



HAL
open science

Contributions à la résolution du transport à la demande fondées sur les systèmes multi-agents

Anas Malas

► **To cite this version:**

Anas Malas. Contributions à la résolution du transport à la demande fondées sur les systèmes multi-agents. Intelligence artificielle [cs.AI]. Normandie Université, 2017. Français. NNT : 2017NORMIR07 . tel-01661358v1

HAL Id: tel-01661358

<https://theses.hal.science/tel-01661358v1>

Submitted on 11 Dec 2017 (v1), last revised 13 Dec 2017 (v2)

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.



Normandie Université

THÈSE

Pour obtenir le diplôme de doctorat

Spécialité INFORMATIQUE

Préparée au sein de l'INSA de Rouen Normandie

Contributions à la résolution du transport à la demande fondées sur les systèmes multi-agents

**Présentée et soutenue par
Anas MALAS**

**Thèse soutenue publiquement le 18 mai 2017
devant le jury composé de**

POLLET Yann	Professeur titulaire de chaire Conservatoire National des Arts et Métiers	Rapporteur
ROUSSEAU Francis	Professeur des Universités Université de Reims	Rapporteur
BOUZID Maroua	Professeur des Universités Université de Caen	Membre du jury
CARDON Alain	Professeur des Universités Université Normandie	Membre du jury
EL FALOU Mohamad	Maître de Conférences Université Libano-Française	Membre du jury
EL FALOU Salah	Maître de Conférences Université Libanaise	Membre du jury
ITMI Mhamed	Maître de Conférences habilité à diriger des recherches Université Normandie	Directeur de thèse

Thèse dirigée par Mhamed ITMI, laboratoire d'Informatique, du Traitement de l'Information et des Systèmes (LITIS)

Remerciements

J'adresse mes remerciements tout d'abord à mon directeur de thèse M. Mhamed ITMI, maître de conférences HDR à l'INSA de Rouen Normandie. Je lui suis reconnaissante pour m'avoir accordé sa confiance. Il s'est toujours montré à l'écoute et très disponible tout au long de la réalisation de ce travail.

Je tiens à remercier M. Alain Cardon. Ses conseils précieux m'ont permis d'avancer et de perfectionner mon travail. J'ai particulièrement apprécié les discussions scientifiques enrichissantes avec lui et aboutissant à un travail réussi.

Je remercie aussi, M. Salah El Falou, maître de conférence à l'université Libanaise et M. Mohamad El Falou, maître de conférence à l'université de Technologies et de Sciences appliquées libano-française, qui m'ont guidée pendant ces trois années de thèse. Je leur suis reconnaissante pour le temps qu'ils m'ont consacré au Liban dans nos nombreuses discussions et pour l'orientation de mes recherches.

J'adresse mes sincères remerciements à M. Yann Pollet, Professeur titulaire de chaire au Conservatoire National des Arts et Métiers et à M. Francis Rousseaux professeur des universités à l'Université de Reims, pour l'intérêt qu'ils ont porté à mon travail en acceptant d'être rapporteurs de cette thèse.

Je tiens à remercier Mme Maroua Bouzid, professeur à l'université de Caen, pour m'avoir fait l'honneur de participer à mon jury.

Mes remerciements s'adressent aux membres du personnel du laboratoire LITIS pour leur accueil et leur aide.

Un grand merci à ma fille et mon mari. Je remercie également ma famille et plus particulièrement ma mère pour sa patience, son soutien et son encouragement.

Ce travail s'inscrit dans le Projet XTerm cofinancé par l'Union Européenne à travers le Fonds Européen de Développement Régional (FEDER) et par la Région Normandie. Il a aussi pu voir le jour grâce aux soutiens de l'association Lebanese Association for Scientific Research (LASeR) et de l'Agence Universitaire de la Francophonie (AUF).



Table des matières

Remerciements	ii
Table des matières	iv
Table des figures	x
Liste des tableaux	xii
Résumé	14
Abstract	15
Introduction Générale.....	16
1 Motivations.....	16
2 Structure du rapport.....	18
2.1 Chapitre 1. Auto-organisation et systèmes multi-agents	18
2.2 Chapitre 2. Le transport à la demande	18
2.3 Chapitre 3. Approche multi-agent décentralisée basée sur l’algorithme de A* pour le transport à la demande.....	19
2.4 Chapitre 4. Approche multi-agent massif pour le transport à la demande dans une ville	19
2.5 Chapitre 5. Approche multi-agent massif auto-organisé pour le transport à la demande dans une ville	20
Chapitre 1. Auto-organisation et systèmes multi-agents.....	21
1 Introduction	21
2 Un peu d’histoire.....	21
3 Agents : définition, types d’agent, propriétés d’environnement	23
3.1 Définition	23
3.2 Types d’agents	24
3.2.1 Agents réactifs.....	24
3.2.2 Agents cognitifs.....	24
3.2.3 Agents hybrides.....	25
3.3 Propriétés d’un environnement	25
4 Définition d’un système multi-agent.....	26
4.1 Interaction entre les agents.....	27
4.2 Objectif de l’interaction	28
5 Auto-organisation.....	29
5.1 Organisation.....	29

5.2	Propriétés d'un système auto-organisé	29
5.3	Mécanismes d'auto-organisation	31
5.3.1	Modification les liens entre les agents.....	31
5.3.2	Réorganisation physique des agents.....	31
6	Conclusion.....	32
Chapitre 2. Le transport à la demande.....		33
1	Introduction	33
2	TAD : définitions et appellations	35
3	TAD : variante du problème de tournées de véhicules	36
3.1	Problème du voyageur de commerce	37
3.2	Problème de tournées de véhicules	38
3.3	Variantes du problème de tournées de véhicules	39
3.3.1	Variante considérant la contrainte de capacité	39
3.3.2	Variante considérant la contrainte de précédence	40
3.3.3	Variante considérant la contrainte temporelle	40
3.3.4	Variante considérant des données dynamiques	41
3.3.5	Variante considérant des données stochastiques	42
3.3.6	Variante considérant des temps de voyage variables	42
3.3.7	Variante qui prend plusieurs contraintes à la fois.....	42
4	Résolution du problème du TAD	43
4.1	Critères d'optimisation du problème du TAD	44
5	Méthodes de résolution du VRP et de ses variantes.....	45
5.1	Méthodes exactes	46
5.2	Méthodes approchées.....	47
5.2.1	Heuristiques.....	47
5.2.2	Métaheuristiques	48
5.3	Approches multi-agents pour la résolution du problème du TAD.....	59
5.3.1	Approches centralisées	59
5.3.2	Approches décentralisées	63
6	Approches de résolution du VRP et de ses variantes considérant l'évolution du trafic 65	
6.1	Collecte, modélisation et stockage des données du trafic routier	67
6.1.1	Technologies de collecte des données du trafic routier	67
6.1.2	Modélisation et stockage des données du trafic routier.....	68
6.2	Approches considérant l'évolution déterministe du trafic	68

6.3	Approches considérant l'évolution stochastique du trafic	70
7	Plateformes et logiciels résolvant le TAD.....	73
7.1	Covoiturage.....	73
7.2	Mobirouter	74
7.3	LITRES-2.....	75
7.4	Uber.....	76
7.5	SAMPO.....	76
8	Conclusion.....	77
Chapitre 3. Approche multi-agent décentralisée basée sur l'algorithme de A* pour le transport à la demande.....		78
1	Introduction	78
2	Premier scénario du problème du TAD : <i>scénario statique</i>	78
3	Rappel de l'algorithme A*	79
3.1	Propriétés de A*	79
4	Structure de données des composants du système du TAD du <i>scénario statique</i>	80
4.1	Infrastructure statique	80
4.2	Demande de transport	81
4.3	Véhicule	82
4.4	Trajectoire	83
4.5	Problème du TAD	83
4.6	Objectif du système du TAD	84
5	Modélisation multi-agent du problème du TAD	85
5.1	Architecture générale	86
5.2	Types d'agents	87
6	Stratégie d'interaction entre les agents pour traiter une demande de transport	87
6.1	Coordination entre les agents Route et Véhicule.....	89
6.1.1	Phase d'initialisation	89
6.1.2	Phase de planification.....	91
6.1.3	Propriétés de l'algorithme A* décentralisé	97
6.2	Rôle de l'agent Véhicule.....	97
6.2.1	Phase d'initialisation	99
6.2.2	Phase de planification.....	100
7	Expérimentations et Résultats	103
7.1	Plate-forme d'expérimentation	103

7.2	Paramètres des expériences.....	105
7.2.1	Infrastructure et données virtuelles	106
7.2.2	Infrastructure et données réelles.....	108
7.3	Les résultats des expérimentations.....	109
7.3.1	Expérimentations de l'infrastructure virtuelle.....	109
7.3.2	Expérimentations de l'infrastructure réelle	113
7.3.3	Analyse.....	115
8	Conclusion.....	116
Chapitre 4. Approche multi-agent massif pour le transport à la demande dans une ville.....		117
1	Introduction	117
2	Deuxième scénario du problème du TAD : <i>scénario dynamique déterministe</i>	118
3	Algorithme général de la résolution du <i>scénario dynamique déterministe</i>	119
4	Structure de données des composants du système du TAD du <i>scénario dynamique déterministe</i>	120
4.1	Infrastructure dynamique déterministe	120
4.2	Demande de transport	123
4.3	Véhicule	123
4.4	Trajectoire	124
4.5	Problème du TAD	124
4.6	Objectif du système du TAD	125
5	Modélisation multi-agents massifs du problème du TAD	126
5.1	Pourquoi l'agentification massive du TAD ?.....	126
5.2	Architecture générale et types d'agents	126
6	Stratégie de coopération entre les agents pour traiter une demande de transport	128
6.1	Description des contraintes temporelles des agents Véhicule	128
6.2	Vérification des contraintes temporelles.....	129
6.3	Méthode d'évaluation d'une route	130
7	Stratégies de négociation entre les agents Véhicule.....	131
7.1	Première stratégie de négociation : " <i>envoyer aux suivants</i> "	132
7.2	Deuxième méthode de négociation : " <i>envoyer au suivant</i> "	134
7.3	Résultats théoriques et expérimentaux.....	136
7.3.1	Résultats théoriques.....	136
7.3.2	Fiabilité des stratégies de négociation.....	137
7.3.3	Résultats expérimentaux.....	138

8	Expérimentations.....	139
9	Conclusion.....	140
Chapitre 5. Approche multi-agent massif auto-organisé pour le transport à la demande dans une ville.....		
142		
1	Introduction	142
2	Troisième scénario du problème du TAD : <i>scénario dynamique stochastique</i>	144
3	Algorithme général de la résolution du scénario <i>dynamique stochastique</i>	144
4	Structure de données des composants du système du TAD	146
4.1	Infrastructure dynamique incertaine	146
4.1.1	Événement imprévu.....	146
4.1.2	Nœud	147
4.1.3	Route	147
4.1.4	Domaine géographique.....	150
4.2	Demande de transport	150
4.3	Véhicule	150
4.4	Trajectoire	150
4.5	Problème du TAD	151
4.6	Objectif du système du TAD	151
5	Modélisation multi-agent massifs du problème du TAD	151
5.1	Types et rôles d'agents.....	151
5.1.1	Agents statiques.....	151
5.1.2	Agents dynamiques	152
6	Exemple introductif simple	157
7	Traitement d'une demande de transport.....	162
7.1	Vérification des contraintes temporelles.....	163
7.2	Négociation entre les agents Véhicule	165
7.3	Méthode d'évaluation d'une route	165
7.4	Listes de chemins usuels	167
7.4.1	Construction des listes de chemins usuels.....	167
7.4.2	Renforcement d'un chemin	168
7.4.3	Affaiblissement d'un chemin	169
7.4.4	Mise à jour des chemins usuels	169
8	Auto-organisation des agents dynamiques pour traiter l'évolution du trafic	169
8.1	Traitement de l'évolution déterministe de la situation dynamique du trafic	170

8.1.1	Détermination des agents dynamiques et de leurs intervalles de temps courants ...	170
8.1.2	Comportement des agents dynamiques durant les intervalles de temps courants ...	171
8.1.3	Comportement des agents dynamiques à la fin des intervalles de temps courants .	171
8.2	Traitement de l'évolution stochastique de la situation dynamique de trafic	174
8.3	Opérations de décomposition et de fusion	175
8.3.1	Décomposition de l'agent Route	175
8.3.2	Fusion des agents Route	178
8.3.3	Décomposition d'un agent Régulateur	179
8.3.4	Fusion des agents Régulateur	180
9	Gestion des connaissances	180
10	Expérimentations	181
10.1	Paramètres des expérimentations	182
10.2	Expérimentations, résultats et analyses	182
10.2.1	Mesure du nombre de clients insatisfaits.....	182
10.2.2	Couverture du réseau routier du Tripoli par des agents Route et Régulateur.....	186
10.2.3	Mesure du nombre d'agents Régulateur contactés	188
11	Conclusion	189
Conclusion générale et perspectives		191
1	Conclusion.....	191
2	Perspectives.....	192
2.1	Enrichissement des demandes de transport.....	192
2.2	Correspondances avec d'autres véhicules du système.....	192
2.3	Utilisation des modes de transports publics.....	193
2.4	Application aux autres problèmes de transport.....	193
2.5	Régulation autonome du trafic routier	193
2.6	Gestion des cas d'urgences	193
Bibliographie.....		194

Table des figures

Figure 1 : Système multi-agent	27
Figure 2 : Représentation du TAD - les demandes	34
Figure 3 : Représentation du TAD - les tournées	34
Figure 4 : Exemple du problème TSP	38
Figure 5 : Exemple du problème de tournées de véhicules	38
Figure 6 : Exemple du problème de tournées de véhicules dynamique	41
Figure 7 : VRP origine du TAD	43
Figure 8 : Comportement des fourmis	58
Figure 9 : Représentation spatio-temporelle de l'environnement (Zargayouna 2005)	60
Figure 10 : La planification multicouche agent (Jin, Abdulrab, et Itmi 2008)	61
Figure 11 : Exemple d'infrastructure statique	80
Figure 12 : Architecture multi-agent du scénario statique	86
Figure 13 : Cycle de vie d'une demande de transport	88
Figure 14 : Calcul de l'heuristique de l'algorithme A* décentralisé	90
Figure 15 : Exploration de l'arbre de A* décentralisé	92
Figure 16 : Première phase de planification	93
Figure 17 : Deuxième phase de planification	94
Figure 18 : Troisième phase de planification	94
Figure 19 : Quatrième phase de planification	95
Figure 20 : Cinquième phase de planification	96
Figure 21 : Trajectoire optimale	96
Figure 22 : Bloc d'ordonnancement	98
Figure 23 : Les deux parties de Mason (Luke 2011)	103
Figure 24 : Triangulation de Delaunay	104
Figure 25 : Infrastructure avec 100 nœuds	106
Figure 26 : La carte de Tripoli	108
Figure 27 : Résultats de l'infrastructure100	112
Figure 28 : Résultats de l'infrastructure500	112
Figure 29 : Nombre de demandes servies	114
Figure 30 : Temps d'exécution des demandes	114
Figure 31 : Pourcentage de demandes servies et leurs temps d'exécution	115

Figure 32 : Infrastructure dynamique déterministe	121
Figure 33 : Architecture SMA du <i>scénario dynamique déterministe</i>	127
Figure 34 : Chercher une trajectoire entre deux lieux	130
Figure 35 : La méthode de négociation " <i>envoyer aux suivants</i> "	133
Figure 36 : La méthode de négociation " <i>envoyer au suivant</i> "	135
Figure 37 : Nombre de messages échangés en fonction du nombre des agents Véhicule	139
Figure 38 : Nombre de demandes servies et nombre de clients insatisfaits	140
Figure 39 : Interactions des agents pour traiter une demande de transport.....	156
Figure 40 : Interactions des agents pour refléter la situation réelle du trafic	156
Figure 41 : Exemple d'une infrastructure simple.....	157
Figure 42 : Agents Route couvrant l'infrastructure à 08h00.....	159
Figure 43 : Agents Régulateur couvrant l'infrastructure à 08h00.....	159
Figure 44 : Agents Route couvrant l'infrastructure à 09h00.....	160
Figure 45 : Chercher une trajectoire entre deux nœuds	164
Figure 46 : Décomposition d'un agent Route	176
Figure 47 : Fusion de deux agents Route	178
Figure 48 : Impact de l'accident#1 sur les clients	184
Figure 49 : Impact de l'accident#2 sur les clients	184
Figure 50 : Impact de l'accident#3 sur les clients	184
Figure 51 : Nombre de clients insatisfaits à cause de l'accident#4.....	185
Figure 52 : Nombre d'agents Route et Régulateur associés aux routes du Tripoli pendant une journée.....	188
Figure 53 : Variation du nombre d'agents Route et Régulateur suite à un accident.....	188
Figure 54 : Différence entre le nombre des agents Routes et Régulateur contactés	189

Liste des tableaux

Tab 1 : Matrice des distances en km	81
Tab 2 : Matrice des temps en minutes.....	81
Tab 3 : Exemple d'une table de planification	83
Tab 4 : Solutions des demandes	85
Tab 5 : La table de planification avant insertion de la demande.....	98
Tab 6 : La table de planification après insertion de la demande	99
Tab 7 : Traitement de demandes dans l'infrastructure100 virtuelle.....	110
Tab 8 : Traitement de demandes dans l'infrastructure500 virtuelle.....	111
Tab 9 : Traitement de demandes dans l'infrastructure réelle	113
Tab 10 : Exemple d'une liste des situations de trafic prévues	122
Tab 11 : Résultats théoriques de la comparaison	137
Tab 12 : Liste d'impact des événements imprévus d'une route R1	149
Tab 13 : Couleur et valeur de pondération des situations du trafic.....	155
Tab 14 : Représentation de la liste des situations de trafic prévues L1	158
Tab 15 : Représentation de la liste des situations de trafic prévues L4	158
Tab 16 : Représentation de la liste des situations de trafic prévues L5	158
Tab 17 : Représentation de la liste des situations de trafic prévues L8	158
Tab 18 : Agents Route associés aux routes de l'infrastructure à partir de 08h00	159
Tab 19 : Agents Régulateur associés aux routes de l'infrastructure à partir de 08h00	159
Tab 20 : Agents Route associés aux routes de l'infrastructure à partir de 09h00	160
Tab 21 : Agents Régulateur associés aux routes de l'infrastructure à partir de 09h00	160
Tab 22 : Agents Route associés aux routes de l'infrastructure à partir de 12h00	161
Tab 23 : Agents Régulateur associés aux routes de l'infrastructure à partir de 12h00	161
Tab 24 : Agents Route associés aux routes de l'infrastructure à partir de 14h00	161
Tab 25 : Agents Régulateur associés aux routes de l'infrastructure à partir de 14h00	161
Tab 26 : Agents Route associés aux routes de l'infrastructure à partir de 16h00	161
Tab 27 : Agents Régulateur associés aux routes de l'infrastructure à partir de 16h00	162
Tab 28 : Agents Route associés aux routes de l'infrastructure à partir de 20h00	162
Tab 29 : Agents Régulateur associés aux routes de l'infrastructure à partir de 20h00	162
Tab 30 : Exemple d'une liste des chemins usuels.....	167
Tab 31 : Liste urgente pour la route R1	175

Tab 32 : Impact des trois accidents sur les demandes de transport.....	183
Tab 33 : Impact de l'accident#4 sur les clients	185
Tab 34 : Nombre d'agents Route et Régulateur associés aux routes du Tripoli pendant une journée.....	187
Tab 35 : Comparaison entre le nombre d'agents Route et celui d'agents Régulateur	189

Résumé

Cette thèse porte sur le problème du transport à la demande (TAD). Nous proposons trois approches décentralisées basées sur les systèmes multi-agents pour résoudre ce problème. La première approche multi-agent utilise l'algorithme A* afin de trouver une solution optimale dans un réseau routier caractérisé par des vitesses de voyage constantes. Des expérimentations sont effectuées sur le réseau routier d'une ville libanaise appelée Tripoli et de bons résultats sont obtenus. Cependant, dans une ville comme Tripoli, les vitesses de voyage dépendent fortement de la situation dynamique du trafic routier. Pour cette raison, la deuxième approche multi-agent massif vient remédier à la première en tenant compte de l'évolution du trafic. Le réseau routier est considéré comme dynamique déterministe. Il se caractérise par des vitesses de voyages dépendantes de la situation habituelle du trafic. Ces vitesses sont pré-calculées en se basant sur des connaissances historiques du trafic routier. Les résultats expérimentaux montrent que le nombre de clients insatisfaits est supérieur à 50 % si les vitesses sont considérées comme constantes. Or, les connaissances historiques ne suffisent pas pour refléter la situation réelle du trafic surtout en cas d'apparition d'un événement imprévu tel qu'un accident sur le réseau. Pour cela, une troisième approche multi-agent massif auto-organisé est proposée. Le réseau routier est considéré comme dynamique stochastique caractérisé par des vitesses de voyage dépendantes de la situation réelle du trafic. Cette approche représente l'organisation dynamique du trafic à son échelle en se basant sur des connaissances historiques du trafic et sur des informations du trafic en temps réel. Les trajectoires des véhicules et leurs durées sont calculées et recalculées en ligne à chaque fois qu'un événement imprévu perturbe la situation habituelle du trafic. Les résultats expérimentaux montrent que jusqu'à 39 % des clients seront insatisfaits si un accident routier n'est pas considéré durant le traitement de leurs demandes. Autrement, 50 % à 100 % de ces clients sont satisfaits.

Mots-clés : auto-organisation, connaissances historiques du trafic, système multi-agent, système multi-agent massif, transport à la demande, vitesses de voyages dépendantes du temps.

Abstract

Title: Contributions to on-demand transport resolution based on multi-agents systems.

This thesis addresses the problem of on-demand transport (ODT). We propose three decentralized approaches based on multi-agent systems to solve this problem. The first multi-agent approach uses the algorithm A* in order to find an optimal solution in a road network characterized by constant travel speeds. Experiments are carried out on the road network of a Lebanese city called Tripoli and good results are obtained. However, in a city like Tripoli, travel speeds depend heavily on the dynamic situation of road traffic. For this reason, the second multi-agent approach comes to remedy the first taking into account the evolution of traffic. The road network is considered as dynamic deterministic. It is characterized by travel speeds dependent on the usual traffic situation. These speeds are pre-calculated on the basis of historical knowledge of road traffic. The experimental results show that the number of dissatisfied customers is greater than 50 % if the speeds are considered to be constant. Nevertheless, historical knowledge is not sufficient to reflect the actual traffic situation, especially in case of an unexpected event (such as an accident) occurring on the network. For this, a self-organized massive multi-agent approach is proposed. The road network is considered as a dynamic stochastic characterized by travel speeds dependent on the actual traffic situation. This approach represents the dynamic organization of traffic on its scale based on historical traffic knowledge and real-time traffic information. Vehicle trajectories and their durations are calculated and recalculated online whenever an unexpected event disrupts the usual traffic situation. The experimental results show that up to 39 % of customers will be dissatisfied if a road accident is not considered during the processing of their demands. Otherwise, 50 % to 100 % of these customers are satisfied.

Keywords: historical traffic knowledge, massive multi-agent system, multi-agent system, on-demand transport, self-organization, time-dependent travel speeds.

Introduction Générale

1 Motivations

La vie dans un environnement sans congestion, sans pollution et sans accidents représente les conditions idéales pour les personnes qui résident dans les zones urbaines. Or, le secteur du transport, et en particulier celui du transport routier, est une cause principale de ces trois nuisances citées précédemment. Dans l'Union européenne (UE), le secteur du transport a le taux de croissance de l'émission de gaz à effet de serre (GES) le plus élevé (Commission of the European Communities 2009). En France en 2007, le transport routier est responsable de plus du quart des émissions de gaz à effet de serre¹. Les émissions dues à ce secteur ont augmenté de 19 % entre 1990 et 2007 à cause de l'accroissement du trafic routier¹. Par ailleurs, le transport routier a causé 39.000 décès dans l'UE en 2008 (Commission of the European Communities 2009).

Le transport à la demande (TAD) se situe à mi-chemin entre le bus et le taxi. Le bus est un type de transport public, collectif et régulier. Il suit des plans temporels (des horaires), des plans spatiaux (des itinéraires) et des points d'arrêts fixes et déterminés à l'avance. Pour cela, il présente des carences selon les dimensions espace et temps. Il ne dessert pas toutes les zones géographiques et il n'est pas disponible tout le temps dans celles qu'il couvre. De plus, durant les heures de pointe, ce type de transport se trouve tellement plein alors que les passagers attendent sur les trottoirs. De même, des gens, comme ceux qui sont à mobilité réduite, ont besoin d'être transportés de porte à porte avec des coûts relativement faibles, etc. Par contre, le taxi est plus flexible. Il suit des plans qui conviennent aux passagers. Cependant, il coûte très cher car il est privé. Un véhicule est réservé pour servir une demande de transport. Pour toutes ces raisons, le TAD a vu le jour comme une alternative qui comble les anomalies des réseaux publics et qui coûte moins cher que le taxi.

Le TAD est un mode de transport collectif, flexible et déclenché par les demandes des clients. Il ne suit ni des horaires, ni des itinéraires, ni des points d'arrêts fixes. Au contraire, ils sont déterminés selon les demandes des clients. Le TAD dans une ville est un problème essentiel dans la logistique urbaine et il a été récemment étudié. Ce problème est caractérisé par la

¹<http://www.statistiques.developpement-durable.gouv.fr/lessentiel/ar/361/1228/emissions-gaz-effet-serre-transports.html>

variation de la situation du trafic routier tout au long d'une journée. Cette variation rend la vitesse de circulation sur les routes de la ville dépendante fortement du temps et du jour de voyage.

Le TAD se développe partout dans le monde car il offre des avantages sociaux, économiques et environnementaux importants individuels et collectifs. A titre d'exemple, il peut servir des zones non desservies par les moyens traditionnels du transport public. Il propose des coûts moins chers que le taxi car il transporte plusieurs personnes par véhicule sur le même trajet. De même, il participe à réduire la congestion routière, les accidents de la route, la pollution (comme l'émission de gaz à effet de serre) car le nombre de véhicules circulants dans les routes diminuent par l'augmentation du nombre des passagers par véhicule.

Si le TAD présente des aspects économiques, sociaux et environnementaux positifs indéniables, son acceptabilité par la population repose sur d'autres facteurs. La réactivité du service à la demande des clients fait partie des paramètres techniques à considérer. La réduction de la durée de la course et le respect des fenêtres des temps pour tout client sont des paramètres toujours recherchés pour améliorer la qualité de service. En favorisant la disponibilité du TAD, nous en augmentons aussi l'acceptabilité par la population, etc. La clé de son acceptation est donc sa réactivité d'offrir des solutions justes, rapides, fiables, respectant les contraintes temporelles des clients et des véhicules, améliorant la qualité de service et reflétant les conditions de la vie réelle (la situation du trafic routier par exemple).

Dans cette thèse, nous nous intéressons à la construction d'un système traitant le problème du TAD dans une ville en temps réel en se basant sur les systèmes multi-agents. Ce système a comme rôle de trouver, pour chaque demande de transport, un véhicule qui transporte les passagers, la trajectoire menée par le véhicule et sa durée du parcours. Cela en respectant des contraintes exigées par les clients et les véhicules (comme les fenêtres de temps des clients, le nombre de places vides des véhicules, etc.), en tenant compte de la situation du trafic en temps réel et en optimisant le nombre de véhicules en circulation et la distance totale parcourue. C'est un système qui se base sur des connaissances historiques du trafic et sur des informations du trafic en temps réel pour représenter l'organisation dynamique du trafic dans la ville à son échelle. De cette manière, le système sera capable d'estimer les durées exactes des trajectoires des véhicules, de construire des solutions fiables et d'éviter des heures de travail supplémentaires pour les conducteurs et le non-respect des fenêtres de temps des demandes des clients. La qualité de service est ainsi optimisée et les clients satisfaits.

Les systèmes multi-agents (SMA) sont modernes et prometteurs dans la modélisation et la résolution des problèmes complexes dynamiques stochastiques comme celui du TAD dans une ville. Ils ont montré un succès dans leur application sur le problème dynamique de tournées de véhicules dont le problème du TAD est l'extension. Le problème de tournées de véhicules et ses variantes sont des problèmes d'optimisation combinatoire NP-difficile (Toth et Vigo 2002). Les méthodes exactes ne traitent que des versions de petite taille de ces problèmes et ne représentant pas des applications réelles. Par contre, les SMA distribuent l'espace de recherche et la charge de calcul sur plusieurs agents autonomes évoluant dans un environnement et interagissant les uns avec les autres. Ce qui permet d'éviter le problème d'explosion de l'espace de recherche. Le modèle le plus convenable pour le problème du TAD est donc celui des systèmes multi-agents. La considération d'un point de vue multi-agent permet d'imaginer de nouvelles mesures et de nouvelles manières de résolution non envisagées par les approches centralisées.

2 Structure du rapport

Le manuscrit est composé de cinq chapitres. Les deux premiers chapitres présentent l'état de l'art des deux domaines sur lesquelles nous travaillons, à savoir : les SMA et l'auto-organisation d'une part, et le transport à la demande de l'autre. Les chapitres 3 à 5 détaillent nos contributions à ces trois domaines.

2.1 Chapitre 1. Auto-organisation et systèmes multi-agents

Dans ce premier chapitre, nous présentons les notions de base des systèmes multi-agents et de l'auto-organisation pour le positionnement et la compréhension de notre travail. Notre présentation n'a pas la prétention d'être exhaustive, mais donne les notions de base d'agents, de l'environnement, d'interaction, d'organisation et d'auto-organisation. L'auto-organisation est une propriété intéressante des systèmes intelligents distribués. Elle suppose que le système est capable de s'organiser sans intervention extérieure.

2.2 Chapitre 2. Le transport à la demande

Le deuxième chapitre donne un état de l'art du problème du transport à la Demande (TAD) et plus généralement du problème de tournées de véhicules qui est son origine. Nous commençons par présenter le problème du TAD et les différentes appellations utilisées pour le designer. Puis, nous expliquons le problème origine du TAD qui est le problème de tournées de véhicules. Nous décrivons les différentes approches de résolution de ces problèmes en

différenciant entre celles qui ont pris la situation du trafic en compte et celles qui ne l'ont pas considérée. Nous exposons finalement certains logiciels et plateformes consacrés au problème du TAD.

2.3 Chapitre 3. Approche multi-agent décentralisée basée sur l'algorithme de A* pour le transport à la demande

Dans ce chapitre, nous proposons une approche multi-agent décentralisée pour modéliser et résoudre le problème du TAD dans un réseau routier statique (El Falou et al. 2014b). Ce réseau routier se caractérise par des vitesses de voyage constantes et indépendantes du temps. Les agents interagissent entre eux en se basant sur l'algorithme A* afin de trouver la solution optimale à chaque demande de transport. Cette solution est exprimée par la trajectoire la plus rapide menée par les véhicules les moins coûteux. Le nombre de changements de véhicules ne doit pas dépasser une limite fixée par le client.

2.4 Chapitre 4. Approche multi-agent massif pour le transport à la demande dans une ville

Dans le chapitre 4, nous proposons une approche multi-agent massif pour résoudre le problème du TAD dans une ville avec un réseau routier dynamique déterministe (Malas et al. 2016a). Ce réseau est caractérisé par des vitesses de voyage dépendantes de la situation habituelle du trafic routier. L'approche se base sur des connaissances historiques du trafic de la ville pour calculer les vitesses de voyage et chercher une solution approchée pour chaque demande de transport. La solution est exprimée par une trajectoire menée par un véhicule. Elle respecte les contraintes temporelles des clients et optimise le nombre des véhicules en circulation et la distance totale parcourue.

De plus, nous proposons et implémentons deux méthodes de négociation entre les agents afin de choisir le véhicule transportant les passagers (Malas et al. 2016b). Ces méthodes de négociation sont comparées avec une troisième proposée dans la littérature. Nous proposons une étude comparative de ces trois méthodes et nous en donnons les avantages et les inconvénients sur des critères étudiés théoriquement et expérimentalement. Ces critères sont le nombre de messages échangés entre les agents, la tolérance aux pannes et la complexité. Une de ces méthodes est plus efficace que les autres, pour cela, nous la retenons dans la suite de cette thèse.

2.5 Chapitre 5. Approche multi-agent massif auto-organisé pour le transport à la demande dans une ville

Dans ce chapitre, nous examinons le problème du TAD dans un réseau routier dynamique stochastique caractérisé par des vitesses de voyage dépendantes de la situation réelle du trafic. Des événements imprévus (comme un accident) peuvent avoir lieu sur le réseau et perturber la situation habituelle du trafic. Dans ce cas, les connaissances historiques du trafic ne suffisent pas pour refléter la situation réelle du trafic. Pour cette raison, nous étendons le modèle multi-agent massif du chapitre 4 pour être capable de prédire la perturbation de la situation habituelle du trafic en cas d'apparition d'un événement imprévu sur l'une des routes. Le modèle étendu se base sur des connaissances historiques du trafic et sur des informations du trafic en temps réel pour représenter l'organisation dynamique du trafic dans la ville à son échelle. Des agents dynamiques sont associés aux routes de la ville selon le trafic. Ces agents s'auto-organisent spatio-temporellement en ligne afin de refléter la situation réelle du trafic et calculer les vitesses de voyage. Ils adaptent en plus les trajectoires des véhicules et leurs durées si nécessaire. De cette manière, le système sera capable de donner une solution basée sur des informations fiables.

Chapitre 1. Auto-organisation et systèmes multi-agents

1 Introduction

Les systèmes multi-agents (SMA) sont l'un des thèmes marquants des années 90 dans la recherche scientifique. Ils sont utilisés comme base pour résoudre plusieurs problèmes dans différentes disciplines comme la communication (le commerce électronique), la télécommunication (la gestion de réseaux), l'industrie (les robots coopératifs), l'information (l'assistance personnelle), la santé (la supervision des malades), le transport (la logistique, l'information du voyage), etc.

Les systèmes multi-agents sont en connexion avec les systèmes répartis, l'intelligence artificielle et le génie logiciel. Ils se basent sur les comportements collectifs produits par les interactions de plusieurs entités autonomes appelées agents afin de résoudre une tâche globale particulière. Dans cette thèse, nous nous intéressons à résoudre un problème du transport routier qui est le problème du transport à la demande en se basant sur les deux notions SMA et auto-organisation. A cette fin, nous proposons dans les chapitres 3, 4 et 5 des modèles multi-agents et des modèles d'auto-organisation pour la résolution de ce problème complexe.

Ce chapitre a pour objectif de rappeler les notions de base des systèmes multi-agents et de l'auto-organisation. Nous présentons tout d'abord dans la section 2 un survol historique des systèmes multi-agents. Puis, nous définissons dans la section 3 la notion d'agent et ses différents types. Les systèmes multi-agents sont présentés dans la section 4 en détaillant les modes et les objectifs d'interaction entre les agents. La section 5 est consacrée à la présentation des concepts d'auto-organisation. Nous concluons ce chapitre dans la section 6.

2 Un peu d'histoire

L'Intelligence Artificielle (IA) est apparue en 1956. C'est le nom donné à l'intelligence des machines et des logiciels. Marvin Lee Minsky, l'un des fondateurs de cette discipline, la définit par: *“the building of computer programs which perform tasks which are, for the moment, performed in a more satisfactory way by humans because they require high level*

*mental processes such as: perception learning, memory organization and critical reasoning*².

L'IA est donc la science qui vise à réaliser par une machine ou un logiciel des tâches accomplies par l'homme en utilisant son intelligence. Les réalisations les plus célèbres de l'IA centralisée sont en robotique, en systèmes d'aide au diagnostic et en systèmes d'aide à la décision.

A cause du progrès de la technologie internet et des réseaux et le besoin croissant de systèmes intelligents dans tous les domaines comme l'industrie, les jeux vidéo, la bureautique, etc., l'IA centralisée devient moins performante en niveau de possibilités et de fiabilité et plus coûteuse au sens de conception et maintenance. De surcroît, des limites conceptuelles et matérielles ont confronté l'IA classique dans la résolution de certains problèmes complexes et distribués (comme le transport, le contrôle du trafic routier, le contrôle aérien, etc.). La décentralisation des connaissances et du contrôle et des efforts collectifs de plusieurs entités mettant ensemble leurs connaissances et leurs compétences, sont exigés pour résoudre de tels problèmes. Pour cette raison, l'idée de l'Intelligence Artificielle Distribuée (IAD) est née pour surmonter les inconvénients de l'IA classique centralisée. L'IAD décompose un problème en des sous-problèmes, chacun étant attribué à un solveur pour le résoudre. Ensuite, les solveurs échangent entre eux des informations pour résoudre le problème global. L'IAD s'intéresse à des comportements intelligents qui sont le produit de l'interaction coopérative de plusieurs entités. Au début des années 80, l'IAD a introduit le concept de système multi-agent (SMA) qui porte sur le modèle de l'agent dont la principale caractéristique est l'autonomie. Les SMA ont un champ d'application beaucoup plus diversifié que les IADs. Trois domaines d'applications des SMA sont définis dans (Boissier, Gitton, et Glize 2004) :

- la simulation des systèmes réels complexes afin de modéliser, expliquer, observer ou prédire leurs comportements ;
- les applications dans lesquelles les agents jouent le rôle d'êtres humains. La notion d'agent simplifie la conception de ces systèmes ;
- la résolution de problèmes dans le domaine de l'intelligence artificielle distribuée.

