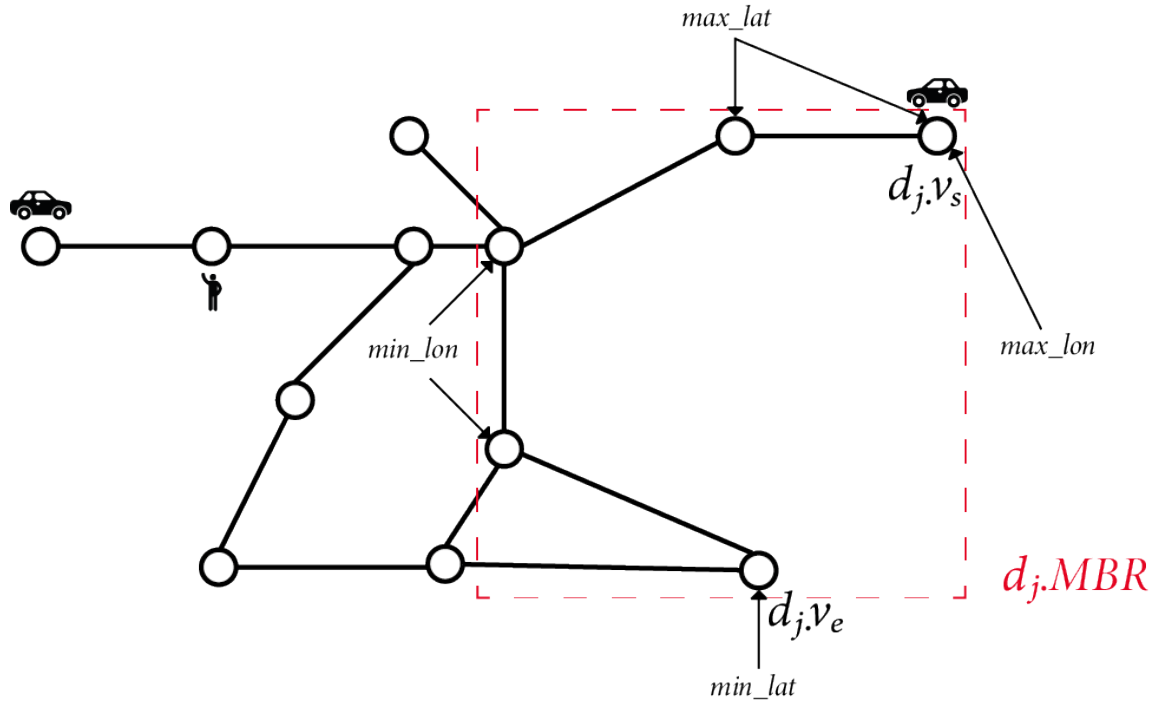


FIGURE 4.3 – Exemple de rectangle englobant minimal (MBR) du plus court chemin d'un conducteur. Un conducteur d_j délimite le MBR en pointillés rouges de son plus court chemin. Ce dernier recueille les longitudes et latitudes maximales et minimales ($min_lat, min_lon, max_lat, max_lon$) à partir des nœuds composant son plus court chemin.

de l'agent. Nous rappelons que cette marge est nommée slk et résulte des préférences de l'agent et de son état courant (détours effectués, temps restant...). Pour effectuer cette extension, nous récupérons la vitesse et la marge de temps de l'agent pour calculer la marge en termes de distance. Nous convertissons cette distance en degrés et étendons le MBR en conséquence (L9-18) afin d'établir le rectangle englobant étendu du trajet de l'agent, nommé $BBOX$ (pour "*bounding box*", en anglais). Enfin, nous restreignons le $BBOX$ pour qu'il ne dépasse pas les limites du graphe étudié (L19-23). Le rectangle englobant étendu $BBOX$ représente alors la perception d'un agent entre deux nœuds comme illustré dans la Figure 4.4.

Cette méthode est utilisée pour la perception des conducteurs, entre chaque arrêt, et pour la perception du passager, de son point de départ à son point d'arrivée.

Pour un agent conducteur, cette perception prend la forme d'une liste de perceptions. Pour chaque arrêt présent dans son ordonnanceur de voyage, il maintient le rectangle englobant $BBOX$ associé. La perception de chaque conducteur est notifiée à l'agent service de transport. Elle peut donc être utilisée par les passagers lors de la phase d'identification des candidats à un covoiturage. Il faut noter que cette

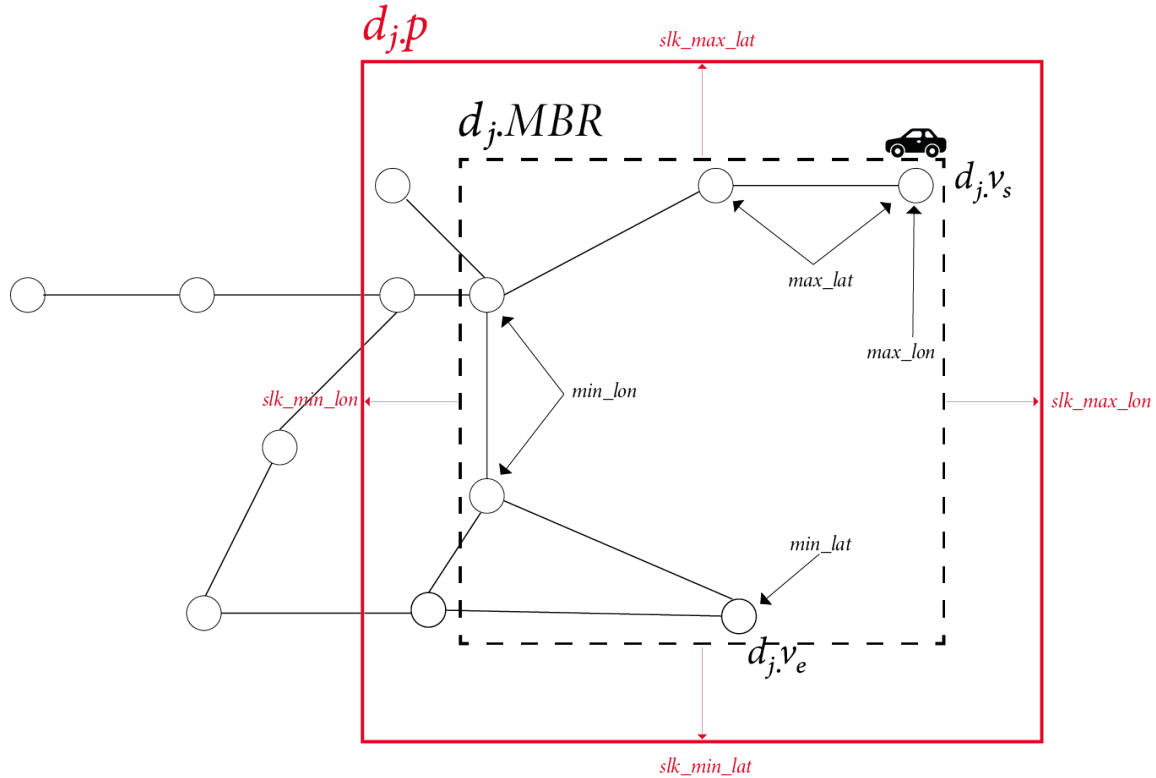


FIGURE 4.4 – Exemple de calcul de la perception d’un agent conducteur. Un conducteur d_j délimite le MBR en pointillés noirs de son plus court chemin, de $d_j.v_s$ à $d_j.v_e$. Ce conducteur collecte ensuite les longitudes et latitudes maximales et minimales (min_lat , min_lon , max_lat , max_lon) des nœuds composant son plus court chemin. À partir de ces coordonnées et de son temps de marge slk , il est capable de calculer sa perception $d_j.p$ représentée par le cadre rouge.

perception est approximative et qu’un test de faisabilité plus rigoureux, basé sur les contraintes de la section 4.1.2, est nécessaire.

2565 La mise à jour de la perception des utilisateurs est une tâche qui peut être lourde en termes de calcul selon l’environnement. Nous avons donc mis en place des mécanismes pour limiter le nombre de ces mises à jour afin que les agents de notre système ne rafraîchissent pas nécessairement leur perception à chaque étape de la simulation, mais seulement si nécessaire.

2570 Les agents conducteurs ne mettent dès lors à jour leurs perceptions que dans deux cas. Le premier cas est lorsqu’ils atteignent un nœud du graphe. Dans notre système, les conducteurs qui s’engagent sur une arête du graphe ne peuvent pas faire demi-tour en cours de route. Ils doivent d’abord atteindre le nœud à l’autre extrémité de l’arête avant de changer de direction. Il n’est donc pas nécessaire de rafraîchir la perception à chaque étape de la simulation. L’autre cas est celui d’un covoiturage dont le contrat
2575 a été conclu. L’insertion de la route d’un agent passager dans l’itinéraire de l’agent

conducteur implique dans la plupart des cas une diminution de sa marge de temps (*slk*), et donc de sa capacité de détour. En effet, à moins que les nœuds de départ et d'arrivée ou de transfert d'un passager ne soient exactement sur l'itinéraire initial du conducteur, ce dernier doit faire un détour et donc utiliser une partie de sa marge de
2580 détour, réduisant sa perception en conséquence. Cette réduction de perception doit être notifiée à l'agent service de transport afin d'éviter que cette perception ne soit inutilement impliquée dans la recherche de candidats d'autres passagers.

Quant aux passagers, ils n'explorent les possibilités de covoiturage que si de nouveaux conducteurs sont présents dans leur perception. En effet, si un conducteur ou
2585 un couple de conducteurs a déjà été traité dans l'étape de simulation précédente et n'a donné aucun résultat, il n'est pas nécessaire de le réévaluer.

Pour les passagers, une étape supplémentaire dans cette mise à jour de la perception est essentielle : l'identification des candidats au covoiturage.

L'identification des candidats au covoiturage

L'identification du meilleur covoiturage pour un passager est un défi majeur. En
2590 effet, il n'est pas envisageable de couvrir toutes les possibilités (paires de conducteurs et nœuds de transfert) en raison de la complexité induite. Il est donc nécessaire de réduire l'espace de recherche, puis de l'explorer efficacement. Chaque objet (nœud du graphe de route, perceptions des utilisateurs...) et agent de notre système peut
2595 être localisé spatialement à l'aide de coordonnées latitudinales et longitudinales. Nous avons donc décidé d'utiliser une structure d'indexation spatiale dynamique appelée R-Arbre (*R-Arbre* en anglais). Comme détaillé dans la section 2.2.2, les R-Arbres [Guttman, 1984] sont des structures de données, sous forme d'arbres, utilisées comme méthodes d'exploration spatiale.

Dans les Figures 4.5 et 4.6, nous proposons un exemple d'application des R-Arbres
2600 au graphe de route qui nous sert d'exemple depuis le début de ce chapitre. Un exemple d'indexation est proposé dans la Figure 4.5, les nœuds du graphe sont indexés selon leur proximité topologique par des rectangles englobant de différentes tailles. La structure R-Arbre résultant de cette indexation est schématisée dans la figure 4.6.

L'intérêt d'une telle structure, dans le cadre de notre problème, est en premier
2605 lieu d'être efficace dans un contexte dynamique. L'ajout, la suppression et la modi-

R1 = Root

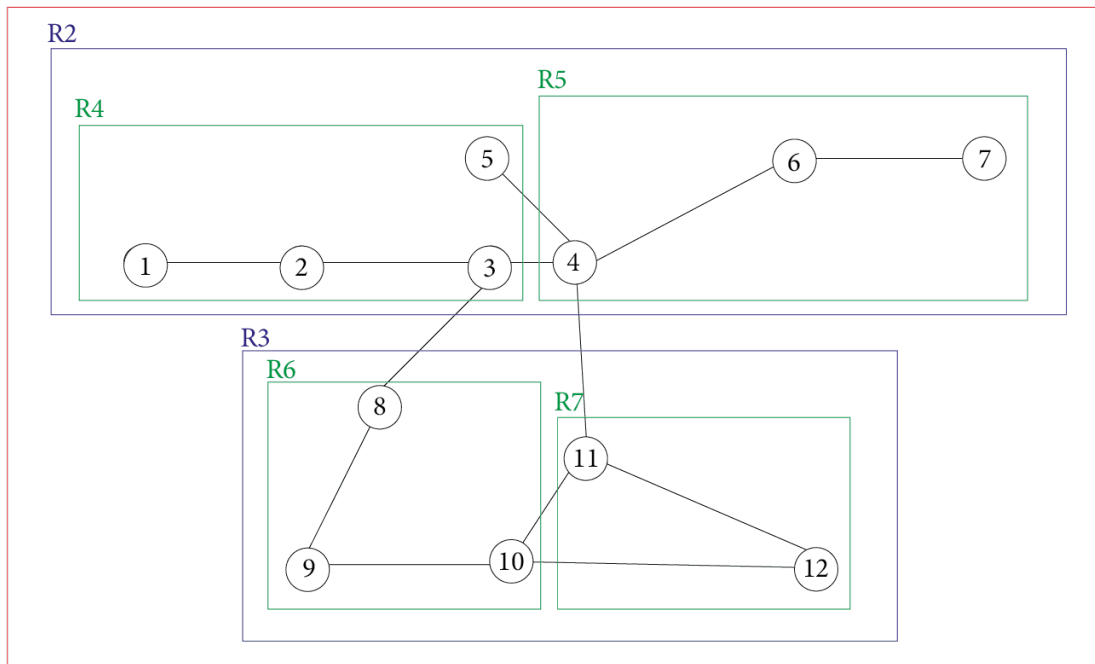


FIGURE 4.5 – Exemple d'indexation R-Arbre sur le graphe de la Figure 4.1. Les nœuds du graphe routier numérotés de 1 à 10 sont indexés par les rectangles de délimitation numérotés de R1 à R7.

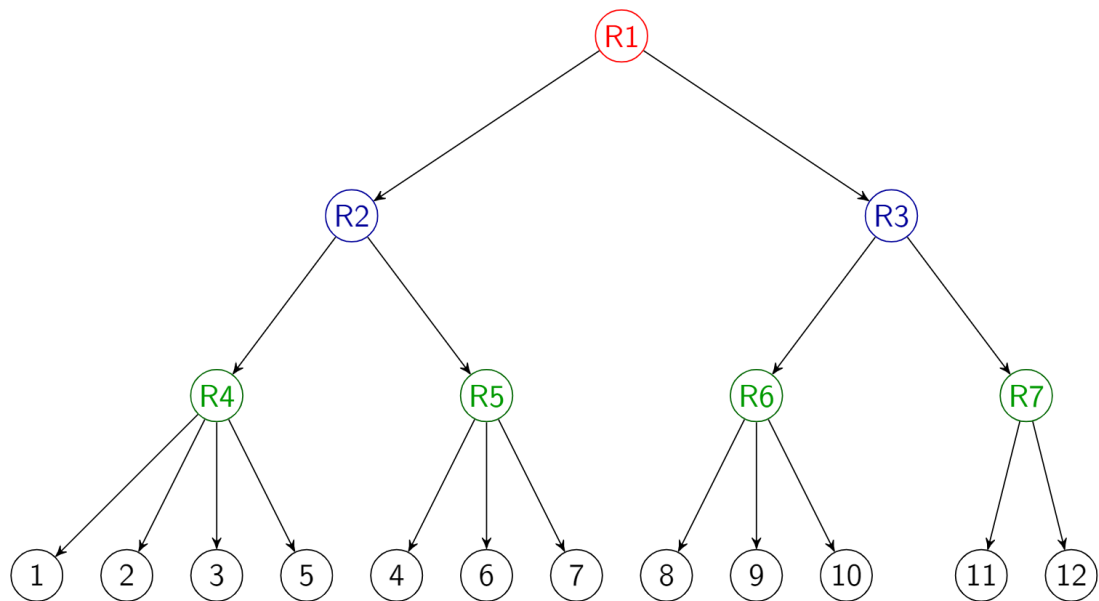


FIGURE 4.6 – Arbre résultant de l'indexation R-Arbre présentée à la Figure 4.5. Les trois premiers niveaux de l'arbre sont des nœuds correspondant aux rectangles englobants. Le dernier niveau comprend les feuilles correspondant aux nœuds du graphe de route.

fication d'informations se font en parcourant l'arbre et en modifiant la disposition
 et le nombre de pages de manière optimisée à l'aide d'heuristiques (ex. : ajout à
 la page nécessitant le moins d'être agrandi, division d'une page en deux pages plus
 2610 petites). L'autre avantage des R-Arbres est le contrôle de la complexité liée à la re-
 cherche d'entités (telles que des agents ou des nœuds du graphe de route) présentes
 dans une zone ou englobant un point, ainsi qu'aux requêtes de plus proches voisins.
 Les algorithmes descendent récursivement dans l'arbre, en écartant les pages qui ne
 concernent pas la requête (hors de la zone de recherche...) afin d'éviter des opéra-
 2615 tions inutiles. Chaque feuille de l'arbre est ensuite testée en fonction de la requête et
 retournée si les conditions sont remplies.

4.2.2 L'algorithme d'optimisation des correspondances par force brute (BFMO)

Dans ce chapitre, nous utilisons les R-Arbres afin d'indexer les perceptions des
 2620 agents conducteurs du système. Ces perceptions sont indexées dans un R-Arbre RT_p
 par l'agent service de transport t_{sa} . Cela permet aux agents passagers d'effectuer
 des requêtes spatiales à l'agent service de transport, sur des zones spécifiques de
 l'environnement afin de restreindre leur espace de recherche. Suite à ces requêtes,
 l'agent service de transport renvoie les agents conducteurs candidats au covoiturage.

2625 Nous distinguons deux phases d'identification des candidats, la première étant
 approximative et la seconde est exacte. Ces phases permettent de réduire progressi-
 vement l'espace des solutions. Ces deux phases sont représentées dans l'algorithme
 4 BFMO, pour *Brute Force Matching Optimization* ou optimisation des correspon-
 dances par force brute (recherche exhaustive). Nous décrivons une simplification de
 2630 cet algorithme dans la suite de cette section.