² Marvin Lee Minsky, citation tirée de la thèse de doctorat de (Hiard 2013)

3 Agents : définition, types d'agent, propriétés d'environnement

Cette section est consacrée à la définition d'un agent, à la présentation des différents types d'agent et à la présentation des propriétés d'un environnement.

3.1 Définition

Le terme agent a eu de nombreuses définitions à cause de la diversité de contexte et d'applications pour lesquelles l'agent est conçu. Nous présentons dans ce qui suit quelques définitions importantes.

Dans (Ferber 1995), un agent est une entité physique ou virtuelle :

- qui est capable d'agir dans un environnement ;
- qui peut communiquer directement avec d'autres agents ;
- qui est mue par un ensemble de tendances (sous la forme d'objectifs individuels ou d'une fonction de satisfaction, voire de survie, qu'elle cherche à optimiser) ;
- qui possède des ressources propres ;
- qui est capable de percevoir (mais de manière limitée) son environnement ;
- qui ne dispose que d'une représentation partielle de cet environnement (et éventuellement aucune) ;
- qui possède des compétences et offre des services ;
- qui peut éventuellement se reproduire ;
- dont le comportement tend à satisfaire ses objectifs, en tenant compte des ressources et des compétences dont elle dispose, et en fonction de sa perception, de ses représentations et des communications qu'elle reçoit.

Par ailleurs, (Jennings, Sycara, et Wooldridge 1998) ont fourni une définition plus courte qui stipule qu'un agent est un système informatique, situé dans un environnement, et qui agit de façon autonome et flexible pour atteindre les objectifs pour lesquels il a été conçu.

Une définition récente proposée par (Cardon et Itmi 2016) disant qu'un agent est :

- une entité intelligente agissant de façon rationnelle et intentionnelle par rapport à ses buts et l'état courant de ses connaissances ;
- une entité de haut niveau, asservie, agissant de façon continue et autonome dans un environnement où des processus prennent place et où d'autres agents existent.

Nous concluons de ces trois définitions qu'un agent est une entité plongée dans un environnement, possédant des accointances (des relations avec les autres agents), autonome en sens qu'elle est capable d'agir sans sollicitation externe et réactive en sens qu'elle est capable de percevoir son environnement et de réagir en fonction de ses perceptions et ses connaissances afin d'accomplir ses objectifs.

3.2 Types d'agents

Selon les capacités et les propriétés d'un agent, il peut être classé en trois familles : les agents réactifs, les agents cognitifs ou les agents hybrides.

3.2.1 Agents réactifs

La structure d'un agent réactif est simple. Ses actions sont des types stimuli/réponse et elles sont déclenchées par des événements (stimuli) qu'ils perçoivent de l'environnement. Il ne possède pas une représentation complète de son environnement et il ne prend pas en considération ses actions passées. En outre, il a une faible capacité de raisonnement, de communication et de planification. Il possède des règles qu'il suit et il est comme un récepteur. (Drogoul 1993) a utilisé un système réactif en simulation pour la reproduction du fonctionnement d'une colonie de fourmis où chaque fourmi est représentée par un agent réactif.

3.2.2 Agents cognitifs

L'agent cognitif est un agent intelligent qui possède une représentation explicite de son environnement, des autres agents et de lui-même. Egalement, il mémorise ses actions passées et il est doté de capacités de raisonnement, de planification, de communication et de prise de décisions afin d'accomplir ses buts explicites.

Le travail de (Rao et Georgeff 1995) basé sur l'architecture BDI (Believe, Desire, Intention) représente cette catégorie d'agents. Dans cette architecture, les agents ont les attitudes mentales suivantes :

- *les croyances* : ce sont les informations que l'agent connaît de son environnement et interprète ;
- *les désirs* : ce sont les buts à réaliser par l'agent ;
- *les intentions* : ce sont les projets de l'agent pour satisfaire ses désirs.

Dans la littérature, plusieurs architectures d'agents cognitifs ont été inspirées du modèle BDI. Nous citons : *Procedural Reasoning System : PRS* (Georgeff et Lansky 1987), *Distributed*

Multi-Agent Reasoning System : dMARS (Singh, Rao, et Wooldridge 1998) et *Intelligent Resource-bounded Machine Architecture : IRMA* (Bratman 1987).

3.2.3 Agents hybrides

L'agent hybride est un compromis entre la capacité de raisonnement des agents cognitifs et la rapidité de réponse des agents réactifs. Généralement, la structure d'un agent hybride est composée de trois couches. La couche la plus basse est une couche réactive qui prend ses décisions en fonction du traitement des données provenant de capteurs sensoriels, une couche intermédiaire qui raisonne sur les connaissances de l'agent de son environnement et enfin, une couche supérieure qui raisonne en prenant en compte l'interaction avec les autres agents du système.

De nombreuses architectures illustrent ce type d'agent. Nous citons : l'architecture ASIC développée par (Boissier et Demazeau 1994) et utilisée pour le traitement numérique d'images, l'architecture de (Ocelllo et Demazeau 1998) et l'architecture ASTRO développée par (Ocelllo et al. 2002) utilisées dans les systèmes multi-agents soumis à des contraintes temporelles.

3.3 Propriétés d'un environnement

L'environnement est le lieu de situation de l'agent où il perçoit de l'information et exerce ses activités. Les propriétés de l'environnement influent intrinsèquement sur le processus de décision des agents. Ces propriétés sont :

- déterministe : un environnement est dit déterministe lorsque l'état futur ne dépend que de son état courant et des actions que les agents réalisent sur lui. L'environnement réel est indéterministe car son état ne dépend pas seulement des actions des agents ;
- statique : un environnement est qualifié statique si son état ne change que lorsqu'un agent exerce une action. Si l'environnement est dynamique, alors l'état de l'environnement perçu par l'agent peut évoluer pendant son raisonnement et l'action que l'agent a l'intention d'effectuer n'aura plus les conséquences désirées ;
- discret : un environnement est discret si le nombre d'actions que l'agent peut effectuer est fini. Sinon, il est qualifié de continu ;
- totalement observable : un environnement est dit totalement observable si l'agent perçoit l'état complet et toutes les informations concernant son environnement. Généralement, ce n'est pas le cas, car un agent perçoit une partie de son

environnement par ses capteurs limités. Dans ce cas, il est dit partiellement observable.

4 Définition d'un système multi-agent

Dans cette section, nous nous intéressons à la présentation des systèmes multi-agents qui seront utilisés ultérieurement dans la modélisation du problème du transport à la demande.

D'après (Ferber 1995), un système multi-agent est un système composé des éléments suivants :

- un environnement, c'est-à-dire un espace disposant généralement d'une métrique ;
- un ensemble d'agents qui représentent les entités actives du système ;
- un ensemble d'objets situés dans l'environnement. Ces objets sont passifs, c'est-à-dire qu'ils peuvent être perçus, créés, détruits et modifiés par les agents ;
- un ensemble de relations qui unissent des objets (et donc des agents) entre eux ;
- un ensemble d'opérations permettant aux agents de percevoir, produire, consommer, transformer et manipuler les objets ;
- un ensemble d'opérateurs chargés de représenter l'application de ces opérations et la réaction du monde à cette tentative de modification, que l'on appellera les lois de l'univers.

Un système multi-agent peut se définir comme un ensemble d'agents en interaction, capable de s'organiser de manière dynamique et adaptative (Briot 2001).

Ainsi, un SMA est un système distribué composé d'un ensemble d'agents en interaction dans un environnement commun, agissant afin de résoudre une tâche commune et cohérente. En fait, chaque agent a ses propres connaissances, une capacité limitée et des informations incomplètes sur son environnement et il possède un but individuel. Pour cela, il est incapable de résoudre seul le problème et il peut entrer en conflit avec les autres agents.

(Cardon, et Itmi 2016) ont ajouté qu'un système sera dit agentifié lorsque toutes ses entités actives sont des agents. L'agentification consiste alors à transformer toutes les connaissances et les relations entre les connaissances en agents.

La figure 1 représente un système multi-agent où les agents sont situés dans un environnement et interagissent entre eux.

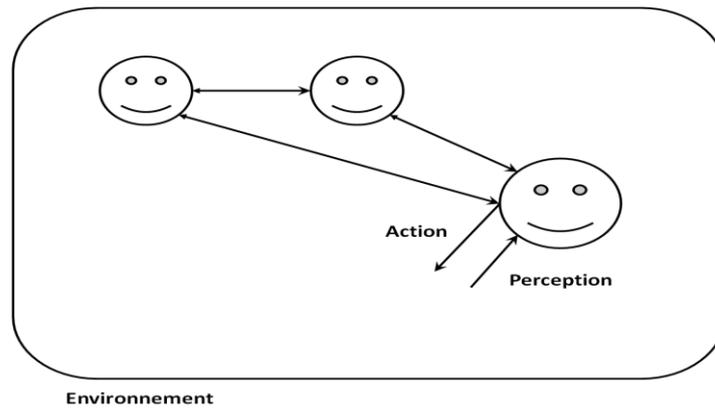


Figure 1 : Système multi-agent

4.1 Interaction entre les agents

L'interaction entre les agents est une partie importante et essentielle dans un système multi-agent. Elle symbolise tout type d'échange d'informations entre les agents. Elle permet aux agents de communiquer, de coordonner, de négocier entre eux et d'agir sur leur environnement pour permettre au système de fonctionner et d'évoluer efficacement afin de réaliser un but particulier.

D'après (Ferber 1995), une interaction est la mise en relation dynamique de deux ou plusieurs agents par le biais d'un ensemble d'actions réciproques. Les interactions sont non seulement la conséquence d'actions effectuées par plusieurs agents en même temps, mais aussi l'élément nécessaire à la constitution d'organisations sociales.

Il existe deux principaux modes d'interaction entre les agents : l'interaction directe et l'interaction indirecte. Dans l'interaction directe, les agents communiquent directement et explicitement via l'échange de messages entre eux. Un agent peut envoyer un message à un seul destinataire ; c'est une communication de point à point. Egalement, il peut envoyer un message vers tous les autres agents du système ; c'est une communication par broadcast. Plusieurs travaux de recherche se sont intéressés à ce type d'interaction et ont développé des langages de communication d'agents ou ACL (Agent Communication Languages) comme KQML (Finin et al. 1994) et FIPA-ACL³.

Dans l'interaction indirecte, les agents partagent les informations via l'environnement (Balbo 2000; Zargayouna 2007) ou via un tableau noir (Hayes-Roth 1985). Le mécanisme de *stigmergie*, introduit par (Grassé 1959), permet aux agents d'interagir de façon indirecte via

³ <http://www.fipa.org>

l'environnement. En effet, les agents modifient leur environnement physique pour faire passer des informations aux autres agents. Dans cette technique, les agents laissent des traces dans leur environnement. Ces traces sont utilisées par les autres agents afin de guider leurs actions. (Hadeli et al. 2003; Hadeli et al. 2004; O'Reilly et Ehlers 2006) utilisent la *stigmergie* comme outil de communication indirecte entre les agents. L'approche alternative de *stigmergie* qui permet aux agents d'échanger leur information indirectement est le tableau noir. Un agent vient chercher les informations dont il a besoin du tableau noir ou vient en déposer pour les mettre à la disposition des autres agents. Les travaux de (Botti et al. 1995) et (Chen et Occeña 2000) illustrent des exemples de la communication indirecte via un tableau noir. En général, un agent réactif communique de façon indirecte (Brooks 1991).

4.2 Objectif de l'interaction

L'objectif de l'interaction entre les agents peut varier de la coordination, à la coopération voire la négociation.

La coordination

Lorsque l'action d'un agent, son but ou même les ressources qu'il utilise sont interdépendants avec ceux des autres agents, le processus de coordination sera nécessaire (Ciancarini 1996). A cette étape, les actions individuelles de chaque agent doivent être coordonnées et articulées de manière à éviter des problèmes dans le comportement global du système et afin que l'ensemble aboutisse à réaliser un travail cohérent. Les protocoles de coordination aident les agents à gérer leurs engagements, à synchroniser leurs activités et à régler les conflits qui existent entre eux. Le protocole de coordination le plus connu dans les systèmes multi-agents est le *Contract Net Protocol* (Smith 1980). Ce dernier décrit un protocole où les agents coordonnent leurs activités en établissant des contrats afin d'atteindre leurs buts.

La négociation

Dans les systèmes multi-agents, un agent entre en conflit avec un second pour plusieurs raisons comme l'accès à des ressources, la proposition de plusieurs solutions différentes pour le même problème, etc. Dans ce cas, la négociation intervient afin que les agents résolvent le conflit, prennent une décision commune et parviennent à un accord acceptable par tous. (Bussmann et Muller 1992) définissent la négociation comme un processus de communication d'un groupe d'agents ayant comme but d'atteindre un accord mutuellement accepté sur un certain sujet.

La coopération

Elle se traduit par un ensemble d'agents qui travaillent ensemble afin de réaliser un but commun. En effet, un agent a besoin de l'aide des autres agents du système car il ne peut pas réaliser son but individuellement. La coopération consiste à répartir des tâches, des ressources et des informations entre les différents agents afin de s'entraider pour réaliser un objectif commun.

5 Auto-organisation

Dans cette section, nous allons présenter les notions d'organisation et d'auto-organisation qui seront utilisées ultérieurement pour la résolution du problème du transport à la demande (TAD).

5.1 Organisation

Une organisation est composée d'un ensemble d'entités qui communiquent pour faciliter les échanges entre eux dans le but de répondre à un objectif donné. Elle peut être considérée comme une structure décrivant les interactions et les relations qui existent entre ces entités (Fox 1981). En général, l'organisation au niveau d'un système multi-agent correspond à la structure du système et aux liens d'échange d'informations entre les agents. C'est un arrangement des agents et de leurs comportements conditionné par les contraintes imposées par l'environnement (Ferber, Stratulat, et Tranier 2009).

L'auto-organisation est un processus où un système modifie son organisation interne sans aucune intervention externe pour s'adapter aux changements de l'environnement (Di Marzo Serugendo, Gleizes, et Karageorgos 2005).

D'ailleurs, (Ishida, Gasser, et Yokoo 1992) définissent l'auto-organisation comme un processus permettant à une organisation d'agents de s'adapter de manière autonome aux situations dynamiquement changeantes. En particulier, l'organisation du SMA doit évoluer à travers le temps quand les circonstances l'exigent. Un SMA est auto-organisé lorsqu'il modifie lui-même son organisation, de manière autonome, pour s'adapter aux changements de son environnement.

5.2 Propriétés d'un système auto-organisé

Afin qu'un système soit auto-organisé, il doit satisfaire à certaines propriétés. Ces propriétés peuvent être classifiées en propriétés obligatoires et propriétés optionnelles (Zeddini 2009).

Les propriétés obligatoires du système auto-organisé sont :

- l'absence de contrôle externe. Le système doit s'organiser en se basant uniquement sur ses décisions internes et sans intervention extérieure ;
- le fonctionnement en mode dynamique. Le système fonctionne en mode dynamique c'est-à-dire qu'il évolue avec le temps. Si l'organisation du système évolue indépendamment de tout contrôle externe, alors l'auto-organisation est toujours exigée.

Concernant les propriétés optionnelles, nous citons :

- le contrôle décentralisé : un système auto-organisé peut travailler sous contrôle centralisé et décentralisé ;
- la dissipation : un système se stabilise dans un certain état lorsqu'il n'y a pas des perturbations externes. C'est ce que nous appelons la dissipation d'une certaine énergie. Sinon, le système change perpétuellement ;
- l'instabilité : lorsque le système fonctionne en mode dynamique non linéaire, il en résulte des modifications significatives dans son comportement global. Nous disons alors que le système est instable ;
- les équilibres multiples : lorsque plusieurs attracteurs possibles pour des états stables se présentent dans le système, des équilibres multiples seront observés ;
- la redondance : quand le système est indifférent aux dommages du fait que ses composants sont répliqués, il possède la caractéristique de redondance ;
- l'auto-maintenance : lorsque le système est capable de se réparer ou de se reproduire, il est dit un système auto-entenu ;
- l'adaptation : lorsque le système a la capacité de réorganiser, cela implique qu'il s'adapte aux changements externes ;
- la complexité : le système auto-organisé est complexe lorsqu'il existe plusieurs valeurs ou objectifs en parallèle. Ainsi, il est impossible de faire combiner plusieurs comportements locaux en un comportement global ;
- la hiérarchie : un système auto-organisé est hiérarchique lorsque plusieurs niveaux d'auto-organisation sont présents ;
- les règles locales simples : des règles de comportement individuel local peuvent être la base de construction d'un comportement global complexe.

5.3 Mécanismes d'auto-organisation

Plusieurs mécanismes d'auto-organisation ont été proposés dans la littérature. Dans les sous-sections suivantes, nous décrivons le mécanisme de modification des liens entre les agents et celui de réorganisation physique des agents.

5.3.1 Modification des liens entre les agents

Un des mécanismes d'auto-organisation consiste à modifier les accointances entre les agents. Lorsque les agents se communiquent, ils acquièrent de nouvelles connaissances, évoluent leurs compétences et découvrent les autres agents. Dans la méthode de modification des liens entre les agents proposés par (Camps et Gleizes 1995; Camps et Gleizes 1996), un agent ne cherche une information que des agents susceptibles de la lui fournir. Lorsqu'il possède une information, il la communique avec les agents intéressés par cette information. De surcroît, une méthode d'évaluation de la pertinence des agents est proposée. Un agent évalue la pertinence des autres suite à une demande d'information. Cette évaluation s'effectue en déterminant le degré de similitude entre la demande et chacun de ces agents. Les agents ayant le degré de similitude supérieur à un seuil défini sont jugés pertinents.

Les accointances d'un agent lui donnent des informations sur son interlocuteur comme la nature de sa relation avec lui, les capacités motivant la relation ainsi que sa préférence : attention ou confiance (Foisel, Chevrier, et Haton 1996). La préférence permet à l'agent de privilégier certains interlocuteurs en leur accordant plus ou moins d'attention ou de confiance et cela selon leurs capacités. Ces auteurs proposent un modèle d'auto-organisation fondé sur la variation de la préférence des agents. L'agent varie ses préférences en fonction des interactions qu'il entretient avec les autres agents. Si ces interactions sont bénéfiques, alors la préférence est renforcée. Sinon, elle est affaiblie.

5.3.2 Réorganisation physique des agents

Un modèle descriptif des interactions entre les composants du système et de leurs évolutions dans le temps est proposé dans (Bourdon 1997; Bourdon 1998). L'espace des relations est représenté par un graphe. Chaque nœud du graphe symbolise un composant du système (processus, serveur, objet, etc.) dont le fonctionnement est caractérisé par l'observation de ses interactions ; chaque arc reliant deux nœuds concrétise les interactions passées et en cours entre ces deux nœuds. C'est un modèle descriptif fondé sur une auto-observation continue du fonctionnement du système et préventif servant de patron pour l'auto-organisation collective du système conformément à ces changements. Pour tester ce modèle, des simulations ont été

proposées. Les résultats de l'expérimentation montrent que le système converge vers un état d'équilibre caractérisé par la satisfaction des besoins de tous les clients et la minimisation des liens d'interaction. Une perturbation, comme l'ajout d'un client, génère une réorganisation spatiale des serveurs pour atteindre un nouvel état d'équilibre.

Dans (Unsal et Bay 1994), une grande population de robots doit trouver un certain nombre d'objets et les ramener à un dépôt. Les robots sont sensibles aux signaux lumineux envoyés par les objets. Un robot est incapable de transporter un objet seul. Il doit coordonner ses actions avec celles des autres robots pour pouvoir ensemble la soulever et la transporter. L'approche est fondée sur un mécanisme d'attraction/répulsion. Le comportement des robots est guidé par des règles simples qui conditionnent leurs déplacements en fonction de la localisation des objets et de leurs voisins les plus proches. Le mécanisme proposé permet aux robots de constituer, sans le savoir, des formes géométriques autour des objets.

6 Conclusion

Dans ce chapitre, nous avons présenté les notions de base des systèmes multi-agents (SMA) et de l'auto-organisation. L'importance de ces deux concepts réside dans les approches que nous développons dans cette thèse. En effet, le paradigme multi-agent est utilisé comme outil de base pour les approches que nous proposons et l'auto-organisation est essentielle puisque nous présentons un système multi-agent auto-organisé pour résoudre le problème du transport à la demande dans une ville. Le chapitre suivant définit le problème du transport à la demande. Ce problème distribué naturellement se prête bien à une modélisation multi-agent et il sera l'application de nos propositions.

Chapitre 2. Le transport à la demande

1 Introduction

Le transport à la demande (TAD) se situe à mi-chemin entre les transports classiques (le bus) et le taxi. Le premier type est public et suit des plans temporels (des horaires) et des plans spatiaux (des chemins) fixés à l'avance. Par contre, le taxi est plus flexible ; il suit des plans qui conviennent aux passagers. Cependant, il est très cher car il est privé. En d'autres termes, pour une demande de transport, un véhicule est réservé pour la servir. Le TAD, déclenché par les demandes de transport des clients, offre un transport public, économe et flexible. Il permet le transport de plusieurs personnes sur le même chemin et il est moins cher que le taxi. De plus, les arrêts, les horaires et les chemins des véhicules de TAD sont déterminés selon les demandes des clients.

Le TAD se développe partout dans le monde car il présente des avantages sociaux, économiques et environnementaux importants individuels et collectifs. A titre d'exemple, il peut servir des zones non desservies par les moyens traditionnels du transport public et il propose des coûts moins chers que le taxi. De plus, il diminue la congestion routière, les accidents de la route et la pollution en augmentant le taux de remplissage des véhicules. C'est ainsi que le nombre de véhicules en circulation diminue.

Quoique le TAD ait des aspects économiques, sociaux et environnementaux positifs indéniables, son acceptabilité par le public est primordiale pour sa réussite. La clé de son acceptation est sa réactivité d'offrir des solutions très rapides, fiables et réelles tout en respectant des contraintes concernant les clients et les véhicules, optimisant la qualité de service et prévoyant tout événement stochastique.

Le problème du TAD a été largement étudié depuis des dizaines d'années et il reste jusqu'à nos jours un thème essentiel car la population s'accroît et sa mobilité augmente le problème du trafic. La figure 2 et la figure 3 représentent le problème du TAD. La figure 2 représente les demandes de transport (flèches pointillées), les temps de parcours des routes reliant deux lieux (pondération des routes). Les heures de départ et d'arrivée souhaitées par chaque client sont indiquées respectivement à côté du lieu de départ et du lieu d'arrivée.

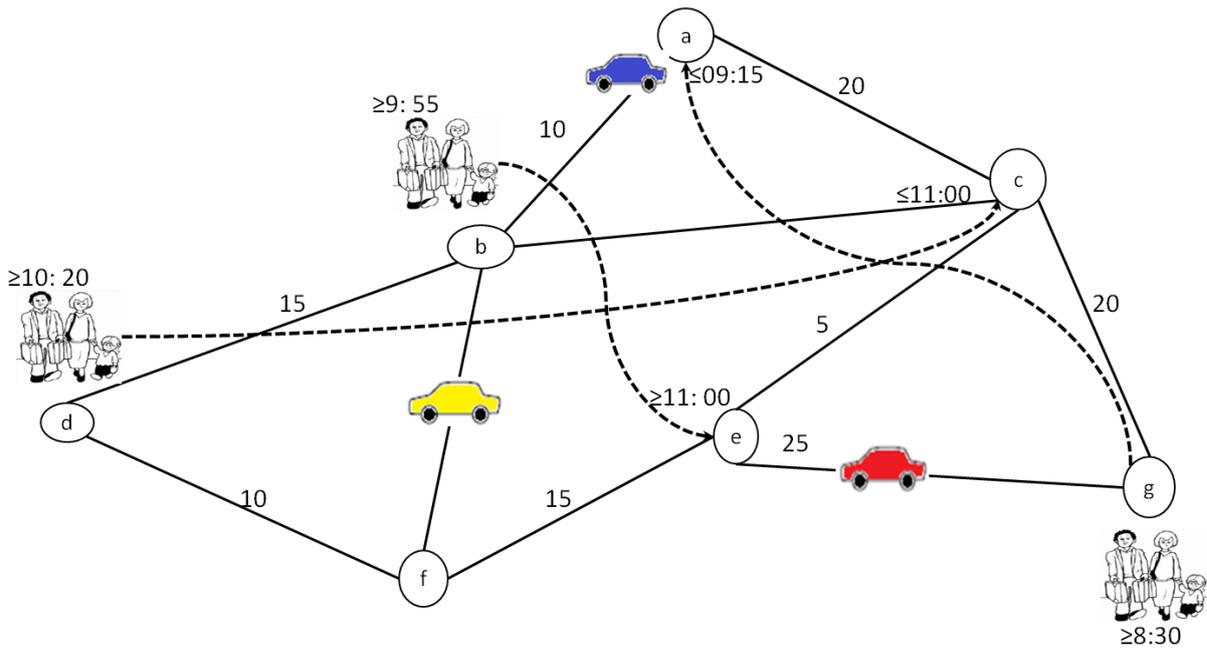


Figure 2 : Représentation du TAD - les demandes

La figure 3 représente les tournées de véhicules répondant aux demandes des clients. La tournée de chaque véhicule est représentée en pointillé (couleur bleu et rouge). L'heure de passage est indiquée pour chaque lieu.

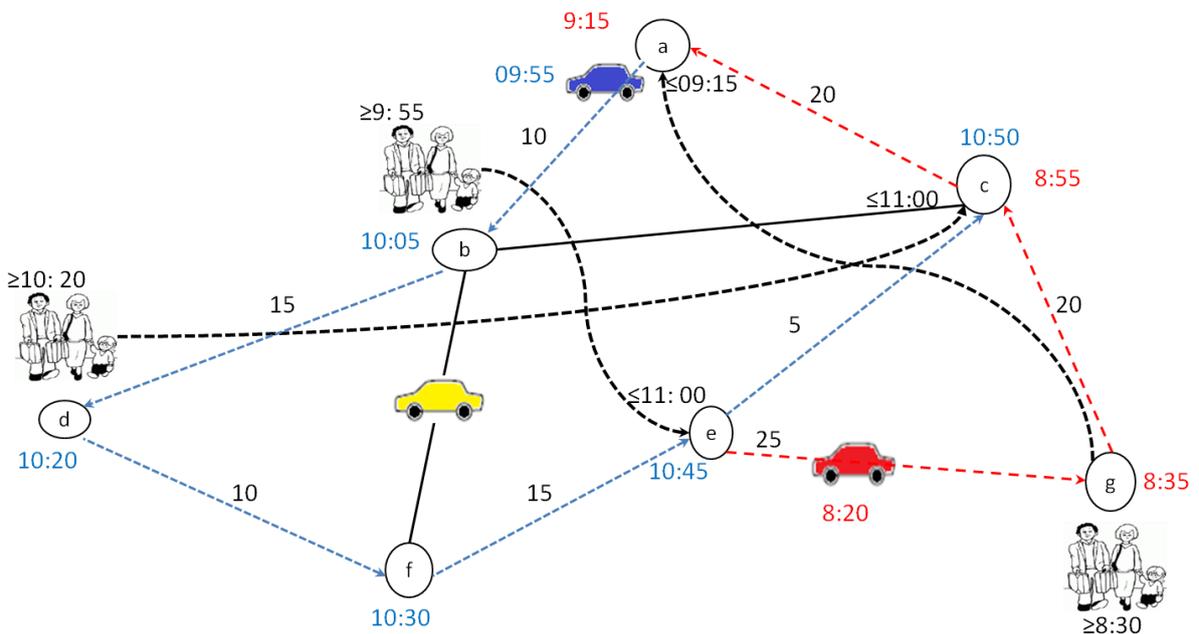


Figure 3 : Représentation du TAD - les tournées

Dans ce chapitre, nous présentons dans la deuxième section différentes définitions et différentes terminologies francophones et anglophones utilisées pour désigner le TAD. Puis, nous expliquons dans la section 3 le problème de tournées de véhicules et nous montrons que le problème du TAD en est une variante. Dans la section 4, nous décrivons les critères

d'optimisation du problème du TAD et nous montrons dans la section 5 et la section 6 respectivement les différentes méthodes de résolution de ces problèmes sans et avec considération de la situation du trafic routier. Dans la section 7, nous exposons quelques logiciels réels consacrés à la résolution du problème du TAD avant de conclure dans la section 8.

2 TAD : définitions et appellations

Le TAD se développe dans plusieurs villes dans le monde surtout aux Etats-Unis et en Europe. Pour cela, à ce type de transport ont été associées plusieurs définitions et appellations aussi bien en français qu'en anglais.

Les définitions attribuées au concept de TAD varient d'une communauté à l'autre. Par exemple, dans la littérature américaine, nous pouvons trouver la définition suivante :

“Non-fixed-route service utilizing vans or buses with passengers boarding and alighting at pre-arranged times at any location within the system's service area. Also called "dial-a-ride"”

l'American Public Transportation Association (APTA⁴).

D'autres définitions ont été recensées dans la littérature européenne. A titre d'exemple, (Le Breton 2001) définit le TAD comme étant :

“Un service de mobilité assuré par des taxis ou des minibus dont la particularité est d'être mis en œuvre sur réservation des utilisateurs. Ces systèmes intermédiaires entre la voiture et le transport collectif semblent adaptés aux nouveaux comportements de mobilité quotidienne des citoyens”.

Pour leur part, (Grosso et al. 2002) supposent que le TAD est :

“Demand Responsive Transport services provide transport “on demand” from passengers using fleets of vehicles scheduled to pick up and drop off people in accordance with their needs. DRT is an intermediate form of transport, somewhere between bus and taxi which covers a wide range of transport services ranging from less formal community transport through to area-wide service networks”.

⁴ http://www.apta.com/resources/reportsandpublications/Documents/Transit_Glossary_1994.pdf (consulté le 29/08/2016)

Finalement, (Josselin 2002) avance que le TAD est un transport terrestre public (associé à un territoire et à une autorité organisatrice des transports), collectif (visant à regrouper les passagers) et individualisé (qui n'est activé que lorsqu'au moins un usager en fait la demande).

D'après toutes ces définitions, il est possible de conclure que le TAD est un mode de transport public, flexible, économe, déclenché par les demandes des clients et nécessitant une réservation préalable. Il se différencie des autres modes de transport public classique (comme le bus) par sa flexibilité. Le TAD s'adapte à la demande des clients. Pour cela, les véhicules ne respectent pas un horaire précis et ne suivent pas des trajets fixes et réguliers. De même, il se distingue des taxis dont les voyages sont en général individuels, en ce qu'il est en mode collectif. Dans le TAD, il est possible de transporter plusieurs personnes sur un même trajet. La manière de regrouper des personnes par véhicule le rend plus optimal que le taxi. Ces différences nous permettent de constater que le TAD est un compromis entre le bus et le taxi regroupant les avantages de ces deux modes de transport.

Concernant l'appellation, le monde anglophone présente des désignations variées servant à dénommer le TAD. Anciennement, les termes *Paratransit* et *Dial-a-ride* (DAR) ont longtemps dominé. Le terme *Paratransit* est le plus ancien, mais aujourd'hui il est réservé pour désigner les services de transports pour des personnes à mobilité réduite (Fitzgerald, Shaunesey, et Stern 2000; Bearnse et al. 2004). Récemment, le terme DAR, utilisé pendant 30 ans, a été abandonné au profit du nouveau terme de DRT qui signifie *Demand Responsive Transport- Transportation- Transit*. De plus, les termes *on-demand transport* ou *transportation on-demand* sont récents et rarement utilisés pour désigner le TAD (Cordeau et al. 2007; Bertelle et al. 2009; El Falou et al. 2014a).

En revanche, en France dans les années 70 et 80, les termes *Bus à la demande* ou *Minibus à la demande* ont été utilisés pour désigner le TAD. Mais rapidement, ces deux termes tendent à disparaître et cèdent leur place à la seule expression dans la littérature française qui est *transport à la demande* : TAD (Le Breton 2001; Zargayouna 2005; Nabaa, Zeddini, et Tranouez 2007; Castex 2007; Garaix 2007; Atahran 2012).

3 TAD : variante du problème de tournées de véhicules

Le problème du TAD est une variante du problème général de tournées de véhicules connu dans la communauté anglophone sous le nom *Vehicle Routing Problem* :VRP (Dantzig et

Ramser 1959; Rego et Roucairol 1994). Le problème d'origine le plus simple du problème de tournées de véhicules est celui du voyageur de commerce connu sous le nom anglophone Travelling Salesman Problem : TSP (Bellmore et Nemhauser 1968). Dans les sous-sections suivantes, nous expliquons le problème de tournées de véhicules et ses variantes aboutissant au problème du TAD. Tout d'abord, nous commençons par le problème du voyageur de commerce. Nous décrivons ensuite le problème classique de tournées de véhicules (VRP). Enfin, nous terminons par la présentation des différentes variantes du VRP dont le problème du TAD fait partie.

Tout au long de cette section, nous choisissons la distribution des médicaments sur les pharmacies comme un exemple que nous allons appliquer sur différentes variantes discutées dans les sous-sections suivantes.

3.1 Problème du voyageur de commerce

Le problème célèbre du voyageur de commerce (TSP) est à la base de tous les problèmes de tournées. Etant donné un ensemble de villes interconnectées entre elles, le problème du voyageur de commerce consiste à trouver le plus court chemin mené par un seul véhicule qui passe exactement une et une seule fois par chaque ville et revient à la ville de départ. Trouver un circuit hamiltonien de longueur minimum semble être la solution. Un circuit hamiltonien est un circuit qui relie l'ensemble des villes d'un réseau de manière à ce que chaque ville soit visitée une et une seule fois.

Si nous cherchons à résoudre la distribution des médicaments comme un problème de TSP, alors la solution sera de trouver le plus court chemin mené par un seul véhicule, qui va passer une et une seule fois par chaque pharmacie pour lui livrer des médicaments et revenir au point de départ.

La figure 4 présente l'exemple du problème de voyageur de commerce. Le graphe à gauche représente l'ensemble des villes à visiter. Celui à droite représente une solution optimale en supposant que la distance entre deux points est représentée par la distance euclidienne entre ces deux points.

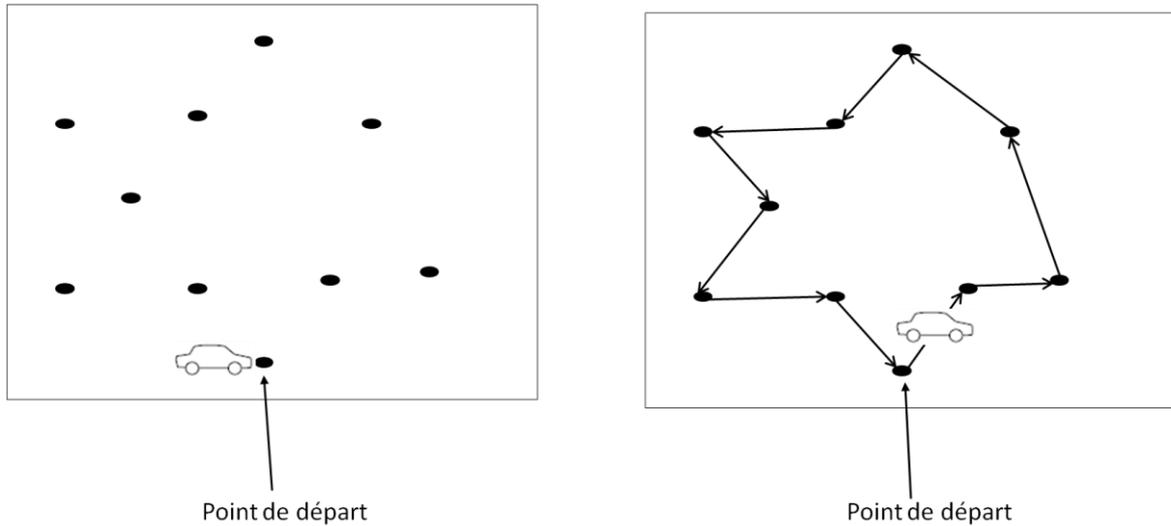


Figure 4 : Exemple du problème TSP

3.2 Problème de tournées de véhicules

Le problème de tournées de véhicules (VRP) a été initialement étudié par (Dantzig et Ramser 1959). Ce problème est une généralisation du problème du voyageur de commerce TSP. Il convient à un m-TSP qui désigne un problème du voyageur de commerce à plusieurs véhicules. Dans VRP, nous disposons d'une flotte de véhicules et un ensemble de villes à visiter. La solution de ce problème se traduit par l'identification d'un ensemble de chemins qui permettent de relier toutes les villes et qui se terminent par la ville de départ tout en optimisant un ou plusieurs critères comme la distance totale parcourue, le nombre de véhicules utilisés, etc. La visite des villes constituant un chemin donné, est assurée par un seul véhicule. Ces chemins sont disjoints et élémentaires (chaque ville appartient à un seul chemin et sera systématiquement visitée une seule fois et par un seul véhicule). La figure 5 représente l'exemple du VRP.