Dans la phase approximative, un agent passager r_i effectue trois requêtes spatiales
 (cf. section 2.2.2) à l'agent t_{sa} (BFMO, L1).

Nous définissons ces trois requêtes comme suit :

- $Q_1 = \text{SpatialPointQuery}(RT_p, r_i.v_s)$
- 2635 \hookrightarrow renvoie tous les conducteurs dont la perception chevauche le nœud de dé-
 part du passager
- $Q_2 = \text{SpatialRangeQuery}(RT_p, r_i.p)$

Algorithm 4 BFMO - Brute Force Matching Optimization

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

```
1: execute query  $Q_1, Q_2, Q_3$  to service agent
2:  $candidates\_s = Q_1 \cap Q_2$  #candidates for the starting node
3:  $candidates\_e = Q_2 \cap Q_3$  #candidates for the ending node
4:  $best\_match.arr\_time, best\_match.nb\_tsf := \infty$ 
5: for all  $d\_s \in candidates\_s$  do
6:   for all  $d\_e \in candidates\_e$  do
7:     if  $d\_s.id = d\_e.id$  then
8:       #single hop
9:       if match is feasible then
10:        compute  $m.arr\_time, m.nb\_tsf$ 
11:         $m.d = d\_s$ 
12:        if  $m.arr\_time < best\_match.arr\_time$  or  $(m.arr\_time = best\_match.arr\_time$  and
13:          $m.nb\_tsf < best\_match.nb\_tsf)$  then
14:           $best\_match = m$ 
15:        end if
16:      end if
17:    else
18:      #multi hop : find the better transfer node
19:       $\tau = r_i.p \cap d\_s.p \cap d\_e.p$  #transfer area
20:      for all  $v\_tsf \in \tau$  do
21:        if match is feasible then
22:          compute  $m.arr\_time, m.nb\_tsf$ 
23:           $m.d_1 = d\_s, m.d_2 = d\_e, m.v\_tsf = v\_tsf$ 
24:          if  $m.arr\_time < best\_match.arr\_time$  or  $(m.arr\_time = best\_match.arr\_time$  and
25:            $m.nb\_tsf < best\_match.nb\_tsf)$  then
26:             $best\_match = m$ 
27:          end if
28:        end if
29:      end for
30:    end if
31:  end for
32: end for
33: return  $best\_match$ 
```

\hookrightarrow renvoie tous les conducteurs dont la perception chevauche la perception du passager

2640 — $Q_3 = \text{SpatialPointQuery}(RT_p, r_i.v_e)$

\hookrightarrow renvoie tous les conducteurs dont la perception chevauche le nœud d'arrivée du passager

À la suite de ces requêtes, l'agent passager r_i obtient deux ensembles de candidats conducteurs. Les candidats dont l'itinéraire passe par, ou près, de son nœud de départ et partageant sa perception $candidates_s = Q_1 \cap Q_2$; et les candidats dont l'itinéraire passe par, ou près, de son nœud d'arrivée et partageant sa perception $candidates_e = Q_2 \cap Q_3$. **Ces ensembles définissent l'espace de recherche de solution de l'agent passager** (BFMO, L2-3). Pour plus de lisibilité dans l'algorithme, nous définissons l'attribut p d'un conducteur candidat comme la ou les perceptions qui sont partagées avec le passager, en excluant les autres. Ainsi, si un conducteur partage la perception du passager avec plusieurs de ses perceptions, il apparaît plusieurs fois dans les candidats.

2650

Nous précisons que le résultat de ces requêtes ne donne, à ce stade, aucune certitude sur la faisabilité temporelle des correspondances. Elles permettent d'identifier rapidement, bien qu'approximativement, des candidats pour une correspondance à un seul saut ($Q_1 \cap Q_2 \cap Q_3$), ou des candidats de départ et d'arrivée pour une correspondance à plusieurs sauts ($Q_1 \cap Q_2$ ou $Q_2 \cap Q_3$). Il est donc nécessaire, avant de considérer tout covoiturage, d'en vérifier la faisabilité vis-à-vis des contraintes exposées dans la section 4.1.2 (BFMO, L9 & L20).

Nous déclarons ensuite la variable *best_match* stockant la meilleure association atteinte jusqu'ici. Notre fonction objectif visant à minimiser l'heure d'arrivée du passager, puis le nombre de transferts, nous instancions deux attributs *best_match.arr_time* et *best_match.nb_tsf* et leur attribuons une valeur infinie. L'algorithme parcourt ensuite les deux ensembles de candidats (BFMO, L5-6) et deux cas se présentent : le covoiturage simple-saut ou le covoiturage multi-saut.

Si les conducteurs candidats sont les mêmes pour le départ et l'arrivée du passager, alors il s'agit d'un covoiturage simple-saut, car il n'y a pas de transfert nécessaire (BFMO, L7). Un test de faisabilité est alors effectué, puis les valeurs des objectifs sont calculées et stockées dans la correspondance *m* (BFMO, L9-11). Ces valeurs sont enfin comparées aux valeurs d'objectifs de la meilleure association enregistrée *best_match*. Il existe deux cas pour lesquels une nouvelle association est meilleure :

1. La nouvelle association minimise l'heure d'arrivée du passager
2. La nouvelle association égalise l'heure d'arrivée du passager, mais minimise le nombre de transferts

Si l'un des cas se présente, la meilleure association est mise à jour avec la nouvelle association (BFMO, L12-13).

Si les conducteurs candidats sont différents, alors il s'agit d'un covoiturage multi-saut. Il est alors nécessaire d'identifier le meilleur nœud de transfert pour les objectifs du passager. Nous commençons par établir la zone de transfert τ (BFMO, L18). Cette zone de transfert contient l'ensemble des nœuds du graphe de route G inclus dans les perceptions des conducteurs candidats et du passager. Elle résulte alors de l'intersection des perceptions des agents : $\tau = r_i.p \cap d_s.p \cap d_e.p$. Un exemple de ce processus est exposé dans la Figure 4.7.

Nous effectuons par la suite une recherche exhaustive de l'association impliquant le meilleur nœud de transfert. Bien que cette heuristique ne soit pas optimale, il

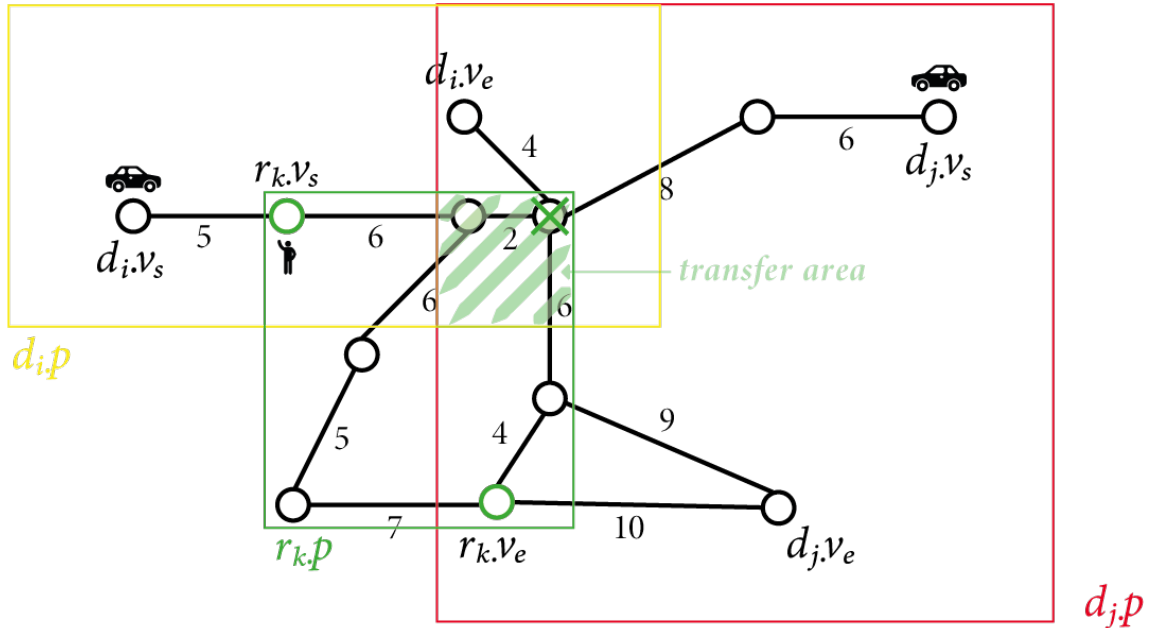


FIGURE 4.7 – Exemple de l'identification des conducteurs et nœuds de transfert candidats pour un covoiturage. La perception du conducteur d_i en jaune chevauche le nœud de départ et une partie de la perception du passager r_k en vert. La perception du second conducteur d_j en rouge chevauche le nœud d'arrivée et une partie de la perception du passager r_k . Les perceptions des agents se chevauchent toutes sur la zone hachurée de vert, alors appelée zone de transfert (*transfer area* en anglais). Deux nœuds sont présents dans cette zone de transfert et sont donc candidats au transfert du passager. Le nœud avec une croix verte représente le lieu de transfert optimal.

s'agit avant tout dans cet article de présenter un premier système de covoiturage avec détours et réservation et de se concentrer sur l'établissement d'une perception dynamique. Nous proposons une bien meilleure solution dans le chapitre 5. Les nœuds de transfert sont parcourus un par un et, après un test de faisabilité, ils sont comparés à la meilleure association *best_match* de la même manière que pour le covoiturage simple-saut (BFMO, L19-24).

Enfin, l'association optimale est retournée à la fin de l'algorithme (BFMO, L31). L'agent passager contacte le ou les conducteurs concernés par la meilleure association de covoiturage afin d'établir le contrat (la réservation) de covoiturage. En conséquence, les agents conducteurs mettent à jour les informations contenues dans leur ordonnanceur de trajet (perceptions, arrêts). Enfin, les conducteurs transmettent les modifications de leurs perceptions à l'agent service de transport. Une telle modification de la perception d'un conducteur pourrait ressembler à la figure 4.8.

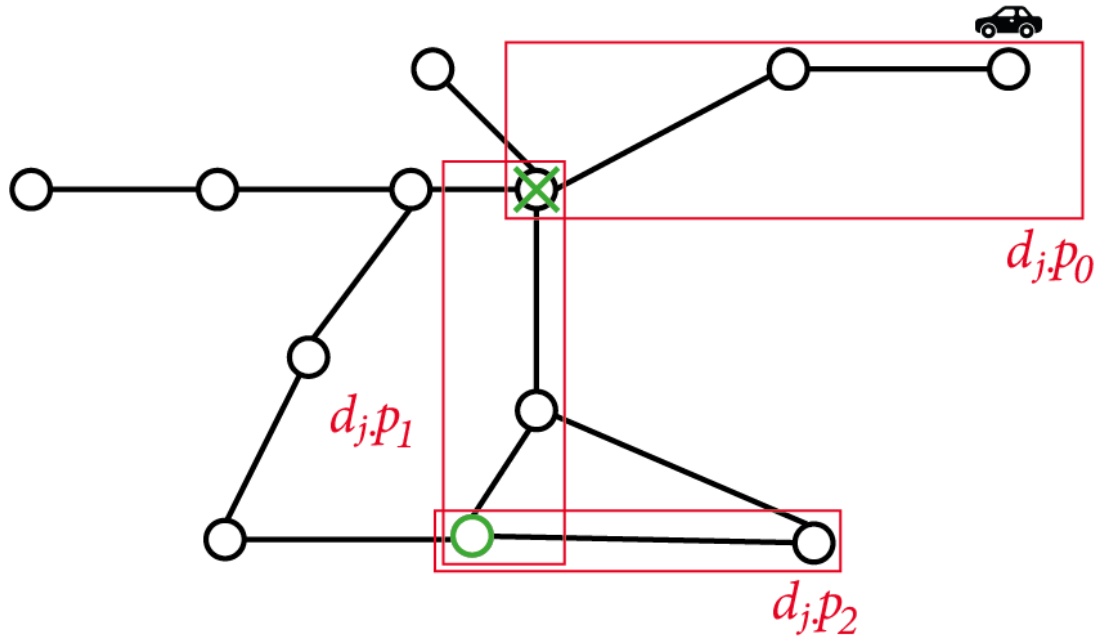


FIGURE 4.8 – Exemple de perceptions multiples après l’acceptation d’un covoiturage par un conducteur. Le conducteur d_j met à jour ses perceptions suite à un contrat de covoiturage avec un passager. Il ne dispose donc plus d’une seule perception, mais de trois, une pour chaque arrêt obligatoire. Ces perceptions étant indexées dans le R-Arbre RT_p , des passagers peuvent encore covoiturer avec le conducteur sur les nœuds présents dans les perceptions.

4.3 Expérimentations et résultats

2700 La principale difficulté dans la comparaison de stratégies de covoiturage est de dé-
finir un cadre expérimental avec peu de biais et de bruit. En effet, la performance de
résolution d’une instance est tout d’abord liée au ratio conducteurs/utilisateurs. Elle
est également liée à la taille de l’environnement et aux itinéraires initiaux des conduc-
2705 teurs et des passagers. Le problème est alors que la performance d’une stratégie par
rapport à une autre est difficile à évaluer, car elle dépend des paramètres de l’instance.
De plus, l’utilisation d’agents, qui offre l’avantage d’une approche individu-centrée,
génère une complexité importante par rapport à d’autres approches plus globali-
santes. Enfin, la recherche exhaustive du meilleur nœud de transfert de l’algorithme
4 BFMO est aussi en partie responsable de la taille contenue des instances étudiées.
2710 La complexité résultant de cette heuristique conduit à des heures de calcul avec de
nombreux passagers ou conducteurs. Nous avons donc dû écarter l’idée d’effectuer
nos expérimentations sur de grandes instances et avons préféré étudier de petites ins-
tances plusieurs fois afin d’obtenir des résultats reproductibles et fiables. De cette
manière, les biais liés aux paramètres du système sont contrôlés.

2715 4.3.1 Protocole expérimental

Nos instances sont composées d'un graphe de 150 nœuds provenant d'un réseau routier réel de San Francisco extrait d'Open Street Map et modélisé à l'aide de la bibliothèque OSMNX [Boeing, 2017]. Ce graphe est illustré dans la Figure 4.9 et présente des propriétés similaires à celles de Manhattan. Les rues sont disposées sous
2720 forme de grille avec quelques petites perturbations, à la manière de notre modèle de graphe artificiel présenté dans le chapitre 3. De nombreuses études dans le domaine du covoiturage sont menées sur Manhattan, et plus particulièrement sur cette structure en grille en raison de sa régularité [Mathieu et Nongaillard, 2018, Tafreshian et Masoud, 2020, Gerte *et al.*, 2018]. En effet, la plupart des rues sont bidirectionnelles
2725 et limitées à la même vitesse. Cela permet d'éviter les impasses et les bruits qui pourraient créer des erreurs dans la simulation.

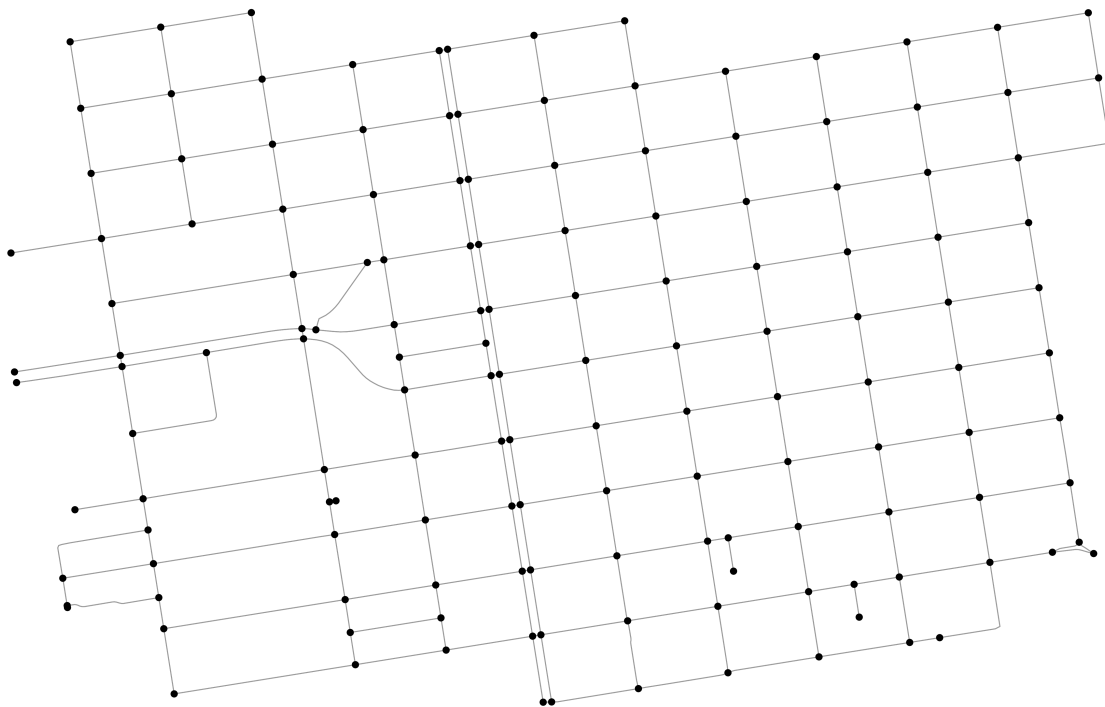


FIGURE 4.9 – Le graphe étudié : une section du côté nord de la ville de San Francisco

Sur ce réseau routier, nous faisons évoluer 10 passagers et 15 conducteurs. Les passagers ont un facteur de détour de 0,2 et un temps d'attente maximum de 3 minutes. Quant aux conducteurs, nous faisons évoluer leur facteur de détours dans nos expériences. Les nœuds de départ et d'arrivée des agents passagers et conducteurs
2730 sont tirés aléatoirement et uniformément dans chaque instance. Les stratégies des passagers sont toutefois comparées sur les mêmes instances afin d'être dans les mêmes

conditions d'offre et de demande.