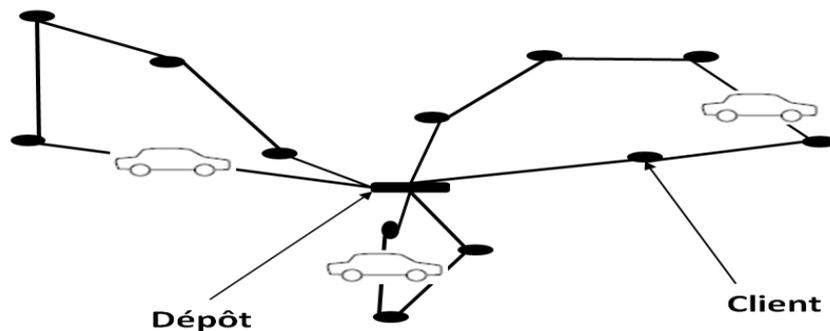


Figure 5 : Exemple du problème de tournées de véhicules

Nous appliquons le problème du VRP à la distribution de médicaments. Un ensemble de véhicules va livrer des médicaments sur un ensemble de pharmacies. La solution de ce

problème consiste à trouver un ensemble de chemins où chacun est mené par un véhicule de telle sorte que toutes les pharmacies soient visitées une et une seule fois.

Le problème du VRP, présenté précédemment, représente le modèle classique du problème de tournées de véhicules. Il est possible d'avoir plusieurs variantes de ce problème en ajoutant des contraintes diverses comme :

- la capacité des véhicules ;
- l'intervalle de temps au cours duquel la visite doit être réalisée ;
- la précedence entre les visites des villes clientes ;
- la panne d'un véhicule ;
- l'apparition/l'annulation d'une demande ;
- l'incertitude du temps de service ou du temps de voyage, etc.

3.3 Variantes du problème de tournées de véhicules

Plusieurs travaux de recherche qui avaient comme objectif de résoudre des problèmes pratiques, ont intégré différentes contraintes dans les problèmes de transport et de distribution (Toth et Vigo 2002; Golden, Raghavan, et Wasil 2008). Pour cette raison, le problème général de tournées de véhicules possède différentes variantes. Dans les sous-sections suivantes, nous allons présenter les principales variantes dérivées du VRP en y ajoutant ou retranchant des contraintes pour arriver à la fin au problème du TAD. Chaque sous-section sera consacrée à la présentation d'une variante du VRP tenant compte d'une ou plusieurs contraintes citées précédemment.

3.3.1 Variante considérant la contrainte de capacité

Cette variante tient compte de la contrainte assurant le respect de la capacité maximale caractérisant chaque véhicule réglementé. La variante du problème du VRP qui considère la contrainte de capacité est nommée CVRP : Capacitated Vehicle Routing Problem (Ralphs et al. 2003). Tous les véhicules dans CVRP sont homogènes et sont caractérisés par la même capacité k . Lorsque les véhicules possèdent des caractéristiques différentes en capacité ou en coût, le problème dans ce cas est appelé HVRP : Heterogeneous fleet Vehicle Routing Problem (Gendreau et al. 1999; Yaman 2005).

Si nous résolvons le problème de la distribution des médicaments comme un problème de CVRP, alors nous l'aurons résolu comme un problème du VRP, mais avec considération de la capacité maximale de chaque véhicule.

3.3.2 Variante considérant la contrainte de précédence

Cette variante exprime le fait qu'une ville doit être visitée avant une autre. Cette contrainte est indispensable pour mettre en œuvre des obligations particulières : par exemple, les visites des fournisseurs doivent précéder les visites de leurs clients. La variante du VRP qui prend en compte la contrainte de précédence est nommée problème de tournées de véhicules avec ramassage et livraison, connu par le terme VRPPD : Vehicle Routing Problem with Pickup and Delivery (Wassan et Nagy 2014).

Si nous appliquons cette variante au problème de la distribution des médicaments qui a les laboratoires comme fournisseurs et les pharmacies comme clients, alors chaque véhicule doit visiter les laboratoires pour ramasser les médicaments avant de les livrer aux pharmacies.

3.3.3 Variante considérant la contrainte temporelle

Nous distinguons deux types de contraintes temporelles : les contraintes liées aux véhicules et celles qui sont liées au service des clients.

Tout d'abord, les contraintes liées aux véhicules permettent d'exprimer l'indisponibilité des véhicules. Cette indisponibilité provient par exemple du nettoyage et de l'entretien technique du véhicule, du temps consacré aux repos des conducteurs, etc. Par contre, les contraintes liées au service des clients permettent de déterminer un intervalle de temps précisant la période de la visite d'un client. Ce dernier impose l'intervalle de visite. Cet intervalle, appelé la fenêtre de temps, est borné par la date de la visite au plus tôt et la date de la visite au plus tard.

La variante du VRP qui prend en compte les contraintes temporelles liées au service des clients est nommée VRPTW : Vehicle Routing Problem with Time Windows (Chang et Chen 2007; Kallehauge 2008). C'est un type de problème de tournées de véhicules dans lequel chaque client précise une fenêtre de temps durant laquelle il souhaite être servi.

Par exemple, dans le problème de la distribution des médicaments, chaque laboratoire et pharmacie précise la date de sa visite. Cette date est représentée par une fenêtre de temps que les véhicules doivent respecter.

Dans la littérature, des travaux de recherche ont considéré les contraintes temporelles imposées par les clients comme des contraintes souples ou fortes. Les problèmes considérant la contrainte de fenêtre de temps comme souple, ont relaxé cette contrainte en permettant de ramasser et de livrer avant le début et après la fin de la fenêtre de temps. Ainsi, les fenêtres de

temps imposées par les clients peuvent être violées avec un coût. En d'autres termes, une pénalité appropriée doit être ajoutée au coût de la solution lorsqu'un client est desservi en dehors de sa fenêtre de temps. Par exemple, un véhicule pouvant servir un client dans un temps qui n'appartient pas à sa fenêtre de temps, est un véhicule qui doit payer une pénalité (Balakrishnan 1993; Koskosidis, Powell, et Solomon 1992; Iqbal, Kaykobad, et Rahman 2015).

Au cas où le problème traiterait la contrainte temporelle comme forte ou rigide, toute solution ne respectant pas les fenêtres de temps imposées par les clients est invalide. Un véhicule, qui ne peut servir un client qu'en dehors des fenêtres de temps exigées, n'est pas une solution. Des demandes de transport peuvent être refusées par le système si la contrainte temporelle n'est satisfaite par aucun véhicule (Cordeau et al. 2002; Andreatta et al. 2016).

3.3.4 Variante considérant des données dynamiques

Cette variante traite le problème de tournées de véhicules dynamique. La dynamique résulte de nouvelles requêtes arrivant au fur et à mesure après l'exécution du système ou/et de nouvelles imprévions apparaissant en temps réel comme la panne d'un véhicule, l'annulation d'une demande, etc. Un problème de tournées de véhicules dynamique sert à ajuster les itinéraires des véhicules en temps réel afin de satisfaire de nouvelles demandes et de nouvelles modifications après le début des tournées de véhicules. Un certain nombre de chemins identifiés au départ doivent être recalculés en un temps acceptable de manière à prendre en considération les nouvelles imprévions. Ce type de problème reflète plus les conditions de la vie pratique dans le domaine du transport. Il est connu en anglais sous le terme de DVRP : Dynamic Vehicle Routing Problem (Pillac et al. 2013).

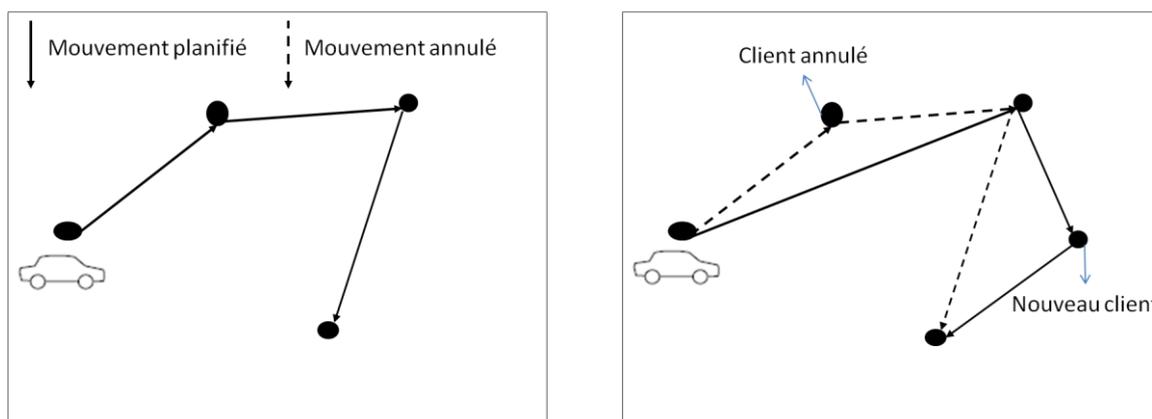


Figure 6 : Exemple du problème de tournées de véhicules dynamique

Exemple : Dans le problème de la distribution des médicaments, il peut arriver que de nouvelles demandes des pharmacies arrivent après le début des tournées de véhicules. Dans ce cas, il faut recalculer les tournées de manière à servir ces demandes. La figure 6 représente l'exemple du problème de tournées de véhicules dynamique.

3.3.5 Variante considérant des données stochastiques

La variante du VRP, dont un des paramètres est aléatoire et incertain durant la phase de planification, est appelée problème de tournées de véhicules stochastique connu par le sigle SVRP : Stochastic Vehicle Routing problem (Berhan et al. 2014). Les paramètres stochastiques peuvent être les requêtes des clients (Novoa et Storer 2009), les temps de service et les temps de voyage (Campbell, Gendreau, et Thomas 2011), etc.

3.3.6 Variante considérant des temps de voyage variables

C'est une variante qui considère que le temps de voyage (ou la vitesse de voyage) entre deux lieux n'est pas fixe ; le temps de voyage dépend du temps de départ et de la situation du trafic. Cette variante est désignée par l'acronyme TDVRP : Time Dependent Vehicle Routing Problem. Dans la littérature, il existe deux types de la variante TDVRP :

- TDVRP déterministe : ce type désigne un problème de tournée de véhicules où le temps de voyage entre deux lieux est dynamique déterministe. Ce temps est affecté par la situation habituelle du trafic routier qui varie selon des événements prévus connus *a priori* comme les heures de pointe (Hashimoto, Yagiura, et Ibaraki 2008; Balseiro, Loiseau, et Ramonet 2011) ;
- TDVRP stochastique : ce type fait référence à un problème de tournées de véhicules où le temps de voyage est dynamique stochastique. Ce temps est affecté par la situation réelle du trafic qui dépend des événements prévus et imprévus tel un accident (Schilde, Doerner, et Hartl 2014).

3.3.7 Variante qui prend plusieurs contraintes à la fois

Il existe des variantes qui prennent deux ou plusieurs contraintes en considération. A titre d'exemple, le problème du ramassage et livraison des marchandises avec fenêtres de temps qui est désigné par le sigle PDPTW : Pick-up and delivery problem with time windows. Cette variante respecte les contraintes de capacité, de précédence et temporelles. Dans ce problème, chaque véhicule doit transporter des marchandises depuis les fournisseurs vers les clients respectifs tout en respectant les fenêtres de temps imposées par chaque client. Deux variantes de PDPTW ont été étudiées dans la littérature : 1-PDPTW (avec un seul véhicule) et m-

PDPTW (avec plusieurs véhicules) (Savelsbergh et Sol 1995). La variante de PDPTW qui transporte des personnes à la place des marchandises, est nommée le problème du transport à la demande (TAD). D'où l'origine du problème du TAD.

La figure 7 représente en synthèse les problèmes origines du problème du TAD.

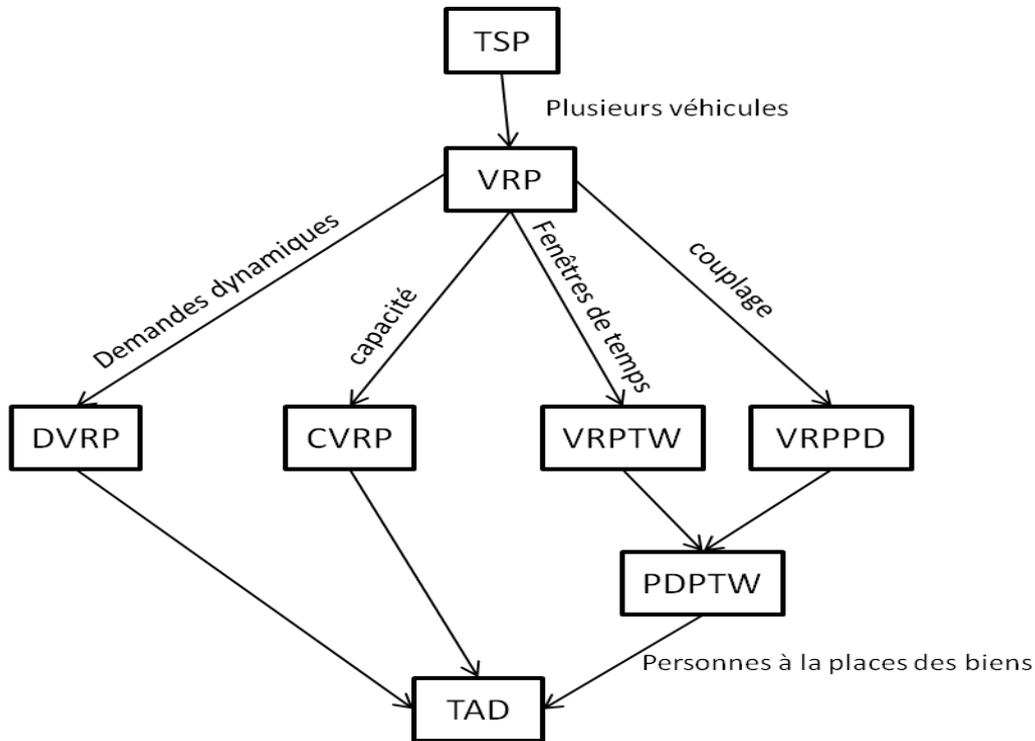


Figure 7 : VRP origine du TAD

4 Résolution du problème du TAD

Le problème du voyageur de commerce et les variantes du problème de tournées de véhicules y compris le TAD sont considérés comme des problèmes d'optimisation combinatoire. Un problème d'optimisation combinatoire a un nombre fini mais généralement très grand de solutions réalisables dans un espace énumérable. Ces solutions doivent respecter un ensemble de contraintes relatives à la nature du problème et doivent être évaluées par une fonction objectif.

Formellement, un problème d'optimisation combinatoire peut être défini par un ensemble discret Ω de solutions admissibles du problème et par une fonction objectif f permettant d'évaluer chaque solution admissible. Le but est de trouver $s \in \Omega$ qui optimise f . Nous parlons d'un problème de maximisation lorsque la fonction objectif est à maximiser et d'un problème de minimisation quand la fonction objectif est à minimiser.

Le problème du voyageur de commerce (TSP) est classé par (Karp 1972) parmi les 21 problèmes NP-complets. Donc, il n'existe pas un algorithme déterministe permettant de trouver la solution optimale au problème TSP en un temps polynomial. La recherche de la solution optimale de ce problème exige l'exploration de toutes les solutions possibles dans l'espace de recherche, puis de calculer le résultat de la fonction d'optimisation de chaque solution afin de choisir la solution optimale. Par exemple, dans le cas d'un problème de TSP avec n villes à visiter, il existe $(n-1)!$ de chemins solutions. C'est un nombre très important de solutions possibles même si n est petit. Supposons que $n=20$ et un ordinateur traitant un million de solutions par seconde. Un algorithme a besoin du 750 siècles de calcul pour trouver la solution exacte (Goldengorin et Pardalos 2012). Donc, faire la recherche exhaustive est impossible en termes de temps.

Le problème de voyageur de commerce (TSP) est un problème d'optimisation combinatoire et appartient à la classe des problèmes NP-difficile (Bouzgarrou 1998; Pang et al. 2004). Alors, le problème de tournées de véhicules (VRP) et ses variantes appartient encore à la classe des problèmes NP-difficile. En effet, dans chacun de ses problèmes, il y a le problème TSP auquel il est ajouté des contraintes qui le complexifient comme les fenêtres de temps, les capacités des véhicules, les demandes dynamiques, le temps de parcours stochastiques, etc. En conséquence, le TAD est un problème NP-difficile (Cubillos, Rodriguez, et Crawford 2007).

4.1 Critères d'optimisation du problème du TAD

La résolution du problème du TAD consiste à déterminer un ensemble de chemins pour une flotte de véhicules pour servir des demandes de transport tout en respectant un ensemble de contraintes imposées par les clients (comme les fenêtres de temps) et tout en optimisant un ou plusieurs critères (Savelsbergh et Sol 1995). Ces critères constituent les objectifs du problème d'optimisation du TAD. Un problème d'optimisation peut être mono-objectif (un seul critère à optimiser) ou multi-objectif (plusieurs critères à optimiser). L'optimisation se révèle par une fonction de minimisation ou de maximisation de ces critères. Les différents critères étudiés dans la littérature sont décrits ci-dessous :

- minimisation de la distance totale parcourue par les véhicules pendant une journée (Kammarti et al. 2004) ;
- minimisation du temps total parcouru par les véhicules pendant une journée (Jih et Hsu 1999). Ce temps peut être composé du :
 - temps de voyage des véhicules (Bruglieri et al. 2014) ;

- temps d'attente des véhicules ;
- minimisation du temps de voyage d'un client qui consiste à minimiser le temps durant lequel le client est à bord du véhicule (Cubillos, Urra, et Rodríguez 2009) ;
- minimisation du nombre de véhicules utilisés et circulant dans les rues. Etant donné que les véhicules et les chauffeurs sont les plus coûteux dans le problème du TAD, ce critère forme un but essentiel de ce problème (Savelsbergh et Sol 1995; Hosny et Mumford 2012) ;
- maximisation du taux d'occupation de chaque véhicule qui consiste à maximiser le nombre de passagers à bord du véhicule ;
- minimisation du désagrément des clients. En d'autres termes, maximisation de la satisfaction des clients (Lin et al. 2012). Le désagrément du client est mesuré par :
 - l'écart entre le temps de départ réel et le temps de départ désiré par le passager ;
 - l'écart entre le temps d'arrivée réel et le temps d'arrivée désiré par le passager ;
 - l'écart entre le temps de trajet réalisé et le temps de trajet direct ;
- maximisation du profit de l'entreprise ou minimisation du coût d'opération(Lin et al. 2012) ;
- maximisation du nombre de demandes servies (Carotenuto et al. 2012).

Nous passons maintenant à décrire les méthodes utilisées dans la résolution du problème de tournées de véhicules et ses variantes.

5 Méthodes de résolution du VRP et de ses variantes

Vue l'importance du problème de tournées de véhicules (VRP) et de ses variantes au niveau économique, social et environnemental, de nombreuses méthodes ont été utilisées dans la littérature pour les résoudre (Savelsbergh et Sol 1995; Cordeau et Laporte 2007; Parragh, Doerner, et Hartl 2007; Laporte 2009; Kumar 2012). Ces méthodes peuvent être classées en trois grandes catégories :

- les méthodes exactes qui garantissent la complétude de la résolution et l'obtention d'une solution optimale ;
- les méthodes approchées qui perdent la complétude pour gagner en efficacité ;

- les méthodes s'appuyant sur les systèmes multi-agents⁵ (SMA) qui assurent le passage à l'échelle. Les SMA résolvent le problème de la complexité, respectent la confidentialité et gardent la distribution naturelle du problème.

Dans les sous-sections suivantes, nous présentons les méthodes exactes et approchées les plus utilisées pour résoudre le problème du VRP et ses variantes. Egalement, nous présentons des travaux de recherche utilisant ces méthodes et les SMA dans la résolution de ces problèmes.

5.1 Méthodes exactes

Les méthodes exactes nécessitent une exploration exhaustive de toutes les solutions possibles de l'espace de recherche pour déterminer la solution exacte optimale du problème. Elles sont disponibles pour les problèmes simples de petite à moyenne taille. Comme le temps de calcul risque d'augmenter exponentiellement avec la taille du problème, ces méthodes rencontrent des difficultés dans la résolution des problèmes complexes et de taille importante. Bien que les méthodes exactes soient les solutions les moins prioritaires à être choisies pour résoudre le problème de tournées de véhicules et ses variantes surtout ceux qui sont dynamiques, ces méthodes ont joué un rôle dans la résolution de ces problèmes. La programmation dynamique (Sniedovich 2010), la méthode de la génération de colonnes (Nemhauser 2012), la méthode de séparation et évaluation ou *branch and bound* (Clausen 1999) et la méthode de coupes ou *branch and cut* (Mitchell 2011) font une partie des méthodes exactes qui sont le plus utilisées dans la résolution du problème de tournées de véhicules et ses variantes.

La méthode de séparation et évaluation se base sur une arborescence pour chercher la solution optimale. Afin d'éviter de parcourir l'arbre des solutions tout entier, cette méthode évalue, pour chacune des branches, la possibilité de trouver la solution optimale. Puis, les sommets qui semblent pouvoir mener à une solution intéressante sont examinés. La méthode de coupes est une variante qui généralise la méthode de séparation et évaluation. Cette méthode génère plusieurs coupes pour restreindre l'espace des solutions et ainsi augmenter la valeur de la relaxation linéaire.

(Psaraftis 1980) est le premier qui a résolu le problème du DAR : *dial a ride* statique et dynamique avec un seul véhicule en utilisant l'algorithme de programmation dynamique. Le problème est résolu avec une instance contenant au maximum 9 requêtes. Ensuite, il a adapté

⁵ Les systèmes multi-agents sont expliqués dans le chapitre 1.

l'algorithme de programmation dynamique pour résoudre le même problème tout en considérant la contrainte des fenêtres de temps. Cette adaptation a permis de résoudre un problème de dizaines de requêtes en minimisant le temps d'attente des passagers, la durée à bord des véhicules pour les passagers et la durée totale de la tournée de chaque véhicule (Psaraftis 1983).

La méthode de coupes (*branch and cut*) exacte basée sur des inégalités valides a été utilisée par (Lu et Dessouky 2004) pour résoudre le problème du ramassage et livraison avec des véhicules multiples (multiple vehicle pickup and delivery problem : MVPDP). Les auteurs ont pu résoudre de façon optimale le problème avec 25 clients. Cette méthode exacte a été utilisée par (Cordeau 2006) pour résoudre le problème de DAR statique. La solution optimale a été obtenue par des instances contenant au maximum 32 requêtes. (Ropke, Cordeau, et Laporte 2007) ont utilisé deux différentes versions de l'algorithme de *branch and cut* pour résoudre le problème du ramassage et livraison des marchandises avec fenêtres de temps : PDPTW. La solution optimale est obtenue avec des instances contenant jusqu'à 96 requêtes. Récemment, la méthode de *branch and cut* est utilisée pour résoudre le problème du DAR avec des caractéristiques et des contraintes supplémentaires qui visent à prendre en compte les spécificités des applications réelles. L'algorithme a pu résoudre une instance composée de 22 requêtes dans 4 heures (Liu, Luo, et Lim 2015).

5.2 Méthodes approchées

Les méthodes approchées sont l'alternative adéquate lorsque la puissance matérielle est assez limitée ou lorsque nous souhaitons résoudre le problème dans un temps limité. Les méthodes de cette approche trouvent des solutions aux problèmes n'ayant pas des algorithmes calculables en temps raisonnable. Elles font des compromis entre l'optimalité de solutions et le temps d'exécution. Elles résolvent les problèmes de grande taille et incluant des contraintes de la vie pratique. Ces méthodes peuvent être divisées en deux catégories : les heuristiques et les métaheuristiques.

Une heuristique n'a pas une définition générale et ne suit pas une règle générale. Elle est conçue pour un problème particulier. Par contre, les métaheuristiques sont des heuristiques générales qui peuvent être appliquées sur plusieurs types de problèmes d'optimisation.

5.2.1 Heuristiques

L'heuristique est une méthode conçue pour un problème d'optimisation particulier et ne suit pas une règle générale. Cependant, elle ne peut pas être généralisée pour résoudre d'autres

types de problèmes. La méthode des gains, la méthode d'insertion séquentielle et les méthodes de construction sont les heuristiques les plus utilisées dans la résolution du problème de tournées de véhicules.

Un algorithme d'insertion séquentielle est présenté par (Jaw et al. 1986) pour résoudre le problème multi-objectif du ramassage et livraison des marchandises avec m véhicules : m-PDP. La fonction objectif est la minimisation du coût total d'insertion des nouvelles requêtes. Ce coût est une somme pondérée de l'insatisfaction des clients et du coût total de l'opérateur. Ce dernier est mesuré par le nombre de véhicules utilisés.

Une méthode de construction se basant sur le principe d'insertion est développée par (Lu et Dessouky 2006) pour résoudre le problème du ramassage et livraison des marchandises avec fenêtres de temps à m véhicules (m-PDPTW). L'objectif est de minimiser un coût total proportionnel à la distance parcourue. Ce coût est composé du coût des véhicules et du coût de voyage.

5.2.2 Métaheuristiques

Les métaheuristiques (DREO et al. 2003; Luke 2009; Talbi 2009) ont été très largement utilisées dans la résolution de plusieurs types de problèmes d'optimisation. Ces méthodes ont présenté leur efficacité dans le cas des problèmes complexes. La difficulté réside dans la modélisation de ces métaheuristiques dans la meilleure manière afin de bien en profiter dans la résolution des problèmes. Les métaheuristiques sont groupées en deux catégories selon le nombre de solutions sur lesquelles elles se basent pour produire la solution finale : les métaheuristiques à solution unique et les métaheuristiques à population de solution.

5.2.2.1 Les métaheuristiques à solution unique

Aussi nommées les méthodes de voisinage, ce sont des métaheuristiques qui partent d'une solution unique initiale pour arriver à la solution finale en se déplaçant de cette solution initiale à une autre voisine. Ce qui caractérise ces méthodes est le choix de la solution voisine. La méthode de descente (Talbi 2009), le recuit simulé (Eglese 1990) et la recherche tabou (Glover 1986) font partie de ce type de méthodes.

La méthode de descente

La méthode de descente se base sur une relation de voisinage. Elle débute par une solution initiale qui sera modifiée successivement par une autre solution voisine de qualité meilleure

pour arriver à la fin à une solution optimale. Cette méthode est rapide et simple à mettre en œuvre, mais elle s'arrête au premier optimum local trouvé (Marmion 2011). L'algorithme 1 explique le fonctionnement de cette méthode.

Algorithme 1 : Algorithme de la méthode de descente

s_0 : solution initiale

$z = f(s_0)$ où f est la fonction d'optimisation

tant que ($\exists s'$ dans *voisinage*(s_0)) **faire**

si ($f(s')$ plus meilleur que z) **alors**

$s_0 = s'$

$z = f(s')$

fin

fin

retourner s_0

Pour éviter la carence de la méthode de descente, d'autres méthodes de recherche locale existent : le recuit simulé et la recherche taboue.

Le recuit simulé

L'algorithme du recuit simulé est inspiré de la procédure du recuit des métaux utilisée en métallurgie. Cette procédure passe par des phases de réchauffage et de refroidissement lent afin de minimiser l'énergie du matériau. L'algorithme de recuit simulé dérive de l'algorithme simple de Metropolis (Metropolis et al. 1953) qui permet de simuler l'évolution d'un système thermodynamique passant d'un état instable vers un état d'équilibre thermique. Par analogie entre le système physique et le problème d'optimisation, la fonction à minimiser deviendra l'énergie du système.

La métaheuristique du recuit simulé est une méthode de recherche locale. A partir d'une solution initiale x , une solution voisine $v(x)$ est générée. Considérons que $\Delta f = f(v(x)) - f(x)$, où f est la fonction d'optimisation. Si la nouvelle solution voisine générée améliore le critère d'optimisation c'est-à-dire que $\Delta f < 0$, nous disons que l'énergie du système est baissée. Sinon, elle la détériore. L'acceptation d'une solution voisine moins bonne (c'est-à-dire que $\Delta f > 0$) peut arriver, mais avec une certaine probabilité $p = e^{-\Delta f/T}$ déterminée en fonction du paramètre température T . Cette acceptation moins bonne vise à explorer l'espace

de recherche de solutions et évite de s'enfermer trop vite dans la recherche d'un optimum local. L'algorithme du recuit simulé est présenté dans l'algorithme 2.

Algorithme 2 : Algorithme du recuit simulé

s : solution initiale

T : une température initiale, $T > 0$

tant que (la condition d'arrêt n'est pas satisfaite) **faire**

 générer un voisin $s' = v(s)$

 calculer $f(s')$

$\Delta f = f(s') - f(s)$

si $\Delta f < 0$ **alors**

 alors $s := s'$

sinon

$p = e^{-\Delta f/T}$

 r : valeur aléatoire dans $[0, 1]$

si $(r \leq p)$ **alors**

$s = s'$

fin

fin

 refroidir la température T

fin

retourner s

(Kirkpatrick 1984; Černý 1985) sont les premiers chercheurs qui ont appliqué la méthode de recuit simulé dans les problèmes d'optimisation combinatoire et surtout dans le problème de voyageur de commerce. (Lin et al. 2012) proposent un modèle basé sur l'algorithme du recuit simulé pour résoudre le problème de covoiturage en taxi, considéré comme un problème de m-DAR statique. Le covoiturage en taxi se distingue du taxi classique par le fait que plusieurs requêtes sont adressées au même véhicule au lieu d'une seule requête. Ce modèle prend en considération les contraintes de fenêtre de temps et a comme objectif de minimiser le coût d'opération représenté par la distance totale parcourue par les véhicules. Il permet également de maximiser la satisfaction des clients en minimisant le temps d'attente et le temps de parcours du client à bord du véhicule. Les résultats des expérimentations ont montré que 29

requêtes ont besoin de 10 taxis et de 302 km de distance parcourue pour être servis dans le modèle proposé. Alors que dans le taxi classique, ils ont besoins de 29 taxis et de 375 km de distance parcourue. Donc, 66 % de taxis et 19 % de distance parcourue ont été réduits dans le modèle proposé.

La recherche Tabou

La recherche taboue est une technique itérative d'optimisation combinatoire développée par (Glover 1986). Cette méthode utilise une technique différente des autres métaheuristiques permettant de faire explorer des solutions en évitant de tomber sur une solution optimale locale et de répéter un cycle. Cette technique consiste à mémoriser les dernières solutions récemment visitées dans une liste appelée liste taboue. Ces solutions déclarées taboues ne sont pas utilisées qu'après un certain temps. Donc, pour éviter de tomber dans le cycle de la solution courante et son meilleur voisin, cette méthode interdit de retourner vers une solution mémorisée dans la liste. Cette liste est de longueur k contenant les k dernières solutions visitées. La solution suivante est un élément de l'ensemble de solutions voisines excluant les solutions de la liste taboue. Lorsque le nombre k est atteint, chaque nouvelle solution sélectionnée remplace la plus ancienne dans la liste. La construction de la liste taboue est basée sur le principe FIFO. L'algorithme 3 présente le fonctionnement de la méthode de recherche taboue.

Algorithme 3 : Algorithme de la recherche taboue

s : solution initiale

listeTabou = \emptyset

tant que (la condition d'arrêt n'est pas satisfaite) **faire**

 générer un voisin $s' = v(s)$ tel que s' n'appartient pas à listeTabou

$$\Delta f = f(s') - f(s)$$

si $\Delta f < 0$ **alors**

$$s = s'$$

fin

 actualiser listeTabou

fin

(Taillard et al. 1997) ont utilisé la méthode de recherche taboue pour résoudre le problème de tournées de véhicules (VRP) avec des fenêtres de temps souples. La liste taboue de cette méthode mémorise les routes de meilleures solutions visitées.

Une méthode de recherche taboue développée pour le problème de tournées de véhicules avec fenêtres de temps (VRPTW) statique a été adaptée par (Gendreau, Michel et al. 1999) pour résoudre le même problème mais en mode dynamique. Afin de raccourcir le temps de calcul, les auteurs ont développé la méthode taboue dans une plateforme parallèle.

(Lau et Liang 2002) ont résolu le problème du ramassage et livraison des marchandises avec fenêtres de temps (PDPTW) en utilisant la méthode de recherche taboue. La solution initiale, générée par une nouvelle méthode constructive hybride, a été améliorée par l'application de la méthode taboue.

(Lai, Caliskan Demirag, et Leung 2016) ont aussi utilisé la méthode de recherche taboue pour résoudre le problème de tournées de véhicules hétérogènes (HVRP) avec fenêtres de temps sur un multi graphe. Ce multi graphe est caractérisé par l'existence des arcs parallèles reliant les paires de sommets. Chacun de ces arcs est pondéré par un critère de voyage comme le temps, la distance et le coût. De plus, les auteurs ont considéré la contrainte temporelle comme une contrainte souple. Donc, une pénalité doit être ajoutée à la fonction objectif en cas d'une violation des fenêtres de temps imposées par les clients.

5.2.2.2 Les métaheuristiques à population de solutions

Contrairement aux métaheuristiques à solution unique, ces métaheuristiques partent d'une population de solutions initiales, et au fur et à mesure, cette population est évoluée jusqu'à atteindre la solution finale. L'algorithme génétique (Luke 2009) et la colonie de fourmis (Colorni et al. 1991) appartiennent à cette famille de métaheuristiques.

L'algorithme génétique

L'algorithme génétique a été inventé par John Holland en 1975 à l'université du Michigan (Holland 1975). Il fait partie des algorithmes évolutionnaires connus sous le terme anglophone Evolutionary Computation. Ces algorithmes servent à trouver une solution approchée aux problèmes qui n'ont pas de solutions calculables en bref délais. Ils sont souvent utilisés pour résoudre les problèmes d'optimisation combinatoire. Ils servent à trouver les extremums d'une fonction objective définie sur une espace de données. L'algorithme génétique est inspiré du mécanisme d'évolution d'une espèce : la génétique. Son concept

repose sur la biologie de la population, de sa génétique et des techniques de son évolution tels que la sélection, le croisement et la mutation.

L'utilisation de l'algorithme génétique exige l'existence des éléments suivants :

- une technique de codage des individus de la population. C'est la première étape de l'application de l'algorithme génétique. Elle s'intègre dans le cadre de la modélisation et de la formalisation du problème à résoudre. Son rôle consiste à associer à chaque individu de la population une structure de données unique sous la forme d'une chaîne de caractères appelée code génétique ou chromosome (une suite de gènes). Plusieurs types de codage existent comme le codage binaire, le codage réel, etc. ;
- une procédure de génération de la population initiale. Cette procédure doit construire une population utile pour la génération des populations successives. La construction de la population initiale doit être adaptée à la nature du problème car elle joue un rôle primordial dans la convergence rapide vers la solution optimale ;
- une fonction d'évaluation pour mesurer l'efficacité de l'individu ou son adaptation au sein de la population. Elle consiste à donner une note à chaque chromosome représentant un individu selon son efficacité au sein de la population. L'évaluation d'un individu est indépendante des autres individus. Le résultat de l'évaluation permettra de garder les individus les plus enclins à la solution du problème et d'éliminer les autres. En général, un individu a une probabilité de vivre et de se reproduire suivant son efficacité au sein de la population, également à dire suivant son adaptation au problème ;
- un ensemble des opérateurs qui servent à diversifier la population en cours et à la transmission d'une partie des propriétés de ses individus à la prochaine génération. Ces opérateurs sont :
 - la sélection : le principe de cet opérateur consiste à déterminer les individus, selon leur fonction d'évaluation, qui vont jouer le rôle des parents et qui vont générer des enfants. Plusieurs techniques de sélection ont été proposées dans la littérature telles que :
 - la sélection par roulette : elle consiste à représenter chaque individu par un segment de longueur proportionnelle à sa valeur de fonction d'évaluation. La concaténation de l'ensemble des segments génère un axe dont la longueur est s (s est la somme totale des valeurs de fonction d'évaluation des individus). La sélection d'un individu se fait en tirant

- aléatoirement une valeur entre 0 et s . L'individu dont le segment est désigné par cette valeur sera sélectionné (Lipowski et Lipowska 2012) ;
- la sélection par tournoi : elle consiste à choisir, parmi k individus tirés aléatoirement, celui qui a la valeur de la fonction d'évaluation la plus efficace (Luke 2009) ;
 - la sélection uniforme où les individus sont sélectionnés de manière aléatoire, uniforme et sans considération de leur fonction d'évaluation ;
 - etc. ;
- le croisement : l'objectif principal du croisement est de recombinaison deux individus pour former deux nouveaux individus. De manière classique, le croisement s'applique à deux individus nommés parents afin de générer deux nouveaux individus nommés enfants qui héritent certaines caractéristiques de leurs parents. Il est mis en place afin que les enfants gardent les meilleures parties de leurs parents. Plusieurs techniques de croisement ont été proposées : croisement à un seul point, croisement à deux points et croisement uniforme (Luke 2009) ;
 - la mutation : en appliquant la sélection et le croisement sur les individus d'une population, il peut avoir lieu, dans certains cas, une convergence prématurée causée par le fait qu'après ces deux opérateurs, tous les individus peuvent devenir identiques. Quand un individu se croise avec lui-même, rien de nouveau n'est généré et la solution optimale n'est jamais atteinte. C'est l'importance du troisième opérateur de mutation qui vient remédier à la convergence prématurée. Celui-ci a comme rôle d'explorer un nouvel état dans l'espace de recherche en assurant qu'un individu change la valeur d'un ou plusieurs gènes de son chromosome aléatoirement. Ce qui garantit l'atteinte de tout point de l'espace de recherche et évite la convergence prématurée. Cet opérateur est défini par une probabilité souvent inférieure à 1 %. Plus cette probabilité est élevée, plus la population est diversifiée. Se présente dans ce contexte le risque de la non-convergence de l'algorithme génétique ;
- des paramètres indispensables pour le fonctionnement de l'algorithme tels que le critère d'arrêt, le nombre total de générations, les probabilités d'applications des opérateurs, etc.