Nous avons décidé d'étudier l'impact du détour des conducteurs sur le système.
2735 En effet, dans ce chapitre, nous cherchons à optimiser le temps d'arrivée des passagers
en utilisant la marge de détour des conducteurs, représentée par leurs perceptions.
Ces perceptions sont directement calculées à partir de la préférence de détour des
conducteurs. Nous faisons donc varier ce facteur de 0,1 – c'est-à-dire un détour très
faible – à 1, c'est-à-dire un détour acceptable d'au plus une fois la taille du trajet
2740 initial du conducteur. Nous proposons de comparer deux stratégies d'agent passager :

- *single-hop* : incapable d'effectuer des transferts et donc effectuant uniquement
des covoiturages simple-sauts ;
- *multi-hop* : capable d'effectuer un transfert et donc effectuant des covoiturages
simple-sauts et multi-sauts.

2745 Ainsi, l'objectif de ces expériences est de mettre en évidence l'impact des détours
et des transferts sur un système de covoiturage. Nous avons décidé d'effectuer une
simulation à génération unique de conducteurs, comme dans la section 3.4.3 du cha-
pitre 3. La simulation avec une génération unique de conducteurs permet d'étudier
les deux stratégies d'agent passager dans une situation où l'offre de covoiturage est de
2750 moins en moins importante. En effet, dans cette simulation, tous les agents conduc-
teurs sont générés à l'instanciation, de la même manière que les agents passagers.
Une fois qu'ils ont atteint leur destination, les agents conducteurs disparaissent du
système et aucun autre agent conducteur n'est généré. La simulation s'arrête lors-
qu'il n'y a plus d'agents conducteurs dans le système. Nous simulons une sorte de
2755 course contre-la-montre afin d'étudier le taux de service, c'est-à-dire la proportion de
passagers atteignant leur destination avant la fin de la simulation.

4.3.2 Résultats

Le premier effet de l'augmentation du facteur de détour du conducteur est, bien
sûr, d'augmenter la distance moyenne de trajet des conducteurs.

2760 Les conducteurs qui font des détours plus importants, pour prendre en charge des
passagers, ont tendance à parcourir plus de kilomètres, comme on peut le voir dans
la Figure 4.10. On remarque que la courbe des résultats des passagers *multi-hop* est
légèrement supérieure à celle des passagers *single-hop*.

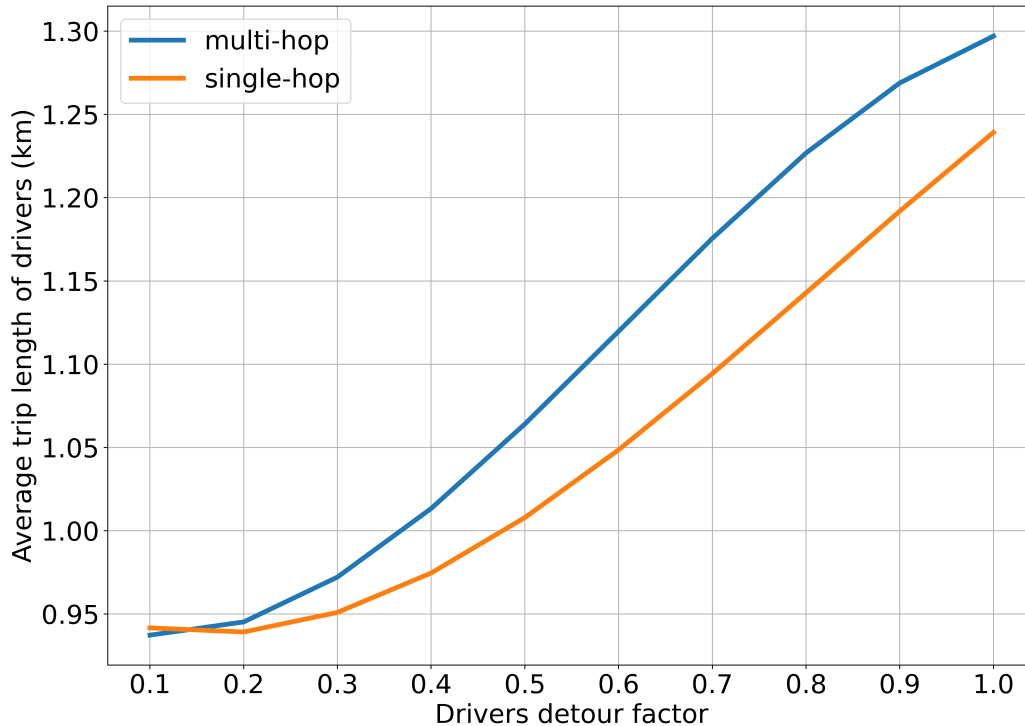


FIGURE 4.10 – Effet du facteur de détournement des conducteurs et des stratégies des passagers sur la distance moyenne de trajet des conducteurs. Les données sont moyennées sur 50 expériences et chaque colonne correspond aux mêmes 50 expériences.

Ce phénomène peut être expliqué par le fait que le covoiturage multi-saut permet d'atteindre un meilleur taux de service, comme en atteste la Figure 4.11. On peut émettre l'hypothèse que les passagers n'ayant pas réussi à rejoindre leur destination sans transferts sont éloignés de leur nœud de destination ou sont dans une zone avec peu d'offre de covoiturage. En leur permettant d'effectuer des transferts, les passagers sont capables d'établir des itinéraires plus complexes dans le but d'atteindre leur destination. En conséquence, les conducteurs responsables de l'acheminement de ces passagers effectuent des détours plus importants et augmentent leur distance de trajet.

En effet, dans cette figure, on observe que le taux de service est meilleur pour le covoiturage multi-saut à un même facteur de détournement, en particulier pour un facteur de 0,4 à 0,6. Plus de la moitié des passagers arrivent à destination à un facteur compris entre 0.4 et 0.5 pour le multi-saut, alors qu'il faut attendre un facteur de 0.5-0.6 pour le simple-saut.

De plus, on constate l'importance du facteur de détournement pour l'acheminement des passagers, notamment dans des situations de pénurie d'offre dues au faible détournement acceptable par les conducteurs. On passe progressivement de 15% de taux de service

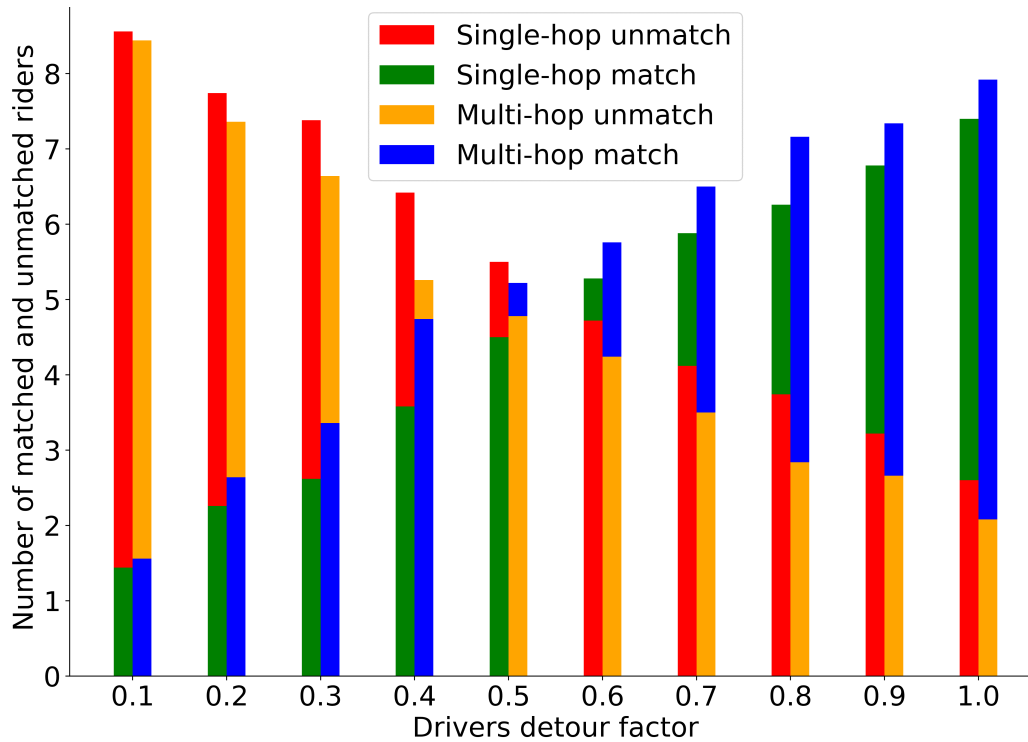


FIGURE 4.11 – Effet du facteur de détour des conducteurs et des stratégies des passagers sur le taux de service : à savoir le nombre de passagers ayant atteint leur destination face au nombre de passagers n’ayant pas réussi. Les données sont moyennées sur 50 expériences et chaque colonne correspond aux mêmes 50 expériences.

2780 pour un facteur de 0,1 à 80% pour un facteur de 1.

Bien que la distance moyenne de déplacement des conducteurs soit allongée, elle est compensée par la distance totale de trajet économisée dans le système, c’est-à-dire la différence entre la distance totale que les passagers du système auraient dû parcourir s’ils avaient fait le trajet en voiture et les somme des détours des conducteurs. En effet, nous économisons entre 800 à 1800m de trajet comme le montre la figure 4.12.

2790 Pour les deux stratégies, nous observons un seuil de rentabilité maximale atteint à un facteur de détour de 0,4 pour le multi-saut et de 0,6 pour le simple-saut, vérifiant que la stratégie multi-saut nécessite moins de détour du conducteur pour la même performance en termes d’économie de distance. En revanche, à partir d’un facteur de détour de 0,5, l’utilisation du multi-saut devient moins rentable pour cet indicateur en raison d’une intensification du détour pour permettre aux passagers d’atteindre leur destination plus rapidement et pour permettre plus de correspondances.

Enfin, le temps d’attente total des passagers varie très peu en fonction du facteur

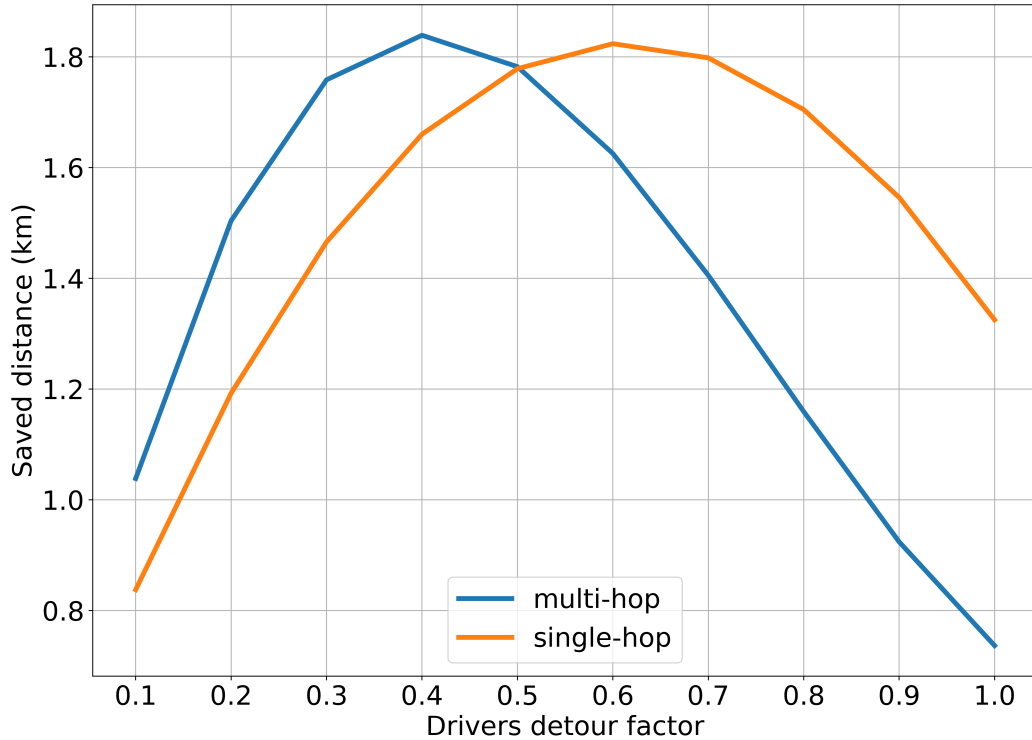


FIGURE 4.12 – Effet du facteur de détournement des conducteurs et des stratégies des passagers sur la distance totale de trajet économisée dans le système. Les données sont moyennées sur 50 expériences et chaque colonne correspond aux mêmes 50 expériences.

de détournement des conducteurs, comme on peut le voir dans la figure 4.13. Cependant, le temps d'attente est plus important pour la stratégie multi-saut, car le passager peut attendre plusieurs secondes sur le nœud de transfert, en plus de l'attente sur le nœud de départ, avant d'être pris en charge par le second conducteur. Il est également important de noter que nous ne considérons pas le temps d'attente des passagers non pris en charge. Les passagers dont la requête n'a pas été pourvue dans les temps par le simple-saut, et donc dans des situations complexes, ont pu attendre plus longtemps afin d'avoir un covoiturage multi-saut leur permettant d'atteindre leur destination.

4.4 Conclusion et perspectives

Dans ce chapitre, nous proposons une approche multi-agents pour résoudre le problème du covoiturage dynamique. Dans notre système, les passagers et les conducteurs sont représentés comme des agents autonomes et rationnels en perpétuelle interaction pour trouver le meilleur équilibre entre l'offre et la demande de covoiturage. Le but est de réduire une fonction objectif qui dépend de plusieurs critères, principalement le

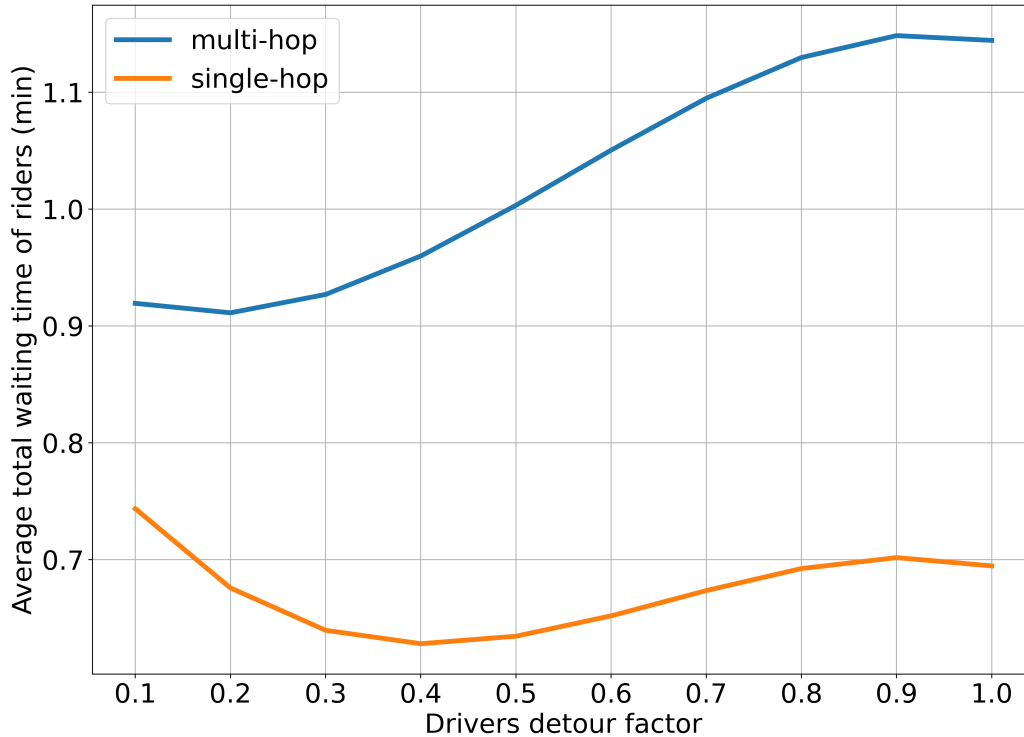


FIGURE 4.13 – Effet du facteur de détour des conducteurs et des stratégies des passagers sur le temps d’attente des passagers. Les données sont moyennées sur 50 expériences et chaque colonne correspond aux mêmes 50 expériences.

temps de trajet et le nombre de nœuds de transfert. Par rapport aux travaux existants dans la littérature traitant du même problème, notre contribution est essentiellement axée sur une modélisation dynamique, individuelle et peu coûteuse en calcul de la perception des utilisateurs. Dans la solution proposée, les agents conducteurs et passagers ont une perception modélisée dynamiquement en utilisant des R-Arbres. Nous modélisons leurs préférences en matière de détour et montrons l’impact de celles-ci sur la résolution d’une instance de covoiturage dynamique. Nous comparons deux stratégies, le covoiturage simple-saut et multi-saut. Nous montrons l’efficacité de la possibilité de transférer des passagers entre plusieurs conducteurs. Notamment sur le facteur de détour minimum nécessaire pour acheminer le plus de passagers possible et la distance totale parcourue. Nous avons réalisé un article et une présentation de ces travaux à l’occasion de la conférence IEEE SMC¹ (*IEEE International Conference on Systems, Man, and Cybernetics*) de 2021 : [Fevre et al., 2021].