L'algorithme 4 montre l'organigramme de l'algorithme génétique.

Algorithme 4 : Algorithme génétique

n : taille de la population

P = {} //population P est vide

meilleur = null

pour i allant de 1 à n **faire**

 P = P union {nouvel individu aléatoire}

fin

tant que le critère d'arrêt n'est pas satisfait **faire**

pour chaque individu pi appartenant à P **faire**

 évaluer (pi)

si meilleur est null ou fitness (pi) > fitness (meilleur) **alors**

 meilleur = pi

fin

fin

 Q = {}

pour i allant de 1 à n/2 **faire**

 Parent pa = SelectionParTounoi (P)

 Parent pb = SelectionParTounoi (P)

 Enfant ea,eb = croisement (pa, pb)

 Enfant eb = croisement (pa, pb)

 Q = Q union {mutation(ea), mutation(eb)}

fin

 P = Q

fin

retourner meilleur

Nous consacrons ce qui suit à la présentation de l'utilisation immense de l'algorithme génétique dans la résolution du problème de tournées de véhicules et ses variantes.

(Kammarti et al. 2004) ont développé un modèle hybride utilisant l'algorithme génétique et la méthode de recherche taboue pour résoudre le problème du ramassage et livraison avec fenêtres de temps à un seul véhicule (1-PDPTW) statique. Avec des opérateurs génétiques (opérateurs de croisement et de mutation) intelligents et spéciaux, les contraintes de précédences et de capacité ont été respectées pour chaque individu de la population. De plus,

trois heuristiques ont été élaborées afin de minimiser la distance totale parcourue, la somme des temps d'attente et la somme des retards de chaque individu généré de la population initiale. Les auteurs ont utilisé la notion de Pareto-dominance ou Pareto optimalité (Voorneveld 2003) pour réaliser une évaluation multicritère pour les solutions en faisant un compromis entre la distance totale du voyage, le temps total d'attente et le temps total de retard. Dans une évaluation multicritère, il existe un équilibre : un critère ne peut pas être amélioré sans détériorer au moins un des autres. Cet équilibre a été appelé optimum de Pareto. Enfin, la méthode de recherche taboue vient pour améliorer les solutions.

Ces travaux ont été étendus par (Kammarti et al. 2005). Ils proposent une nouvelle approche basée sur la méthode de Pareto-dominance et sur l'utilisation de bornes inférieures pour l'évaluation multicritère de solutions. Cette nouvelle approche a minimisé le compromis entre la distance totale parcourue et la somme des retards. (Kammarti et al. 2006) ont par la suite résolu le problème de 1-PDPTW dynamique en utilisant les algorithmes évolutionnistes avec des opérateurs génétiques adaptés pour le problème afin de réduire le temps de calcul. La méthode de Pareto-dominance est également utilisée dans le but de fournir un ensemble de solutions satisfaisantes en minimisant le compromis entre la distance totale parcourue et la somme des retards.

De même, (Harbaoui Dridi et al. 2008) ont utilisé l'algorithme génétique pour résoudre le problème du ramassage et livraison avec fenêtres de temps à m véhicules (m -PDPTW). Le but était de minimiser le coût total de transport et la distance totale de voyage. Ensuite, ils ont étendu ce travail dans (Harbaoui Dridi et al. 2009) en utilisant les algorithmes génétiques avec la notion de Pareto-dominance afin de résoudre le même problème tout en minimisant le compromis entre le nombre de véhicules utilisés, la somme des retards et le coût total de transport. Cette version a été étendue en proposant un autre modèle qui évalue de manière multicritère les solutions. En effet, les auteurs se sont basés sur la notion de Pareto-dominance et sur la méthode de borne inférieure pour trouver une solution satisfaisante pour le problème m -PDPTW dynamique, tout en minimisant le compromis entre le coût de voyage total et le temps de retard total (Harbaoui Dridi et al. 2010).

De plus, (Carotenuto, Paradisi, et Storchi 2014) présentent une autre approche pour résoudre le problème du DRT : *Demand Responsive Transport* (en français le TAD). C'est une approche flexible qui traite les demandes de transport dans un mode dynamique. Son objectif est d'optimiser la qualité de service en minimisant le nombre de déviation et le temps de

parcours et en respectant les fenêtres de temps de chaque passager. Pour réaliser l'objectif, les auteurs ont développé un algorithme à deux niveaux de calculs. Pour un premier niveau, une réponse d'acceptation ou de refus primaire est donnée aux clients le plus tôt possible par un algorithme utilisant l'heuristique constructive. Pour un second niveau, un autre algorithme vient améliorer et perfectionner la solution donnée par le premier niveau. Cet algorithme se base sur le principe de l'algorithme génétique mais avec une fonction d'évaluation spéciale inspirée de l'heuristique de construction.

Quant à (Guerriero et al. 2014), ils ont résolu le problème de DARP : Dial-a-Ride problem (en français le TAD) en optimisant le temps d'attente total des véhicules et le temps de parcours total des passagers. La route de chaque véhicule a une durée maximale qui ne doit pas être dépassée et le nombre de véhicules utilisés est contrôlé. Ce problème a été traité en suivant deux étapes : la première étape est composée des métaheuristicques (recuit simulé et algorithme génétique) qui trouvent un ensemble de solutions faisables respectant les fenêtres de temps, la capacité des véhicules, la contrainte de précédence, etc. Ces solutions sont ensuite traitées par la méthode *epsilon-contrainte*, utilisée pour résoudre les problèmes multi-objectifs, et ceci dans le but de générer des solutions améliorées optimisant les deux objectifs cités précédemment.

L'algorithme de colonie de fourmis

L'algorithme de colonie de fourmis est récemment développé par (Colorni et al. 1991). Il apporte une nouvelle manière de résoudre les problèmes d'optimisation combinatoire. C'est une méthode itérative à population dont les individus partagent un savoir commun leur permettant d'orienter leurs futurs choix et d'indiquer aux autres individus des choix à suivre ou à éviter. Le principe de cette métaheuristique est inspiré du comportement collectif réel et la méthode de communication adoptés par les fourmis. Ces insectes trouvent le plus court chemin entre leur nid et la source de nourriture grâce à un moyen de communication et à une forme d'apprentissage : la phéromone. La phéromone est une substance déposée sur le sol par une fourmi lors de son déplacement. Ainsi, plus il y a de fourmis qui utilisent le même trajet, plus il y aura de phéromones sur ce dernier et plus il y a de chance que celui-ci soit choisi par les prochaines fourmis qui passeront.

La figure 8 montre l'exemple du comportement des fourmis. Une fourmi trouve au hasard une source de nourriture F à travers un chemin quelconque "a". Puis, elle revient au nid N à travers un autre chemin "b" en laissant derrière elle la substance appelée phéromone (1). Les

autres fourmis empruntent indifféremment les quatre chemins possibles pour atteindre la même source de nourriture (2). Mais le chemin le plus court sera parcouru par plus de fourmis que les autres chemins. Alors, le plus court chemin sera plus renforcé par des phéromones. Comme ces phéromones sont attractives, les fourmis tendent à suivre le chemin le plus court et les autres chemins plus longs disparaissent (3).

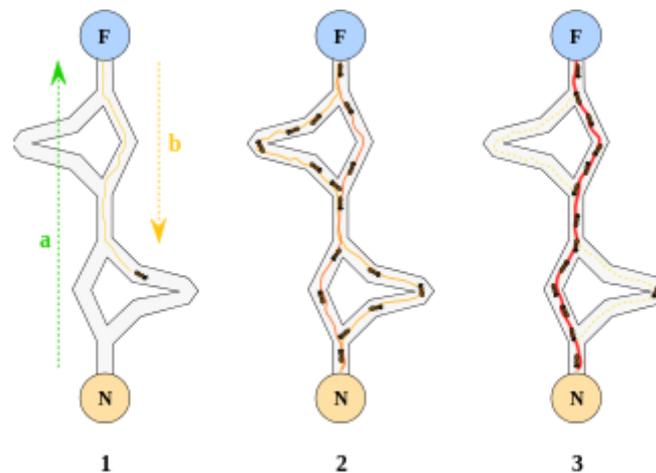


Figure 8 : Comportement des fourmis

L'algorithme 5 représente l'algorithme de colonies de fourmis pour la résolution du problème TSP.

Algorithme 5 : L'algorithme de colonies de fourmis pour le TSP [(Gherboudj 2013)]

initialiser une population de m fourmis ;

évaluer les m fourmis ;

tant que la condition d'arrêt n'est pas satisfaite **faire**

pour i = 1 à m **faire**

 construire le trajet de la fourmi i ;

 déposer des phéromones sur le trajet de la fourmi i ;

fin

 évaluer les m fourmis ;

 évaporer les pistes de phéromones ;

fin

retourner la ou les meilleures solutions

L'algorithme de colonies de fourmis a été proposé pour la première fois pour résoudre le problème du voyageur de commerce (TSP) par (Colormi et al. 1991). (Montemanni et al.

2002) ont, pour leur part, développé un algorithme basé sur la colonie de fourmis pour résoudre le problème de tournées de véhicules dynamique (DVRP). Egalement, le problème de tournées de véhicules avec de temps de voyage dynamique stochastique (TDVRP) est résolu par l'algorithme de colonie de fourmis (Duan et al. 2015).

5.3 Approches multi-agents pour la résolution du problème du TAD

Plusieurs travaux de recherche ont montré que le paradigme systèmes multi-agents (SMA) est une technique prometteuse de nos jours pour la modélisation et la résolution de problèmes du TAD. Les systèmes multi-agents (SMA) sont adaptés aux problèmes dynamiques, complexes et naturellement distribués. Ils accélèrent l'exécution et permettent de raccourcir les temps de réponse du système à des clients en ligne. Dans les sous-sections suivantes, nous allons présenter un ensemble des approches qui résolvent les problèmes du TAD en se basant sur le SMA. Ces approches sont qualifiées centralisées ou décentralisées. La centralisation ou la décentralisation reflète le contrôle ou le calcul pour trouver la solution.

5.3.1 Approches centralisées

Dans (Zargayouna 2005), l'auteur propose un modèle multi-agent pour résoudre le problème du TAD. Dans cette approche, trois agents sont identifiés : un agent Client, un agent Interface et un agent Véhicule. L'agent Client représente le passager. Lorsqu'un agent Client reçoit une requête, il l'envoie à l'agent Interface. Ce dernier teste les paramètres de la requête. Si les paramètres sont incorrects, il la rend à l'agent Client. Dans le cas contraire, il l'envoie vers l'environnement pour être exécuté. L'agent Véhicule est associé à chaque véhicule. Il présente son offre pour desservir une requête. L'environnement du problème a été représenté dans ce modèle de manière spatio-temporelle (voir figure 9) dans le but d'éviter de diffuser chaque nouvelle requête vers tous les agents Véhicule. L'idée est la suivante : l'agent Véhicule met des écouteurs sur les nœuds futurs qu'il va visiter. Chaque agent Véhicule aura ainsi une zone de couverture dynamique constituée par l'ensemble des nœuds de visite future où l'agent a déjà déposé ses écouteurs. Alors, lorsqu'une nouvelle requête arrive, seuls les agents Véhicule, qui ont des écouteurs pendant le temps de départ sur le nœud de départ de la requête et pendant le temps d'arrivée sur le nœud d'arrivée, reçoivent la requête de l'agent Interface. Ensuite, ces agents présentent leur offre pour desservir cette requête à l'agent Interface. L'offre est calculée suivant une heuristique d'insertion. Enfin, l'agent Interface choisit le véhicule le mieux positionné à desservir le client. Le véhicule le mieux positionné est celui qui perd moins des écouteurs à cause de l'insertion d'une nouvelle requête, c'est-à-

dire que celui qui optimise le nombre de nœuds couverts par la zone de l'agent véhicule. Par conséquent, le nombre de requêtes desservies sera optimisé. Le point faible de cette idée est qu'une requête dont le point de départ ou d'arrivée n'est pas couvert par aucun agent véhicule, sera rejetée. La figure 9 ci-dessous représente la modélisation de l'environnement spatio-temporelle proposée.

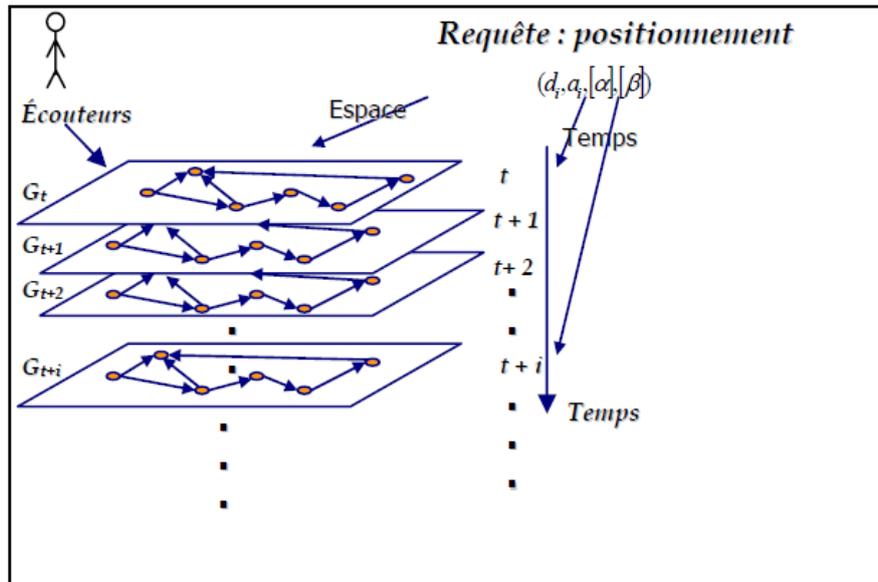


Figure 9 : Représentation spatio-temporelle de l'environnement (Zargayouna 2005)

Par ailleurs, (Jin, Abdulrab, et Itmi 2008) proposent une approche multicouche hybride et distribuée basée sur la technique multi-agents pour la résolution DRT. Cette approche optimise la période totale creuse des véhicules, le nombre de sièges utilisés dans chaque véhicule et le nombre de véhicules utilisés. Les auteurs distinguent quatre types d'agents :

- agent Utilisateur : il représente le client, sa requête de transport et ses interactions avec le système DRT. Il reçoit toutes les demandes du client concernant sa requête (changement de lieu, annulation de la requête, etc.) ;
- agent Nœud-station : il modélise chaque nœud-station géographique réelle. Il est responsable du traitement, de l'attribution d'un véhicule et de l'établissement de l'horaire de chaque requête reçue tout en optimisant un ensemble de critères comme le nombre de véhicules utilisés, etc. Il est responsable de la coordination entre les différents agents du système ;
- agent Taxi : il représente chaque véhicule réel du système. Il a comme rôle d'assurer l'interopérabilité entre le véhicule et le système DRT. Il propose un plan puis l'actualise suite à des échanges des informations avec l'agent Nœud-station ;

- agent Transport-admin : il est responsable (en charge) de tous les autres agents.

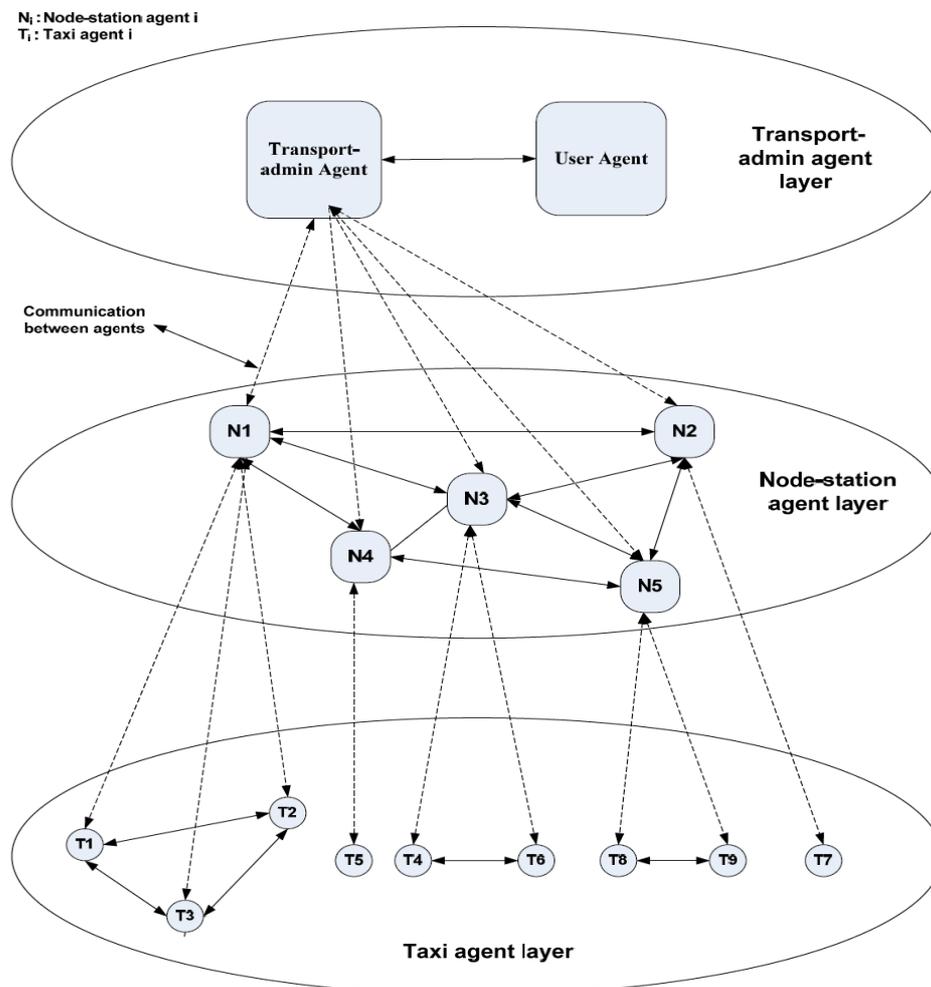


Figure 10 : La planification multicouche agent (Jin, Abdulrab, et Itmi 2008)

La figure 10 présente les différentes couches et types agents de chaque couche. Nous voyons dans la première couche les agents Utilisateur et Transport-admin qui communiquent à propos de la requête du client et de toute modification qui subit cette requête comme changement du lieu, annulation de la requête, etc. La deuxième couche contient les agents de types Nœud-station. La dernière couche contient les agents Taxi.

L'idée de cette approche est la suivante : l'agent Taxi trouve un ensemble de chemins pour servir les requêtes par une recherche locale. Ensuite, il coopère avec les agents de type Taxi et Nœud-station existants dans son domaine de planification afin d'échanger d'informations à propos des événements imprévus comme la situation du trafic et de trouver son meilleur chemin dans une recherche globale. Cette approche ne modélise pas le problème de manière décentralisée, mais l'auteur a utilisé la plateforme A-globe qui représente un environnement distribué pour exécuter ses algorithmes.

(Abdulrab, Babkin, et Satunin 2010) proposent une approche pour la résolution du problème DRT en fusionnant deux différents modèles. Le premier est un modèle multicouche, distribué et hybride basé sur la technique multi-agent et proposé dans (Jin, Abdulrab, et Itmi 2008). Il permet d'améliorer la fiabilité du système en prenant en considération les facteurs stochastiques et les problèmes de temps réel comme le trafic, etc. Le second est un modèle centralisé basé sur l'enchère combinatoire. Il aide à trouver la solution optimale dans une recherche globale. L'hybridation de ces deux modèles améliore l'efficacité et la fiabilité du système.

Quant à (Zidi et al. 2011), ils développent une approche multi-agent centralisée basée sur l'algorithme de recuit simulé multi-objectif (Multi-Objective Simulated Annealing) pour résoudre le problème du TAD statique. L'algorithme de recuit simulé multi-objectif utilise le concept de Pareto-dominance pour donner des solutions optimisant en même temps les deux critères suivants : le coût de voyage total et la qualité de service constitué par la durée de voyage et le nombre de stations visitées. Dans cette approche, trois types d'agents sont distingués : l'agent Client, l'agent Dispatching-centre et l'agent Véhicule. L'idée de cette approche se résume par le suivant : un agent Client envoie une requête de transport à l'agent Dispatching-centre. Ce dernier diffuse la requête vers tous les agents Véhicule du système. Ensuite, chaque agent Véhicule applique l'algorithme de recuit simulé multi-objectif pour trouver le chemin optimal et envoie la réponse à l'agent Dispatching-centre. Lorsque cet agent reçoit toutes les réponses, il choisit l'agent Véhicule qui présente la réponse optimale. Les auteurs ont utilisé le benchmark généré par (Cordeau et Laporte 2003) pour tester l'approche proposée. Ils ont également comparé les résultats de cette approche avec ceux de l'approche proposée par (Cordeau et Laporte 2003) et l'approche proposée par (Cubillos, Urrea, et Rodríguez 2009). Dans la première approche, les auteurs ont utilisé la méthode de recherche taboue pour résoudre le problème du TAD. Dans la deuxième, ils ont utilisé l'algorithme génétique pour résoudre le même problème. Deux critères de comparaison ont été étudiés : la durée totale de la route et le temps total de trajet pour chaque client. L'approche proposée présente de mauvaises solutions par rapport aux deux autres en comparant leur durée totale de la route. Par contre, elle présente de meilleures solutions que la première et de mauvaises solutions que la deuxième en utilisant le temps total de trajet des clients comme critère de comparaison.

De même, une autre approche multi-agents est proposée par (Mguis et al. 2012) pour la résolution du problème de tournées de véhicules dynamique dans une situation d'urgence.

C'est le problème d'établissement d'un ensemble de tournées de véhicules pour servir des clients dans une situation d'urgence comme le cas d'une catastrophe naturelle. Cette approche considère l'aspect dynamique du problème comme l'arrivée d'une nouvelle demande après le démarrage des tournées et l'apparition d'une perturbation. Dans leur article, les auteurs présentent une modélisation de l'approche multi-agents proposée en utilisant la méthodologie O-MaSE : Organization based Multiagents System Engineering (DeLoach 2005). L'architecture SMA proposée est composée de trois types d'agents : l'agent Interface, l'agent Superviseur et l'agent Véhicule. L'agent Interface est chargé de la communication du système avec le monde extérieur, donc c'est lui qui reçoit les demandes des clients et les envoie vers l'agent Superviseur. L'agent Superviseur supervise tout le système. Dans un premier temps, il regroupe les demandes selon différents critères comme leurs types. Dans un deuxième temps, il affecte chaque groupe à un agent Véhicule. Ce dernier utilise l'algorithme génétique pour trouver le chemin le plus court qui servira toutes les demandes affectées à lui en respectant les fenêtres de temps de chacune d'elles. Comme c'est un problème dynamique, alors l'agent Interface peut recevoir une nouvelle demande après le démarrage des tournées. Dans ce cas, il l'envoie à l'agent Superviseur qui à son tour l'envoie aux agents Véhicule. Chaque agent Véhicule calcule le coût d'insertion supplémentaire de cette nouvelle demande dans son chemin et l'envoie à l'agent Superviseur. Ce dernier choisit comme agent Véhicule qui va desservir la demande, celui qui a présenté le meilleur coût parmi les coûts reçus. Ces agents communiquent suivant le protocole *Contract Net Protocol* afin de résoudre le problème.

Nous notons la présence d'un point commun entre ces approches. Ce point commun est l'existence d'un agent central. Ce dernier prend la décision finale et choisit le véhicule qui desservira une nouvelle demande. D'où la notion de la centralisation de ces approches. Les méthodes décentralisées seront présentées dans le paragraphe suivant.

5.3.2 Approches décentralisées

(Cubillos, Guidi-Polanco, et Demartini 2004) proposent un modèle décentralisé basé sur le concept multi-agent pour la résolution du problème DRT. Ce modèle est composé de deux phases : la première est une phase qui s'occupe de l'optimisation. Dans cette phase, les solutions, qui maximisent les intérêts de l'opérateur en prenant compte de la demande du client, sont identifiées. La deuxième est une phase de prise de décision qui se base sur un processus de négociation entre des différents agents. Ce processus de négociation vise à obtenir un résultat qui fait un compromis entre les demandes des clients et l'offre de

l'opérateur tout en intégrant le client dans la prise de décision finale. L'architecture multi-agent est décomposée en deux couches :

- la couche internet : cette couche joue le rôle d'interface entre le monde extérieur et le système. Elle contient :
 - un agent Véhicule associé à chaque véhicule du système ;
 - un agent Client qui agit comme une interface entre le client et le système ;
 - un agent Broker responsable de l'enregistrement des véhicules et de la configuration de leurs descriptions dans la base de données du système. Cela afin de chercher ultérieurement la liste des véhicules les mieux adaptés à une demande d'un client ;
- la couche de planification : cette couche est responsable du traitement des requêtes reçues, de l'attribution des véhicules les plus convenables aux requêtes et de l'établissement de l'horaire pour chaque requête. Elle contient quatre types d'agents :
 - un agent Schedule créé par l'agent Véhicule qui symbolise la planification et l'horaire de chaque véhicule ;
 - un agent Trip-request créé par l'agent Client qui symbolise les spécifications de la tour désirée par le client ;
 - un agent Planner responsable de présenter un ensemble de propositions d'attribution des véhicules pour chaque requête et de choisir la meilleure après négociation avec l'agent Trip-request ;
 - un agent Map qui représente l'espace géographique et ses données. Il possède des informations concernant les distances entre les différents nœuds et le trajet entre deux nœuds.

Une autre approche multi-agent décentralisée dynamique basée sur une auto-organisation et sur un processus de négociation pour résoudre le problème du TAD, a été proposée par (Bertelle et al. 2009). Le but de l'auto-organisation est de faire face au manque de service dans des zones et à la concentration dans d'autres. Cette approche comporte trois types d'agents :

- un agent Client représentant l'utilisateur avec les paramètres de sa demande de transport (point de départ, point d'arrivée, etc.) ;
- un agent Interface ayant comme rôle la diffusion de la demande de l'utilisateur à tous les agents Véhicule situés dans un environnement. Cet environnement symbolise le

réseau urbain qui est modélisé par un graphe pondéré, orienté et découpé en zones dynamiques auxquelles il est rattaché un certain nombre de véhicules ;

- un agent Véhicule associé à chaque véhicule.

Cette approche s'auto-organise pour faire face au manque de service dans des zones et à la concentration dans d'autres. Et cela en déterminant des zones dynamiques couvrant des parties du graphe et auxquelles un certain nombre de véhicules est rattaché. Pour tout véhicule, qui quitte sa zone, il doit payer une pénalité pour toujours se rappeler qu'il doit être rattaché à sa zone. En outre, cette approche se base sur un processus de négociation appliquée par les agents Véhicule afin de prendre la décision finale et préciser le véhicule vainqueur qui va desservir une nouvelle demande. L'idée du processus de négociation est la suivante : à la réception d'une nouvelle demande, chaque agent Véhicule calcule un surcoût dû à l'exécution ou à l'insertion de la demande dans son trajet. Ce surcoût symbolise une fonction d'optimisation composée de trois paramètres qui sont le temps supplémentaire à ajouter pour servir la nouvelle demande, le nombre de clients à bord et la distance pour arriver au client. L'agent Véhicule collecte les réponses des autres agents Véhicule et classe son surcoût par rapport aux surcoûts des autres dans une liste par ordre croissant. Enfin, il diffuse la tête de la liste. L'agent Véhicule le plus fréquemment cité est le vainqueur.

La notion de la décentralisation des travaux de recherche présentés précédemment vient de l'inexistence d'un agent central ayant comme rôle de prendre la décision finale.

Nous passons maintenant à décrire des méthodes utilisées dans la résolution du problème de tournées de véhicules et ses variantes avec considération de l'évolution du trafic.

6 Approches de résolution du VRP et de ses variantes considérant l'évolution du trafic

La plupart des travaux de recherche présentés dans la section 5 considèrent que les temps de voyage (ou les vitesses de voyage) sur un réseau routier sont fixes tout le temps. Il en résulte que le temps de voyage (ou la vitesse de voyage) entre deux endroits est toujours constant. En revanche, le temps du voyage entre deux endroits d'une ville n'est pas le même à deux instants différents. Dans une étude statistique faite au Royaume-Uni, 85 % de la variation et de l'imprécision du temps de voyage nécessaire pour traverser un chemin, est dû à la

congestion du trafic routier (Eglese, Maden, et Slater 2006). Donc, le temps de voyage dépend principalement de la situation dynamique du trafic routier.

La considération de la relation entre la vitesse de voyage (ou le temps de voyage) et la situation du trafic est un facteur principal qui a des effets positifs sur les applications urbaines. L'ignorance de cette relation a des effets négatifs environnementaux, sociaux et économiques. Par ailleurs, elle aboutit à avoir des problèmes dans les opérations de la logistique urbaine. Ces problèmes peuvent se résumer en : (i) la nécessité des heures de travaux supplémentaires pour les chauffeurs et (ii) la diminution de la qualité de service grâce à la violation des fenêtres de temps des clients. (Maden, Eglese, et Black 2010) montrent que la moitié des tournées de véhicules ne respectent pas les heures de travail si le facteur de la variation du temps de voyage n'est pas pris en considération. De plus, d'autres chercheurs comme (Ehmke, Steinert, et Mattfeld 2012) stipulent que la négligence de ce facteur aboutit à 20 % d'erreur dans l'estimation de la durée des routes.

Deux types du trafic routier sont distingués dans la littérature : un trafic déterministe et un trafic stochastique. Le trafic déterministe est causé par des facteurs prévus comme les heures de pointe. Dans ce cas, la variation de la vitesse peut être connue à priori à partir des données historiques. D'autre part, le trafic stochastique est causé par des facteurs imprévus tels qu'un accident, la panne d'un véhicule, les protestations, les conditions météo, etc. Dans ce cas, la situation du trafic ne peut être connue par avance. Elle est connue à l'instant où le facteur apparaît à travers des données réelles.

De nouveaux travaux de recherche considèrent la variation du temps de voyage selon la situation dynamique du trafic dans le problème de tournées de véhicules dans une ville ou TDVRP : "time dependent vehicle routing problem" (Malandraki 1989; Crainic, Ricciardi, et Storchi 2009b; Kim et al. 2015; Cattaruzza et al. 2015). Certaines approches retiennent uniquement les facteurs prévus comme les heures de pointe. Elles se basent sur des données historiques du trafic pour estimer le trafic routier et calculer le temps de voyage. D'autres approches rajoutent les facteurs imprévus et stochastiques et se basent sur des informations du trafic réel. (Hu et Sun 2012) ont proposé un modèle pour la résolution de la distribution urbaine. Ce modèle se base sur les expériences du passé afin de répondre rapidement à toute perturbation. De plus, (Kong et al. 2012) utilisent l'information historique et réelle pour trouver le routage optimal.

Quelle que soit l'approche proposée, elle utilise les méthodes présentées dans la section 5 (méthodes exactes, méthodes approchées ou SMA) pour résoudre le problème de tournées de véhicules et ses variantes tout en considérant l'état du trafic. Les sous-sections suivantes seront consacrées à la présentation des méthodologies utilisées pour la collection, modélisation et stockage des données issues du trafic routier et à la présentation des approches résolvant des problèmes de tournées de véhicules et leurs variantes tout en considérant la situation dynamique du trafic.

6.1 Collecte, modélisation et stockage des données du trafic routier

Des difficultés empêchent les auteurs d'étudier le problème de tournées de véhicules avec des temps de voyage dynamique (TDVRP). La considération de la contrainte des vitesses (ou des temps de voyage) dynamiques complexifient le problème de tournées de véhicules. Sans oublier la difficulté de collecter et de gérer les données du trafic routier (Hill et Benton 1992).

6.1.1 Technologies de collecte des données du trafic routier

Les données historiques du trafic sont utilisées pour prédire la situation du trafic routier. Historiquement, le recueil des données du trafic était fait sous la forme des enquêtes terrains menées auprès des usagers en leur demandant des informations par questionnaire. Ces enquêtes sont parfois encore utilisées, mais restent très ponctuelles. Le recueil des données du trafic est une mission très difficile, coûteuse et nécessite beaucoup de temps. En revanche, au cours des dernières années, le domaine de collecte des données du trafic connaît un progrès technologique considérable. Dans ce qui suit, les principales technologies de capteurs utilisées par les systèmes de recueil de données du trafic sont décrites brièvement.

Trois catégories de capteurs sont distinguées (Guilbert 2015) :

- les capteurs intrusifs qui sont posés sur ou dans la chaussée. Nous distinguons les capteurs à boucles inductives, les capteurs magnétomètres, capteurs à jauges de contraintes, les capteurs à fibres optiques, etc. ;
- les capteurs non-intrusifs ne nécessitent pas de travail sur la chaussée. Nous distinguons les capteurs Laser, les capteurs infrarouge, les capteurs vidéo, etc. Dans les capteurs vidéo, nous trouvons ceux qui sont spécialisés dans la détection automatique des incidents et ceux qui sont spécialisés dans la détection automatique des bouchons. Cela se fait par l'analyse et le traitement des images acquises en temps réel ;

- les capteurs embarqués qui se trouvent à bord des véhicules. Nous distinguons deux types :
 - un véhicule traceur : c'est un véhicule transmettant au moins une information sur sa position grâce à un équipement embarqué. Cet équipement peut être un système de positionnement par satellites pour calculer sa localisation géographique (GPS : Global Positionning System). Ces véhicules fournissent des données appelées données flottantes automobiles connues en termes anglophone par FCD : Floating Car Data ;
 - les réseaux des opérateurs de téléphonie mobile qui fournissent les données des téléphones mobiles FMD : Floating Mobile Data à cause de la localisation des téléphones mobiles dans les cellules du réseau GSM lors de leurs événements. Ces données peuvent fournir des informations utiles pour la connaissance et la gestion du trafic.

6.1.2 Modélisation et stockage des données du trafic routier

Le stockage et la consolidation des données historiques issues du trafic routier présentent des difficultés importantes (Cattaruzza et al. 2015). Pour cela, plusieurs méthodes ont été proposées. Dans (Ehmke et Mattfeld 2010), les données sont stockées de manière homogène en les regroupant en fonction de la variation des vitesses quotidiennes. (Van Woensel et al. 2008) présentent quant à eux une approche de file d'attente pour obtenir les vitesses sur un arc à partir des données du trafic routier dans un problème de tournées de véhicules avec des temps de voyage dynamiques. (Ehmke et Mattfeld 2011) ont proposé une approche basée sur les techniques de l'entrepôt et de la fouille de données pour filtrer et analyser les données énormes du trafic routier. Les auteurs collectent des données automobiles flottantes FCD. Ces FCD proviennent des capteurs embarqués dans un véhicule traceur. L'utilisation de ces techniques aboutit à une réduction importante du volume de données collectées (Ehmke, Meisel, et Mattfeld 2012). De plus, les temps de voyage sont produits et sauvegardés dans une mémoire de manière efficace sans perdre la qualité et la fiabilité du routage.

6.2 Approches considérant l'évolution déterministe du trafic

Les approches décrites par (Ichoua, Gendreau, et Potvin 2003; Fleischmann, Gietz, et Gnutzmann 2004; Eglese, Maden, et Slater 2006) résolvent le problème TDVRP : Time Dependent Vehicle Routing Problem. C'est le problème de tournées de véhicules considérant la variation des temps du voyage. Ces approches ont uniquement retenu les données

historiques du trafic afin d'identifier le temps du voyage à un instant donné. Sur la même route à deux instants différents, il est possible d'avoir deux temps de voyage différents. Le problème de la variation des temps de voyage dû à des événements prévus a été résolu par la décomposition d'une journée en des intervalles de temps. A chaque intervalle, une vitesse constante est assignée. Quand un véhicule traverse une route, sa vitesse varie à chaque fois qu'il passe d'un intervalle de temps à un autre.

Les auteurs (Ichoua, Gendreau, et Potvin 2003) présentent une approche pour résoudre le problème de TDVRP statique et dynamique, tout en garantissant qu'un véhicule qui part à un temps t_1 va arriver à un temps t_2 plus petit que le temps d'arrivée d'un autre qui part du même point, mais dans un temps t_3 plus grand que t_1 . Pour arriver à cette garantie, il faut toujours prendre la variation de la vitesse en considération tout au long du voyage d'un véhicule. En passant par deux intervalles de temps dans lesquels la vitesse varie, il faut recalculer le temps de voyage de la nouvelle vitesse ou bien la garantie sera ratée. Dans cette approche, ils ont adapté, d'une part, la méthode de recherche taboue parallèle utilisée par (Taillard et al. 1997) pour résoudre le TDVRP statique. D'autre part, ils ont adapté l'heuristique utilisée par (Gendreau, Michel et al. 1999) pour résoudre le TDVRP dynamique. Enfin, ils ont montré par des expérimentations faites sur des données virtuelles que la prise en considération de la variation des vitesses en fonction du temps aboutit à obtenir de meilleurs résultats par rapport à ceux obtenus par les modèles présentés par (Taillard et al. 1997) et (Gendreau, Michel et al. 1999). Ces derniers considèrent que la vitesse de voyage est toujours constante.

(Fleischmann, Gietz, et Gnutzmann 2004) expliquent le mécanisme de calcul des temps de voyage sur les chemins en se basant sur des systèmes d'informations du trafic modernes. En outre, des expérimentations basées sur des données réelles de temps de voyage, dérivées d'un système d'information du trafic à Berlin, ont été effectuées. Les résultats expérimentaux ont montré que la déconsidération de la variation des temps de voyage dans un problème de tournées de véhicules aboutit à une sous-estimation de 10 % des temps de voyage et à la violation de 10 % des fenêtres de temps des clients.

L'approche proposée par (Eglese, Maden, et Slater 2006) est basée sur la catégorisation des jours dans le but de rassembler ceux qui ressemblent en termes d'état du trafic quotidien. Selon le trafic routier, une route peut avoir une à plusieurs catégories de jour et un jour d'une année appartient à une des catégories des jours de la route. La journée de chaque catégorie du

jour est discrétisée en des intervalles de temps hétérogènes couvrant la variation de l'état du trafic de la route durant les 24 heures de la journée. Chaque intervalle de temps contient une vitesse de voyage fixe. Après la catégorisation des jours et la fixation d'une vitesse pour chaque intervalle de temps, les auteurs ont construit une base de données très immense dénommée *Road Timetable*. Ils se basent sur l'idée qu'il y a moins de 5 % de différence entre les vitesses de voyages prises sur la même route pendant le même temps en un même jour pour construire la table *Road Timetable*. Cette base de données contient le temps de voyage minimum et le chemin le plus court en termes de temps entre deux localisations selon le temps, le jour et la semaine d'une année. Suite à des expérimentations faites en utilisant des informations réelles de l'infrastructure au nord-ouest de l'Angleterre, les auteurs ont démontré que l'utilisation des informations concernant le trafic, la *Road Timetable* et la variation du temps de voyage aboutit à obtenir des résultats respectant les fenêtres de temps des clients et à éviter que les chauffeurs font des heures de travail supplémentaires.

L'approche proposée par (Eglese, Maden, et Slater 2006) a été utilisée par (Maden, Eglese, et Black 2010) pour mettre en œuvre un système de distribution des marchandises à UK. Ce système a été construit en se basant sur une heuristique et en créant la *Road timetable* conçue par (Eglese, Maden, et Slater 2006). Il sert à récupérer le plus court trajet en termes de temps entre les clients et de les affecter aux véhicules. Egalement, ce système a comme objectif de minimiser le temps total nécessaire pour livrer des marchandises d'un dépôt vers un ensemble des clients en respectant plusieurs contraintes comme la capacité des véhicules et les contraintes temporelles pour les clients, les véhicules et les chauffeurs. Dans les expérimentations, les auteurs ont montré que l'utilisation de la table *Road timetable* aboutit à réduire de 7 % l'émission de CO₂.

6.3 Approches considérant l'évolution stochastique du trafic

Les approches décrites dans la section précédente résolvent le problème de tournées de véhicules et ses variantes en considérant que les vitesses de voyage varient selon le temps à cause des situations déterministes du trafic. Les variations sont connues à priori et ne sont pas influencées par des événements imprévus qui peuvent survenir sur un chemin comme un accident. Dans cette section, nous passons à présenter des approches résolvant le problème de tournées de véhicules et ses variantes en considérant les situations stochastiques du trafic.

(Kim, Lewis, et White 2005) étudient l'effet de la prise en considération des informations du trafic en temps réel sur l'optimalité de tournées de véhicules dans un réseau routier incertain.

Ce réseau se particularise par la présence de certains chemins avec un temps de voyage instable et qui varie selon le temps. En outre, des procédures de prise de décision basées sur le processus de décision markovien ont été développées pour déterminer pour chaque conducteur d'un véhicule le temps optimal de sa présence et le temps optimal de son départ de chaque point du réseau selon la situation du trafic routier. Effectivement, un ensemble de chemins du réseau routier est observé continuellement en temps réel pour déterminer l'état de la congestion pour chacun d'eux. Par suite, en se basant sur les données historiques et réelles de la situation du trafic de ces chemins, le temps de départ optimal de certains points et celui de points suivants à visiter ultérieurement sont indiqués jusqu'à la fin du voyage pour chaque conducteur. Les résultats des instances générées sur un réseau du monde réel montrent la nécessité de l'utilisation des informations du trafic en temps réel. Ces informations aboutissent à réduire le nombre de véhicules utilisés jusqu'à 6.88 % et le coût jusqu'à 3.65 % par rapport aux résultats obtenus dans un modèle ne prenant que les données historiques du trafic.

L'approche proposée par (Potvin, Xu, et Benyahia 2006) traite le problème de routage et d'ordonnancement des véhicules qui ramassent des courriers depuis les localisations des clients et les apportent à un dépôt central. Dans cette approche, les auteurs supposent que la communication entre le centre de distribution et les chauffeurs se déroule justement au temps où les chauffeurs sont dans une des localisations des clients et cela pour qu'ils sachent leur destination suivante. Le point faible de cette approche réside dans le fait que le chauffeur n'arrivera probablement pas à sa destination suivante en temps voulu au cas où un facteur imprévu apparaîtrait en cours de route. Dans ce cas, le temps de tolérance représentant le retard maximal d'arrivée à la destination est dépassé. A cause de ce retard, le centre ne reçoit pas un rapport concernant l'arrivée du chauffeur à sa destination au temps convenable. A ce moment, le centre change la planification des trajets des véhicules et envoie un autre chauffeur à cette destination pour remédier à ce retard et de servir le client dans un temps convenable. Ainsi, il est très possible que ces deux chauffeurs viennent pour servir le même client en même temps.

(Lorini, Potvin, et Zufferey 2011) ont amélioré l'approche proposée par (Potvin, Xu, et Benyahia 2006) en prenant en compte deux contraintes : (1) la réception d'une nouvelle requête et (2) la variation du temps du voyage en fonction du temps entre deux localisations des clients. Ces deux contraintes parviennent après que les véhicules commencent leur chemin planifié. Pour traiter la première contrainte, les auteurs ont amélioré le protocole de

communication entre le centre et les chauffeurs. Le centre communique avec les chauffeurs tout au long de leur voyage entre les localisations des clients pour connaître leurs localisations géographiques. Lorsque la limite du temps de tolérance maximum d'un client est dépassée, le centre peut demander au chauffeur concerné de faire une déviation de sa destination courante vers une autre destination pour servir un autre client. Il demande également à un autre chauffeur approprié de se diriger vers la destination du premier. En cas de réception d'une nouvelle requête, le centre communique avec le chauffeur convenable pour faire une déviation de sa destination courante et servir la nouvelle requête. Concernant la deuxième contrainte, le temps de voyage varie selon des facteurs prévus comme les heures de pointe et des facteurs imprévus comme les accidents. Les auteurs ont pris les facteurs prévus en considération durant le calcul du temps du voyage entre deux points. Ils ont décomposé la journée en trois tranches et ils ont affecté à chaque tranche un temps de voyage constant. Afin de représenter le retard dû à un facteur imprévu, ils ont ajouté au temps de voyage une valeur symbolisant une perturbation dynamique et générée en se basant sur la loi normale.

(Schilde, Doerner, et Hartl 2014) proposent des méthodes basées sur des heuristiques pour traiter le DARP : dial-a-ride problem avec des temps de voyages dynamiques stochastiques. L'effet d'un accident sur le trafic routier est considéré dans ces méthodes. Les auteurs s'appuient sur des statistiques dérivées des informations historiques concernant les accidents pour déterminer l'influence de cet événement sur le temps de voyage. De plus, ils ont utilisé la technique proposée par (Lecluyse, Sörensen, et Peremans 2013) pour estimer la sévérité de l'accident sur la situation du trafic routier en termes de temps et d'espace (surface). Dans ces approches, le facteur stochastique est considéré pendant la planification et le calcul des temps de voyage des chemins des véhicules afin de ne pas violer les contraintes temporelles. De plus, les auteurs imposent une limite de temps à ne pas dépasser dans la tournée de chaque client. L'objectif des approches proposées est de minimiser l'insatisfaction des clients, le nombre de véhicules utilisés et la durée totale du chemin.

(Groß et al. 2015) ont utilisé le modèle de l'intervalle de temps de voyage (ITT : Interval Travel Time) pour prévoir le temps de voyage entre deux places dans le problème de routage. Chaque lien reliant deux places est pondéré par un intervalle de temps de voyage ITT. La borne inférieure de l'intervalle de temps représente le temps de voyage entre les deux places dans le pire des cas et la borne supérieure représente le temps de voyage entre les deux places dans le meilleur des cas. Les différents temps de voyage sur un lien sont générés par la loi de distribution shifted gamma. Ensuite, la borne inférieure de chaque ITT est déterminée par le

quantile à 5 % et la borne supérieure par le quantile à 95 %. L'objectif de cette approche est de minimiser la durée de voyage total et le retard de chaque client.

Dans ce qui suit, nous effectuons un survol rapide sur les différentes plateformes et logiciels de résolution du problème du TAD.

7 Plateformes et logiciels résolvant le TAD

Plusieurs plateformes ou services résolvant le TAD ont été mises en œuvre. Certaines d'entre elles ont été commercialisées dans plusieurs villes dans différents pays comme les États-Unis, Finlande, Belgique, Italie, France, etc. Ces plateformes ou services s'appuient sur un smartphone, une tablette ou un ordinateur pour sélectionner le véhicule et réserver celui qui convient le plus au client. Dans les sous-sections suivantes, nous présentons un ensemble de services et des plateformes résolvant le TAD.

7.1 Covoiturage

Le covoiturage, un service qui ressemble au TAD, a été introduit dans les années 1950 dans plusieurs pays occidentaux tels que Taxistop en Belgique, Allostop créée en France, en 1958 et Allo-Stop au Canada. Le service de covoiturage existe partout dans le monde comme les États-Unis⁶, le Canada⁷, la Suisse⁸, etc. Il consiste à partager un même véhicule par un conducteur non professionnel et d'autres passagers dans le but d'effectuer tout ou une partie d'un trajet en commun. Depuis les années 1980, il représente une alternative aux transports classiques. La vulgarisation de l'Internet a beaucoup contribué à l'émergence du covoiturage en facilitant les contacts entre conducteurs et passagers. Contrairement au transport classique où le passager choisit sa destination, en covoiturage, c'est le conducteur qui propose de partager son véhicule avec les passagers. Un conducteur offre aux passagers de les ramener à bord de sa voiture pour une partie de son trajet ou pour le trajet complet qu'il doit lui-même effectuer. Ainsi, la date, le lieu de départ et l'heure sont généralement déterminés à l'avance, et s'avèrent les mêmes pour tous les passagers. À l'arrivée, le conducteur peut laisser les passagers là où lui-même s'arrête. Il peut aussi déposer ses passagers à proximité d'un transport en commun ou bien là où un membre de leurs familles ou un ami pourront les prendre en charge.

⁶ <https://www.carpoolworld.com>

⁷ <https://www.covoiturage.ca/>

⁸ <http://www.e-carpooling.ch/index.php/en/>

L'entreprise BlaBlaCar est le leader mondial du covoiturage. Elle propose une plateforme payante pour le covoiturage dans la majorité des pays occidentaux tels que le Royaume-Uni⁹, la France¹⁰, l'Allemagne, l'Italie, la Belgique, le Brésil¹¹, le Portugal, l'Espagne, la Turquie et le Mexique. Via le site web de l'entreprise, elle permet de mettre en relation des conducteurs et des passagers souhaitant partager un trajet et les frais associés. Les conducteurs publient leurs places disponibles et les passagers les achètent en ligne. À la base, la mise en relation des conducteurs et des passagers a été gratuite pendant de nombreuses années. Mais, à partir de 2011, elle est devenue payante (15 % du prix du covoiturage).

7.2 Mobirouter

Le logiciel *MobiRouter* a été développé en Finlande par *Mobisoft Oy* et il est commercialisé et rendu opérationnel au Royaume-Uni par *Mobisoft*. Ce logiciel fonctionne en temps réel et reçoit les demandes de transport des clients jusqu'à 15 minutes avant le temps de départ. Ensuite, il réalise l'ordonnancement nécessaire avant l'expédition des véhicules convenables chez les clients. La sélection des véhicules proposés à un client donné est basée sur : (i) les exigences particulières des clients prises au téléphone, (ii) les lieux et les horaires de départ et d'arrivée, (iii) le temps calculé du voyage, et (iv) les engagements faits avec les autres clients. Une fois la sélection du véhicule terminée, le conducteur du véhicule sélectionné reçoit automatiquement les clients et les détails de voyage à travers son écran d'affichage du véhicule. Lorsqu'un client appelle pour réserver un voyage, l'agent rentre les détails du voyage et le logiciel attribue automatiquement le meilleur véhicule en offrant des alternatives lorsque cela est possible.

Les principales étapes opérationnelles du *MobiRouter* sont la saisie des demandes de transport, la définition du service et la définition du véhicule. La saisie d'une demande se fait à l'aide d'un écran affichant la carte numérique qui assiste à la localisation des adresses et de voir les positions du véhicule dans toutes les fonctions. De plus, les exigences des clients sont automatiquement envoyées au conducteur. Des messages spécifiques au sujet d'un client peuvent également être envoyés pour aider le conducteur. *MobiRouter* permet de réaliser des réservations de groupe le rendant facile à réserver de grands groupes tels que les enfants des écoles ou des participants à une activité de jour.

⁹ <https://www.blablacar.co.uk/>

¹⁰ <https://www.blablacar.fr/>

¹¹ <https://www.blablacar.com.br/>

La définition du véhicule permet de faire entrer dans la base de données du logiciel, de grandes flottes de véhicules comme les minibus, les taxis, etc. Les capacités peuvent être définies en fonction de l'offre de service, par exemple fauteuil roulant, porte-vélos. Des capacités multiples peuvent être définies pour refléter les configurations de véhicules différents. *MobiRouter* peut également être utilisé pour planifier les taxis. Une variété de coûts de fonctionnement peut être associée aux taxis et l'algorithme du logiciel indiquera qui est le moins cher pour un voyage donné. Le logiciel peut également indiquer automatiquement la possibilité du partage d'un taxi.

Quant à la définition du service, une variété de prestations peut être définie pour le même véhicule, variant d'heure en heure ou au jour le jour. Toute combinaison de services adaptés à la demande peut être définie; y compris «fixe», «semi-fixe» ou «voie libre». De plus, il permet de concevoir des zones de service en indiquant le temps de voyage entre les points sélectionnés ou les zones d'équidistance à partir d'un point central. Les aires de service et les zones sont définies en utilisant un simple outil de dessin sur la carte numérique.

7.3 LITRES-2

LITRES est un projet de recherche et de développement qui a été lancé en 1993 en Australie. Il vise à développer une technologie concernant la simulation et l'ordonnancement de nouvelles formes de transport public tel que le transport à la demande. Le premier simulateur LITRES (1993-94) a été appliqué dans les études de l'ordonnancement pour Canberra, Sydney et Perth. La version la plus récente, appelée LITRES-2 (1996-97), distingue clairement entre les modules de la simulation et de l'ordonnancement d'une part, et l'ordonnanceur conçu pour une utilisation autonome dans les applications en temps réel dans le transport à la demande, d'autre part. Les méthodes utilisées ici permettent la superposition d'une grande variété de configurations spatiales, temporelles et socio-économiques des demandes de voyage. Ces demandes de voyages sont passées du simulateur à l'ordonnanceur. Un ordonnanceur central est nécessaire pour répondre aux demandes de voyages, et de déployer efficacement les véhicules. LITRES-2 comprend deux modules intégrés pour deux types critiques : L2sched et L2jplan. Le module L2sched contrôle le déploiement d'une flotte de véhicules de tourisme répondant à la demande. Les fonctions incluent la réservation dynamique, l'ordonnancement, la répartition et le routage. Des heuristiques sont utilisées avec l'objectif de minimiser les coûts d'exploitation des véhicules. Ce module dispose d'applications potentielles en tant que système d'ordonnancement en temps réel pour une flotte

de véhicules de tourisme, ou pour des services de livraison ou pour le service de distribution de courrier. Quant au module L2jplan, il émule les processus de prise de décision des clients potentiels. Cela implique de minimiser pour chaque client le coût total du voyage. L2jplan dispose d'applications potentielles en tant que composant d'une information ou réservation de gestion de services en temps réel.

7.4 Uber

Uber¹², le nom d'origine du service UberCab, est un service application mobile visant à mettre en contact des utilisateurs avec des conducteurs réalisant des services de transport. L'application est lancée dans la ville californienne de San Francisco, aux États-Unis en 2010 sur iOS et Android. Par la suite, l'entreprise s'est étendue progressivement et l'application est commercialisée dans plus de 488 villes dans le monde. Les services proposés par Uber sont particulièrement utilisés dans les périodes où les moyens de transport traditionnels sont indisponibles comme durant la nuit ou le week-end. Dès que le client soumet sa requête de transport, la plate-forme est capable de suivre les coordonnées GPS du client, même si ce dernier ne connaît pas sa localisation. En quelques minutes, un conducteur d'un véhicule Uber le prendra en charge. Le client à son tour est en mesure de suivre l'arrivée du conducteur Uber et reçoit un message textuel confirmant l'arrivée du conducteur. Finalement, le conducteur ne dispose pas du numéro de téléphone du client, mais il est en mesure de communiquer avec lui s'il est incapable de le trouver. La plate-forme de Uber utilise des algorithmes et des "heat maps" pour estimer le nombre d'utilisateurs potentiels qui existent dans une zone, qui correspond alors au nombre de conducteurs présents dans une ville. Uber est capable d'analyser des statistiques importantes comme le nombre de fois que l'application smartphone est ouverte afin d'évaluer les potentielles demandes de clients pour les taxis. En conséquence, des délais d'attente pour les clients sont fournis et les pilotes peuvent être plus efficaces dans l'utilisation de leur temps libre.

7.5 SAMPO

L'objectif du projet SAMPO (System for Advanced Management for Public Transport Operations) est de mettre en place et de tester un système de transport à la demande à *Tuusula* en Finlande et dans d'autres villes européennes en Belgique, en Italie et en Suède. Le projet a été lancé entre 1996 et 1997 et a été poursuivi en 1998 et 1999 dans le cadre du projet

¹² <https://www.uber.com/>

SAMPLUS (System for Advanced Management for Public Transport Operations). L'objectif du SAMPO est de combler les lacunes géographiques dans les services publics de transport existant en Finlande pour garantir l'égalité d'accès aux transports en commun. Ce système sert à optimiser le transport des groupes ayant des besoins particuliers comme les personnes âgées ou handicapées qui ne sont pas en mesure d'accéder aux transports publics. Il peut être utilisé pour améliorer ou établir des transports publics flexibles dans les zones rurales ou peu denses. D'un point de vue technique, le système conçu dispose d'un centre d'expédition central avec un ordinateur où est installé le logiciel du transport à la demande développée. Les clients appellent le centre d'expédition central pour réserver les voyages simples ou pour organiser des voyages réguliers au moins trois heures à l'avance. Le système SAMPO recommande les meilleures alternatives combinant bus et services de taxi. La flotte de transport, qui comprend des bus, des minibus et des taxis, possède des dispositifs de communication pour recevoir les informations concernant les passagers et les itinéraires des voyages.

8 Conclusion

Dans ce chapitre, nous avons présenté le problème du TAD comme une variante du problème célèbre de tournées de véhicules VRP. Plusieurs approches de résolution ont été générées pour résoudre ces types de problèmes. Les méthodes exactes, les heuristiques, les métaheuristiques et les systèmes multi-agents ont joué un rôle essentiel dans la résolution de tels problèmes. Nous avons vu aussi, que récemment, la situation du trafic dans la résolution de ces problèmes a commencé à être prise en considération. Enfin, nous avons présenté différentes plateformes et logiciels créés pour résoudre le problème du TAD.

Chapitre 3. Approche multi-agent décentralisée basée sur l'algorithme de A* pour le transport à la demande

1 Introduction

Le problème que nous cherchons à étudier et à concevoir est le problème du transport à la demande (TAD). Le TAD offre un transport public, économe et flexible. Comme nous l'avons vu, il permet le transport de plusieurs personnes sur un même trajet et il est moins cher que le taxi. En outre, il se déclenche par les demandes des clients. Selon ces demandes, les arrêts, les horaires et les itinéraires des véhicules sont déterminés.

Dans ce chapitre, le problème du TAD est modélisé et résolu par un système multi-agent décentralisé. Les agents communiquent entre eux en se basant sur une variante décentralisée de l'algorithme A* (algorithme de recherche de chemin avec minimum coût dans un graphe) pour trouver une solution optimale à chaque demande de transport. La solution optimale consiste à chercher la trajectoire la plus rapide menée par les véhicules les moins chers dans un réseau routier statique. Ce réseau est caractérisé par des vitesses de voyage constantes et indépendantes du temps.

Ce chapitre est structuré comme suit. Dans la deuxième section, nous présentons le scénario statique du problème du TAD. La section 3 contient un petit rappel de l'algorithme A* et de ses propriétés. La section 4 présente le cadre formel du problème du TAD en décrivant la structure de données de chacun de ses composants essentiels. Dans la cinquième section, nous exhibons la modélisation du problème du TAD par les systèmes multi-agents (SMA) en décrivant les différents types d'agents. La sixième section explique l'utilisation des agents de l'algorithme A* de manière décentralisée pour traiter une demande de transport. La section 7 montre les expérimentations faites pour tester l'approche proposée. La dernière section conclut le chapitre.

2 Premier scénario du problème du TAD : *scénario statique*

Le premier scénario du TAD est intitulé *scénario statique*. L'idée de ce scénario se résume comme suit : des clients sont dispersés partout dans une ville. Cette ville est caractérisée par

un réseau routier ayant des vitesses de voyage constantes. Chaque client formule une demande de transport pour qu'un ou plusieurs passagers soient amenés d'un lieu de départ à un temps de départ au plus tôt vers un lieu d'arrivée à un temps d'arrivée au plus tard. Il indique également le nombre maximal de véhicules par lesquels il souhaite être transporté (le nombre maximal de correspondances). L'objectif du système du TAD est de trouver la solution optimale qui répond à la demande du client. Cette solution se traduit par la trajectoire la plus rapide menée par les véhicules les moins chers. Elle doit respecter les contraintes du client (le temps de départ au plus tôt, le temps d'arrivée au plus tard et le nombre maximal de véhicules exigés) et des véhicules (le nombre de places vides et les contraintes temporelles concernant leurs engagements avec d'autres clients) et minimiser les deux critères suivants : la durée totale des trajectoires et leur coût. A cette fin, nous proposons une approche basée sur les systèmes multi-agents (SMA) et sur l'algorithme A*.

3 Rappel de l'algorithme A*

L'algorithme A* est une extension de l'algorithme de Dijkstra (Dijkstra, E.W. 1959). C'est un algorithme de recherche du chemin à coût minimum dans un graphe, mais avec une heuristique. Chaque nœud n dans le graphe est évalué par une fonction d'évaluation $f(n)=g(n)+h(n)$. Cette fonction d'évaluation est le coût total estimé du chemin passant par le nœud n pour se rendre au but. $g(n)$ est le coût du nœud de départ jusqu'au nœud n et $h(n)$ est l'heuristique qui est égale au coût estimé du nœud n jusqu'au but.

3.1 Propriétés de A*

L'algorithme A* (Hart, Nilsson, et Raphael 1968) est admissible lorsqu'il utilise une heuristique admissible. Une heuristique est admissible lorsqu'elle ne surestime jamais le coût pour atteindre le but. En d'autres termes, pour tout nœud n du graph, $h(n) \leq h'(n)$ où $h(n)$ est l'heuristique de n et $h'(n)$ est le véritable coût pour se rendre de n au but.

L'algorithme A* admissible se caractérise par les propriétés suivantes :

- la complétude : l'algorithme A* est complet, il garantit de trouver une solution si elle existe ;
- l'optimalité : l'algorithme A* est optimal, il garantit de trouver la solution optimale ;
- la complexité en temps de l'algorithme A* est exponentielle. Elle est égale à $O(b^d)$ où b est le facteur de ramification et d est la longueur (profondeur) de la solution.

La preuve de détails peut être trouvée dans (Hart, Nilsson, et Raphael 1968).

4 Structure de données des composants du système du TAD du scénario statique

Dans cette section, nous décrivons la structure de données des quatre composants fondamentaux du système du *scénario statique* : (1) l'infrastructure de la ville ; (2) la demande de transport du client ; (3) le véhicule ; (4) et la trajectoire. Nous définissons ensuite le problème du TAD et son objectif.

4.1 Infrastructure statique

L'environnement physique du système du TAD du *scénario statique* est une infrastructure statique. Cette infrastructure symbolise un réseau routier d'une ville caractérisé par des vitesses de voyage constantes et indépendantes du temps.

Pour représenter l'infrastructure statique, nous nous basons sur les conventions standards de la théorie des graphes. Nous définissons l'infrastructure par un graphe $G=(P, R)$. $P = \{p_1, \dots, p_n\}$ est l'ensemble de nœuds routiers désignant les lieux de départ et d'arrivée possibles des clients dans la ville et $R = \{r_1, \dots, r_n\}$ est l'ensemble d'arcs désignant les routes reliant les lieux de la ville. Chaque route $r_i = p_i \xrightarrow{(t_i, d_i)} p_j$ relie deux lieux p_i et p_j et elle est pondérée par le temps moyen fixe pour la traverser t_i et par sa distance d_i . La figure 11 illustre l'exemple d'une infrastructure statique. Nous remarquons que les deux lieux "a" et "b" sont reliés par une route de distance 25 km et le temps requis pour la parcourir est 30 min.

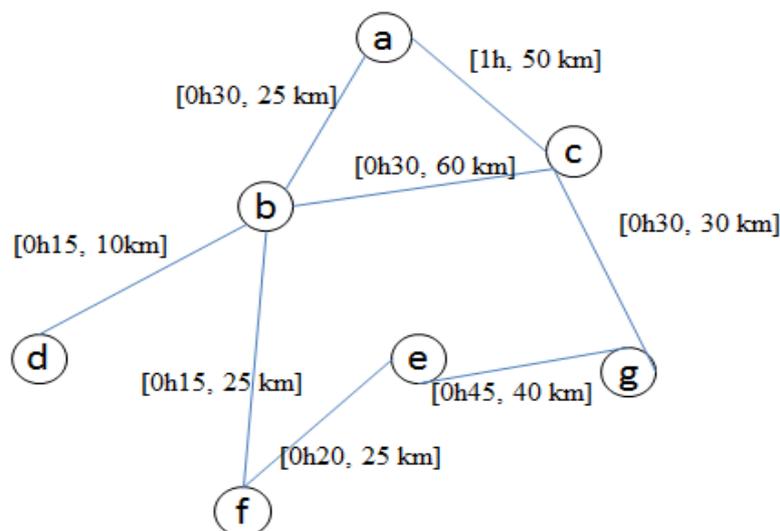


Figure 11 : Exemple d'infrastructure statique

Pour représenter la distance et le temps moyen fixe de chaque route, nous utilisons la notion de matrice. Ainsi, deux matrices statiques sont identifiées : la première matrice contient les temps moyens de parcours entre deux lieux du réseau et la deuxième représente les distances les séparant.

Le tab 1 représente la matrice contenant la distance en km entre deux lieux de l'infrastructure représentée dans la figure 11. Le tab 2 représente la matrice contenant le temps le plus court (en minutes) entre deux lieux de cette infrastructure.

Nœuds	A	B	C	D	E	F	G
A	0	25	50	35	75	50	80
B	25	0	60	10	50	25	90
C	50	60	0	70	70	85	30
D	35	10	70	0	60	35	100
E	75	50	70	60	0	25	40
F	50	25	85	35	25	0	65
G	80	90	30	100	40	65	0

Tab 1 : Matrice des distances en km

Nœuds	A	B	C	D	E	F	G
A	0	30	60	45	65	45	90
B	30	0	30	15	35	15	60
C	60	30	0	45	65	45	30
D	45	15	45	0	50	30	75
E	65	35	65	50	0	20	45
F	45	15	45	30	20	0	65
G	90	60	30	75	45	65	0

Tab 2 : Matrice des temps en minutes

4.2 Demande de transport

La demande de transport d'un client est définie par :

$$D = \left((dep, t_{dep}) \xrightarrow{n_{passagers}} (arr, t_{arr}), n_{véhicules} \right) \text{ où}$$

- dep et arr sont les lieux de départ et d'arrivée ;
- t_{dep} représente le temps de départ au plus tôt ;
- t_{arr} représente le temps d'arrivée au plus tard ;
- $n_{passagers}$ représente le nombre de passagers à transporter ;

- $n_{véhicules}$ représente le nombre maximum de changement de véhicules qui doit être effectué pour transporter les passagers.

Pour mieux comprendre les différents paramètres, voici l'exemple d'une demande $D1$:

$$D1 = \left((A, 08h00) \xrightarrow{2} (D, 09h00), 3 \right)$$

$D1$ est une nouvelle demande de transport où deux passagers veulent partir du lieu "A" au plus tôt à 08h00 et veulent arriver au lieu "D" au plus tard à 09h00. Ils veulent être servis par trois véhicules au maximum.

Notons que le temps de départ, le temps d'arrivée et le nombre de changement de véhicules indiqués par le client sont des exigences rigides.

4.3 Véhicule

Nous supposons que tous les véhicules du système du TAD sont homogènes en termes de type et de nombre de sièges. Un véhicule est défini par 5-uplet (nom, capacité, garage, coût, table de planification) où :

- nom est le nom du véhicule ;
- capacité représente le nombre de sièges du véhicule ;
- garage est la place où le véhicule est garé chaque jour ;
- coût est le coût par kilomètre de l'utilisation du véhicule ;
- la table de planification est utilisée pour maintenir l'occupation du véhicule par jour et pour résumer son emploi du temps quotidien. Cette table aide à connaître, à chaque instant, le lieu du véhicule qui peut être un lieu ou une route et le nombre de passagers à bord.

Chaque ligne de la table de planification est définie par le triplet (t, lieu, |passagers|) où :

- t est le temps ;
- lieu est le lieu du véhicule au temps t ;
- |passagers| est le nombre de passagers au temps t.

Ceci est l'exemple d'un véhicule v_1 : (v_1 , 4, "A", 0,3 euro, tab 3). Le véhicule v_1 a une capacité de 4 passagers. Il se gare chaque jour à "A" et coûte 0.3€ par kilomètre. La table de planification du véhicule v_1 est représentée par le tab 3. Au moment de 0h00, v_1 se trouve dans son garage "A". A 07h00, le nombre de passagers est égal à 3. L'information sur le

véhicule au moment de 07h20 peut être facilement déduite du tab 3. A 07h20, v_1 est sur la route allant de "A" → "B" et le nombre de passagers est 3. À 07h45, le véhicule dépose le passager au point "D" et prend la route "D" → "B" → "A" pour arriver à son garage vers 08h30 sans passagers.

temps	Lieux	passagers
0h00	"A"	0
07h00	"A"	3
07h30	"B"	2
07h45	"D"	0
08h00	"B"	0
08h30	"A"	0

Tab 3 : Exemple d'une table de planification

4.4 Trajectoire

La trajectoire est définie dans le système du TAD du *scénario statique* par :

$$trajectoire = \left((dep, t_{dep}) \xrightarrow{v_1} \dots (x_i, t_{x_i}) \xrightarrow{v_i} \dots (arr, t_{arr}) \right) \text{ où :}$$

- $chemin = (dep \rightarrow \dots x_i \rightarrow \dots arr)$ est la liste des lieux et des routes traversés par les véhicules pour atteindre le lieu d'arrivée arr à partir du lieu de départ dep . Les lieux d'un chemin sont classés en deux types : un lieu principal et un lieu transitoire. Les lieux de départ et d'arrivée d'une demande sont des lieux principaux. Tandis que les lieux reliant deux lieux principaux sont des lieux transitoires ;
- t_{dep} est le temps de départ du nœud dep , t_{x_i} est le temps de départ du nœud x_i (ou le temps de début du parcours de la route $x_i \xrightarrow{(t_i, d_i)} x_{i+1}$) et t_{arr} est le temps d'arrivée au nœud arr ;
- v_i est le véhicule qui emmène les passagers sur la route $x_i \xrightarrow{(t_i, d_i)} x_{i+1}$.

En supposant que $x_1 = dep$ et $x_n = arr$, nous définissons :

- $temps(traj) = \sum_{i=1}^{n-1} temps(x_i \rightarrow x_{i+1}) = \sum_{i=1}^{n-1} t_i$ le temps nécessaire pour traverser les routes de la trajectoire ;
- $coût(traj) = \sum_{i=1}^{n-1} coût(v_i) * distance(x_i \rightarrow x_{i+1}) = \sum_{i=1}^{n-1} coût(v_i) * d_i$ la somme totale des coûts pour traverser les routes de la trajectoire.

4.5 Problème du TAD

Soient une infrastructure $G=(P; R)$ et un ensemble de véhicules $V=\{v_1, \dots, v_n\}$.

Le problème du TAD est défini par une demande de client :

$$D = \left((dep, t_{dep}) \xrightarrow{n_{passagers}} (arr, t_{arr}), n_{véhicules} \right)$$

Une trajectoire $traj = \left((dep, t_{dep'}) \xrightarrow{v_1} \dots (x_i, t_i) \xrightarrow{v_i} \dots (arr, t_{arr'}) \right)$ est une solution valide de la demande D si :

- $t_{dep'} \geq t_{dep}$ et $t_{arr'} \leq t_{arr}$;
- $t_{i+1} > t_i$ avec $i \in [1, \dots, n-1]$;
- $\forall v_i \in traj, pVide(v_i)_{[t_i, t_{i+1}]} \geq n_{passagers}$ où $pVide(v_i)_{[t_i, t_{i+1}]}$ représente le nombre de places vides de v_i durant $[t_i, t_{i+1}]$;
- $n_{véhiculesEchan} \leq n_{véhicules}$ où $n_{véhiculesEchan}$ est le nombre de véhicules échangés¹³.

Dans ce cas, nous pouvons affirmer que $traj$ satisfait D .

Nous remarquons que le temps de départ t_{dep} et le temps d'arrivée t_{arr} de la demande sont deux contraintes temporelles considérées comme des contraintes fortes et elles doivent être respectées par le véhicule qui va servir la demande. Donc, la possibilité qu'une demande soit refusée par le système est très probable lorsque tous les véhicules du système ne réussissent pas à respecter ces contraintes temporelles.

4.6 Objectif du système du TAD

L'objectif du système du TAD du *scénario statique* est de trouver la trajectoire optimale satisfaisante à la demande du client. La trajectoire optimale est la plus rapide menée par les véhicules les moins coûteux. A cette fin, nous étudions deux critères d'optimisation : le temps et le coût.