Ces travaux ont toutefois plusieurs limitations. La complexité associée à la recherche exhaustive des différents nœuds de transfert possibles pour un partage de

1. <https://www.ieeesmc.org/>

trajet est beaucoup trop importante. C'est pourquoi nous proposons un nouvel algorithme, dans le prochain chapitre 5, afin d'accélérer ce processus. Autre limitation de ces précédents travaux, il s'agissait uniquement d'optimiser le trajet des passagers. Or, dans le monde réel, un conducteur est autant un utilisateur du service de covoiturage qu'un passager. Il est alors nécessaire d'impliquer son trajet dans l'optimisation du covoiturage, notamment en minimisant l'inconvénient lié à ses détours. Cela permettrait, d'une part, de conserver l'attractivité du service de covoiturage auprès des conducteurs et donc de disposer d'une offre importante. D'autre part, en minimisant les détours, on préserve le patrimoine de détour des conducteurs du système et donc l'offre de transport. Nous proposons une telle optimisation dans nos travaux présentés dans le prochain chapitre 5.

Chapitre 5

2835 Le covoiturage dynamique multi-saut avec gestion optimisée des nœuds de transfert

Ce chapitre s’articule autour des limitations identifiées à la suite de nos travaux présentés dans le chapitre 4. Nous proposons une évolution de notre algorithme 4
2840 BFMO, notamment sur l’identification du meilleur nœud de transfert à l’occasion d’un covoiturage multi-saut. Nous utilisons une nouvelle fois la puissance des R-Arbres afin de guider la recherche de ce nœud de façon optimale.

De plus, nous développons les préférences de nos agents passagers en détaillant une fonction multi-objectif pondérée par leurs préférences permettant de simuler un
2845 large éventail de comportements réalistes. Les conducteurs sont cette fois considérés comme des utilisateurs à part entière du système de covoiturage et nous proposons des techniques de minimisation de leur distance de détour.

Enfin, nous présentons divers scénarios de simulation en faisant varier les préférences des utilisateurs et les paramètres des instances. Nos expériences montrent
2850 que notre modèle peut simuler efficacement des comportements du monde réel et est capable d’optimiser le covoiturage au cas par cas.

5.1 Formulation du problème

Alors que la majorité de la littérature sur le covoiturage traite de l'optimalité globale du système, nous nous intéressons à une optimisation préférentielle pour chaque individu du système. Dans le chapitre précédent, les préférences d'un passager étaient essentiellement orientées autour du temps de trajet. Nous proposons, dans ce chapitre, de permettre à un passager d'exprimer ses préférences sur plusieurs objectifs tels que le temps de trajet, le temps d'attente, le nombre de transferts ou encore la distance additionnelle de détour.

5.1.1 Spécification du réseau routier et des agents du système

Dans l'ensemble, le modèle du réseau routier et des agents utilisateurs sont similaires à celui du chapitre 4. Ainsi, nous ne redétaillons pas chacun de ces éléments, leur descriptif étant disponible dans la section 4.1.1. Nous rappelons toutefois brièvement cette description pour plus de compréhension et nous concentrons sur les évolutions de notre modèle.

Le réseau routier

Le réseau routier est modélisé par un graphe orienté et connexe $G = \langle V, E \rangle$ dans lequel V est l'ensemble des nœuds du graphe et E est l'ensemble des arêtes liant ces nœuds. Les arêtes représentent les routes et comportent différents attributs tels que la distance, la vitesse maximale autorisée et le temps de parcours entre deux nœuds. Un nœud représente une intersection de routes et est référencé spatialement avec un couple de coordonnées.

Les agents utilisateurs

Un agent utilisateur du système $u_i = \langle v_s, v_l, v_e, det_{max} \rangle$, avec $u_i \in U$ représentant un agent passager ou conducteur dans le système de covoiturage. Il est initialisé par quatre éléments : le nœud de départ v_s , le nœud de localisation actuel v_l , le nœud d'arrivée v_e et le facteur de détour det_{max} . Le facteur de détour quantifie la volonté de détour d'un agent utilisateur. L'ensemble des agents utilisateurs est noté

$U = \{u_1, u_2, \dots, u_n\}$ et cet ensemble représente la totalité des agents conducteurs D
2880 et passagers R du système : $U = D \cup R$.

Les agents conducteurs

Un agent conducteur $d_i = \langle v_s, v_l, v_e, det_{max}, c_{max} \rangle$, avec $d_i \in D$, est initialisé par cinq éléments. Un conducteur étant un utilisateur, il hérite des quatre premiers paramètres définis précédemment. Le dernier paramètre c_{max} définit le nombre maximal
2885 de sièges disponibles dans le véhicule du conducteur. Un conducteur a pour objectif d'atteindre sa destination avant son heure d'arrivée la plus tardive. Il suit donc son trajet de nœud en nœud, en prenant systématiquement le chemin le plus court. S'il doit faire un détour pour prendre et déposer un passager, il doit être capable de déterminer si cette action est possible vis-à-vis de son heure d'arrivée la plus tardive.
2890 Pour ce faire, cet agent maintient un ordonnanceur de voyage où il stocke différentes informations telles que ses prochains arrêts (nœuds du graphe de route), les temps de trajets minimum, marge de détours ou capacité restante (nombre de sièges disponibles) entre ses arrêts.

Les agents passagers

Un agent passager $r_i = \langle v_s, v_l, v_e, det_{max}, wt_{max}, pref \rangle$, avec $r_i \in R$, est cette fois
2895 initialisé par six éléments. Un passager étant un utilisateur du système, il hérite des quatre premiers paramètres définis précédemment, auxquels on ajoute le temps d'attente maximum de trajet wt_{max} . La principale évolution de l'agent passager est la capacité d'exercer ses préférences de trajet via le nouveau paramètre $pref$. Ce paramètre
2900 représente le vecteur de préférence du passager, c'est-à-dire son profil/comportement. Chaque élément $pref_i \in [0, \alpha]$ et la somme des éléments de $pref$ doit être égale à α : $sum(pref_i, \dots, pref_k) = \alpha$. On peut alors, par exemple, fixer α à 10 et faire varier la pondération des préférences d'un agent passager entre 0 et 10. De cette manière, nous
2905 pouvons attribuer un poids à chaque objectif contenu dans cette liste de préférences et détailler différents profils de passagers. Ceci implique que, si $pref_i > pref_j$, alors le passager accorde plus d'importance à la préférence $pref_i$ qu'à la préférence $pref_j$ lors de l'optimisation de son trajet. Un passager a donc cette fois pour objectif d'atteindre sa destination avant son heure d'arrivée la plus tardive tout en sélectionnant le ou les meilleurs covoiturages en fonction de ses préférences.

2910 L'ensemble des interactions des agents du système est résumé dans la Figure 5.1.
Comme on peut le voir, le comportement d'un agent passager est basé sur deux al-
gorithmes : PMO (section 5.2.1) et CIER (section 5.2.2). L'algorithme PMO gère
le processus d'optimisation en maintenant les meilleures associations et en appe-
lant l'algorithme CIER, lui-même responsable de l'identification du meilleur nœud
2915 de transfert dans le contexte d'un covoiturage multi-saut.

5.1.2 Contraintes et Formulations

Comme dans le chapitre 4, et pour les mêmes raisons, nous limitons notre étude au
covoiturage à un et deux sauts, c'est-à-dire au covoiturage limité à deux conducteurs
pour un transfert.

2920 Les contraintes sont identiques à celles présentées dans la section 4.1.2 à savoir :

- **Contrainte temporelle** : Respect de l'heure limite d'arrivée des utilisateurs,
respect du temps d'attente du passager ;
- **Contrainte de capacité** : Respect de la capacité maximale d'un véhicule ;
- **Contrainte de l'ordre des arrêts** : Respect de l'ordre des arrêts à l'occasion
2925 d'un covoiturage (nœud de prise en charge, puis nœud de dépôt).

La définition d'une association évolue pour prendre en compte quatre objectifs.
Une association ("*a match*" en anglais) $m \in r_i.M$ est défini par plusieurs variables :
un ou deux conducteurs $d_j, d_k \in D$ (simple-saut : $j = k$, multi-saut : $j \neq k$), un
nœud de transfert v_tsf le cas échéant, un nombre de transferts nb_tsf (0 ou 1 dans
2930 cette étude), un temps d'attente total $total_wt$, un temps d'arrivée arr_time , et une
distance additionnelle de détour add_dist . Chaque passager a dorénavant pour but
de choisir l'association la plus optimale dans M selon ses préférences $pref$ présentées
précédemment dans la section 5.1.1.

Le problème étudié est donc multi-objectif en impliquant la minimisation simul-
2935 tanée de quatre critères :

- $total_wt = start_wt + tsf_wt$: le temps d'attente total du passager corres-
pond à la somme de son temps d'attente $start_wt$ au nœud d'origine et au
nœud de transfert tsf_wt . Cette valeur doit être inférieure au temps d'attente
maximal, c'est-à-dire que pour un agent passager r_i : $r_i.total_wt \leq r_i.wt_{max}$.
- $arr_time = t_s + total_wt + travel_time$: l'heure d'arrivée de l'agent passager.
2940

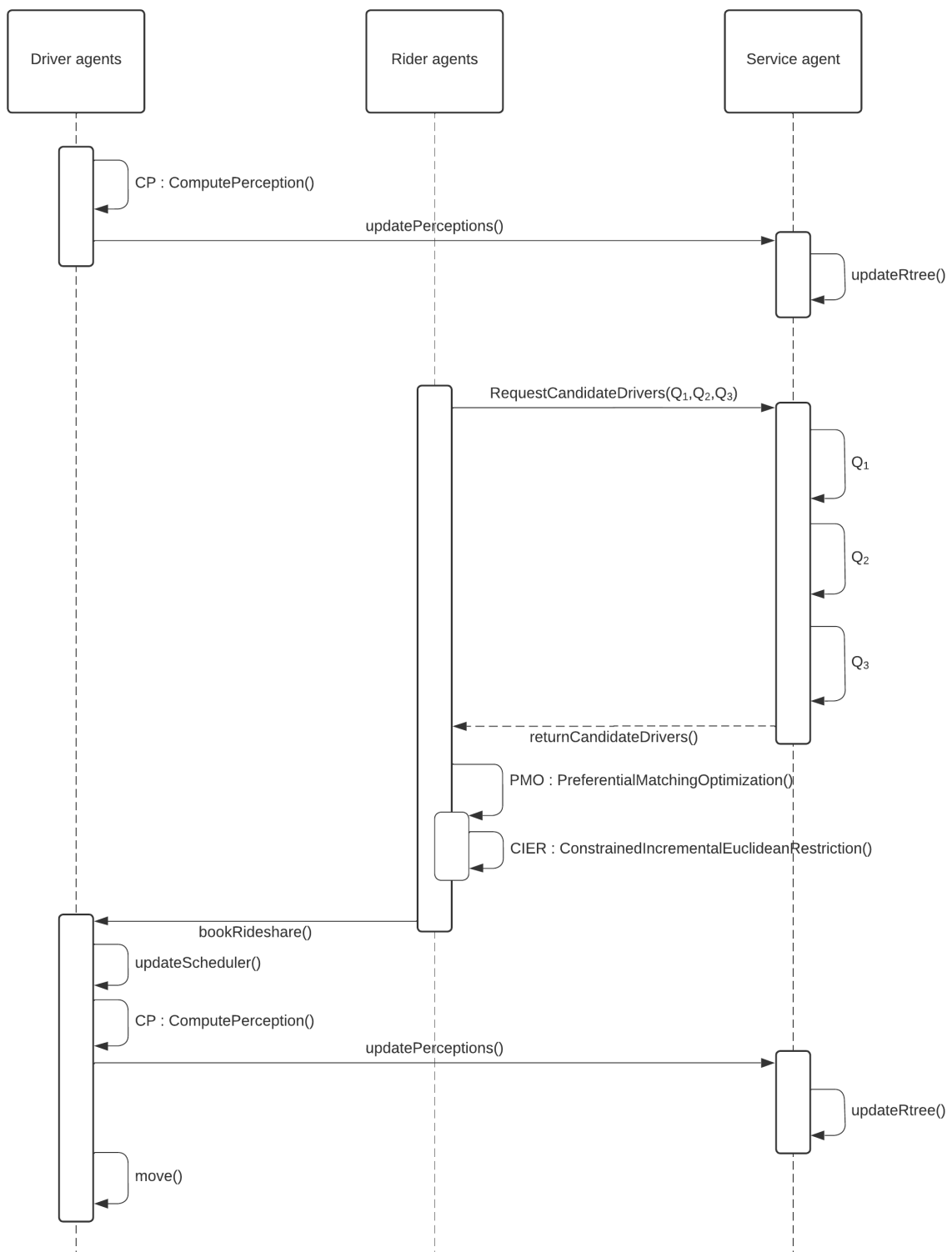


FIGURE 5.1 – Le diagramme séquentiel des agents du système. Les agents sont représentés par des lignes de vie. Les différentes actions et communications entre agents sont ordonnées chronologiquement de haut en bas. Les phases d'activation des agents sont représentées par les plus petits rectangles et les flèches représentent les appels de fonction et les communications interagents.

Il correspond à la somme de son temps d'apparition t_s , de son temps d'attente $total_wt$ et de son temps de trajet $travel_time$.

- $add_dist = \sum_{i=1}^{nb_d} (d_i.ridesharing_dist - d_i.initial_dist)$: la somme des distances additionnelles de détour des conducteurs impliqués dans le covoiturage du passager. La somme est effectuée pour le nombre de conducteurs nb_d impliqué dans le covoiturage. Une distance additionnelle de détour est calculée en effectuant la différence entre la distance de trajet du conducteur en covoiturant $ridesharing_dist$, et donc avec les éventuels détours, et sa distance de trajet initiale $initial_dist$. On ne considère alors pas tout le trajet du conducteur (car il pourrait y avoir d'autre covoiturage par la suite), mais uniquement le tronçon de covoiturage avec le passager.
- $nb_tsf = nb_d - 1$: le nombre total de transferts d'un agent passager pour atteindre sa destination. Il correspond au nombre de conducteurs impliqués dans le covoiturage nb_d , soustrait de 1. En effet, un covoiturage avec 1 seul conducteur n'implique pas de transfert (0) alors qu'un covoiturage avec 2 conducteurs implique 1 transfert.

5.2 Le système multi-agents proposé

Dans ce chapitre, nous utilisons les R-Arbres à deux fins. D'une part, les perceptions des agents conducteurs du système sont indexées dans un R-Arbre RT_p par l'agent service de transport tsa . Cela permet aux agents passagers d'effectuer des requêtes spatiales à l'agent service de transport, sur des zones spécifiques de l'environnement, et ainsi de restreindre leur espace de recherche. Suite à ces requêtes, l'agent service de transport renvoie les agents conducteurs candidats au covoiturage. D'autre part, si le covoiturage à évaluer est multi-saut, nous indexons les différents nœuds de transfert possibles dans un autre R-Arbre RT_t afin de pouvoir transformer le problème en la recherche du plus proche voisin d'un groupe (*GNN/ANN : Group/Aggregate nearest neighbour* en anglais, cf. section 2.4.6). Ces opérations sont décrites dans l'algorithme 5, que nous nommons PMO pour *Preferential Matching Optimization*, et l'algorithme 6, que nous nommons CIER pour *Constrained Incremental Euclidean Restriction*.

5.2.1 L'algorithme d'optimisation préférentielle du covoiturage (PMO)

L'algorithme 5 PMO est une évolution de l'algorithme 4 BFMO présenté en section 4.2.2.

Algorithm 5 PMO - Preferential Matching Optimization

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

```

1: execute query  $Q_1, Q_2, Q_3$  to service agent
2:  $candidates\_s = Q_1 \cap Q_2$  #candidates for the starting node
3:  $candidates\_e = Q_2 \cap Q_3$  #candidates for the ending node
4:  $best\_add\_dist, best\_wt, best\_arrival\_time, best\_nb\_tsf := \infty,$ 
5:  $best\_matches := list() \#r_i.M$ 
6: for all  $d\_s \in candidates\_s$  do
7:   for all  $d\_e \in candidates\_e$  do
8:     if  $d\_s.id = d\_e.id$  and match is feasible then
9:       #single hop
10:      compute  $m.add\_dist = d\_s.ridesharing\_dist - d\_s.initial\_dist$ 
11:      compute  $m.total\_wt, m.arr\_time, m.nb\_tsf$ 
12:       $m.d = d\_s$ 
13:     else
14:       #multi hop
15:        $m = CIER()$ 
16:     end if
17:     if  $m$  minimizes all current best values of the objectives (pareto optimal) then
18:       overwrite  $best\_matches := set(m)$  #the only element of the set is now  $m$ 
19:       update all objectives :  $best\_add\_dist, best\_wt, best\_arrival\_time, best\_nb\_tsf$ 
20:     else if  $m$  minimizes at least one objective (pareto dominant) then
21:       add  $m$  to  $best\_matches$ 
22:       update concerned objectives
23:     end if
24:   end for
25: end for
26:  $best\_match =$  best match among  $best\_matches$  according rider preferences
27: return  $best\_match$ 

```

2975 Pour établir son espace de recherche, un agent passager r_i effectue trois requêtes spatiales (cf. section 2.2.2) à l'agent tsa : une requête Q_2 de chevauchement d'une zone dans l'espace avec sa perception et deux requêtes de chevauchement d'un point dans l'espace : Q_1 sur son origine et Q_3 sur sa destination (BFMO, L1).

2980 À la suite de ces requêtes, l'agent passager r_i obtient deux ensembles de candidats conducteurs. Les candidats dont l'itinéraire passe par, ou près, de son nœud de départ et partageant sa perception $candidates_s = Q_1 \cap Q_2$ et les candidats dont l'itinéraire passe par, ou près, de son nœud d'arrivée et partageant sa perception $candidates_e = Q_2 \cap Q_3$ (PMO, L2-3). Ce processus est exposé dans la Figure 4.7.

2985 Une fois l'espace de recherche obtenu, l'agent passager doit être capable d'identifier rapidement et efficacement le meilleur covoiturage. Il existe deux possibilités : un covoiturage simple-saut (s-hop), c'est-à-dire un seul conducteur acheminant le passa-

ger de son origine à sa destination, et un covoiturage multi-saut (m-hop), c'est-à-dire plusieurs conducteurs (deux dans notre étude) acheminant le passager à sa destination via un transfert. L'agent passager parcourt les deux ensembles de candidats et agit différemment selon le type de covoiturage (PMO, L6-7).

Pour qu'un covoiturage simple-saut soit considéré, un agent conducteur doit appartenir à la fois à l'ensemble *candidats_s* et à l'ensemble *candidats_e* car il doit être candidat au départ et à l'arrivée du passager. Si cette condition est satisfaite et que les contraintes spatio-temporelles définies dans la section 4.1.2 ne sont pas violées, nous calculons les valeurs des objectifs et générons la solution de covoiturage m (PMO, L8-12).

En revanche, si les conducteurs sont différents, alors il s'agit d'un covoiturage multi-saut. Comme précisé dans la section 2.1.5, le problème du covoiturage multi-saut est beaucoup plus complexe que celui du simple-saut. En effet, la recherche du nœud de transfert optimal génère de nombreux calculs supplémentaires. Une évaluation de l'ensemble des nœuds de transfert possibles est trop coûteuse computationnellement, comme nous l'avons expérimenté dans le chapitre 4. C'est pourquoi nous proposons une approche différente, présentée dans l'algorithme 6 nommé CIER (PMO, L15).

5.2.2 L'algorithme de restriction euclidienne incrémentale sous contraintes (CIER)

Basé sur l'approche R-Arbres, l'algorithme CIER que nous proposons est une extension de l'algorithme de référence dans la littérature IER [Papadias *et al.*, 2003] pour résoudre le problème GNN/ANN (cf. section 2.4.6).

Le problème du plus proche voisin d'un groupe ou d'un agrégat (GNN/ANN) et l'identification du nœud de transfert optimal

Des travaux antérieurs ont mis en évidence le parallèle entre les réseaux routiers sous forme de graphe [Chen *et al.*, 2020, Yiu *et al.*, 2005], le problème du covoiturage dynamique multi-saut [Xu *et al.*, 2020] et le problème du plus proche voisin d'un groupe ou d'un agrégat (d'abord appelé GNN puis ANN) [Papadias *et al.*, 2004]. Si l'on considère T comme l'ensemble des nœuds de transfert disponibles et N comme l'ensemble des nœuds de départ et d'arrivée des utilisateurs, le problème ANN (*ag-*

Algorithm 6 CIER - Constrained Incremental Euclidean Restriction

Input : G : road infrastructure graph, r_i : the rider, d_s : the origin driver, d_e : the destination driver, $best_add_dist$: optimal additional distance

Output : $best_match$: multi-hop match of d_s and d_e with transfer node optimality

- 1: $\tau = r_i.p \cap d_s.p \cap d_e.p$ #transfer area
- 2: $RT_\tau :=$ spatial indexing of τ in a rtree
- 3: $N := \{r.v_s, r.v_e, d_s.dest, d_e.origin\}$
- 4: $H :=$ new priority queue
- 5: $best_match := None$
- 6: compute $non_gnn_dist, init_dist$
- 7: **for all** object o in $RT_\tau.Root$ **do**
- 8: compute $add_dist^\mathcal{E} = dist_{agg}^\mathcal{E}(o, N) + non_gnn_dist - init_dist$ #use of euclidean distances
- 9: enqueue ($H, o, add_dist^\mathcal{E}$)
- 10: **end for**
- 11: **while** H is not empty **do**
- 12: $o :=$ dequeue(H)
- 13: **if** ($add_dist^\mathcal{E} \geq best_add_dist$) **then**
- 14: return $best_match$
- 15: **end if**
- 16: **if** o is a leaf (a transfer node) **then**
- 17: compute $add_dist = dist_{agg}(o, N) + non_gnn_dist - init_dist$ #use of real distances
- 18: **if** $add_dist < best_add_dist$ **then**
- 19: **if** match is feasible according schedule inserting constraint **then**
- 20: compute $best_match.wt, best_match.arr_time$ and $best_match.nb_tsf$
- 21: $best_match.v_tsf = o, best_match.add_dist = add_dist, best_match.d_1 = d_s, best_match.d_2 =$
 d_e
- 22: $best_add_dist = add_dist$
- 23: **end if**
- 24: **end if**
- 25: **else**
- 26: **for** $child$ in o **do**
- 27: compute $add_dist^\mathcal{E} = dist_{agg}^\mathcal{E}(child, N) + non_gnn_dist - init_dist$
- 28: enqueue ($H, child, add_dist^\mathcal{E}$)
- 29: **end for**
- 30: **end if**
- 31: **end while**
- 32: **return** $best_match$

gregate nearest neighbor) peut être formulé comme suit : une requête ANN permet de retrouver le ou les nœuds de T minimisant la somme des distances par rapport à tous les nœuds de N . En effet, lorsque l'on recherche le nœud de transfert optimal, on cherche le nœud qui minimise la somme des distances de détour des conducteurs associés au covoiturage.

En considérant au maximum un transfert, un transfert est possible s'il y a au moins un nœud de transfert présent dans la perception de l'agent passager r_i et dans la perception d'un conducteur dans $candidates_s$ et dans $candidates_e$. Nous rappelons que cette zone de rencontre des perceptions des trois agents est nommée la zone de transfert τ (voir Figure 4.7) et est calculée comme suit : $\tau = r_i.p \cap d_s.p \cap d_e.p$ avec $\tau \subseteq V$, $d_s \in candidates_s$ et $d_e \in candidates_e$ (CIER, L1). Les nœuds inclus dans τ sont ensuite indexés dans un nouveau R-Arbre RT_τ afin de permettre à nouveau des opérations rapides sur ces données (CIER, L2).

3030 Concernant l'ensemble N , six nœuds sont candidats : l'origine et la destination de
 chacun des participants au covoiturage, r_i , d_s et d_e . Nous avons toutefois fait le
 choix de ne considérer que quatre de ces nœuds. En effet, deux distances de trajet
 ne sont pas impactées par la position du nœud de transfert : celle entre l'origine du
 premier conducteur d_s et l'origine du passager r_i , et celle entre la destination du
 3035 passager r_i et la destination du second conducteur d_e . Le nœud d'origine du conduc-
 teur candidat au départ et le nœud de destination du conducteur candidat à l'arrivée
 n'ont donc pas à être considérés dans le problème. Pire encore, ils apporteraient une
 sorte de "bruit" lors de la minimisation des distances et fausseraient le résultat. En
 revanche, pour les autres nœuds, la distance de chaque segment dépend de la position
 3040 du nœud de transfert. Ainsi, $N = \{r.v_s, r.v_e, d_s.dest, d_e.origin\}$, avec *origin* et
dest correspondant respectivement au nœud d'origine et de destination des conduc-
 teurs pour la session de covoiturage (CIER, L3). En effet, ces nœuds ne sont pas
 nécessairement les nœuds de départ v_s et d'arrivée v_e des conducteurs, ils peuvent
 être des nœuds intermédiaires de leur ordonnanceur de voyage (liste *stops*).

3045 Une fois cette notion de dépendance à la position du nœud de transfert mise
 en évidence, nous pouvons définir trois types de distances constituant la distance
 additionnelle de détour d'un covoiturage multi-saut :

- les distances variables de détour ***gnn_dist*** : distances variant en fonction de
 la sélection du nœud de transfert ;
- 3050 — les distances fixes initiales ***init_dist*** : distances précalculées initialement pré-
 vues par les conducteurs pour rejoindre leur destination sans détours ;
- les distances fixes de détour ***non_gnn_dist*** : distances précalculées liées au
 détour des conducteurs, mais indépendantes du nœud de transfert.

La distance additionnelle correspond à l'écart entre la distance initiale de trajet
 3055 des conducteurs, c'est-à-dire la distance sans détours, et la nouvelle distance de tra-
 jet. La distance additionnelle de détour à minimiser est donc redéfinie comme suit :
 $add_dist = gnn_dist + non_gnn_dist - init_dist$.

Extension de l'algorithme IER

Il existe de nombreuses approches pour résoudre le problème ANN de manière
 3060 puissante. L'une des plus populaires est l'algorithme IER (*Incremental Euclidean
 Reduction*) [Yiu *et al.*, 2005]. Il s'agit d'une heuristique de type *Best-First* (le meilleur

en premier) basée sur une file d'attente prioritaire H et une indexation des nœuds de transfert dans un R-Arbre. L'algorithme parcourt les différents nœuds de l'arbre en priorisant les éléments les plus proches afin que la première feuille identifiée soit la meilleure. Dans ce chapitre, nous proposons d'étendre cet algorithme en conservant la meilleure distance additionnelle add_dist trouvée (la plus petite), à la fois pour le covoiturage à saut unique et à saut multiple. Nous nommons donc cette variable $best_add_dist$ et elle est utilisée comme entrée de l'algorithme 6 CIER.

Dans la solution que nous proposons, nous privilégions l'utilisation des distances euclidiennes afin de contenir la complexité. En effet, le calcul d'une distance euclidienne entre deux nœuds se fait en $O(1)$ alors que le calcul d'une distance réelle se fait en résolvant le plus court chemin entre ces deux nœuds et est de l'ordre de $O(E + V \log(V))$. Dans l'algorithme CIER que nous proposons (algorithme 6), nous utilisons principalement les distances euclidiennes pour gérer l'ordre des éléments de la file d'attente prioritaire (CIER, L8-9 & L27-28).

Ainsi, au début de l'algorithme 6, nous évaluons la distance euclidienne agrégée $dist_{agg}^E$ des différents enfants de la racine du R-Arbre $RT_τ$. Cette distance est calculée à l'aide du MINDIST [Adler et Heeringa, 2008]. Le principe du MINDIST est de calculer la distance euclidienne minimale entre un nœud de $N = \{r.v_s, r.v_e, d_s.dest, d_e.orig\}$ et un rectangle englobant du R-Arbre en utilisant les coordonnées longitudinales et latitudinales comme on peut le voir dans la figure 5.2.

On agrège donc ces distances minimales pour chaque enfant o de la racine du R-Arbre $RT_τ$ (CIER, L6-8). Le total de ces distances constitue la gnn_dist du calcul final de la distance additionnelle. On additionne donc les non_gnn_dist et on soustrait les $init_dist$ précalculées pour obtenir la distance euclidienne additionnelle add_dist^E (CIER, L8). Chaque enfant est ensuite enfilé dans H et ordonné de façon croissante en fonction de sa distance additionnelle euclidienne (CIER, L9). De cette façon, le rectangle englobant le moins éloigné des nœuds de N sera étudié en premier.

Une fois cette phase d'étude de la racine de l'arbre effectuée, nous rentrons dans le cœur de l'algorithme. Tant que la file H n'est pas vide, l'objet (le rectangle englobant du R-Arbre ou le nœud de transfert) ayant la valeur la plus petite est extrait de la file.

Un premier test est alors effectué quant à la distance euclidienne additionnelle de l'objet extrait de la file. Si cette distance est supérieure à la meilleure distance

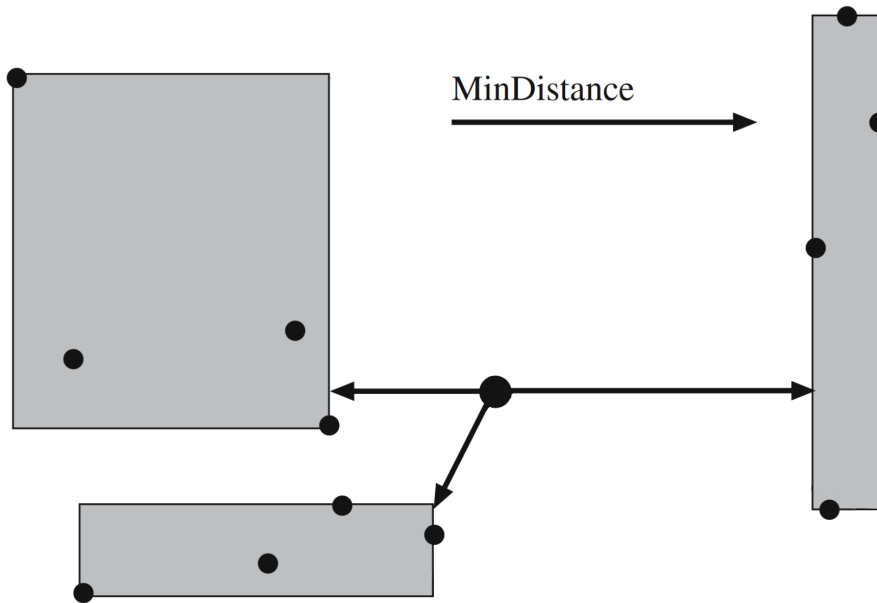


FIGURE 5.2 – Exemple d’un MINDIST : un calcul des distances euclidiennes minimales entre un nœud et les rectangles englobants d’un R-Arbre indexant d’autres nœuds.

3095 additionnelle réelle enregistrée jusqu’ici pour le pas de simulation et l’agent passager, à savoir *best_add_dist*, la boucle est interrompue. En effet, cela veut dire qu’à partir d’ici, on ne trouvera plus de solution améliorante.

S’il est encore possible d’améliorer la solution courante, deux cas se présentent :

- Si l’objet extrait de la file est une feuille du R-Arbre RT_τ , et donc un nœud de transfert, alors nous vérifions si la solution est meilleure en calculant et en considérant la distance additionnelle réelle *add_dist*. Si elle est meilleure et ne viole pas les contraintes de la section 4.1.2, nous mettons à jour le *best_match* (CIER, L16-23).
- Si l’objet extrait de la file est un nœud du R-Arbre RT_τ , et donc un rectangle englobant, nous effectuons le MINDIST sur ce dernier et l’insérons dans H (CIER, L26-29).

Par conséquent, pour être considérée comme la meilleure association, la distance additionnelle de détour d’un candidat au covoiturage *add_dist* (L17) doit être inférieure à la meilleure distance additionnelle enregistrée *best_add_dist*. L’avantage d’une telle approche est qu’elle permet de réduire considérablement l’espace de recherche en limitant les calculs et l’évaluation des candidats non optimaux. Cependant, le risque de tomber sur un optimum local est considérable. En optimisant uniquement la distance de détour supplémentaire, on risque de négliger les objectifs liés aux préférences des passagers. Cette méthode tend à favoriser les agents conducteurs du

3115 système en minimisant leurs détours. Elle tend aussi à favoriser le bien commun, en
effet, en minimisant les détours, on préserve le patrimoine de détour des conducteurs
du système et donc l'offre de transport.

5.2.3 L'aspect multi-objectif de la solution proposée

Précédemment, nous avons souligné que l'algorithme CIER favorise la performance
3120 de recherche des conducteurs candidats au covoiturage et la minimisation de la dis-
tance de détour des agents conducteurs par rapport aux préférences des passagers.
Pour répondre à ce constat et diversifier l'offre de covoiturage, la solution que nous
proposons est basée sur le mécanisme de dominance de Pareto [Pareto, 1964]. L'idée
est de maintenir une liste des meilleures solutions de covoiturage *best_matches*. Une
3125 meilleure solution de covoiturage est une solution minimisant au moins un des ob-
jectifs présentés dans la section 5.1.2 (temps d'attente, heure d'arrivée, distance ad-
ditionnelle et nombre de transferts). Elle est, dans ce cas, Pareto dominante et est
ajoutée à la liste *best_matches*. De cette façon, nous préservons des solutions opti-
males pour chaque objectif et le passager peut, selon ses préférences, choisir celle qui
3130 lui convient le mieux. Si une solution de covoiturage optimise tous les objectifs, alors
elle est Pareto optimale, la liste est alors vidée et cette solution lui est ajoutée. Ce
processus est détaillé dans les lignes 17 à 22 de l'algorithme 5 (PMO, L17-22).

En plus de cette liste des meilleures solutions, nous maintenons deux autres listes :
la liste des pires valeurs *worst_goals* et des meilleures valeurs *best_goals* atteintes
3135 pour chaque objectif. Ces listes permettent à un agent passager d'évaluer la perfor-
mance d'une solution. En effet, en effectuant une différence proportionnelle, on obtient
un pourcentage de performance pour chaque objectif et en moyenne. Ce pourcentage
est calculé à l'aide de la formule 5.1 :

$$\frac{\sum_i^k \left(1 - \frac{\text{abs}(\text{candidate.goals}_i - \text{worst_goals}_i)}{\text{worst_goals}_i - \text{best_goals}_i} * 100\right)}{k} \quad (5.1)$$

Avec k le nombre d'objectifs à considérer.

3140 Par exemple, pour un objectif dont la pire valeur est 0 et la meilleure valeur est 10,

et pour une solution présentant une valeur de 5 pour ce même objectif, le pourcentage de performance pour cet objectif est de 50%. Une solution présentant 60%, 40%, 30% et 50% de performance pour $k = 4$ objectifs, dispose d'une performance moyenne de 45%.

3145 Cette différence proportionnelle ne prend, à ce stade, pas encore en compte les préférences des agents passagers. Les préférences d'un agent passager sont stockées dans une liste $pref$ dont chaque élément $pref_i \in [0, \alpha]$, avec $\alpha \in \mathbb{R}_+$. La somme des éléments de $pref$ est égale à α : $sum(pref_i, \dots, pref_k) = \alpha$. Dans notre implémentation, nous supposons que $\alpha = 10$. La formule 5.1 est donc mise à jour dans la formule 3150 5.2 afin de prendre en compte les préférences de l'utilisateur pour chaque objectif.

$$\frac{\sum_i^k (1 - \frac{abs(candidate.goals_i - worst_goals_i)}{worst_goals_i - best_goals_i} * 100) * pref_i}{k + sum(pref)} \quad (5.2)$$

De cette façon, un objectif plus important pour le passager a plus d'impact sur le pourcentage final de performance de la solution. Le passager peut enfin sélectionner la solution la plus performante (PMO, L26).