Soit $Traj = \{traj_1, \dots, traj_n\}$ l'ensemble de trajectoires qui satisfont une demande D . $traj_i \in Traj$ est optimale en temps si et seulement si : $\forall traj_j \in Traj \setminus \{traj_i\}, temps(traj_i) \leq temps(traj_j)$. Alors, $traj_i$ est la trajectoire la plus rapide. $traj_i \in Traj$ est optimale en coût si et seulement si : $\forall traj_j \in Traj \setminus \{traj_i\}, coût(traj_i) \leq coût(traj_j)$. Alors, $traj_i$ est la trajectoire la moins coûteuse.

Ainsi, une trajectoire est une solution pour une demande D , si elle satisfait D et si elle est optimale.

¹³ Nous notons que sortir d'un véhicule v_i pour prendre un autre v_j puis reprendre v_i est considéré comme 3 changements de véhicules

Demandes	Trajectoires	Coûts
$d_1 = \left(('A', 7h) \xrightarrow{1} ('D', 7h45), 3 \right)$	$('A', 7h) \xrightarrow{v_1} ('B', 7h30) \xrightarrow{v_1} ('D', 7h45)$	10.5 €
$d_2 = \left(('D', 8h) \xrightarrow{1} ('F', 8h45), 1 \right)$	$('D', 8h) \xrightarrow{v_2} ('B', 8h15) \xrightarrow{v_2} ('F', 8h30)$	10.5 €
$d_3 = \left(('C', 9h) \xrightarrow{2} ('E', 10h05), 2 \right)$	$('C', 9h) \xrightarrow{v_1} ('B', 9h30) \xrightarrow{v_2} ('F', 9h45)$ $\xrightarrow{v_2} ('E', 10h05)$	33 €

Tab 4 : Solutions des demandes

Le tab 4 présente des trajectoires optimales et leurs coûts pour des demandes de transport.

5 Modélisation multi-agent du problème du TAD

Nous avons décidé d'utiliser le paradigme multi-agent comme outil de modélisation et de résolution du problème du transport à la demande (TAD). Ce problème est dynamique, physiquement distribué (véhicules, clients, lieux, routes, trajectoires, etc.), fonctionnellement hétérogène (la recherche du plus court chemin, la vérification de la disponibilité des véhicules, le lancement de la demande, la réflexion de l'état du trafic, etc.), caractérisé par des contraintes complexes (contraintes spatio-temporelles des clients et des véhicules) et confrontant les exigences de la vie réelle. Les méthodes exactes sont une solution inappropriée pour résoudre un tel problème. Elles sont incapables en temps de traiter une version réelle ou de grande taille de ce type de problème. De même, l'intelligence artificielle centralisée et la recherche opérationnelle sont incapables de donner un algorithme efficace pour résoudre un problème de routage et d'ordonnancement dynamique. Tandis que les techniques de l'intelligence artificielle distribuée démontrent toute leur puissance dans la résolution des problèmes dynamiques (Bonabeau, Dorigo, et Theraulaz 1999). Les systèmes multi-agents, qui font partie de l'intelligence artificielle distribuée, constituent une solution appropriée pour résoudre le problème de tournées de véhicules dynamique (Shah 2012).

Les systèmes multi-agents (SMA) permettent de réduire la complexité de la résolution d'un problème en divisant le savoir nécessaire en sous-ensembles, en associant un agent intelligent indépendant à chacun de ces sous-ensembles et en coordonnant l'activité de ces agents (Ferber 1995). Ces agents communiquent entre eux, échangent les informations correspondantes, planifient pour travailler ensemble et enfin, analysent et synthétisent leurs solutions afin de réaliser un but particulier. Par conséquent, les systèmes multi-agents sont capables de résoudre la complexité des problèmes de transport engendrée par l'explosion de l'espace de

recherche à cause du grand nombre de véhicules, des demandes, etc. Ils répartissent le traitement, la charge de calcul et la prise de décision sur plusieurs agents en tenant compte de la nature de la distribution du problème. Cela aide à pallier au problème de la complexité et à donner une solution satisfaisante dans un temps acceptable pour les demandes des clients.

Bien que nous avons démontré l'efficacité des systèmes multi-agents à résoudre des systèmes de transport intelligents (comme les systèmes du TAD), la difficulté réside de savoir comment construire un modèle à base d'agents pour un tel problème afin de profiter au maximum des avantages du SMA. Si nous distribuons la résolution du problème tout en gardant les mêmes procédures que les méthodes classiques alors ceci augmente le temps d'exécution des demandes sans aucune amélioration dans la qualité des solutions (Zeddini 2009). Il faut bien identifier des entités (agents) intelligents autonomes, modéliser leurs interactions et définir leurs buts individuels tendant à améliorer les performances globales du système et à résoudre le problème du TAD.

5.1 Architecture générale

Nous modélisons le problème du TAD par un système multi-agent décentralisé (El Falou et al. 2014b). La décentralisation signifie l'inexistence d'un agent central s'occupant de la prise de décision finale, de la gestion des demandes de transports et de la réservation des véhicules et de leurs chemins.

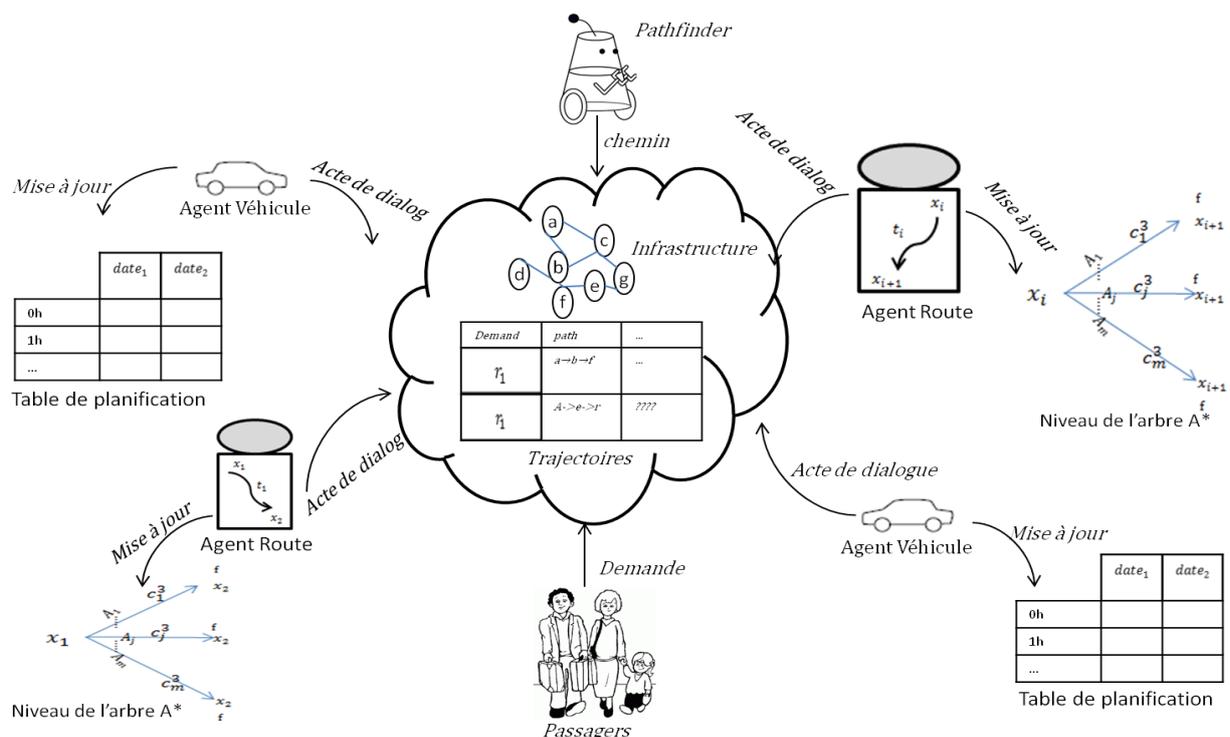


Figure 12 : Architecture multi-agent du scénario statique

Différents types d'agents sont identifiés. Ces agents interagissent ensemble afin de trouver une solution optimale pour le *scénario statique* du problème du TAD. L'architecture du système multi-agent est illustrée dans la figure 12. Dans le centre de la figure, nous montrons l'espace de dialogue. Les discussions des agents se déroulent autour des véhicules et des trajectoires à adopter pour servir les demandes de transport.

5.2 Types d'agents

Nous identifions quatre types d'agents :

Agent Client : il agit comme une interface entre les clients et le système du TAD. Il envoie une demande de transport au système et reçoit une trajectoire optimale, si elle existe.

Agent Pathfinder : cet agent est responsable de trouver le chemin le plus rapide entre le lieu de départ et le lieu d'arrivée d'une demande. Un certain nombre d'algorithmes de recherche graphique classique ont été développés pour résoudre ce genre de problème sur un graphe pondéré. L'agent Pathfinder applique l'algorithme de Dijkstra sur le graphe représentant l'infrastructure statique du système du TAD.

Agent Route : il est associé à une route d'un chemin $r_i = x_i \xrightarrow{(t_i, d_i)} x_{i+1}$. Il demande à chaque agent Véhicule de calculer le coût pour traverser sa route et collabore avec d'autres agents Route pour trouver la trajectoire optimale (la trajectoire la plus rapide menée par les véhicules les moins chers).

Agent Véhicule : cet agent est associé à chaque véhicule du système du TAD. Son rôle principal est d'affirmer à un agent Route s'il est capable de transporter des passagers à travers sa route et lui transmet le coût de transport.

6 Stratégie d'interaction entre les agents pour traiter une demande de transport

Etant donné que le système est décentralisé et distribué, les différents agents communiquent entre eux afin de trouver la trajectoire optimale pour une demande de transport.

Soit D une demande de transport lancée :

$$D = \left((dep, t_{dep}) \xrightarrow{n_{passagers}} (arr, t_{arr}), n_{véhicules} \right)$$

La figure 13 illustre le cycle de vie de la demande D . L'idée principale du cycle de vie est la suivante. L'agent Client reçoit la demande D et l'envoie à l'agent Pathfinder. Ce dernier cherche le chemin le plus rapide reliant le lieu de départ dep et le lieu d'arrivée arr en utilisant l'algorithme de Dijkstra. Un agent Route Ar_i est associé à chaque route r_i du chemin. L'agent Pathfinder envoie ce chemin vers les agents Route associés à ses routes. Ces agents Route coordonnent avec les agents Véhicule du système du TAD en se basant sur l'algorithme A* décentralisé, afin de trouver la trajectoire optimale. En effet, chaque agent Route Ar_i demande, à tour de rôle, à l'ensemble des agents Véhicule du système, de calculer le coût de parcours de sa route $r_i = x_i \xrightarrow{(t_i, d_i)} x_{i+1}$. Puis, il évalue chaque nœud x_{i+1} qu'il atteint. Cette étape fournit une partie du graphe illustré dans la figure 15. Il donne son rôle par la suite à un autre agent Route choisi en se basant sur une heuristique. Cette opération est répétée jusqu'à soit atteindre le nœud objectif arr et trouver la trajectoire optimale, soit échouer.

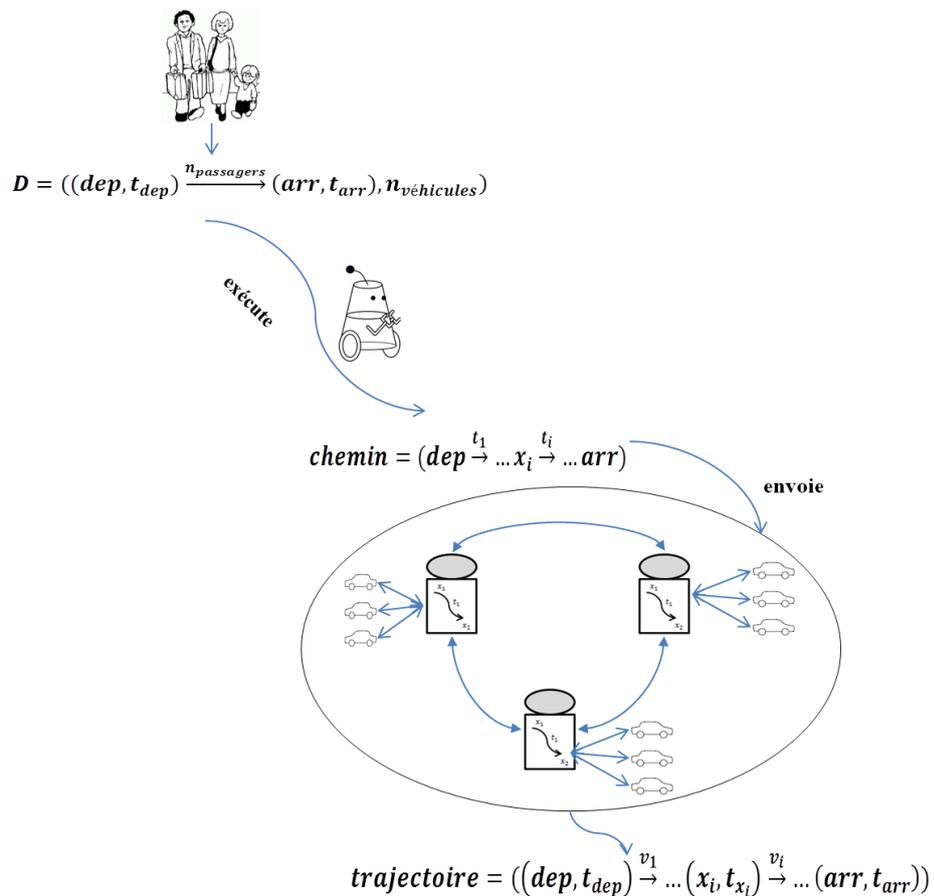


Figure 13 : Cycle de vie d'une demande de transport

Dans ce qui suit, nous détaillons la procédure de coordination entre les agents Route et Véhicule afin de trouver la trajectoire optimale visée.

6.1 Coordination entre les agents Route et Véhicule

Quand une nouvelle demande arrive, l'agent Client l'envoie à l'agent Pathfinder. Ce dernier cherche le chemin le plus rapide entre le lieu de départ et le lieu d'arrivée de la demande en appliquant l'algorithme de Dijkstra sur l'infrastructure statique. Ce chemin est composé d'un ensemble de routes à traverser. Chaque route est associée à un agent Route. Les agents Route associés aux routes du chemin le plus rapide et l'ensemble des agents Véhicule du système se coordonnent ensemble pour trouver la trajectoire optimale à cette demande. En d'autres termes, ils se coordonnent pour trouver les véhicules avec minimum coût qui peuvent transporter les clients sur les routes du chemin le plus rapide tout en respectant le nombre de changement de véhicules demandé par le client. Cette coordination se base sur l'algorithme A* décentralisé et elle est divisée en deux phases : la phase d'initialisation et la phase de planification. La phase d'initialisation vise à calculer l'heuristique de l'algorithme A* décentralisé. La phase de planification vise à trouver la trajectoire optimale. Ces deux phases sont expliquées dans les deux sous-sections suivantes. Dans la troisième sous-section, nous présentons les propriétés de l'algorithme A* décentralisé.

6.1.1 Phase d'initialisation

Le but de la phase d'initialisation est de calculer une heuristique admissible pour l'algorithme A* décentralisé lancé pour chaque demande de transport. Cette heuristique sera utilisée lors de la phase de planification.

Par définition, une heuristique admissible est une heuristique qui ne surestime jamais le coût pour atteindre le but. En d'autres termes, l'heuristique de tout nœud du graphe est plus petite ou égale au vrai coût pour se rendre de ce nœud au but.

Pour que l'heuristique de l'algorithme A* décentralisé soit admissible, elle doit représenter le coût minimal pour arriver au but à partir de n'importe quel nœud du graphe. Pour cette raison, nous supposons que l'heuristique d'un nœud n est égale à la somme des minimums des coûts proposés par les véhicules pour le parcours de chaque route du chemin reliant le nœud n avec le nœud d'arrivée. Notons que le nombre maximal de changement de véhicules $n_{véhicules}$ exigé par le client, n'est pas pris en considération dans le calcul de l'heuristique.

La figure 14 explique la méthode de calcul de l'heuristique. Chaque agent Route Ar_i du chemin demande aux agents Véhicule du système de vérifier s'ils peuvent exécuter sa route

$r_i = x_i \xrightarrow{(t_i, d_i)} x_{i+1}$ à partir d'un temps t_{x_i} . Chaque agent Véhicule Av_j vérifie si le nombre de

places vides, entre le temps de début du parcours de la route t_{x_i} et celui de fin ($t_{x_i} + t_i$), est plus grand ou égal au nombre de passagers à transporter. Et, il envoie sa réponse à l'agent Route Ar_i . S'il a des places, la réponse sera le coût d'exécution c_i^j qui est égal à la distance de la route d_i multipliée par le coût du véhicule v_j . Dans le cas contraire, la réponse sera "-" qui signifie que cet agent Véhicule Av_j ne peut pas exécuter la route r_i . Ar_i sauvegarde toutes les réponses obtenues des agents Véhicule dans un vecteur $C_i=[c_i^1, \dots, c_i^m]$. Puis, il calcule le minimum de C_i $\min(C_i)=c_i^j$ où c_i^j représente le coût minimal proposé par l'agent Véhicule Av_j pour parcourir la route r_i .

Une fois que tous les agents Route ont calculé le coût minimal du parcours de leur route, Ar_n l'agent associé à la route $r_n=(x_n \rightarrow x_{arr})$ initialise son heuristique $h_n=\min(C_n)$. Ensuite, il transmet cette valeur à Ar_{n-1} . Ce dernier initialise son heuristique h_{n-1} par la valeur $\min(C_{n-1}) + h_n$, ($h_{n-1} = \min(C_{n-1}) + h_n$) et ainsi de suite jusqu'à calculer l'heuristique h_2 de l'agent Route Ar_2 associé à $r_2=(x_2 \rightarrow x_3)$.

Soit $H=\{h_2, \dots, h_n\}$ l'heuristique de l'algorithme A* décentralisé.

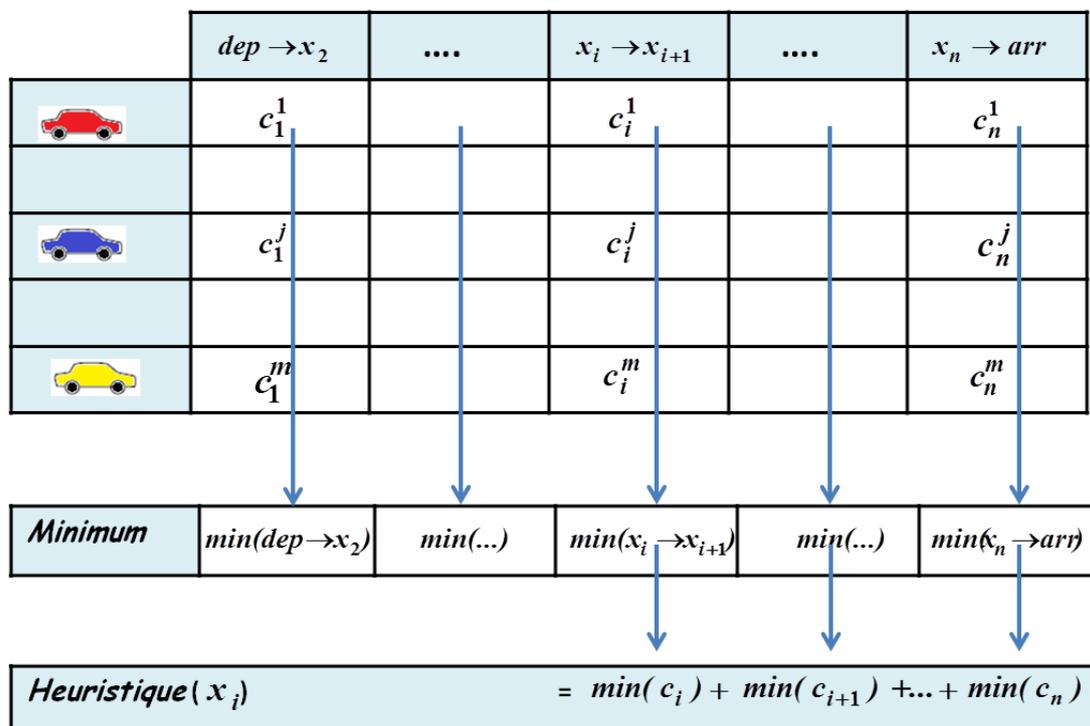


Figure 14 : Calcul de l'heuristique de l'algorithme A* décentralisé

6.1.2 Phase de planification

Après le calcul de l'heuristique, la phase de planification commence. Dans cette phase, les agents Route associés aux routes du chemin et les agents Véhicule du système interagissent ensemble afin de trouver la trajectoire optimale. A cette fin, ces agents suivent une stratégie de coordination décentralisée basée sur l'algorithme A*. Cet algorithme est une recherche arborescente qui s'effectue en utilisant l'heuristique calculée dans la phase d'initialisation et un jeton partagé entre les agents Route pour trouver la trajectoire visée de manière décentralisée. En effet, l'algorithme A* décentralisé est modélisé par un arbre distribué entre les agents Route. Il n'y a ni contrôle ni mémorisation centraux sur l'arbre de A* décentralisé. Chaque agent Route du chemin est responsable de développer un niveau de l'arbre. Donc, pour le bon fonctionnement de l'algorithme A* décentralisé, un jeton partagé entre les agents Route existe. Ce jeton permet de définir à chaque stade du développement de l'arbre A* décentralisé, l'agent Route suivant¹⁴ qui va explorer l'arbre. Ce jeton est représenté par un vecteur $F = \{f_2, \dots, f_n\}$ généré par l'agent Route Ar_1 associé à la première route $r_1 = dep \rightarrow x_2$. Ce vecteur a une dimension égale au nombre de routes du chemin moins un. Chaque agent Route, hormis Ar_1 , correspond à une entrée du jeton (Ar_2 correspond à l'entrée f_2). Une entrée f_i du vecteur jeton contiendra la plus petite valeur (la valeur minimale) atteinte par une fonction f appliquée aux nœuds inexplorés x_i de l'arbre : $f_i = \text{minimum } f(x_i)$. En effet, chaque nœud de cet arbre est évalué selon une fonction heuristique f appelée *distance-plus-coût*. Cette fonction détermine l'ordre dans lequel les nœuds de l'arbre de A* décentralisé seront explorés. La fonction heuristique du nœud x_i est $f(x_i) = g(x_i) + h(x_i)$ où :

- $g(x_i)$ est la somme des coûts proposés par les véhicules pour aller du nœud de départ *dep* jusqu'à x_i calculé grâce à l'arbre de A* décentralisé. Dans la figure 15, le coût c_i^j proposé par le véhicule Av_j pour l'exécution de la route r_i est placé en son milieu. Par exemple, $g(x_4)$ de x_4 encerclé est égal à $c_1^j + c_2^1 + c_3^1$;
- $h(x_i)$ est l'estimation heuristique de la distance entre le nœud x_i et le nœud d'arrivée *arr*. Elle est définie par le coût minimal pour atteindre le nœud d'arrivée *arr* à partir du nœud x_i lorsque la contrainte du nombre maximal de changement de véhicules exigé par le client n'est pas considérée. En d'autres termes, $n_{véhicules}$ de la demande

¹⁴ Notons qu'à un temps t , il n'y a qu'un seul agent Route qui développe un niveau de l'arbre. Cela afin d'éviter les problèmes de redondance, etc.

de transport sera considéré égal à ∞ . $h(x_i)=h_i$ où h_i est une valeur extraite du vecteur H (l'heuristique de l'algorithme A* décentralisé) défini ci-dessus.

Sur la figure 15, $f(x_i)$ est placé au-dessus de x_i . Par exemple, $f(x_4)$ est égal à $g(x_4)+h_4$.

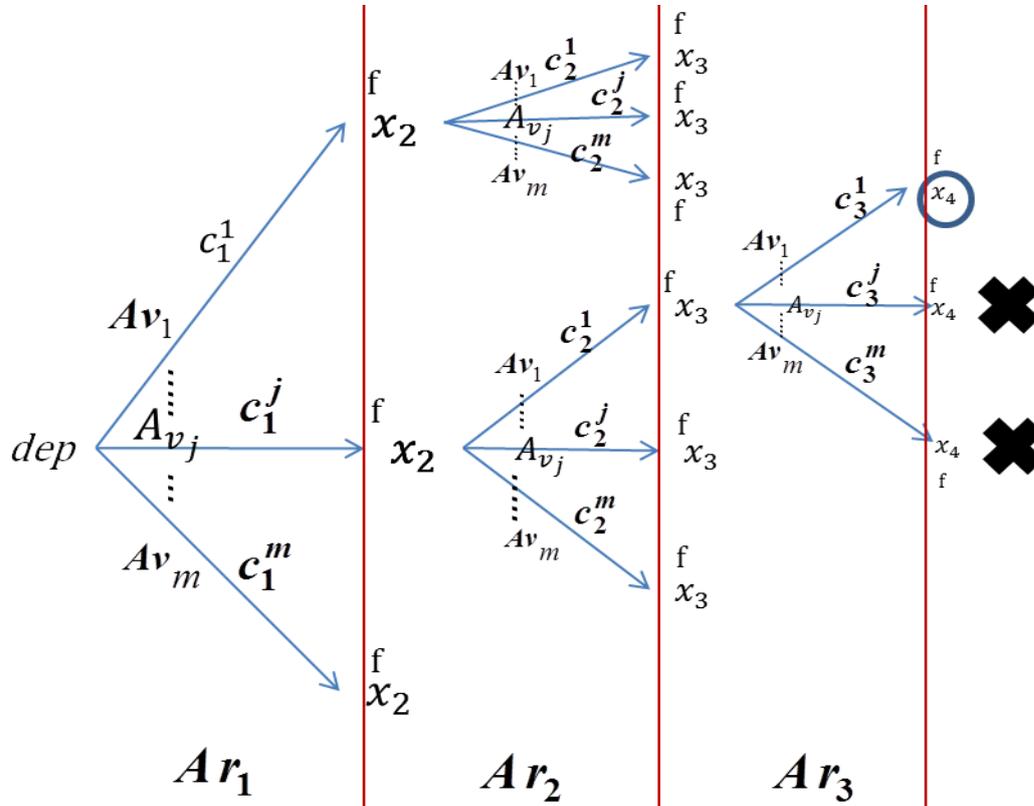


Figure 15 : Exploration de l'arbre de A* décentralisé

Après la définition de l'heuristique, du jeton et de la fonction d'évaluation de chaque nœud, nous passons maintenant à l'explication du fonctionnement de l'exploration de l'arbre de A* décentralisé.

Exploration de l'arbre de A* décentralisé

Afin de trouver la trajectoire optimale, il faut explorer les nœuds de l'arbre de A* décentralisé. Et cela, en se basant sur l'heuristique calculée et le jeton. Ce dernier va contenir, dans chacune de ses entrées, le minimum de la fonction f appliquée aux nœuds inexplorés de chaque route.

Dans la première phase de planification, l'agent Route Ar_1 commence à contacter chaque agent Véhicule Av_j . Il lui demande le coût d'exécution de $r_1 = dep \rightarrow x_2$ à partir du moment initial t_{dep} comme le montre la figure 16. Nous obtenons le premier niveau du graphe de la figure 15. Puis, il applique la fonction heuristique $f(x_2)$ sur les nœuds inexplorés x_2 atteints

par chaque véhicule. Il assigne ensuite à la première entrée du jeton f_2 , la valeur minimale de $f(x_2)$. Enfin, Ar_1 contacte Ar_2 en lui notifiant que c'est son rôle de développer le nœud x_2 et lui transmet le jeton. En faisant cela, l'agent Route Ar_1 effectue la première phase de planification représentée par la figure 16.

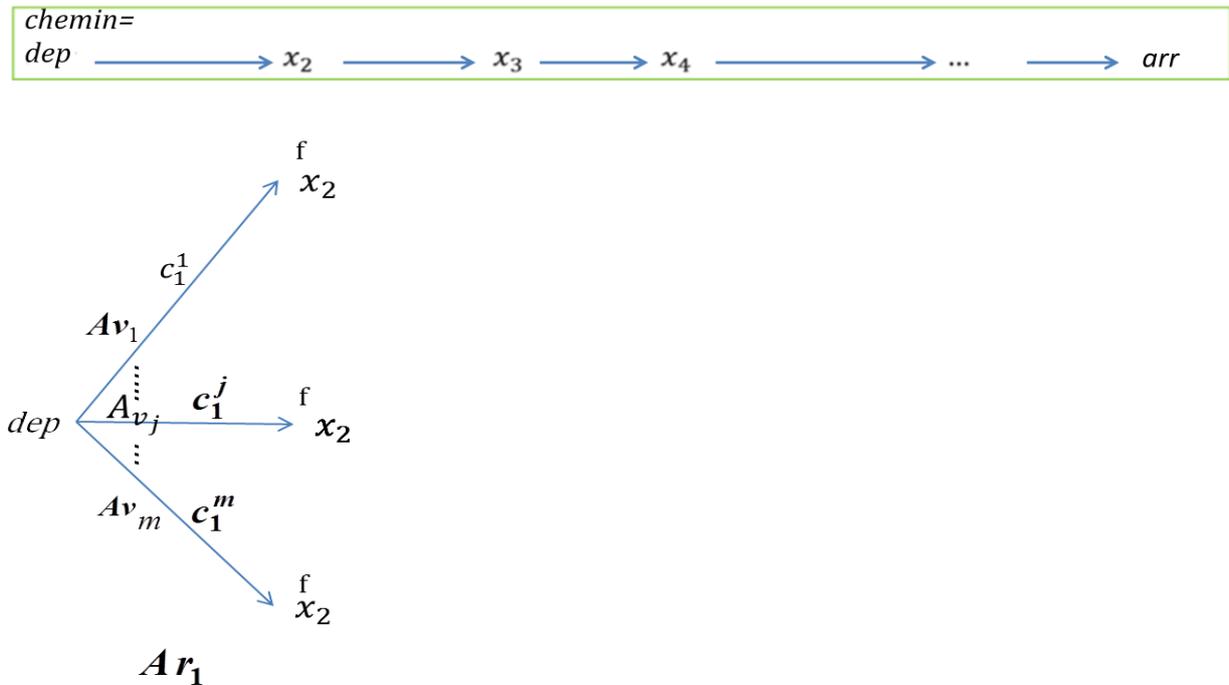


Figure 16 : Première phase de planification

Dans la deuxième phase de planification, l'agent Route Ar_2 explore le nœud x_2 ayant $f(x_2) = f_2$. La figure 17 montre le développement de l'arbre durant cette phase en supposant que le nœud x_2 encerclé a $f(x_2) = f_2$. Le nœud inexploré devient ainsi exploré. L'agent Ar_2 doit remplacer la valeur f_2 du jeton par le nouveau minimum de la fonction f appliquée aux nœuds x_2 inexplorés.

A la i -ème phase de planification, supposons que l'agent Ar_i (Ar_3 dans la figure 18) est notifié de développer un des nœuds x_i . Il développe celui ayant $f(x_i) = f_i$ (c'est le x_3 encerclé dans la figure 18). Ensuite, il remplace f_i par le minimum de la fonction f appliquée aux nœuds inexplorés x_i . Puis, il contacte chaque agent Véhicule Av_j , et lui demande de calculer le coût d'exécution de sa route $r_i = x_i \rightarrow x_{i+1}$ à partir de t_{x_i} le moment de l'arrivée à x_i . De même, Ar_i doit informer chaque agent Véhicule à propos de sa participation pour atteindre x_i . Par exemple, sur la figure 18, au cours de l'exploration du nœud x_3 encerclé, Ar_3 doit informer Av_j et Av_1 qu'ils se sont engagés à exécuter r_1 à partir de t_1 et r_2 à t_2 respectivement. Cette

information permet à l'agent Véhicule de connaître ses engagements qui sont nécessaires pour avoir une trajectoire valide.

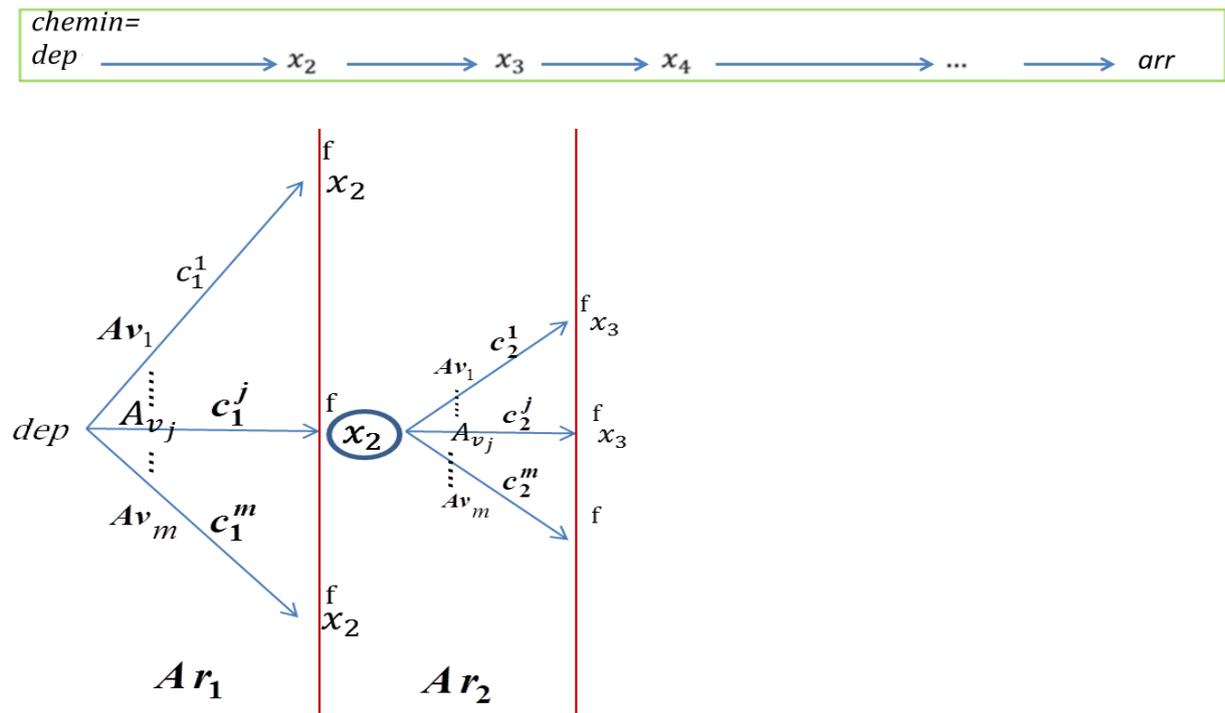


Figure 17 : Deuxième phase de planification

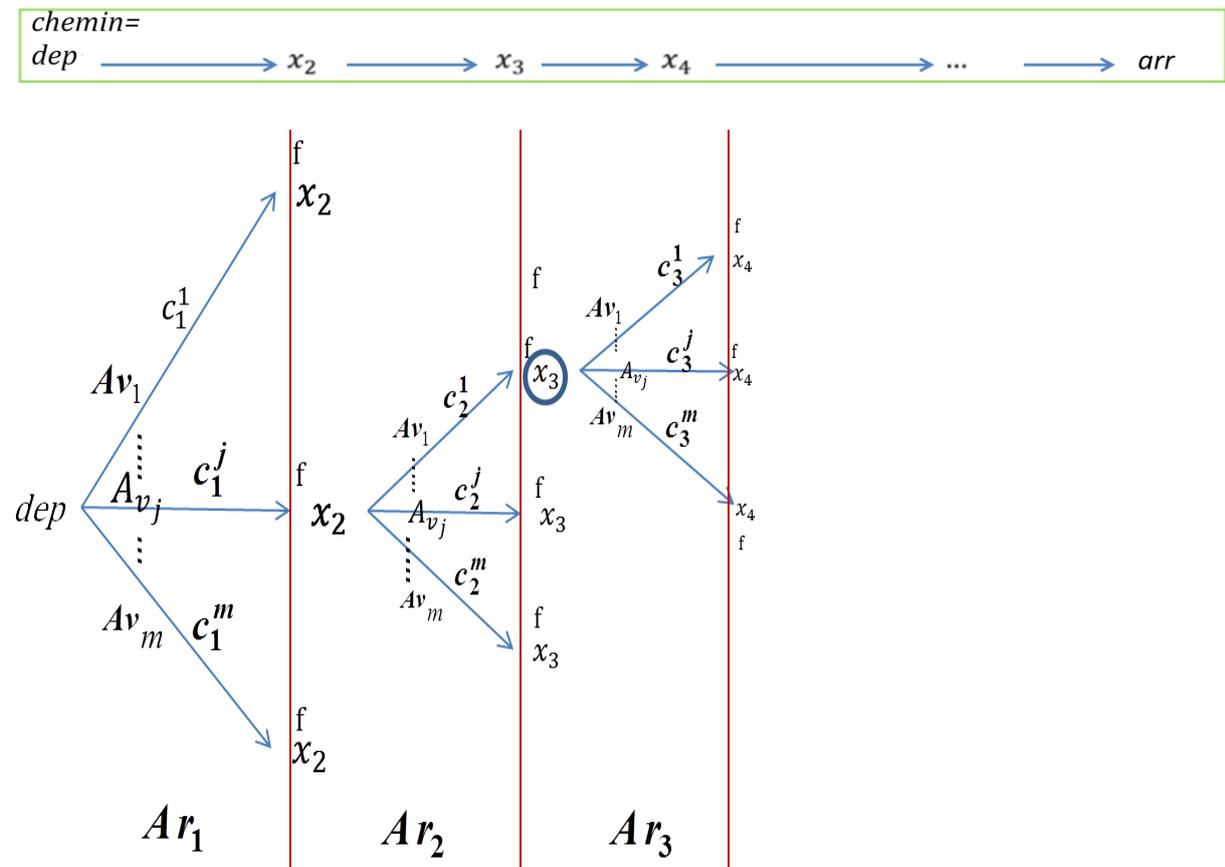


Figure 18 : Troisième phase de planification

Après avoir reçu le coût d'exécution de la route r_i de chaque agent Véhicule, Ar_i développe x_i et calcule les valeurs $f(x_{i+1})$ pour les nœuds x_{i+1} atteints par chaque véhicule. Ensuite, il calcule le minimum de la fonction $f(x_{i+1})$ et l'assigne à sa place dans la $(i+1)^{\text{ème}}$ entrée du vecteur de jeton F (minimum $f(x_{i+1})=f_{i+1}$).

Basé sur le vecteur jeton F, Ar_i doit choisir l'agent Route qui devrait continuer à développer l'arbre A* décentralisé à partir d'un nœud inexploré. Ce serait l'agent Route Ar_k qui correspond à l'entrée f_k possédant la valeur minimale du vecteur jeton F. Ce serait l'agent Ar_2 comme l'indique la figure 19.

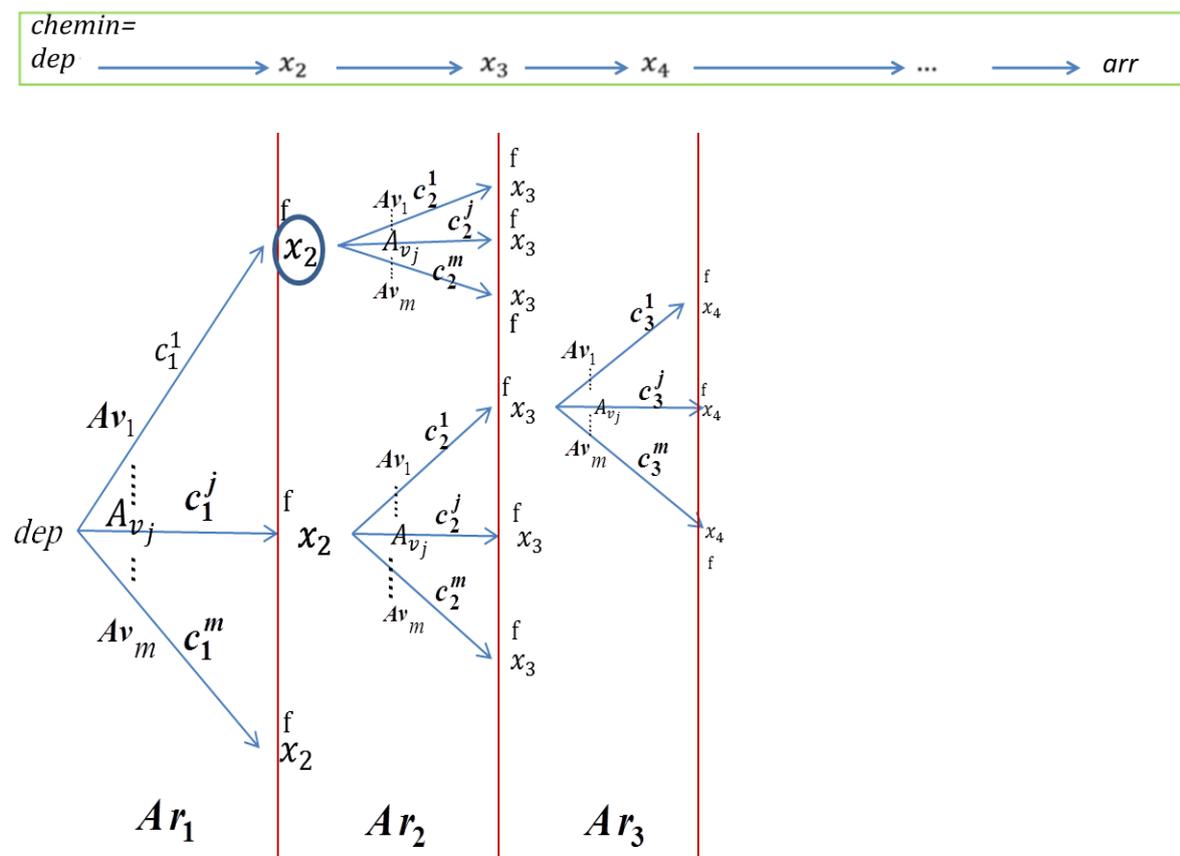


Figure 19 : Quatrième phase de planification

Ar_i transmet le vecteur jeton F à Ar_k et le notifie de son tour de développer le nœud minimisant $f(x_k)$. Cette phase de planification est répétée jusqu'à atteindre le but ou l'échec.

Une adaptation finale de A* décentralisé est d'ajouter un facteur d'élagage basé sur le nombre maximal de changement de véhicules $n_{véhicules}$ déterminé par le client. Un nœud x_i atteint avec un nombre d'agents Véhicule supérieur à $n_{véhicules}$ est élagué comme le montre la figure 20. En d'autres termes, l'exploration à partir de ce nœud est arrêtée. Tout agent Véhicule utilisé pour atteindre chaque nœud est placé à l'extrémité de la route. Soit $n_{véhicules}=2$ (voir la

chemin=
 $dep \longrightarrow x_2 \longrightarrow x_3 \longrightarrow x_4 \longrightarrow \dots \longrightarrow arr$

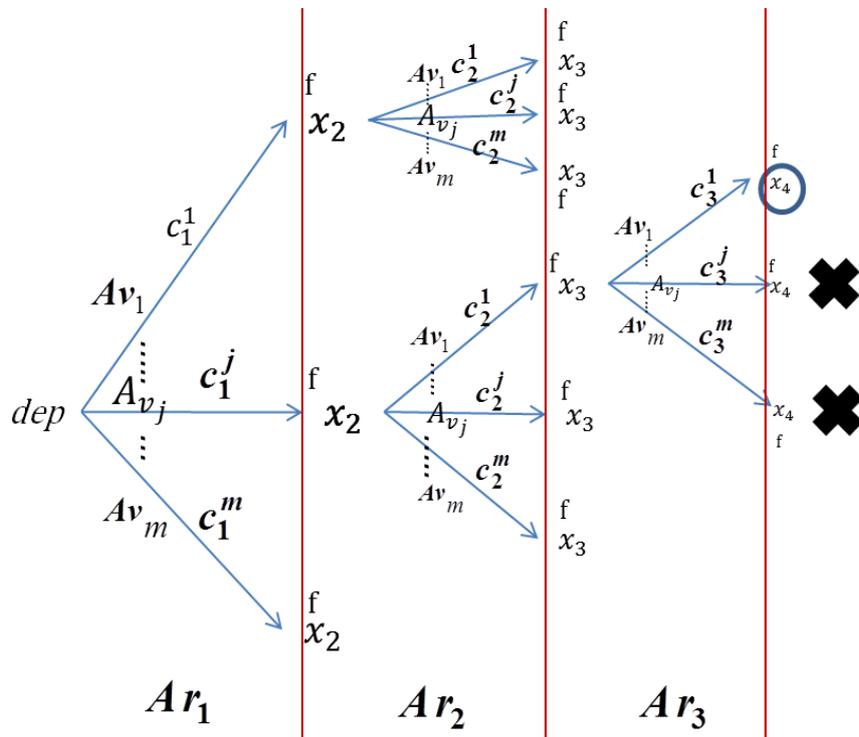


Figure 20 : Cinquième phase de planification

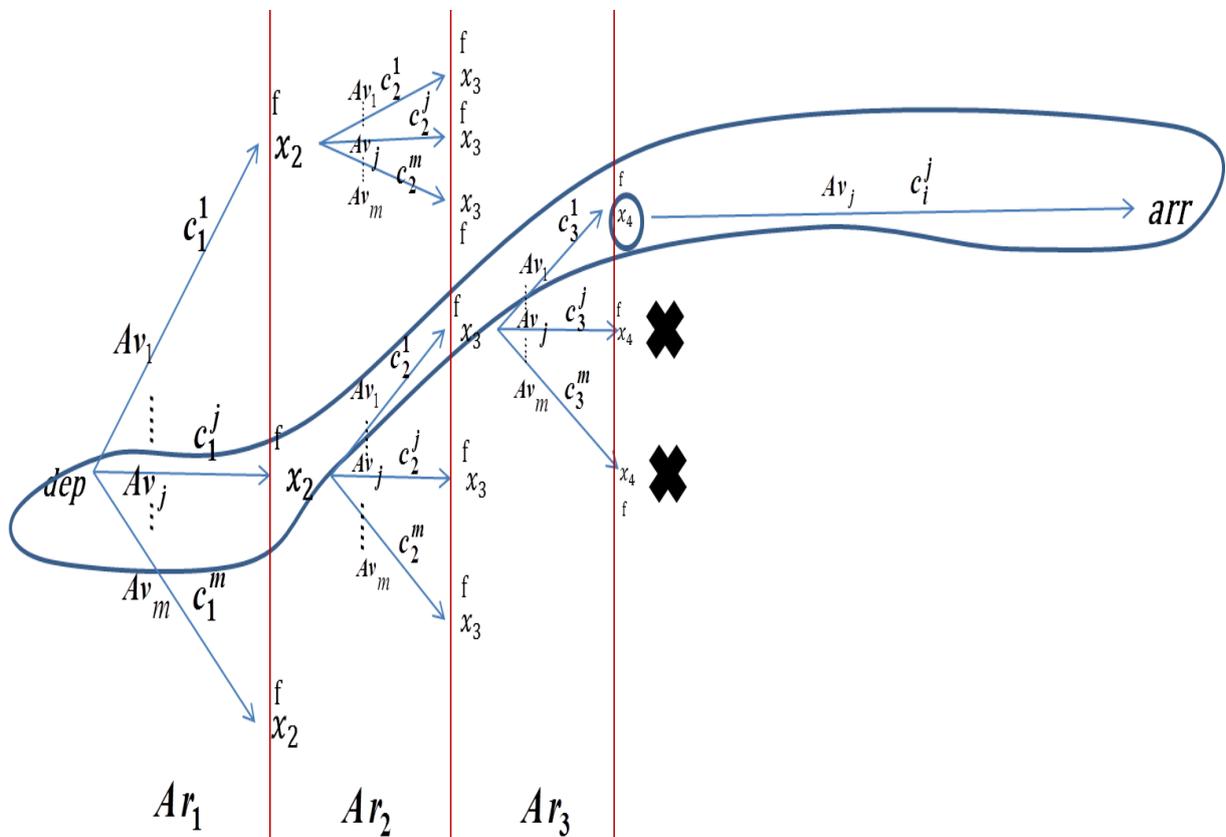


Figure 21 : Trajectoire optimale

figure 20), deux nœuds x_4 sont taillés parce qu'ils sont atteints avec 3 agents Véhicule. Nous notons que sortir d'un véhicule v_1 pour prendre un autre v_2 puis reprendre v_1 est considéré comme 3 changements de véhicules. La trajectoire optimale est tracée dans la figure 21.

6.1.3 Propriétés de l'algorithme A* décentralisé

L'heuristique calculée pour l'algorithme A* décentralisé durant la phase d'initialisation est admissible. Puisque l'heuristique est admissible, cet algorithme devient admissible et il garantit de trouver une solution optimale (si elle existe).

La complexité en temps de l'algorithme classique A* est exponentielle. Elle est égale à $O(b^d)$, où b est le facteur de ramification et d est la longueur (profondeur) de la solution. Dans l'algorithme A* décentralisé, le facteur de ramification est égal au nombre d'agents Véhicule disponibles m et la profondeur de la solution est égale au nombre de routes du chemin n .

Par ailleurs, l'algorithme A* décentralisé est exécuté par les agents Route. Chaque agent calcule et stocke une étape de l'algorithme, ce qui réduit le temps réel de calcul.

Le rôle de l'agent Véhicule sera détaillé dans ce qui suit.

6.2 Rôle de l'agent Véhicule

Le rôle d'un agent Véhicule est de vérifier sa disponibilité à parcourir une route associée à un agent Route. La disponibilité de l'agent Véhicule se traduit par les caractéristiques suivantes :

- l'agent Véhicule a un nombre de places vides égal ou plus grand que le nombre de passagers à transporter entre le temps de début et le temps de fin du parcours de la route ;
- l'agent Véhicule peut insérer la route dans sa table de planification de manière à respecter les contraintes temporelles de ses anciennes routes pour lesquelles il est engagé en exécution et celles de la nouvelle.

L'insertion d'une nouvelle route dans la table de planification d'un véhicule est inspirée de l'algorithme d'ordonnement proposé par (Jaw et al. 1986). L'idée générale de l'algorithme d'ordonnement est la suivante : un bloc d'ordonnement débute par le premier passager qui va être transporté par le véhicule et se termine quand le dernier passager à bord arrive à destination. La figure 22 donne l'exemple du bloc d'ordonnement d'un véhicule engagé à

transporter trois clients. Chaque client a un lieu de départ désigné par le signe ‘-’ et un autre d’arrivée désigné par le signe ‘+’.

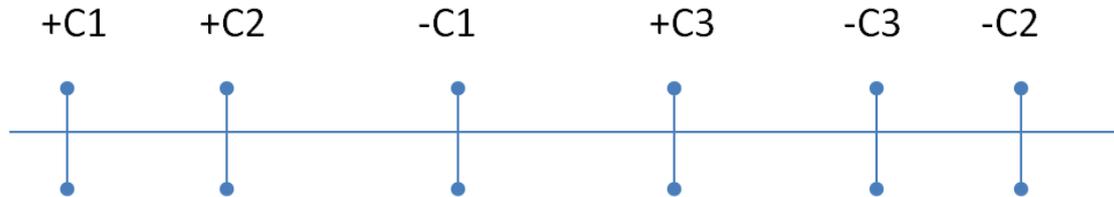


Figure 22 : Bloc d’ordonnancement

Dans le bloc d’ordonnancement, si un véhicule est engagé avec k passagers durant des périodes de temps différents, nous obtiendrons $l=2*k$ arrêts (départ et arrivée). Il y a donc $\sum_{i=1}^{l+1} i = \frac{(l+1)(l+2)}{2}$ possibilités pour qu’un nouveau passager soit inséré dans le bloc ayant l arrêts.

Dans notre cas, la table de planification d’un véhicule est inspirée du bloc d’ordonnancement. Elle commence toujours par le premier passager qui va être transporté par le véhicule et se termine quand le dernier passager à bord arrive à sa destination. Les tab 5 et tab 6 expliquent la procédure d’insertion dans la table de planification.

Le nombre théorique de possibilités d’insertion dans le bloc d’ordonnancement proposé par (Jaw et al. 1986) est le maximum que nous pouvons avoir dans un problème de tournées de véhicules. Plus nous ajoutons des contraintes, plus ce nombre diminue. En effet, les contraintes temporelles imposées par les clients et la contrainte de capacité de chaque véhicule réduisent le nombre théorique d’insertion pour devenir 1 ou zéro. La valeur zéro signifie que l’insertion est infaisable dans la table de planification du véhicule.

Supposons qu’un agent Véhicule reçoit la route $r_i = x_i \rightarrow x_{i+1}$. Le temps t_{x_i} de départ de x_i est 08h15 et le temps $t_{x_{i+1}}$ d’arrivée à x_{i+1} est 08h25. L’agent Véhicule a la table de planification représentée par le tab 5 ci-dessous. Cet agent doit vérifier s’il peut exécuter cette route et l’insérer dans sa table de planification.

	Temps	Place	Passagers
	07h30	A	2
←	08h05	B	1
←	08h20	E	3
	09h00	F	2
	09h30	B	0

Tab 5 : La table de planification avant insertion de la demande

Nous remarquons qu'il n'y a qu'une seule possibilité pour chaque insertion, à cause des contraintes temporelles exigées par le passager.

La table de planification, représentée par le tab 5, devient après insertion, la table représentée par le tab 6.

Temps	Place	Passagers
07h30	A	2
08h05	B	1
08h15	x_j	2
08h20	E	4
08h25	x_{j+1}	3
09h00	F	2
09h30	B	0

Tab 6 : La table de planification après insertion de la demande

Rappelons qu'un agent Route demande à un agent Véhicule de vérifier sa disponibilité de transporter des passagers sur sa route durant la phase d'initialisation et la phase de planification. Dans la phase d'initialisation, l'agent Véhicule vérifie qu'entre les temps de début et de fin du parcours de la route, il a assez de places à bord du véhicule pour transporter les passagers sur cette route. Dans la phase de planification, il teste de plus s'il peut arriver au point de début de la route avant le temps de début et s'il peut arriver à son point de fin avant le temps de fin sans perturber les engagements avec d'autres agents Route. Dans ce qui suit, nous détaillons, le rôle de l'agent Véhicule durant les deux phases.

6.2.1 Phase d'initialisation

Dans la *phase d'initialisation*, l'agent Véhicule reçoit la route $r_i = x_i \xrightarrow{t_i, d_i} x_{i+1}$ avec t_{x_i} le temps de départ de x_i et $t_{x_{i+1}}$ le temps d'arrivée à x_{i+1} . Tout d'abord, il calcule, de sa table de planification, l'ensemble des intervalles de temps $Ts = \{ts_1, \dots, ts_m\}$ entre t_{x_i} et $t_{x_{i+1}}$. Puis, il vérifie pour chaque intervalle ts_i , le nombre de places libres dans le véhicule pendant cet intervalle. Dans le cas où ce nombre serait plus grand ou égal à $n_{passagers}$ pour tous les intervalles, l'agent Véhicule calcule le coût d'exécution de la route en multipliant sa distance d_i par le coût du véhicule par Km ($d_i * \text{coût}$). Autrement, il renvoie "-" (au moins durant un intervalle de temps, l'agent Véhicule n'a pas de places suffisantes). Dans ce cas, l'exécution de la route par ce véhicule n'est pas possible.

Le calcul du coût d'exécution d'une route pendant la phase d'initialisation est illustré dans l'algorithme 6.

Algorithme 6 : Algorithme de calcul du coût d'exécution d'une route par un agent Véhicule durant la phase d'initialisation

$$r_i = x_i \xrightarrow{t_i, d_i} x_{i+1}$$

Ts : ensemble des intervalles de temps

$c = "-"$

pour chaque $ts_i \in Ts$ **faire**

si (nb places vides durant $ts_i < n_{passagers}$) **alors**

retourner c ;

fin

fin

$c = d_i * \text{coût}$

retourner c ;

6.2.2 Phase de planification

Durant la *phase de planification*, l'agent Véhicule reçoit une route $r_i = x_i \xrightarrow{t_i, d_i} x_{i+1}$ avec un temps de début d'exécution (ou un temps de départ de x_i) t_{x_i} et un temps de fin d'exécution (ou un temps d'arrivée à x_{i+1}) $t_{x_{i+1}}$. Cet agent doit vérifier s'il peut insérer la nouvelle route dans sa table de planification de manière à respecter le nombre de places vides, les temps de début et de fin de cette route et de ses anciennes routes qu'il est engagé d'exécuter. Pour cela, il doit vérifier un ensemble de contraintes décrites dans le paragraphe suivant.

6.2.2.1 Description des contraintes

L'agent Véhicule calcule l'ensemble des intervalles $Ts = \{ts_1, \dots, ts_m\}$ entre t_{x_i} et $t_{x_{i+1}}$ de la route r_i . Pour chaque intervalle ts_i , il vérifie si le nombre de places vides dans le véhicule est plus grand ou égal à $n_{passagers}$. Si pour un de ces intervalles, cette contrainte n'est pas vérifiée, alors cet agent Véhicule ne peut pas exécuter cette route et il renvoie à l'agent associé à la route "-". Autrement, il vérifie pour le premier intervalle $ts_1 = [ta_1, tb_1]$ où il va insérer le temps de départ t_{x_i} et pour le dernier intervalle $ts_m = [ta_m, tb_m]$ où il va insérer le temps d'arrivée $t_{x_{i+1}}$, les quatre conditions suivantes :

$$t_{a_1} + t_{chemin(a_1 \rightarrow x_i)} \leq t_{x_i} \quad (1)$$

$$t_{x_i} + t_{chemin(x_i \rightarrow b_1)} \leq t_{b_1} \quad (2)$$

$$t_{a_n} + t_{chemin(a_n \rightarrow x_{i+1})} \leq t_{x_{i+1}} \quad (3)$$

$$t_{x_{i+1}} + t_{chemin(x_{i+1} \rightarrow b_n)} \leq t_{b_n} \quad (4)$$

La condition (1) vérifie que le véhicule arrive au point de départ x_i de la route à un temps ne dépassant pas son temps de départ t_{x_i} .

La condition (2) vérifie que les passagers sont ramassés en respectant le temps de l'engagement suivant.

La condition (3) vérifie que le véhicule arrive au point d'arrivée x_{i+1} de la route à un temps ne dépassant pas son temps d'arrivée $t_{x_{i+1}}$.

La condition (4) vérifie que les passagers sont livrés en respectant le temps de l'engagement suivant.

Dans le cas où il ne trouverait qu'un seul intervalle $ts_1 = [ta_1, tb_1]$, il est suffisant de vérifier les deux conditions suivantes :

$$t_{a_1} + t_{chemin(a_1 \rightarrow x_i)} \leq t_{x_i} \quad (5)$$

$$t_{x_{i+1}} + t_{chemin(x_{i+1} \rightarrow b_1)} \leq t_{b_1} \quad (6)$$

La condition (5) vérifie que le véhicule arrive au point de départ x_i de la route à un temps ne dépassant pas son temps de départ t_{x_i} . La condition (6) vérifie que les passagers sont livrés en respectant le temps de l'engagement suivant.

Nous donnons l'exemple de vérification des contraintes. Supposons qu'un agent Véhicule, ayant la table de planification représentée par le tab 5, reçoit durant la phase de planification, la route $r_i = x_i \rightarrow x_{i+1}$ avec $t_i=8h15$ et $t_{i+1}=8h25$. Tout d'abord, l'agent Véhicule calcule $Ts=\{[08h05, 08h20], [08h20,09h00]\}$. Après la vérification du nombre de places vides, l'agent Véhicule doit vérifier les quatre contraintes suivantes comme le montre le tab 6 :

- arriver du point "B" au point " x_i " avant 8h15 traduisant la contrainte (1)

$$t_B + t_{chemin(B \rightarrow x_i)} \leq t_{x_i}$$

- arriver du point " x_i " au point "E" avant 8h20 traduisant la contrainte (2)

$$t_{x_i} + t_{chemin(x_i \rightarrow E)} \leq t_E$$

- arriver du point "E" au point " x_{i+1} " avant 8h25 traduisant la contrainte (3)

$$t_E + t_{chemin(E \rightarrow x_{i+1})} \leq t_{x_{i+1}}$$

- arriver du point " x_{i+1} " au point "F" avant 9h00 traduisant la contrainte (4)

$$t_{x_{i+1}} + t_{chemin(x_{i+1} \rightarrow F)} \leq t_F$$

6.2.2.2 Vérification des contraintes temporelles

Chaque agent Véhicule doit vérifier les contraintes citées ci-dessus afin de déterminer sa disponibilité d'exécuter une route. Pour chaque contrainte, l'agent Véhicule contacte l'agent Pathfinder pour lui trouver le temps nécessaire entre la paire de nœuds de cette contrainte. L'agent Pathfinder calcule le temps du chemin le plus rapide entre ces deux nœuds en se basant sur l'algorithme de Dijkstra. Puis, il envoie la réponse à l'agent Véhicule. Ce dernier vérifie si le temps du chemin respecte la contrainte à vérifier. Si la condition n'est pas vérifiée, cet agent Véhicule ne peut pas exécuter la route et il envoie à l'agent associé à la route "-". Dans le cas contraire, l'agent Véhicule continue à vérifier les autres contraintes.

Après la vérification de toutes les contraintes convenables, l'agent Véhicule calcule le coût d'exécution de la route en multipliant sa distance d_i par le coût de son véhicule. Le calcul du coût d'exécution d'une route pendant la phase de planification est illustré dans l'algorithme 7.

Algorithme 7 : Algorithme de calcul du coût d'exécution d'une route par un agent Véhicule durant la phase de planification

$$r_i = x_i \xrightarrow{t_i, d_i} x_{i+1}$$

Ts : ensemble des intervalles de temps

$c = "-"$

pour chaque $ts_i \in Ts$ **faire**

si (nb places vides durant $ts_i < n_{passagers}$) **alors**

retourner c ;

fin

fin

$v = \text{verificationContraintes}()$;

si ($v == \text{vrai}$) **alors**

$c = d_i * \text{coût}$

fin

retourner c ;

7 Expérimentations et Résultats

Cette section détaille les expérimentations faites pour valider l'approche multi-agent décentralisée proposée. L'objectif de ces expérimentations est de déterminer :

- la durée du traitement (le temps d'exécution) d'une demande de transport ;
- la durée du traitement d'un ensemble de demandes ;
- le nombre de demandes servies parmi un ensemble ;
- le nombre de véhicules utilisés parmi un ensemble pour servir les demandes.

Deux types d'infrastructure sont adaptés : une infrastructure virtuelle et une infrastructure réelle. Dans ce qui suit, nous présentons les plateformes utilisées pour réaliser nos expérimentations, puis, nous détaillons comment les paramètres de données des ces expérimentations sont générés. Enfin, nous exhibons les résultats des expérimentations.

7.1 Plate-forme d'expérimentation

La plateforme d'expérimentation se base sur : MASON, la méthode de Monte Carlo et la triangulation de Delaunay.

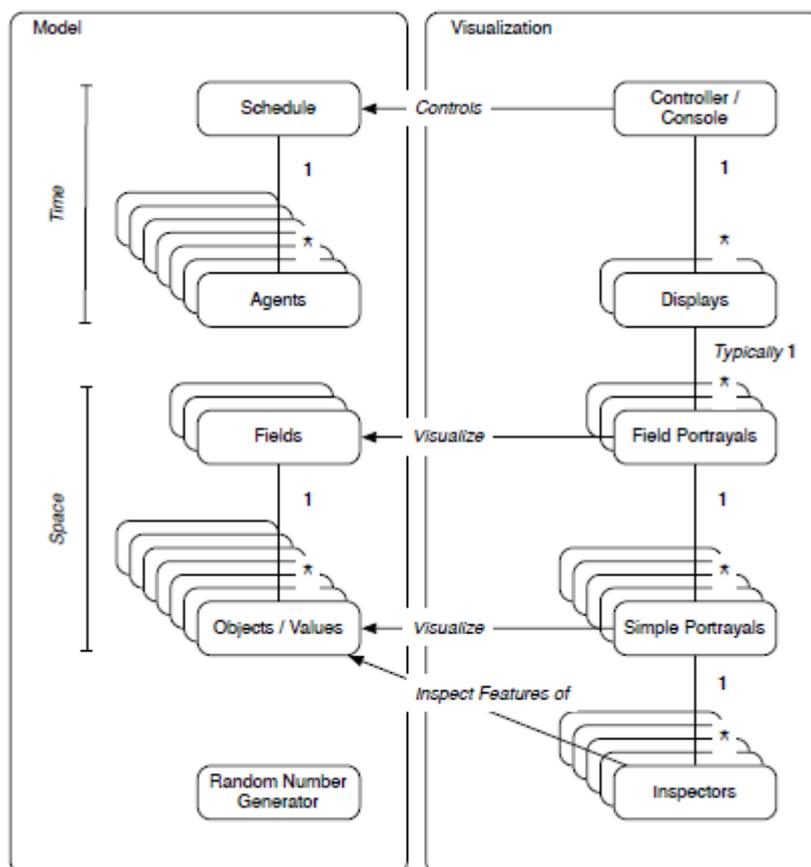


Figure 23 : Les deux parties de Mason (Luke 2011)

MASON (Multi-Agent Simulation Of Networks) est un environnement de simulation multi-agent (Luke 2011). Il a été développé par l'Université George Mason. MASON est écrit en Java. Il peut prendre en charge un grand nombre d'agents et dispose de plusieurs fonctionnalités avancées : il est rapide, facilement extensible, portable, hautement modulaire, flexible et son modèle de base est facile à comprendre. Il peut également produire des résultats identiques à travers différentes plates-formes. Il est divisé en deux parties, comme le montre la figure 23 : la première représente le modèle et la seconde symbolise la visualisation qui est présenté en 2D ou en 3D. Les modèles sont complètement séparés des outils de visualisation. Ils peuvent être ajoutés, supprimés ou modifiés à tout moment.

La technique de la triangulation de Delaunay (Shamos et Hoey 1975; Lee et Schachter 1980; Lee et Lin 1986) est utilisée pour construire un graphe représentant l'infrastructure virtuelle. Ce type de triangulation a été inventé par le mathématicien russe Boris Delaunay en 1934 (Delaunay 1934). La triangulation de Delaunay d'un ensemble de N points d'un plan est un ensemble de triangles tels qu'aucun point du plan n'est à l'intérieur du cercle circonscrit d'un des triangles. En d'autres termes, le cercle circonscrit de chaque triangle de la triangulation ne doit contenir aucun point du plan. La figure 24 représente l'exemple d'une triangulation de Delaunay. La triangulation de Delaunay d'un ensemble de N points est construite avec un temps de $O(n \log n)$ (Lee et Lin 1986).

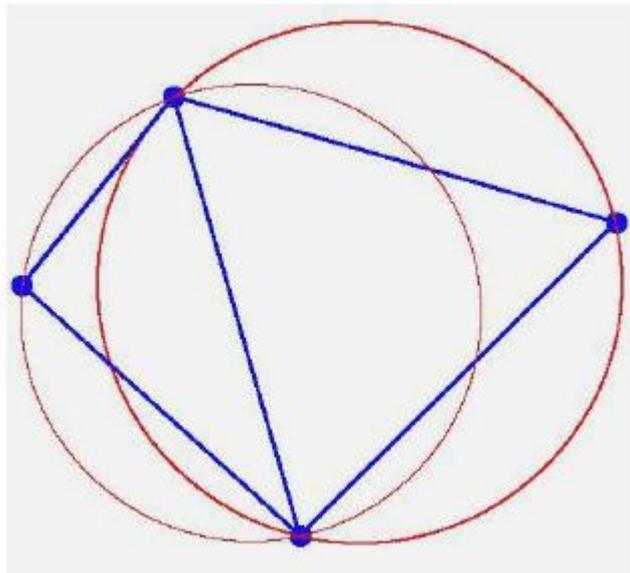


Figure 24 : Triangulation de Delaunay

La méthode de Monte Carlo (Eckhardt 1987) est utilisée pour générer les lieux et les temps de départ et d'arrivée de chaque demande de transport. Cette méthode, inventée en 1947 par

Nicholas Metropolis, est une méthode numérique qui génère un nombre aléatoire en utilisant des techniques probabilistes.

Maintenant, nous passons à la description des paramètres de données utilisées dans nos expérimentations.

7.2 Paramètres des expériences

Dans la littérature, il existe des *benchmarks* (une banque de données) créés pour expérimenter le problème de tournées de véhicules et ses variantes. Ces *benchmarks* existent pour analyser et comparer les performances d'un nouveau système avec celles des systèmes proposés dans la littérature. Nous citons le *benchmark* proposé par (Solomon 1987) pour le problème de tournées de véhicules avec fenêtre de temps (VRPTW) et celui de Li et Lim¹⁵ pour le problème du ramassage et de livraison avec fenêtre de temps PDPTW. Ce sont des benchmarks destinés surtout à la résolution des problèmes statiques et qui étudient des critères classiques comme la distance totale parcourue et le nombre de véhicules utilisés.

Afin d'évaluer toutes les approches proposées dans le présent chapitre et les chapitres suivants, nous générons de nouvelles données. Dans nos approches proposées, de nouvelles contraintes sont considérées (comme le nombre de changement de véhicules pour transporter les passagers d'une demande, le trafic routier, la table de planification des véhicules, etc.) et de nouveaux critères d'évaluation sont étudiés (comme le temps d'exécution du système, la satisfaction des clients, l'impact du trafic sur le réseau routier, etc.). Pour ces raisons, nous décidons de ne pas utiliser les benchmarks existants pour pouvoir évaluer des nouveaux critères et prendre en compte de nouvelles contraintes.

Pour mesurer quantitativement les résultats de l'approche proposée, deux types d'infrastructure sont créés. Le premier type est une infrastructure virtuelle et la seconde une infrastructure réelle. L'infrastructure réelle est générée dans le but de valider les résultats obtenus dans le cas de l'infrastructure virtuelle. Chaque type d'infrastructure a ses propres données des demandes de transports et des véhicules. Dans ce qui suit, nous détaillons la génération de ces deux types d'infrastructure et présentons les paramètres des demandes de transports et des véhicules de chaque infrastructure.

¹⁵ <http://www.sintef.no/projectweb/top/pdptw/li-lim-benchmark>

7.2.1 Infrastructure et données virtuelles

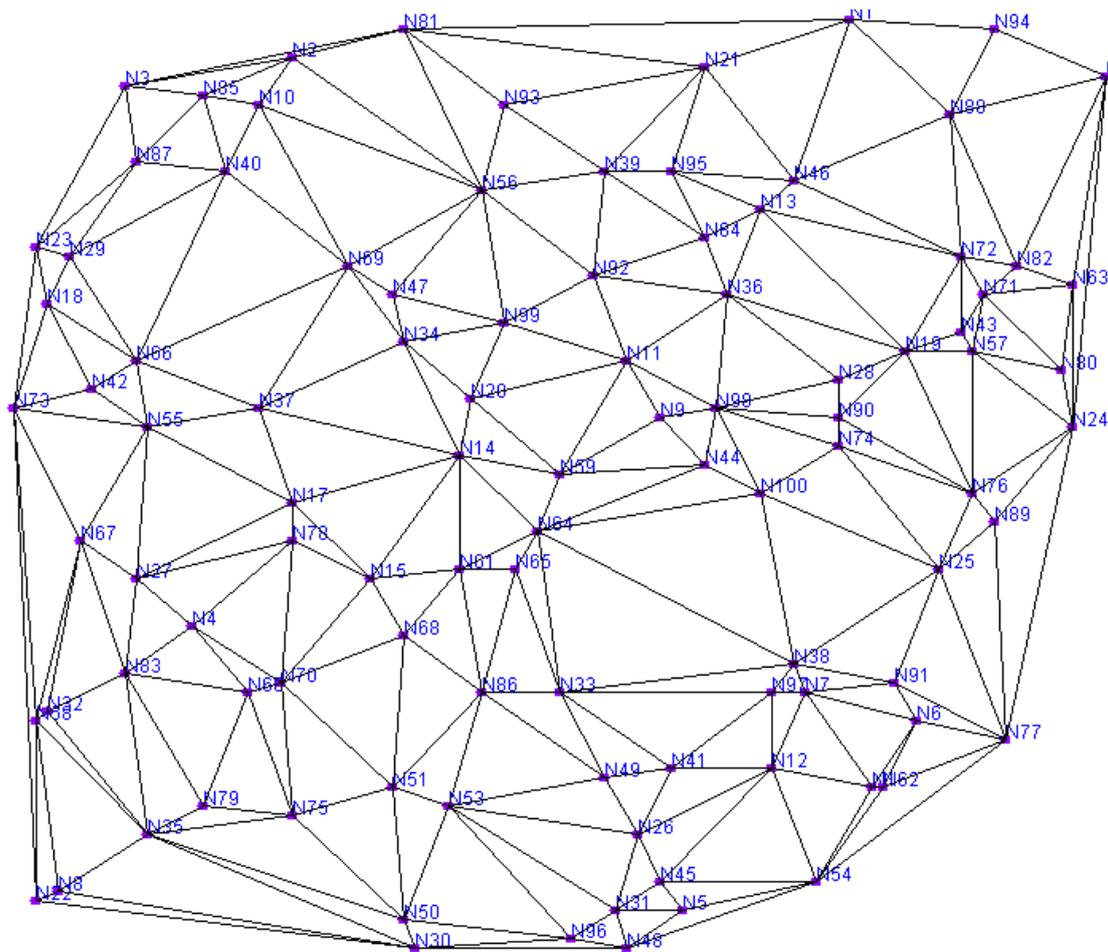


Figure 25 : Infrastructure avec 100 nœuds

Nous générons deux infrastructures de type virtuel pour tester l'approche proposée. La première, appelée *infrastructure100*, est définie par un graphe composé de 100 nœuds et 280 liens. La deuxième, appelée *infrastructure500*, est un graphe composé de 500 nœuds et 1474 liens. La première infrastructure est représentée dans la figure 25. Ces deux infrastructures virtuelles sont générées par l'algorithme de la triangulation de Delaunay (Shamos et Hoey 1975; Lee et Schachter 1980; Lee et Lin 1986). Chaque nœud de ces infrastructures a ses propres coordonnées (abscisse et ordonnée) qui sont créés de manière aléatoire en appliquant une distribution uniforme sur un ensemble contenant 100 entiers variant de 1 à 100. Par conséquent, les coordonnées de chaque point varient entre 1 et 100. De plus, les routes reliant l'ensemble des nœuds dans chaque infrastructure sont générées par l'application de l'algorithme de la triangulation de Delaunay sur les nœuds. Les sommets et les côtés des triangles représentent respectivement les lieux et les routes de l'infrastructure. Chaque route est pondérée par deux valeurs entières : la première valeur représente la distance euclidienne entre les deux nœuds et la seconde représente le temps nécessaire pour traverser la route. Ce

temps est calculé en divisant la distance par la vitesse. Nous considérons trois catégories de routes. La première catégorie comprend des routes dont la vitesse limite est 40 km/h. La deuxième catégorie comprend des routes dont la vitesse limite est 70 km/h. Enfin, la troisième comprend les routes de vitesse limite 100km/h.