5.3 Expérimentations et résultats

3155 Dans cette section, nous prouvons la pertinence de l'approche proposée. Tout d'abord, nous définissons des profils d'utilisateurs pour les agents (conducteurs et passagers) composant notre système de covoiturage dynamique. Ces profils visent à simuler des comportements d'utilisateurs évoluant dans un système de covoiturage dynamique dans le monde réel. Nous proposons ensuite d'évaluer ces comportements 3160 sur des graphes de routes réels. Nous terminons enfin cette section par la présentation et l'analyse de nos résultats.

5.3.1 Création des profils utilisateurs

Nous appelons profil utilisateur un ensemble de préférences visant à simuler un comportement réel. Pour les conducteurs, nous faisons essentiellement varier la préférence de détour det_{max} entre 0 (c'est-à-dire pas de détour) et 1, correspondant à une capacité de détour de la taille du trajet initial du conducteur. Nous avons donc plusieurs profils de conducteurs, allant du plus égoïste au plus altruiste. Concernant les passagers, nous définissons 5 profils utilisateurs présentés dans le tableau 5.1 en fonction des 4 objectifs présentés en section 5.1.2.

Profil	Temps d'attente	Distance additionnelle	Heure d'arrivée	Nombre de transferts
<i>Équilibré</i>	3	3	3	1
<i>Pluie</i>	7	1	1	1
<i>Écologique</i>	1	7	1	1
<i>Pressé</i>	1	1	7	1
<i>Confort</i>	1	1	1	7

TABLE 5.1 – Tableau des différents profils des agents passagers

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

Bien que les valeurs associées aux objectifs pour chaque profil soient purement empiriques, notre objectif est avant tout de mettre en évidence l'impact de l'intensification de l'optimisation d'un objectif sur le système.

5.3.2 Protocole expérimental

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

Nos instances sont composées du même graphe que le chapitre 4. Ce graphe de 150 nœuds provient d'un réseau routier réel de la ville de San Francisco, extrait d'Open Street Map et modélisé à l'aide de la bibliothèque OSMNX. Ce graphe est illustré dans la Figure 4.9. Sur ce graphe en grille, 30 conducteurs et 20 passagers interagissent. Leurs nœuds de départ et de destination sont générés de manière aléatoire et uniforme. Les itinéraires initiaux des conducteurs sont dérivés du plus court chemin entre leurs nœuds de départ et d'arrivée. Nous supposons que les passagers ont un temps d'attente maximal de 3 minutes et un facteur de détour fixé à 0,2 pour créer une situation difficile.

Nous notons que nous avons également effectué des simulations sur des graphes plus importants et complexes tels que la ville de Piedmont en Californie, représentée sur la Figure 2.10, mais nous ne l'avons pas sélectionnée pour les raisons mentionnées ci-dessus (nous nous concentrons sur un cadre expérimental avec peu de biais et peu de bruit). Une simulation sur un tel graphe était impossible avec la recherche exhaustive du chapitre 4, du fait de la complexité d'une telle approche.

5.3.3 Résultats

Dans cette sous-section, nous détaillons les résultats obtenus avec notre modèle. L'objectif est de mettre en évidence l'impact des différentes stratégies sur des indicateurs tels que : le temps d'attente moyen, le taux de service (succès du covoiturage), la distance additionnelle moyenne et le nombre moyen de véhicules nécessaires au covoiturage. Chaque figure comprend plusieurs courbes correspondant aux profils des

3220 passagers mentionnés précédemment. Chaque point d'une courbe est le résultat de
quel que soit le profil du passager. Enfin, l'axe des abscisses est indexé par le facteur
de détour des conducteurs. À l'exception de la position de départ et d'arrivée des
agents conducteurs et passagers, tout est déterministe.

L'impact du profil "écologiste"

3225 La Figure 5.3 montre comment le profil "écologiste" minimise la distance addi-
tionnelle moyenne de détour des conducteurs. Ce profil, représenté par la courbe verte
avec des triangles, minimise considérablement cette distance par rapport aux autres
courbes. Ce phénomène est amplifié par l'augmentation du facteur de détour des
conducteurs. On observe ainsi des détours 2 à 3 fois moins importants avec ce profil
3230 qu'avec les profils concurrents. On constate que, en conséquence, les autres objectifs
sont très largement négligés par ce profil, à savoir, le temps d'attente (Figure 5.5), le
nombre moyen de véhicules impliqués dans le covoiturage (Figure 5.4) et le taux de
service (Figure 5.6).

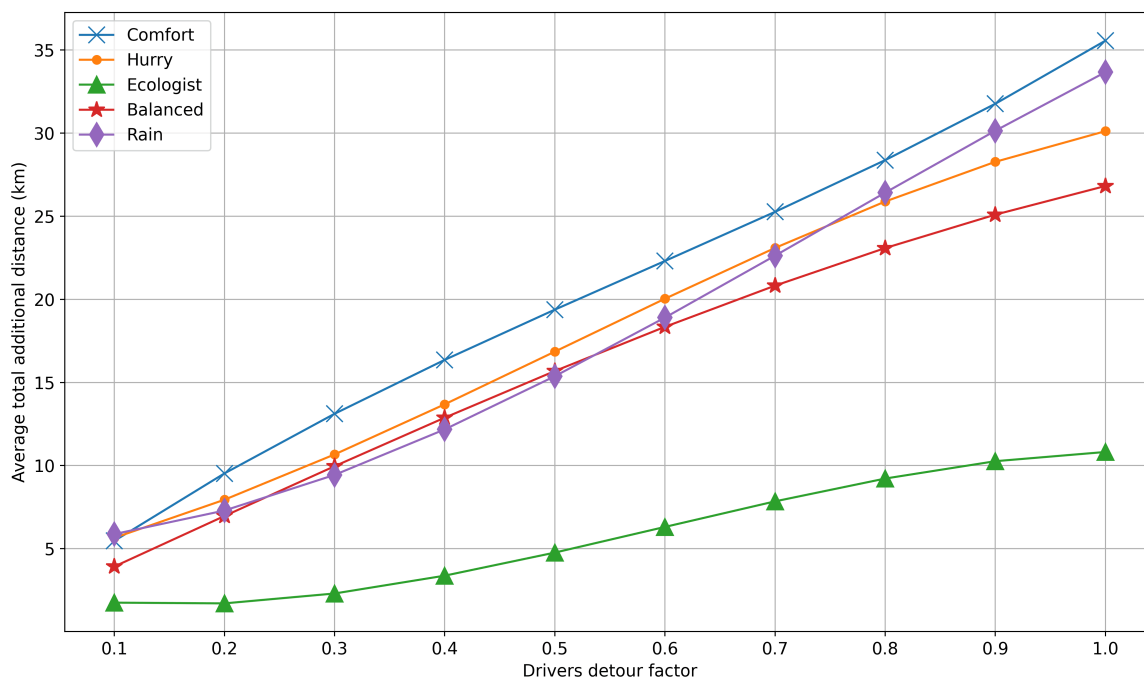


FIGURE 5.3 – Effet du facteur de détour des conducteurs et des profils des passagers sur la distance additionnelle moyenne de détour des conducteurs. Les données sont moyennées sur 50 expériences et chaque colonne correspond aux mêmes 50 expériences. Le profil "écologiste" (courbe verte avec des triangles) est plus performant que les autres profils pour cet indicateur.

L'impact du profil "confort"

3235 Le profil "confort", représenté par la courbe bleue avec des croix dans la Figure 5.4, vise à minimiser le nombre de transferts (changements de véhicules) pour atteindre la destination cible. Cette figure, qui représente le nombre moyen de véhicules impliqués dans un covoiturage en fonction du facteur de détour, révèle l'efficacité de ce profil. En effet, ce profil est capable de minimiser sa cible de plus en plus (environ de 1.4 à 1.1 véhicule impliqué en moyenne) avec l'augmentation du facteur de détour des conducteurs. Il est également bien meilleur que les autres profils en montrant entre 20% et 70% de performance supplémentaire sur cet objectif. En revanche, il est le plus faible pour la distance de détour supplémentaire (Figure 5.3) et est similaire à la majorité des autres profils pour les autres objectifs (Figure 5.6 et Figure 5.5). Nous
3240
3245 pouvons conclure ici que la réduction du nombre de transferts entre véhicules et la favorisation du covoiturage simple-saut augmente la distance de détour.

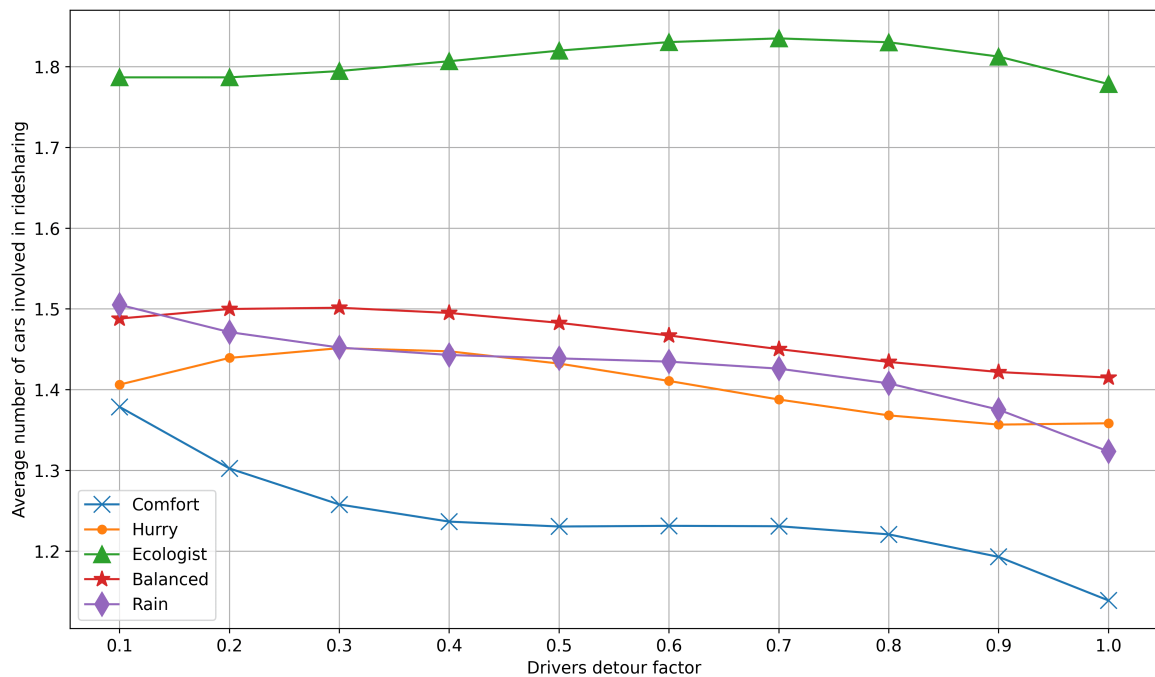


FIGURE 5.4 – Effet du facteur de détour des conducteurs et des profils des passagers sur le nombre moyen de véhicules impliqués dans le covoiturage. Les données sont moyennées sur 50 expériences et chaque colonne correspond aux mêmes 50 expériences. Le profil écologiste (courbe verte avec triangles) est sous-performant tandis que le profil confort (courbe bleue avec croix) surpasse les autres profils pour cet indicateur.

L'impact des profils "pluie", "pressé" et "équilibré".

Nous avons choisi de traiter ces trois profils conjointement, car, dans la majorité des cas, la tendance des courbes et leurs résultats sont similaires. Cela s'explique par la difficulté de dissocier le temps de parcours, le temps d'attente et la distance de parcours. Ces derniers sont en effet étroitement liés et interdépendants. Une attente trop longue a un impact sur le temps de trajet, de même qu'un trajet trop long en distance a le même effet. Néanmoins, le profil "pluie" réussit à réduire le temps d'attente dans les situations extrêmes (peu de détours). En effet, ce profil - tracé en violet avec des losanges - minimise au mieux le temps d'attente des passagers dans la Figure 5.5 lorsque les conducteurs ont un facteur de détour inférieur à 0,5. Il se confond ensuite avec les autres profils.

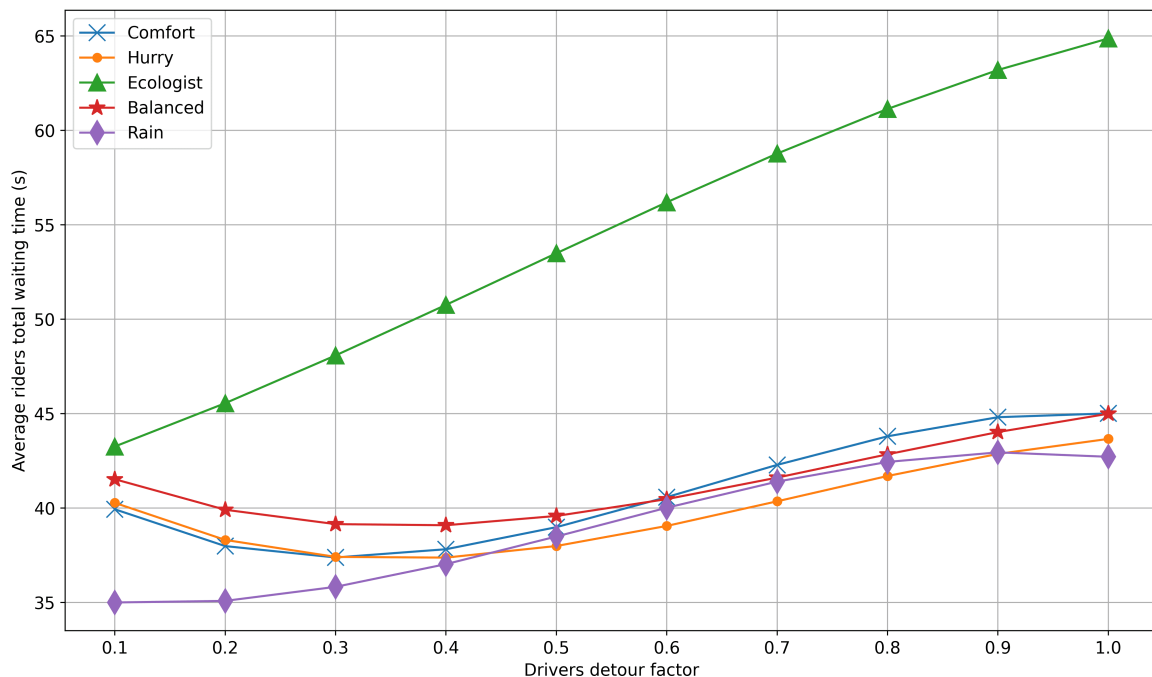


FIGURE 5.5 – Effet du facteur de détour des conducteurs et du profil des passagers sur le temps d'attente moyen d'un passager. Les données sont calculées en moyenne sur 50 expériences et chaque colonne correspond aux mêmes 50 expériences. Le profil écologiste (courbe verte avec triangles) est nettement moins performant. Le profil "pluie" (courbe violette avec des diamants) surpasse les autres profils sur l'indicateur en considérant un faible facteur de détour du conducteur.

Quant au profil "Pressé", cherchant à arriver le plus rapidement possible et donc à minimiser son temps d'arrivée, il cherche indirectement à réduire son temps d'attente. C'est pourquoi nous le voyons se positionner très légèrement comme le leader de la minimisation du temps d'attente sur la Figure 5.5 pour un facteur de détour du

conducteur de 0.5 à 0.9. Notre profil " équilibré " remplit son objectif d'être le juste milieu entre les différents profils et leurs objectifs, comme on peut le voir sur les différentes figures.

3265 La performance globale des profils : le taux de service

Pour conclure sur l'analyse de ces résultats, nous pouvons considérer la performance des profils du point de vue du taux de service. Le taux de service représente la part des passagers qui ont réussi à se rendre à leur destination en covoiturant. Les résultats concernant ce taux sont présentés dans la Figure 5.6 et sont fonction du

3270 facteur de détour des conducteurs.

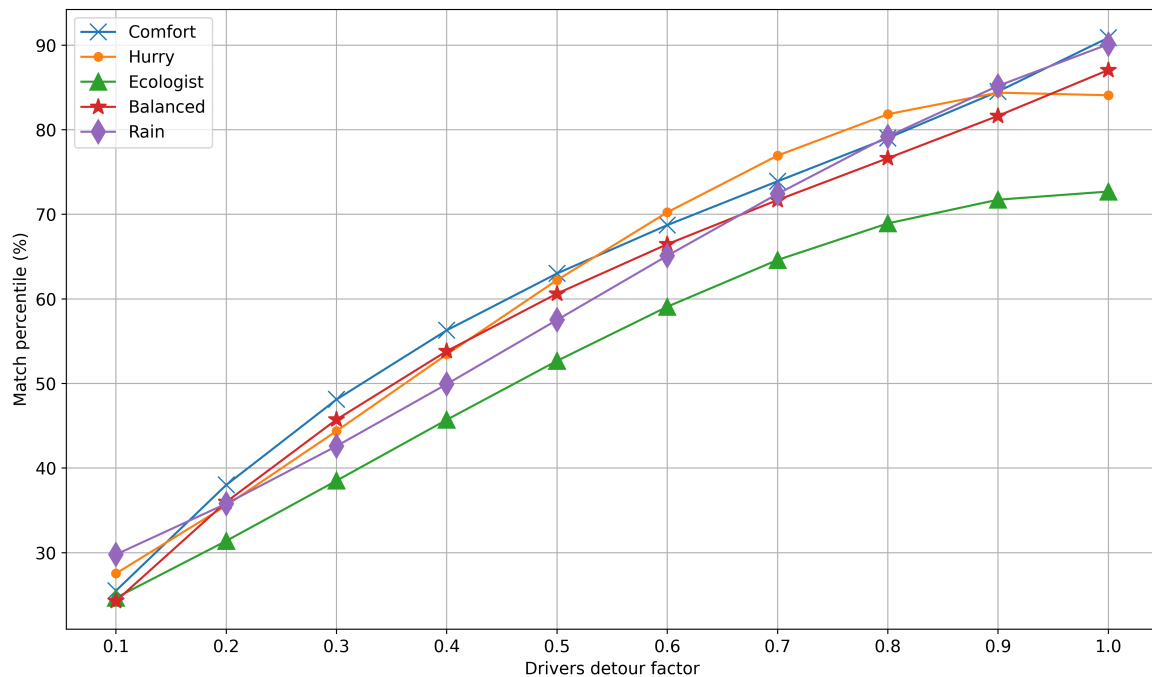


FIGURE 5.6 – Effet du facteur de détour des conducteurs et du profil du passager sur le taux de service du système. Les données sont calculées en moyenne sur 50 expériences et chaque colonne correspond aux mêmes 50 expériences. Le profil écologiste (courbe verte avec triangles) présente une sous-performance significative avec une différence moyenne de 5 à 15%.

Le premier élément d'étude est la tendance générale des courbes. En effet, plus le facteur de détour augmente, plus il existe de solutions potentielles et donc plus le taux de service augmente. Ce phénomène est normal et démontre le bon fonctionnement de notre modèle. Nous remarquons que le profil "écologiste" (triangles verts) présente un

3275 taux de service relativement faible par rapport au reste des profils avec une différence

moyenne de 5 à 15%. Aussi, il représente le pire taux de service atteint avec environ 72% pour un facteur de détour des conducteurs de 1. Si on émet l'hypothèse que les utilisateurs qui souhaitaient être passagers ont finalement pris leur véhicule pour rejoindre leur destination, il se peut que ce profil ne permette pas de réduire, autant qu'espéré, les émissions globales du système, curieux paradoxe !

Concernant les autres profils, leurs courbes sont similaires et se rejoignent à 5% près. Le meilleur taux de service atteint est de 90% avec les profils "pluie" et "confort" respectivement soulignés en violet par des losanges et en bleu par des croix.

5.4 Conclusion et perspectives

Nous avons étudié dans ce chapitre le covoiturage dynamique multi-saut avec les préférences des utilisateurs. Nous proposons un système multi-agent pour modéliser les différents acteurs d'un système de covoiturage et leur spectre de comportements possibles. Nous avons utilisé la puissance de l'indexation spatiale R-Arbre pour atteindre deux objectifs : 1) fournir une gestion et un accès à l'espace avec un faible coût de calcul, et 2) transformer le problème du covoiturage en un problème ANN afin de pouvoir le résoudre à l'aide d'une extension de l'algorithme de référence dans la littérature. Les expériences présentées ont montré que l'approche proposée permet une optimisation individuelle et efficace.

Cette approche révèle certaines limitations pour les profils favorisant le temps d'attente et le temps de trajet. En effet, l'optimisation par agrégation des pondérations semble insuffisante lorsque les objectifs sont interdépendants. Pour répondre à ce constat, d'autres techniques peuvent être envisagées, comme l'agrégation par l'intégrale de Choquet, qui s'est avérée efficace pour l'optimisation multicritère [Cheikh et Hammadi, 2014, Ben Cheikh-Graiet *et al.*, 2020]. Une autre perspective consiste à rendre le comportement des agents dynamique. En effet, le comportement des agents est pour l'instant défini à leur apparition et ne change pas au cours du temps. En fonction de l'état de leur environnement (trafic, météo, manque d'approvisionnement, etc.), leur comportement pourrait évoluer les mener à changer leurs objectifs principaux. En étendant notre approche à des agents au comportement dynamique, nous nous rapprocherions encore davantage du monde réel et de ses caractéristiques.

Enfin, il serait fructueux de considérer également la complémentarité avec d'autres

modalités de transport, en particulier les transports publics tels que le bus et le métro. Nous passerions alors de la mono-modalité à la comodalité, c'est-à-dire la multimodalité associée au covoiturage.

Conclusion générale

Synthèse

Dans ce travail de recherche, nous proposons un système d'aide à la décision de covoiturage basé sur une approche à la croisée des domaines de la simulation multi-agents et de l'optimisation.

3315

Nous identifions trois mécanismes ayant une influence majeure sur la complexité du problème du covoiturage dynamique. (i) Les détours, impliquant de devoir générer une perception dynamique de l'espace accessible par chaque utilisateur, qu'il soit passager ou conducteur. (ii) La réservation, qui prend la forme d'un contrat entre un passager et un conducteur et limite l'optimisation dans le temps. (iii) Les transferts, qui permettent aux passagers de changer de véhicule pendant leur trajet, mais induisent une complexité accrue du fait de la nécessité d'identifier le nœud de transfert optimal.

3320

Nous faisons, pas-à-pas, évoluer notre modèle afin de prendre en main de façon progressive ces mécanismes et d'atteindre notre objectif d'un système de covoiturage dynamique adapté au monde réel, à sa population hétérogène et à ses contraintes de réactivité. Notre démarche scientifique ainsi que nos contributions peuvent être résumées en trois étapes :

3325

1. Dans le chapitre 3, nous mettons en avant l'avantage d'un système d'information et d'aide à la décision de covoiturage pour des autostoppeurs. Nous détaillons un modèle de covoiturage minimal, basé sur des graphes artificiels, permettant une infinité de transferts de passager, mais excluant toute réservation ou détour des conducteurs. Nous comparons un autostoppeur connecté à notre système d'aide à la décision face à un autostoppeur du monde réel, et donc non connecté. Nos résultats démontrent l'efficacité de notre approche avec

3330

3335 une plus grande rapidité dans la résolution des instances et une plus grande
part de passagers rejoignant leur destination quand ils sont connectés [Fèvre
et al., 2022].

2. Dans le chapitre 4, nous enrichissons notre modèle en intégrant la réservation
et les détours, tout en faisant évoluer nos agents utilisateurs sur un graphe
3340 construit à partir de données réelles. Après avoir fait face à diverses problé-
matiques de discrétisation de l’espace, nous proposons une méthode basée sur
l’indexation spatiale R-Arbre afin de modéliser des perceptions dynamiques
pour les utilisateurs. Ces perceptions nous permettent d’identifier efficacement
les candidats au covoiturage pour un passager. Nos résultats montrent le béné-
3345 fice de la capacité d’effectuer des transferts en témoignant d’un meilleur taux
de service et d’une moindre distance totale parcourue face à des passagers
n’effectuant pas de transferts [Fèvre *et al.*, 2021].

3. Enfin, dans le chapitre 5, nous augmentons le niveau de réalisme de notre mo-
dèle : d’abord en proposant un algorithme performant pour l’identification du
3350 nœud de transfert optimal, gage d’une réponse rapide dans le monde réel ; en-
suite en détaillant un nouveau système de préférences pour les passagers, afin
de simuler la variété des comportements présents dans la population. Nous
transformons le problème du covoiturage en un problème ANN afin de pouvoir
le résoudre à l’aide d’une extension de l’algorithme de référence de la litté-
3355 rature. Nos expériences montrent que notre modèle peut simuler efficacement
des comportements du monde réel et est capable d’optimiser le covoiturage au
cas par cas [Fèvre *et al.*, 2022].

Perspectives

Notre système ayant été construit de façon à pouvoir intégrer facilement de nou-
3360 velles mobilités et préférences dans le processus d’optimisation, il serait intéressant
d’étudier les enjeux et apports du covoiturage dynamique en tant que modalité alter-
native dans un système multimodal classique, et ainsi atteindre la co-modalité. Une
comparaison entre un système multimodal perturbé (ex : panne de transport), avec
ou sans covoiturage dynamique, pourrait mettre en évidence la complémentarité de
3365 ces offres de transport.

De plus, l’intégration des parkings relais (P+R) en tant que passerelle entre le

covoiturage et l'offre de transport public est une piste intéressante à explorer. Ces parkings se positionneraient alors comme des portes d'entrée et de sortie du système de transport public, aussi bien que comme des nœuds de concentration pour l'offre et la demande de covoiturage.

Enfin, il serait intéressant de faire évoluer le comportement des agents en fonction de l'environnement. On disposerait ainsi d'un environnement dynamique dans lequel des perturbations pourraient survenir et, en conséquence, les agents du système changeraient la pondération de leurs préférences en temps réel. Cette évolution permettrait de se rapprocher de situations réalistes et d'étudier l'impact d'une perturbation sur la population.

Bibliographie

- [Adler et Heeringa, 2008] ADLER, M. et HEERINGA, B. (2008). Search Space Reductions for Nearest-Neighbor Queries. *In* AGRAWAL, M., DU, D., DUAN, Z. et LI, A., éditeurs : *Theory and Applications of Models of Computation*, volume 4978, pages 554–567. Springer Berlin Heidelberg, Berlin, Heidelberg.
- [Agatz *et al.*, 2012] AGATZ, N., ERERA, A., SAVELSBERGH, M. et WANG, X. (2012). Optimization for dynamic ride-sharing : A review. *European Journal of Operational Research*, 223(2):295–303.
- [Agatz *et al.*, 2011] AGATZ, N., ERERA, A. L., SAVELSBERGH, M. W. et WANG, X. (2011). Dynamic Ride-Sharing : A Simulation Study in Metro Atlanta. *Procedia - Social and Behavioral Sciences*, 17:532–550.
- [Alonso-Mora *et al.*, 2017] ALONSO-MORA, J., SAMARANAYAKE, S., WALLAR, A., FRAZZOLI, E. et RUS, D. (2017). On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. *Proceedings of the National Academy of Sciences*, 114(3):462–467.
- [Ayed *et al.*, 2011] AYED, H., GALVEZ-FERNANDEZ, C., HABBAS, Z. et KHADRAOUI, D. (2011). Solving time-dependent multimodal transport problems using a transfer graph model. *Computers & Industrial Engineering*, 61(2):391–401.
- [Balbo et Pinson, 2005] BALBO, F. et PINSON, S. (2005). Dynamic modeling of a disturbance in a multi-agent system for traffic regulation. *Decision Support Systems*, 41(1):131–146.
- [Baldacci *et al.*, 2004] BALDACCI, R., MANIEZZO, V. et MINGOZZI, A. (2004). An Exact Method for the Car Pooling Problem Based on Lagrangean Column Generation. *Operations Research*, 52:422–439.
- [Beckmann *et al.*, 1990] BECKMANN, N., KRIEGEL, H.-P., SCHNEIDER, R. et SEEGER, B. (1990). The R*-tree : An efficient and robust access method for points and rectangles. *ACM SIGMOD Record*, 19(2):322–331.

- [Bellman, 1958] BELLMAN, R. (1958). On a Routing Problem. *Quarterly of Applied Mathematics*, 16(1):87–90.
- 3405
- [Ben Cheikh-Graiet *et al.*, 2020] BEN CHEIKH-GRAIET, S., DOTOLI, M. et HAMMADI, S. (2020). A Tabu Search Based Metaheuristic for Dynamic Carpooling Optimization. *Computers & Industrial Engineering*.
- [Bessghaier *et al.*, 2012] BESSGHAIER, N., ZARGAYOUNA, M. et BALBO, F. (2012). Management of urban parking : An agent-based approach. *In International Conference on Artificial Intelligence : Methodology, Systems, and Applications*, pages 276–285. Springer.
- 3410
- [Bistaffa *et al.*, 2015] BISTAFFA, F., FARINELLI, A. et RAMCHURN, S. (2015). *Sharing Rides with Friends : A Coalition Formation Algorithm for Ridesharing*.
- [Boccaletti *et al.*, 2014] BOCCALETTI, S., BIANCONI, G., CRIADO, R., DEL GENIO, C. I., GÓMEZ-GARDEÑES, J., ROMANCE, M., SENDIÑA-NADAL, I., WANG, Z. et ZANIN, M. (2014). The structure and dynamics of multilayer networks. *Physics Reports*, 544(1):1–122.
- 3415
- [Boeing, 2017] BOEING, G. (2017). OSMnx : New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Computers, Environment and Urban Systems*, 65:126–139.
- 3420
- [Cheikh et Hammadi, 2014] CHEIKH, S. B. et HAMMADI, S. (2014). Multi-criterion Tabu Search to Solve the Dynamic Carpooling Based on the Choquet Integral Aggregation. *Journal of Traffic and Logistics Engineering*, 2(2):126–132.
- [Cheikh et Hammadi, 2016] CHEIKH, S. B. et HAMMADI, S. (2016). Multi-Hop Ridematching optimization problem : Intelligent chromosome agent-driven approach. *Expert Systems with Applications*, 62:161–176.
- 3425
- [Cheikh *et al.*, 2015] CHEIKH, S. B., HAMMADI, S. et TAHON, C. (2015). Agent-based Evolutionary Cooperative Approach for Dynamic Multi-Hop Ridematching Problem. *IFAC-PapersOnLine*, 48(3):887–892.
- 3430
- [Chen *et al.*, 2019] CHEN, Z., LIU, X. C. et WEI, R. (2019). Agent-based approach to analyzing the effects of dynamic ridesharing in a multimodal network. *Computers, Environment and Urban Systems*, 74:126–135.
- [Chen *et al.*, 2020] CHEN, Z., YAO, B., WANG, Z.-J., GAO, X., SHANG, S., MA, S. et GUO, M. (2020). Flexible Aggregate Nearest Neighbor Queries and its Keyword-Aware Variant on Road Networks. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1.
- 3435

- [Clarke, 1971] CLARKE, E. H. (1971). Multipart pricing of public goods. *Public Choice*, 11(1):17–33.
- 3440 [Coltin et Veloso, 2014] COLTIN, B. et VELOSO, M. (2014). Ridesharing with passenger transfers. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3278–3283.
- [Cordeau et Laporte, 2007] CORDEAU, J.-F. et LAPORTE, G. (2007). The dial-a-ride problem : Models and algorithms. *Ann Oper Res*, page 18.
- 3445 [Cortés et al., 2010] CORTÉS, C. E., MATAMALA, M. et CONTARDO, C. (2010). The pickup and delivery problem with transfers : Formulation and a branch-and-cut solution method. *European Journal of Operational Research*, 200(3):711–724.
- [Daoud et al., 2020] DAOUD, A., BALBO, F., GIANESSI, P. et PICARD, G. (2020). Decentralized Insertion Heuristic with Runtime Optimization for On-demand Transport Scheduling. In *11th International Workshop on Agents in Traffic and Transportation (ATT 2020)*, Santiago de Compostela, Spain.
- 3450 [Di Febbraro et al., 2013] DI FEBBRARO, A., GATTORNA, E. et SACCO, N. (2013). Optimization of Dynamic Ridesharing Systems. *Transportation Research Record : Journal of the Transportation Research Board*, 2359(1):44–50.
- 3455 [Dijkstra, 1959] DIJKSTRA, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271.
- [Dong et al., 2005] DONG, J., CHEN, S. et JENG, J.-J. (2005). Event-based blackboard architecture for multi-agent systems. In *International Conference on Information Technology : Coding and Computing (ITCC'05) - Volume II*, volume 2, pages 379–384 Vol. 2.
- 3460 [Dotoli et al., 2017] DOTOLI, M., ZGAYA, H., RUSSO, C. et HAMMADI, S. (2017). A Multi-Agent Advanced Traveler Information System for Optimal Trip Planning in a Co-Modal Framework. *IEEE Transactions on Intelligent Transportation Systems*, 18(9):2397–2412.
- 3465 [Emmerink et van Beek, 1997] EMMERINK, R. H. M. et VAN BEEK, P. (1997). Empirical Analysis of Work Schedule Flexibility : Implications for Road Pricing and Driver Information Systems. *Urban Studies*, 34(2):217–234.
- [Feng et al., 2017] FENG, F., PANG, Y., LODEWIJKS, G. et LI, W. (2017). Collaborative framework of an intelligent agent system for efficient logistics transport planning. *Computers & Industrial Engineering*, 112:551–567.
- 3470 [Ferber, 1997] FERBER, J. (1997). *Les Systèmes Multi-Agents : Vers Une Intelligence Collective*. InterEditions.

- [Ferber et Weiss, 1999] FERBER, J. et WEISS, G. (1999). *Multi-Agent Systems : An Introduction to Distributed Artificial Intelligence*, volume 1. Addison-wesley Reading. 3475
- [Fevre et al., 2021] FEVRE, C., ZGAYA-BIAU, H., MATHIEU, P. et HAMMADI, S. (2021). Multi-agent Systems and R-Trees for Dynamic and Optimised Ridesharing. In *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 1352–1358.
- [Fèvre et al., 2022] FÈVRE, C., ZGAYA-BIAU, H., MATHIEU, P. et HAMMADI, S. 3480 (2022). Agent-based Modeling for Dynamic Hitchhiking Simulation and Optimization. In *14th International Conference on Agents and Artificial Intelligence*, pages 322–329.
- [Fevre et al., 2022] FEVRE, C., ZGAYA-BIAU, H., MATHIEU, P. et HAMMADI, S. 3485 (2022). Preferential Optimization of Multi-hop Dynamic Ridesharing based on R-Trees and Multi-Agent Systems. *Under review*.
- [Ford, 1956] FORD, L. R. (1956). Network Flow Theory. Rapport technique, RAND Corporation.
- [Furuhata et al., 2013] FURUHATA, M., DESSOUKY, M., ORDÓÑEZ, F., BRUNET, M.-E., WANG, X. et KOENIG, S. (2013). Ridesharing : The state-of-the-art and future 3490 directions. *Transportation Research Part B : Methodological*, 57:28–46.
- [Gerte et al., 2018] GERTE, R., KONDURI, K. C. et ELURU, N. (2018). Is There a Limit to Adoption of Dynamic Ridesharing Systems? Evidence from Analysis of Uber Demand Data from New York City. *Transportation Research Record*, 2672(42):127– 3495 136.
- [Groves, 1973] GROVES, T. (1973). Incentives in Teams. *Econometrica*, 41(4):617–631.
- [Guériau et al., 2016] GUÉRIAU, M., BILLOT, R., EL FAOUZI, N.-E., MONTEIL, J., ARMETTA, F. et HASSAS, S. (2016). How to assess the benefits of connected 3500 vehicles? A simulation framework for the design of cooperative traffic management strategies. *Transportation research part C : emerging technologies*, 67:266–279.
- [Guttman, 1984] GUTTMAN, A. (1984). *R Trees : A Dynamic Index Structure for Spatial Searching*, volume 14.
- [Hart et al., 1968] HART, P. E., NILSSON, N. J. et RAPHAEL, B. (1968). A Formal 3505 Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107.

- [Herbawi et Weber, 2012a] HERBAWI, W. et WEBER, M. (2012a). *Modeling the Multihop Ridematching Problem with Time Windows and Solving It Using Genetic Algorithms*, volume 1.
- 3510 [Herbawi et Weber, 2012b] HERBAWI, W. et WEBER, M. (2012b). The ridematching problem with time windows in dynamic ridesharing : A model and a genetic algorithm. *In 2012 IEEE Congress on Evolutionary Computation*, pages 1–8, Brisbane, Australia. IEEE.
- [Jeribi et al., 2011] JERIBI, K., MEJRI, H., ZGAYA, H. et HAMMADI, S. (2011). Distributed graphs for solving co-modal transport problems. *In 2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 1150–1155.
- 3515 [Jonuschat et al., 2015] JONUSCHAT, H., STEPHAN, K. et SCHELEWSKY, M. (2015). Understanding Multimodal and Intermodal Mobility. *Transport and Sustainability*, 7:149–176.
- 3520 [Kamar et Horvitz, 2009] KAMAR, E. et HORVITZ, E. (2009). Collaboration and Shared Plans in the Open World : Studies of Ridesharing. *In IJCAI International Joint Conference on Artificial Intelligence*, page 187.
- [Karp, 1972] KARP, R. M. (1972). Reducibility among Combinatorial Problems. *In MILLER, R. E., THATCHER, J. W. et BOHLINGER, J. D., éditeurs : Complexity of Computer Computations : Proceedings of a Symposium on the Complexity of Computer Computations, Held March 20–22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, and Sponsored by the Office of Naval Research, Mathematics Program, IBM World Trade Corporation, and the IBM Research Mathematical Sciences Department, The IBM Research Symposia*
- 3525 *Series*, pages 85–103. Springer US, Boston, MA.
- 3530 [Kelly et al., 1998] KELLY, F. P., MAULLOO, A. K. et TAN, D. K. H. (1998). Rate control for communication networks : Shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49(3):237–252.
- [Kivela et al., 2014] KIVELA, M., ARENAS, A., BARTHELEMY, M., GLEESON, J. P.,
- 3535 MORENO, Y. et PORTER, M. A. (2014). Multilayer networks. *Journal of Complex Networks*, 2(3):203–271.
- [Kleiner et al., 2011] KLEINER, A., NEBEL, B. et ZIPARO, V. A. (2011). A Mechanism for Dynamic Ride Sharing Based on Parallel Auctions. *In Twenty-Second International Joint Conference on Artificial Intelligence*.
- 3540 [Lacroix et al., 2009a] LACROIX, B., MATHIEU, P. et KEMENY, A. (2009a). Generating Various and Consistent Behaviors in Simulations. *In DEMAZEAU, Y., PAVÓN,*

- J., CORCHADO, J. M. et BAJO, J., éditeurs : *7th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 2009)*, Advances in Intelligent and Soft Computing, pages 110–119, Berlin, Heidelberg. Springer.
- 3545 [Lacroix *et al.*, 2009b] LACROIX, B., MATHIEU, P. et KEMENY, A. (2009b). The Use of Norms Violations to Model Agents Behavioral Variety. In HÜBNER, J. F., MATSON, E., BOISSIER, O. et DIGNUM, V., éditeurs : *Coordination, Organizations, Institutions and Norms in Agent Systems IV*, Lecture Notes in Computer Science, pages 220–234, Berlin, Heidelberg. Springer.
- 3550 [Lacroix *et al.*, 2013] LACROIX, B., MATHIEU, P. et KEMENY, A. (2013). Formalizing the Construction of Populations in Multi-Agent Simulations. *Engineering Applications of Artificial Intelligence*, 26:211–226.
- [Li *et al.*, 2019] LI, M., ZHIWEI, QIN, JIAO, Y., YANG, Y., GONG, Z., WANG, J., WANG, C., WU, G. et YE, J. (2019). Efficient Ridesharing Order Dispatching with
3555 Mean Field Multi-Agent Reinforcement Learning. *arXiv :1901.11454 [cs]*.
- [Lin *et al.*, 2016] LIN, J., SASIDHARAN, S., MA, S. et WOLFSON, O. (2016). A Model of Multimodal Ridesharing and Its Analysis. In *2016 17th IEEE International Conference on Mobile Data Management (MDM)*, pages 164–173, Porto. IEEE.
- [Lloret-Batlle *et al.*, 2017] LLORET-BATLLE, R., MASOUD, N. et NAM, D. (2017).
3560 Peer-to-Peer Ridesharing with Ride-Back on High-Occupancy-Vehicle Lanes : Toward a Practical Alternative Mode for Daily Commuting. *Transportation Research Record : Journal of the Transportation Research Board*, 2668:21–28.
- [Lopes Silva *et al.*, 2018] LOPES SILVA, M. A., DE SOUZA, S. R., FREITAS SOUZA, M. J. et DE FRANÇA FILHO, M. F. (2018). Hybrid metaheuristics and multi-
3565 agent systems for solving optimization problems : A review of frameworks and a comparative analysis. *Applied Soft Computing*, 71:433–459.
- [Ma et Wolfson, 2013] MA, S. et WOLFSON, O. (2013). Analysis and evaluation of the slugging form of ridesharing. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - SIG-
3570 SPATIAL'13*, pages 64–73, Orlando, Florida. ACM Press.
- [Manolopoulos *et al.*, 2010] MANOLOPOULOS, Y., NANOPOULOS, A., PAPADOPOULOS, A. N. et THEODORIDIS, Y. (2010). *R-Trees : Theory and Applications*. Springer Science & Business Media.
- [Manolopoulos *et al.*, 2014] MANOLOPOULOS, Y., THEODORIDIS, Y. et TSOTRAS, V. J. (2014). Spatial Indexing Techniques. In LIU, L. et ÖZSU, M. T., éditeurs : *Encyclopedia of Database Systems*, pages 1–7. Springer New York, New York, NY.

- [Masoud et Jayakrishnan, 2017] MASOUD, N. et JAYAKRISHNAN, R. (2017). A real-time algorithm to solve the peer-to-peer ride-matching problem in a flexible ride-sharing system. *Transportation Research Part B : Methodological*, 106:218–236.
- 3580 [Mathieu et Nongailard, 2018] MATHIEU, P. et NONGAILLARD, A. (2018). Effective Evaluation of Autonomous Taxi Fleets :. *In Proceedings of the 10th International Conference on Agents and Artificial Intelligence*, pages 297–304, Funchal, Madeira, Portugal. SCITEPRESS - Science and Technology Publications.
- [Mathieu et al., 2015] MATHIEU, P., PICAULT, S. et SECQ, Y. (2015). Design Pat-
3585 terns for Environments in Multi-agent Simulations. *In CHEN, Q., TORRONI, P., VILLATA, S., HSU, J. et OMICINI, A., éditeurs : PRIMA 2015 : Principles and Practice of Multi-Agent Systems*, volume 9387, pages 678–686. Springer International Publishing, Cham.
- [Mathieu et al., 2016] MATHIEU, P., PICAULT, S. et SECQ, Y. (2016). Design Pat-
3590 terns pour les environnements dans les simulations multi-agents. *Revue des Sciences et Technologies de l'Information - Série RIA : Revue d'Intelligence Artificielle*, 30(1-2):133.
- [Moore, 1957] MOORE, E. F. (1957). The Shortest Path through a Maze. *Harvard University Press, Cambridge*.
- 3595 [Moulin et Chaib-draa, 1996] MOULIN, B. et CHAIB-DRAA, B. (1996). An overview of distributed artificial intelligence. *undefined*.
- [Moulin, 1988] MOULIN, H. (1988). *Axioms of Cooperative Decision Making*, volume 15. Cambridge University Press, Cambridge, UK.
- [Moulin, 2004] MOULIN, H. (2004). *Fair Division and Collective Welfare*, volume 1.
3600 The MIT Press.
- [Nash, 1950] NASH, J. F. (1950). The Bargaining Problem. *Econometrica*, 18(2):155–162.
- [Nicosia et al., 2017] NICOSIA, G., PACIFICI, A. et PFERSCHY, U. (2017). Price of
3605 Fairness for allocating a bounded resource. *European Journal of Operational Research*, 257(3):933–943.
- [Nisan et Ronen, 2007] NISAN, N. et RONEN, A. (2007). Computationally Feasible VCG Mechanisms. *Journal of Artificial Intelligence Research*, 29:19–47.
- [Nourinejad et Roorda, 2016] NOURINEJAD, M. et ROORDA, M. J. (2016). Agent
3610 based model for dynamic ridesharing. *Transportation Research Part C : Emerging Technologies*, 64:117–132.

- [Orozco *et al.*, 2021] OROZCO, L. G. N., ALESSANDRETTI, L., SABERI, M., SZELL, M. et BATTISTON, F. (2021). Multimodal urban mobility and multilayer transport networks. *arXiv :2111.02152 [physics]*.
- [Orozco *et al.*, 2020] OROZCO, L. G. N., BATTISTON, F., IÑIGUEZ, G. et SZELL, M. (2020). Extracting the Multimodal Fingerprint of Urban Transportation Networks. *Findings*, page 13171.
- [Papadias *et al.*, 2004] PAPADIAS, D., SHEN, Q., TAO, Y. et MOURATIDIS, K. (2004). *Group Nearest Neighbor Queries*, volume 20.
- [Papadias *et al.*, 2005] PAPADIAS, D., TAO, Y., MOURATIDIS, K. et HUI, C. K. (2005). Aggregate nearest neighbor queries in spatial databases. *ACM Transactions on Database Systems*, 30(2):529–576.
- [Papadias *et al.*, 2003] PAPADIAS, D., ZHANG, J., MAMOULIS, N. et TAO, Y. (2003). Query Processing in Spatial Network Databases. *In Proceedings 2003 VLDB Conference*, pages 802–813. Elsevier.
- [Pareto, 1964] PARETO, V. (1964). *Cours d'économie politique*. Librairie Droz.
- [Pelzer *et al.*, 2015] PELZER, D., XIAO, J., ZEHE, D., LEES, M. H., KNOLL, A. C. et AYDT, H. (2015). A Partition-Based Match Making Algorithm for Dynamic Ridesharing. *IEEE Transactions on Intelligent Transportation Systems*, 16(5):2587–2598.
- [Rao et Georgeff, 1995] RAO, A. S. et GEORGEFF, M. P. (1995). BDI Agents : From Theory to Practice. page 8.
- [Russell et Norvig, 2021] RUSSELL, S. et NORVIG, P. (2021). Artificial intelligence : A modern approach, global edition 4th. *Foundations*, 19:23.
- [Russell et Norvig, 1995] RUSSELL, S. J. et NORVIG, P. (1995). *Artificial Intelligence : A Modern Approach*. Prentice-Hall.
- [Russell et Norvig, 2010] RUSSELL, S. J. et NORVIG, P. (2010). *Artificial Intelligence a Modern Approach*. Pearson Education, Inc.
- [Sayarshad et Chow, 2015] SAYARSHAD, H. R. et CHOW, J. Y. J. (2015). A scalable non-myopic dynamic dial-a-ride and pricing problem. *Transportation Research Part B : Methodological*, 81:539–554.
- [Shorten, 2020] SHORTEN, S. (2020). *Computing Multimodal Journeys : A Distributed Approach*. Thèse de doctorat.

- [Shuo Ma *et al.*, 2013] SHUO MA, YU ZHENG et WOLFSON, O. (2013). T-share :
A large-scale dynamic taxi ridesharing service. *In 2013 IEEE 29th International
3645 Conference on Data Engineering (ICDE)*, pages 410–421, Brisbane, QLD. IEEE.
- [Solomon, 1987] SOLOMON, M. M. (1987). Algorithms for the Vehicle Routing and
Scheduling Problems with Time Window Constraints. *Operations Research*.
- [Tafreshian et Masoud, 2020] TAFRESHIAN, A. et MASOUD, N. (2020). Trip-based
graph partitioning in dynamic ridesharing. *Transportation Research Part C : Emer-
3650 ging Technologies*, 114:532–553.
- [Tong *et al.*, 2018] TONG, Y., ZENG, Y., ZHOU, Z., CHEN, L., YE, J. et XU, K.
(2018). A unified approach to route planning for shared mobility. *Proceedings of
the VLDB Endowment*, 11(11):1633–1646.
- [Vickrey, 1961] VICKREY, W. (1961). Counterspeculation, Auctions, and Competi-
3655 tive Sealed Tenders. *The Journal of Finance*, 16(1):8–37.
- [Weiss, 1999] WEISS, G. (1999). *Multiagent Systems : A Modern Approach to Dis-
tributed Artificial Intelligence*. MIT press.
- [Wolfler Calvo *et al.*, 2004] WOLFLER CALVO, R., DE LUIGI, F., HAASTRUP, P. et
MANIEZZO, V. (2004). A distributed geographic information system for the daily
3660 car pooling problem. *Computers & Operations Research*, 31(13):2263–2278.
- [Wooldridge, 2009] WOOLDRIDGE, M. (2009). *An Introduction to MultiAgent Sys-
tems*. John Wiley & Sons.
- [Xu *et al.*, 2020] XU, Y., KULIK, L., BOROVICA-GAJIC, R., ALDWYISH, A. et QI, J.
(2020). Highly Efficient and Scalable Multi-hop Ride-sharing. *In Proceedings of
3665 the 28th International Conference on Advances in Geographic Information Systems*,
pages 215–226, Seattle WA USA. ACM.
- [Xu *et al.*, 2019] XU, Y., QI, J., BOROVICA-GAJIC, R. et KULIK, L. (2019). Geo-
Prune : Efficiently Finding Shareable Vehicles Based on Geometric Properties.
arXiv :1907.01831 [cs, eess].
- 3670 [Yiu *et al.*, 2005] YIU, M., MAMOULIS, N. et PAPADIAS, D. (2005). Aggregate near-
est neighbor queries in road networks. *Knowledge and Data Engineering, IEEE
Transactions on*, 17:820–833.

Titre en français : Alliance entre les SMA et l'Optimisation au service du Covoiturage Dynamique

3675 **Résumé en français :** La crise écologique et énergétique actuelle mène à la nécessité de trouver des alternatives à l'autosolisme (conduite solitaire). Le covoiturage représente une solution prometteuse, notamment sous sa forme dynamique. Un système de covoiturage dynamique s'adapte en temps réel aux changements d'état de l'environnement afin de satisfaire les utilisateurs, à savoir les passagers et les conducteurs. Pour respecter ce caractère temps réel, il est nécessaire, d'une part, 3680 de maîtriser la complexité liée à l'optimisation des trajets, et d'autre part, de veiller à éviter la concentration de données sur un même système. En outre, il est important de prendre en compte les préférences de chaque utilisateur afin d'optimiser de façon personnalisée des objectifs tels que le temps d'attente, le temps de trajet ou encore le nombre de transferts. Dans cette thèse, nous nous intéressons aux divers mécanismes et variantes de cette forme de covoiturage. Nous défendons 3685 une approche individu-centrée, afin de mieux respecter le caractère hétérogène des utilisateurs, et décentralisée, pour permettre une meilleure répartition du flux de données, et donc un temps de réponse plus rapide. Nous proposons d'allier les systèmes multi-agents - dont la nature est intrinsèquement individu-centrée et décentralisée - à différents algorithmes d'optimisation dans le but de répondre aux problématiques énoncées. Les différents résultats issus de nos simulations attestent de 3690 la performance et de la robustesse de notre approche, notamment dans des situations complexes telles que la pénurie d'offres de covoiturage. Le système proposé dans cette thèse permet de traiter dynamiquement des requêtes complexes de passagers tout en minimisant l'impact du partage de trajet pour les conducteurs, et ce, pour un large spectre de préférences et de comportements.

3695 **Mots-clefs en français :** Covoiturage dynamique, Systèmes multi-agents, Optimisation, Simulation, Mobilité, Transport

Titre en anglais : Multi-Agent Systems and Optimization systems for Dynamic Ridesharing

Résumé en anglais : The current ecological and energy crisis leads to the necessity to find alternatives to autosolism (solitary driving). Ridesharing represents a promising solution, especially 3700 in its dynamic form. A dynamic ridesharing system adapts in real time to the changing state of the environment in order to satisfy the users, namely the riders and the drivers. To respect this real-time nature, it is necessary, on the one hand, to control the complexity related to the optimization of the trips, and on the other hand, to ensure that the concentration of data on a single system is avoided. Furthermore, it is important to consider the preferences of each user in order to optimize 3705 in a personalized way objectives such as waiting time, travel time or number of transfers. In this thesis, we focus on the various mechanisms and variants of this form of ridesharing. We advocate an individual-centric approach, in order to better respect the heterogeneous nature of the users, and a decentralized approach, to allow a better distribution of the data flow and thus a faster response time. We propose to combine multi-agent systems - whose nature is intrinsically individual-centric 3710 and decentralized - with different optimization algorithms in order to address the stated problems. The various results of our simulations attest to the performance and robustness of our approach, especially in complex situations such as the scarcity of ridesharing offers. The system proposed in this thesis allows to dynamically handle complex requests from riders while minimizing the impact of ride sharing for drivers, and this, for a wide spectrum of preferences and behaviors.

3715 **Mots-clefs en anglais :** Dynamic Ridesharing, Multi-agent systems, Optimization systems, Simulation, Mobility, Transport