Concernant les véhicules, chacun d'entre eux dispose de quatre sièges, une place de parking pour se garer chaque jour et enfin un coût par kilomètre. Celui-ci est calculé de manière aléatoire en appliquant une distribution uniforme sur un ensemble comprenant 10 entiers allant entre 1 et 10. Par conséquent, le coût de chaque véhicule varie entre 1 euro/km et 10 euro/km. Tous les véhicules fonctionnent toute la journée. Ainsi, ils ont les mêmes horaires de travail.

Nous décrivons dès à présent comment les demandes de transport sont générées. Les demandes des clients sont très diverses et ceci à cause des paramètres suivants :

- le temps de départ et le temps d'arrivée ;
- le lieu de départ et le lieu d'arrivée ;
- le nombre de passagers à transporter ;
- le nombre de routes à travers lesquelles les véhicules doivent passer ;
- le nombre maximal de changements de véhicules.

Tout d'abord, le lieu de départ et le lieu d'arrivée de chaque demande constituent l'un des nœuds de l'infrastructure. Ces lieux sont choisis aléatoirement en appliquant la distribution uniforme sur l'ensemble des nœuds. Par ailleurs, le nombre de passagers varie entre 1 et 4.

Les heures de pointe en France varient entre les intervalles suivants : [07h00, 09h00] et [16h00, 19h00]¹⁶. Ainsi, la plupart des demandes devraient avoir un temps de départ entre ces deux intervalles. Pour générer le temps de départ de chaque demande, nous appliquons la méthode de Monte Carlo sur un ensemble d'heures qui varient entre 07h00 et 22h00. Les heures comprises dans les deux intervalles [07h00, 09h00] et [16h00, 19h00] ont une probabilité d'être choisies plus forte (75 %) que les autres. Enfin, 500 demandes sont générées et sauvegardées dans un fichier XML.

¹⁶ <http://www.insee.fr> « Concilier environnement et mobilité : un enjeu pour le territoire du SCoT de l'Angoumois », décimal n°325, avril 2013 »

7.2.2 Infrastructure et données réelles

Afin de valider les résultats obtenus dans une infrastructure virtuelle et d'évaluer réellement l'approche proposée, nous utilisons la carte d'une ville libanaise appelée Tripoli. C'est une ville au nord du Liban. Sa superficie est 2,0248 km². Elle possède 20,7 % de la population du Liban qui avoisine les 850.000 personnes. La carte de Tripoli est représentée dans la figure 26. Cette figure montre le réseau routier et les endroits principaux de cette ville.

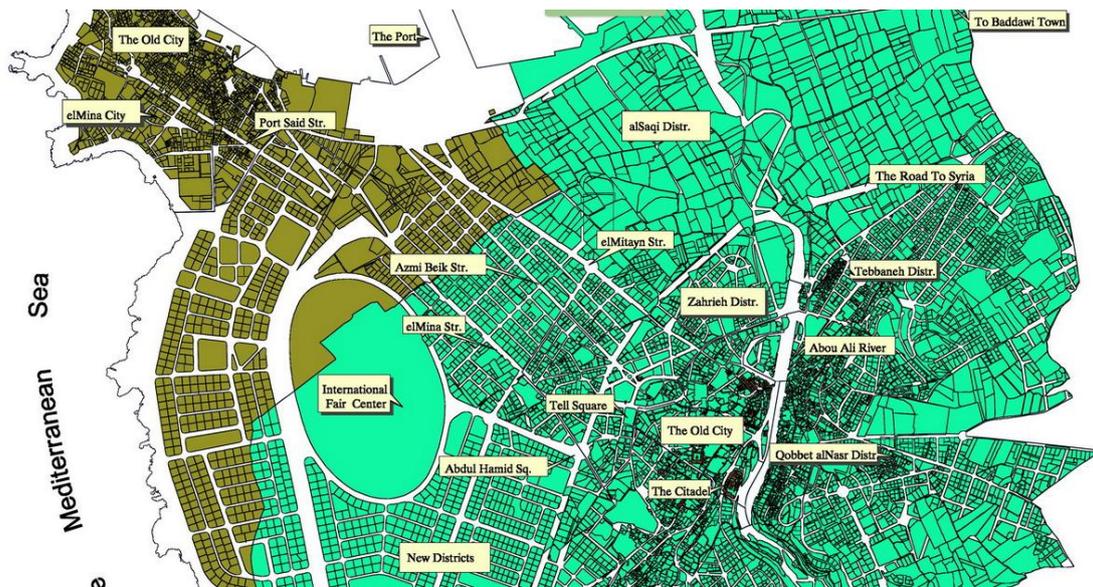


Figure 26 : La carte de Tripoli

Tripoli possède une municipalité, un port, 144 écoles, 10 établissements d'enseignement supérieur, 11 hôpitaux, etc. De cette ville, nous avons pris 117 nœuds routiers et 315 routes pour faire nos expérimentations. Chaque route a deux paramètres : une distance et une vitesse limite qui varie entre 40 et 100 km /h. Ces paramètres parviennent de la municipalité de Tripoli.

De plus, nous avons choisi une compagnie de transport très célèbre à Tripoli appelée *Connexion*. La compagnie *Connexion* a 45 conducteurs et 30 véhicules réservés pour transporter les clients dans les quartiers de Tripoli. L'entretien réalisé avec le directeur général nous a permis de déceler les besoins de la population tripolitaine pour définir le nombre de demandes quotidiennes, les lieux de départ et d'arrivée et les temps de départ les plus demandés.

En ce qui concerne les véhicules, chaque conducteur dispose de quatre sièges et se gare chaque jour dans le parking de la compagnie. De plus, tous les véhicules fonctionnent toute la

journée, ayant ainsi le même horaire de travail. Cependant, les conducteurs ont des horaires de travail différents.

Concernant les demandes des clients, comme nous l'avons déjà évoqué, elles sont très diverses dépendant de plusieurs paramètres. Tout d'abord, les lieux de départ et d'arrivée de chaque demande représentent l'un des lieux à Tripoli. Ces derniers sont générés par l'application de la méthode de Monte Carlo sur l'ensemble des lieux. Ceux qui sont les plus demandés selon les informations données par le directeur général de la compagnie de transport *Connexion*, ont une probabilité plus forte (75 %) d'être choisis en comparaison des autres lieux. En outre, le nombre de passagers est choisi de façon aléatoire en appliquant la distribution uniforme sur un ensemble composé de quatre entiers qui varient entre 1 et 4. Selon les données fournies par le directeur général, le temps de départ de la plupart des demandes de transport à Tripoli varie entre les intervalles suivants : [07h00, 09h00] et [14h00, 16h00]. Pour générer le temps de départ de chaque demande, nous appliquons la méthode de Monte Carlo sur un ensemble contenant des heures qui varient entre 07h00 et 22h00. Les heures de pointe comprises dans les deux intervalles [7h00, 09h00] et [14h00, 16h00] ont une probabilité plus forte (75 %) d'être choisies en comparaison des autres heures. Enfin, 500 demandes sont générées et sauvegardées dans un fichier XML.

7.3 Les résultats des expérimentations

Nos expériences sont réalisées dans les conditions suivantes : un processeur Intel i3 1.80 GHz, une mémoire vive de 6 Go et Windows 7 comme système d'exploitation. Dans le cadre de nos expériences, les demandes sont exécutées en mode statique.

7.3.1 Expérimentations de l'infrastructure virtuelle

Nous remarquons que l'exécution d'une demande desservie par un seul véhicule prend 0,9 seconde. Deux types d'expérimentation sont effectués.

Dans la première expérimentation, nous mesurons le nombre de demandes servies par le système et le temps d'exécution total en fixant le nombre de demandes et en variant le nombre de véhicules. Pour cela, nous distinguons trois cas : le premier cas comprend 70 demandes, le deuxième est composé de 140 demandes, y compris les 70 demandes précédentes et la troisième est composée de 200 demandes, y compris les 140 demandes précédentes. Pour chaque cas, les demandes sont exécutées :

- sept fois de façon aléatoire. Puis, le résultat moyen de ces sept exécutions est calculé ;

- une seule fois de manière sélective. Dans le traitement (l'exécution) sélectif, le système attribue à chaque demande une priorité en commençant par la demande ayant le plus petit nombre de routes (dans son chemin), et ainsi de suite.

Enfin, les résultats obtenus dans le traitement aléatoire sont comparés à ceux obtenus par le traitement sélectif.

Traitement de 70 demandes			
Nombre de véhicules	10	20	30
Moyenne du nombre de demandes servies aléatoirement	(40 D, 2 sec)	(65 D, 7 sec)	(70 D, 42 sec)
Nombre de demandes servies en priorité	(44 D, 1.3 sec)	(68 D, 8 sec)	(70 D, 707 sec)
Traitement de 140 demandes			
Nombre de véhicules	10	20	30
Moyenne du nombre de demandes servies aléatoirement	(58 D, 3.3 sec)	(96 D, 10 sec)	(118 D, 44 sec)
Nombre de demandes servies en priorité	(63 D, 2.4 sec)	(100 D, 7.5 sec)	(122 D, 32 sec)
Traitement de 200 demandes			
Nombre de véhicules	10	20	30
Moyenne du nombre de demandes servies aléatoirement	(62 D, 3 sec)	(105 D, 8.5 sec)	(137 D, 30 sec)
Nombre de demandes servies en priorité	(71 D, 2.6 sec)	(116 D, 5 sec)	(146 D, 17 sec)

Tab 7 : Traitement de demandes dans l'infrastructure100 virtuelle

Nous montrons les résultats de la première expérience faite sur l'infrastructure de 100 nœuds (*infras₁₀₀*) dans le tab 7. Le tab 8 concernent les expérimentations appliquées sur l'infrastructure de 500 nœuds (*infras₅₀₀*). Pour le traitement des demandes de chaque cas, nous représentons les résultats en trois lignes. La première ligne représente le nombre de véhicules disponibles. La deuxième représente la moyenne du nombre de demandes servies de façon aléatoire et le temps moyen pris pour les exécuter. La troisième ligne représente le nombre de demandes servies en priorité et le temps nécessaire pour les exécuter. Chaque cellule du tableau contient le nombre de demandes servies (par exemple 40 D) et le temps de traitement en seconde (par exemple, 2 sec).

Traitement de 70 demandes			
Nombre de véhicules	10	20	30
Moyenne du nombre de demandes servies aléatoirement	(48 D, 16 sec)	(68 D, 122 sec)	(70 D, 508 sec)
Nombre de demandes servies en priorité	(48 D, 23 sec)	(68 D, 183 sec)	(70 D, 123 sec)
Traitement de 140 demandes			
Nombre de véhicules	10	20	30
Moyenne du nombre de demandes servies aléatoirement	(70 D, 29 sec)	(103 D, 155 sec)	(125 D, 616 sec)
Nombre de demandes servies en priorité	(71 D, 25 sec)	(108 D, 129 sec)	(128 D, 646 s)
Traitement de 200 demandes			
Nombre de véhicule	10	20	30
Moyenne du nombre de demandes servies aléatoirement	(80 D, 27 sec)	(116 D, 132 sec)	(150 D, 456 sec)
Nombre de demandes servies en priorité	(84 D, 21 sec)	(125 D, 102 sec)	(154 D, 481 sec)

Tab 8 : Traitement de demandes dans l'infrastructure500 virtuelle

Dans la deuxième expérimentation, nous étudions le nombre de demandes servies par le système et le temps d'exécution en fixant le nombre de véhicules et en variant le nombre de demandes. Nous distinguons quatre cas : le premier cas comprend 10 véhicules, le deuxième comprend 20, le troisième comprend 30 et le quatrième en comprend 40. Pour chaque cas, les demandes sont traitées en priorité. Comme mentionné précédemment, le système assigne à chaque demande une priorité à partir de la demande ayant le plus petit nombre de routes dans son chemin. Le résultat de la seconde expérimentation appliquée à l'infrastructure composée de 100 nœuds et celle composée de 500 nœuds est montré respectivement dans la figure 27 et la figure 28. Le premier graphe des deux figures montre la variation du pourcentage de demandes servies selon le nombre de demandes et le nombre de véhicules. Le second montre la variation du temps d'exécution selon le nombre de demandes et le nombre de véhicules.

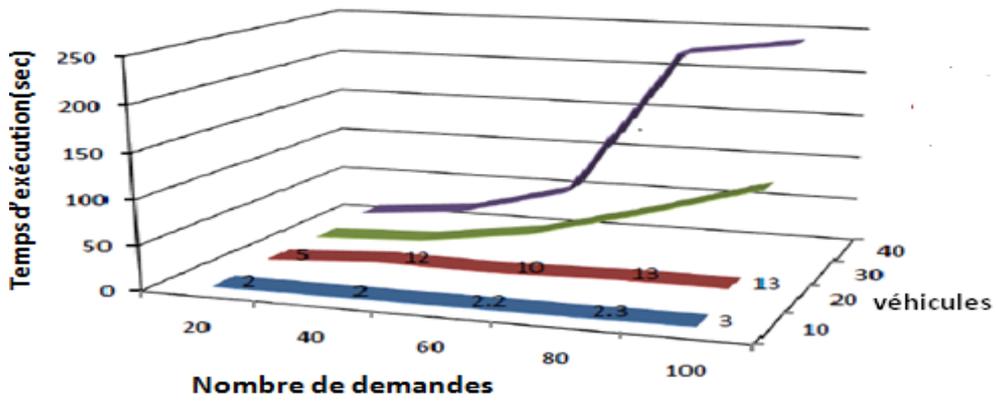
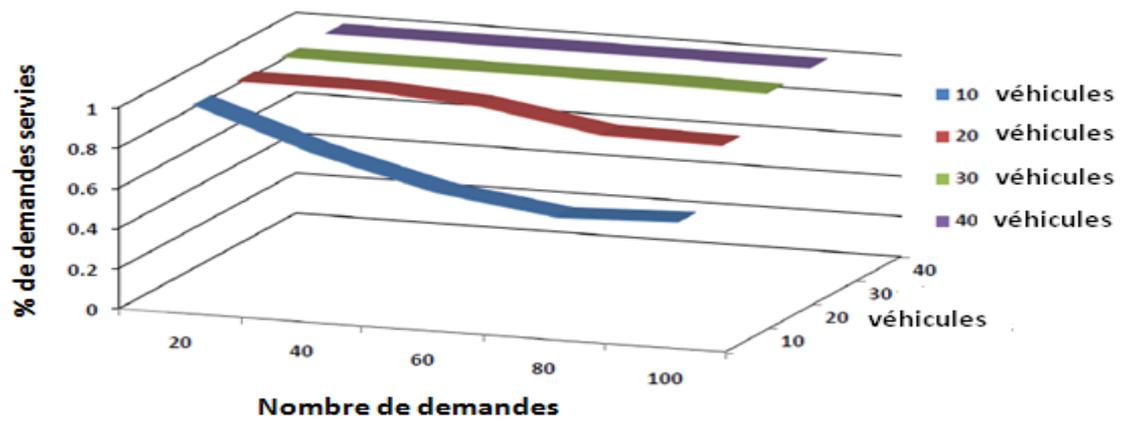


Figure 27 : Résultats de l'infrastructure100

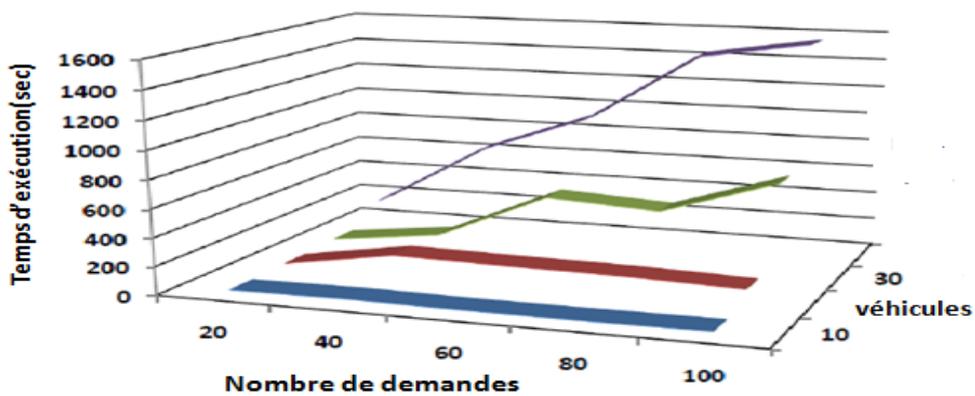
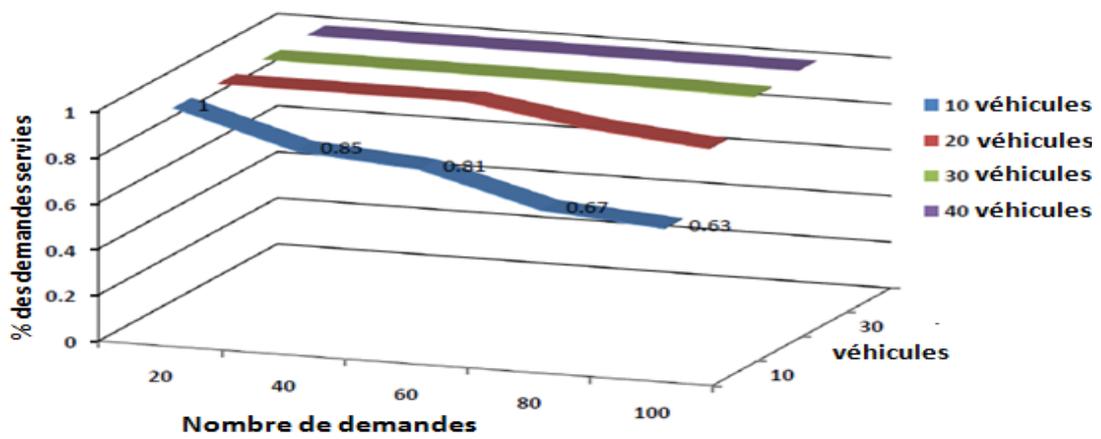


Figure 28 : Résultats de l'infrastructure500

7.3.2 Expérimentations de l'infrastructure réelle

Afin de valider les résultats dans un cas réaliste, nous avons appliqué les deux expérimentations faites dans le paragraphe 7.3.1 sur une infrastructure réelle. Cette infrastructure représente le réseau routier de la ville de Tripoli en supposant que la vitesse de voyage sur chaque route est constante. Elle est égale à la vitesse limite de la route.

Traitement de 40 demandes			
Nombre de véhicules	10	20	30
Moyenne du nombre de demandes servies aléatoirement	(33 R, 2.5 sec)	(40 R, 8.6 sec)	(40 R, 37 sec)
Nombre de demandes servies en priorité	(33 R, 2.7 sec)	(40 R, 11.4 sec)	(40 R, 60 sec)
Traitement de 80 demandes			
Nombre de véhicules	10	20	30
Moyenne du nombre de demandes servies aléatoirement	(42 R, 2.2 sec)	(70 R, 16 sec)	(80 R, 97 sec)
Nombre de demandes servies en priorité	(45 R, 3.25 sec)	(71 R, 16.5 sec)	(79 R, 264 sec)
Traitement de 120 demandes			
Nombre de véhicules	10	20	30
Moyenne du nombre de demandes servies aléatoirement	(58 R, 3.7 sec)	(88 R, 21.5 sec)	(111 R, 107 sec)
Nombre de demandes servies en priorité	(62 R, 3.7 sec)	(92 R, 11.6 sec)	(110 R, 82.2 sec)
Traitement de 160 demandes			
Nombre de véhicules	10	20	30
Moyenne du nombre de demandes servies aléatoirement	(65 R, 4.22 sec)	(97 R, 20.6 sec)	(128 R, 74 sec)
Nombre de demandes servies en priorité	(69 R, 4 sec)	(108 R, 13.5 sec)	(131 R, 38.6 sec)
Traitement de 200 demandes			
Nombre de véhicules	10	20	30
Moyenne du nombre de demandes servies aléatoirement	(66 R, 4.5 sec)	(104 R, 18 sec)	(137 R, 70 sec)
Nombre de demandes servies en priorité	(70 R, 3.4 sec)	(115 R, 7.8 sec)	(146 R, 28 sec)

Tab 9 : Traitement de demandes dans l'infrastructure réelle

Dans la première expérimentation, nous distinguons cinq cas : le premier cas comprend 40 demandes, le deuxième est composé de 80 demandes, y compris les 40 demandes précédentes, le troisième est composé de 120 demandes, y compris les 80 demandes précédentes, le quatrième comprend 160 demandes et le dernier comprend 200 demandes y compris toutes les 160 demandes. Pour chaque cas, les demandes sont exécutées 7 fois de façon aléatoire, puis le résultat moyen de ces 7 cas est calculé. De plus, ces demandes sont exécutées une seule fois de manière sélective. Le tab 9 représente les résultats du traitement de demandes dans l'infrastructure réelle de la ville de Tripoli.

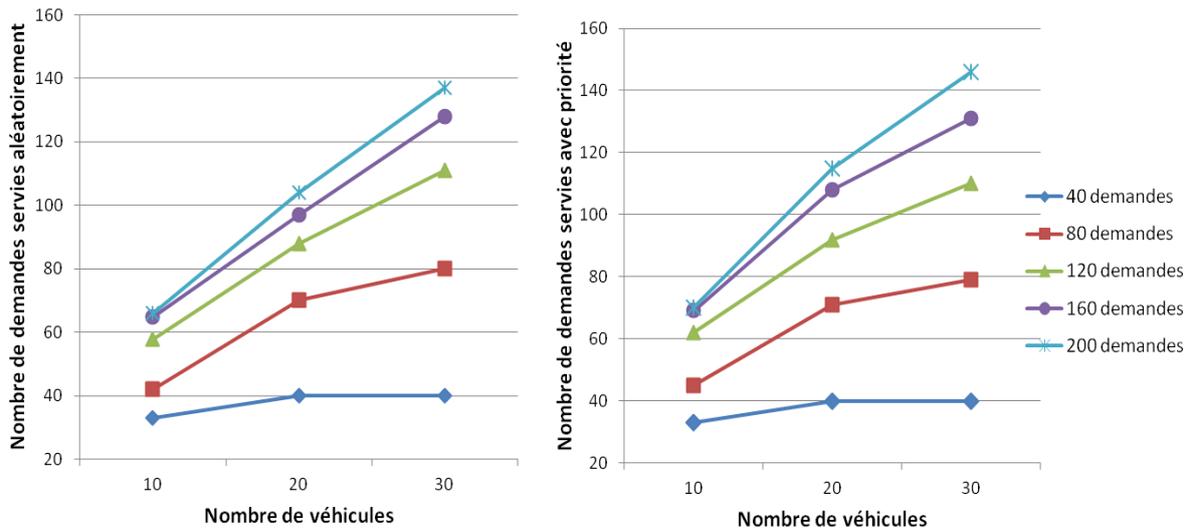


Figure 29 : Nombre de demandes servies

Les résultats du tab 9 sont représentés par les quatre graphiques de la figure 29 et la figure 30. Les deux premiers graphiques de la figure 29 représentent le nombre de demandes servies en variant le nombre de véhicules. Le premier représente l'exécution des demandes de manière aléatoire et le second représente l'exécution des demandes de manière sélective.

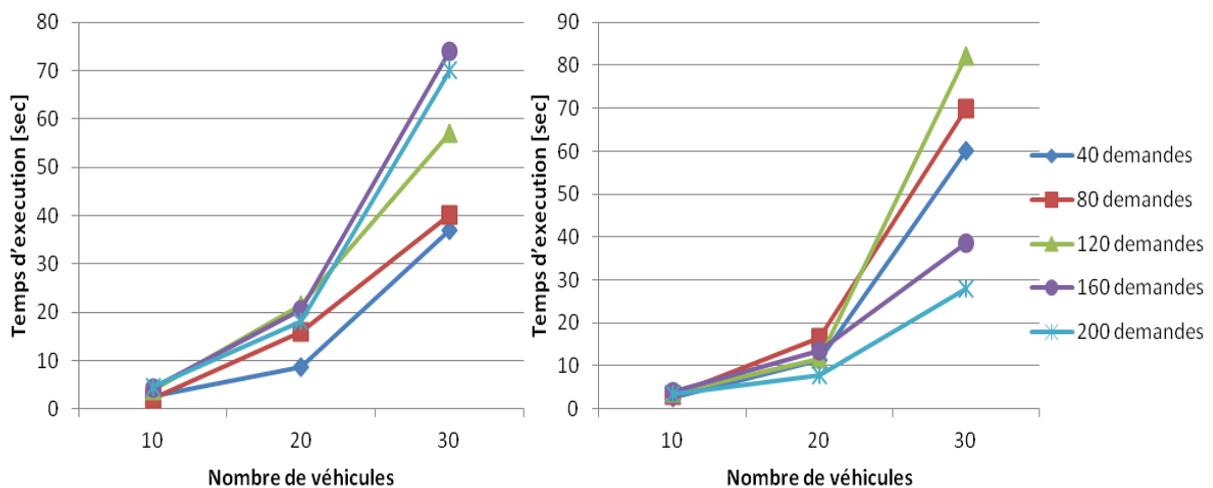


Figure 30 : Temps d'exécution des demandes

Les deux graphiques de la figure 30 représentent le temps d'exécution d'un nombre fixe de demandes en variant le nombre de véhicules disponibles. Le premier représente le traitement des demandes de manière aléatoire. Le second montre leur traitement de manière sélective.

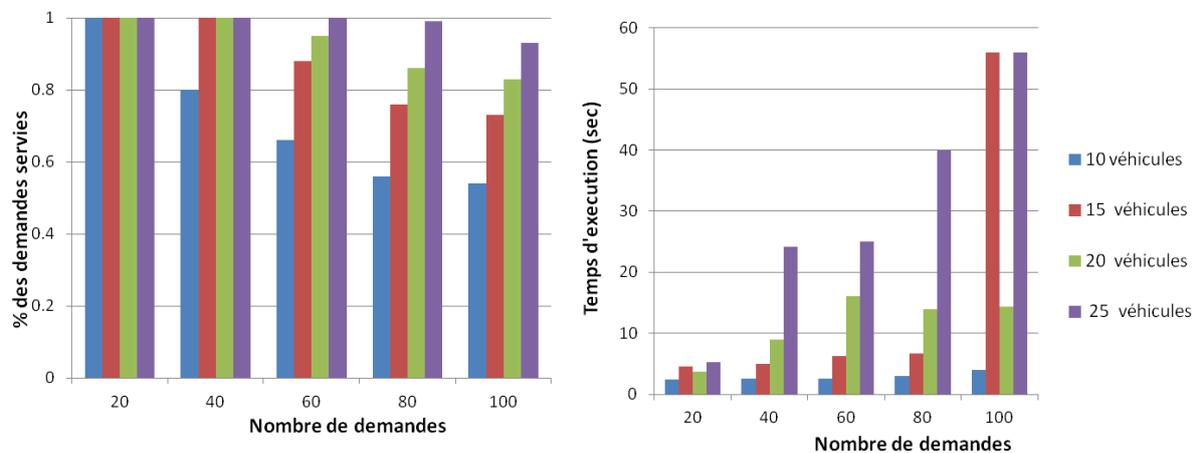


Figure 31 : Pourcentage de demandes servies et leurs temps d'exécution

La seconde expérimentation suit la même logique que celle de la deuxième traitée dans le paragraphe 7.3.1. Toutefois, sa particularité réside dans le fait que le nombre de véhicules varie entre 10 et 25. Les résultats sont représentés par les deux graphes de la figure 31.

7.3.3 Analyse

D'après les expérimentations faites sur les deux infrastructures virtuelles et l'infrastructure réelle, nous remarquons que :

- les résultats des expérimentations effectuées sur l'infrastructure réelle confirment ceux obtenus dans les deux infrastructures virtuelles ;
- l'augmentation du nombre de véhicules fait accroître le nombre de demandes servies, ainsi que le temps de traitement ;
- l'augmentation du nombre de demandes fait élever également le nombre de demandes servies et le temps de traitement ;
- l'introduction d'une stratégie de priorité intelligente dans l'exécution des demandes augmente le nombre de demandes servies pour plus de 66 % des cas et réduit le temps d'exécution ainsi que le nombre de véhicules utilisés pour plus de 53 % des cas.

Finalement, il est judicieux d'émettre les deux remarques suivantes :

- si le traitement des demandes se fait avec une certaine priorité, des solutions optimales peuvent émerger ;

- le temps d'exécution et le nombre de demandes servies dépendent de plusieurs facteurs : le nombre de routes du chemin de chaque demande, l'ordre de traitement des demandes, le temps de départ de chaque demande, le nombre de véhicules disponibles et le nombre total de demandes à servir.

8 Conclusion

Dans ce chapitre, nous avons proposé une approche multi-agent décentralisée pour un *scénario statique* du problème du transport à la demande (TAD). Ce scénario considère que les vitesses de voyage du réseau routier sont indépendantes du temps. Les agents communiquent de manière décentralisée en se basant sur l'algorithme A^* pour trouver la trajectoire la plus rapide menée par les véhicules les moins coûteux convenant à chaque demande de transport. Des expérimentations sont faites pour valider l'approche proposée dans des infrastructures statiques réelles et virtuelles. Ces infrastructures sont caractérisées par des vitesses constantes.

Néanmoins, nous savons que la vitesse de voyage est subjective et peut varier selon la situation du trafic. Le fait de considérer qu'elle est constante, dans les services de transport des personnes, aboutit à l'échec des horaires planifiés. Les fenêtres de temps sont violées, les temps de voyage des clients estimés à bord des véhicules sont dépassés et les chauffeurs travaillent des heures supplémentaires. Par conséquent, l'entreprise paye le coût de la détérioration du service, de l'insatisfaction de clients et des heures supplémentaires de travail pour les véhicules et les chauffeurs.

Dans les chapitres suivants, nous allons résoudre le problème du TAD en mode dynamique et dans un réseau routier dynamique où les vitesses de voyage s'évaluent avec le temps. Nous allons exploiter les systèmes multi-agents (SMA) afin de résoudre opérationnellement le problème tout en considérant la situation du trafic. Nous allons agentifier tout le problème sans utilisation de A^* et chercher une solution satisfaisante pour chaque demande de transport.

Chapitre 4. Approche multi-agent massif pour le transport à la demande dans une ville

1 Introduction

L'intérêt croissant porté à la recherche dans le domaine de la logistique urbaine est principalement dû à l'augmentation de la population dans les villes, aux problèmes du trafic routier et à la pression du public (Crainic, Ricciardi, et Storchi 2009). Le problème de tournées de véhicules dans les villes (Time Dependant Vehicle Routing Problem : TDVRP), un problème essentiel dans la logistique urbaine, a été récemment examiné (Malandraki 1989; Duan et al. 2015).

Un des principaux défis pris en considération dans l'optimisation du problème de tournées de véhicules dans les villes est la variation de la situation du trafic qui engendre une variation des vitesses de voyage (ou des temps de voyage) durant la journée (Cattaruzza et al. 2015). Un système, traitant un problème réel de tournées de véhicules dans une ville, doit considérer le trafic routier afin qu'il soit capable d'estimer les vitesses de voyage et par suite rendre des résultats fiables.

La majorité des recherches étudient le problème classique de tournées de véhicules. Elles supposent que les vitesses de voyage ne varient pas dans le temps (Savelsbergh et Sol 1995; Kammarti et al. 2004; Jin, Abdulrab, et Itmi 2008; Dridi et al. 2010; Guerriero et al. 2014; Carotenuto, Paradisi, et Storchi 2014). Cette supposition est contradictoire avec la réalité de la circulation dans les villes et aboutit à des solutions non fiables, infaisables et très coûteuses. Effectivement, une solution fondée sur cette supposition induit l'insatisfaction des clients lesquels la fenêtre de temps de leurs demandes est ratée et des heures de travail supplémentaires pour les conducteurs. De cette manière, la qualité de service diminue. La considération que les vitesses de voyage sont indépendantes du temps aboutit à rater les heures de travail par la moitié des voyages des véhicules (Maden, Eglese, et Black 2010) et à obtenir 20 % d'erreur dans l'estimation des durées des itinéraires (Ehmke, Steinert, et Mattfeld 2012).

Dans ce chapitre, nous proposons une nouvelle approche multi-agent massif décentralisée pour le problème du TAD dans une ville caractérisée par un réseau routier dynamique

déterministe. Il est dynamique car les vitesses de voyages sur les routes varient avec le temps selon le trafic et déterministe car ces vitesses sont déterminées par avance en se basant sur les connaissances historiques du trafic de la ville. L'approche considère l'impact du trafic habituel sur les vitesses de voyage. Cela devrait faire que notre système de TAD donne des résultats plus fiables. Nous montrons dans la partie expérimentale que le nombre de clients insatisfaits est supérieur à 50 % lorsque la situation du trafic routier n'est pas considérée (Malas et al. 2016a). En outre, deux méthodes de négociation, permettant de trouver le bon véhicule transportant les passagers d'une demande, sont présentées (Malas et al. 2016b). Cette approche vise à réduire l'insatisfaction des clients et le coût d'opération. Ce dernier décroît en diminuant la distance totale parcourue et le nombre de véhicules en circulation.

Dans la deuxième section, nous décrivons le nouveau scénario du problème du TAD qui va être modélisé et résolu. L'algorithme général pour la résolution du scénario est expliqué dans la section 3. Nous définissons la structure de donnée des composants du TAD dans une quatrième section. Les différents types d'agents de la nouvelle approche sont explicités dans la section 5, pour passer ensuite aux mécanismes de coopération de ces agents afin de traiter une nouvelle demande dans la section 6. Les négociations réalisées dans le but de trouver la bonne solution et les expérimentations sont traitées simultanément dans les sections 7 et 8. La dernière section conclut le chapitre.

2 Deuxième scénario du problème du TAD : *scénario dynamique déterministe*

Le deuxième scénario du problème du TAD est nommé *scénario dynamique déterministe*. Son idée se résume comme suit : des clients sont dispersés dans une ville. Chacun d'eux déclenche une demande de transport en indiquant le lieu de départ, le lieu d'arrivée, le temps de départ au plus tôt, le temps d'arrivée au plus tard et le nombre de passagers à transporter. Cette demande doit être servie par un seul véhicule.

La ville est caractérisée par des routes avec des vitesses de voyage dynamiques déterministes. Ces vitesses sont dynamiques car elles varient avec le temps selon le trafic et déterministes car elles sont déterminées par avance en fonction des situations habituelles du trafic. Ce scénario considère l'effet des situations habituelles du trafic sur les vitesses de voyage. Ces situations sont causées par des événements prévus et elles sont connues par avance par les connaissances historiques du trafic.

L'objectif du système du TAD est de chercher pour chaque demande de transport, le bon véhicule avec la trajectoire qui va suivre pour transporter les passagers tout en :

- respectant les contraintes du client (le temps de départ au plus tôt, le temps d'arrivée au plus tard) et des véhicules (le nombre de places vides et les contraintes temporelles concernant leurs engagements avec d'autres clients) ;
- minimisant le nombre de véhicules en circulation en priorité puis la distance totale parcourue ;
- prenant en considération la situation habituelle du trafic en se basant sur des connaissances historiques du trafic.

Dans la section suivante, nous présentons l'algorithme général de l'idée du *scénario dynamique déterministe*.

3 Algorithme général de la résolution du scénario dynamique déterministe

Ci-après l'algorithme général du système du TAD du *scénario dynamique déterministe*.

Début

Partie 1 : Formulation d'une nouvelle demande de transport

Un sous-système prend les informations de la demande d'un client connecté pour aller d'un point 'x' à un temps de départ au plus tôt t_x et arriver au point 'y' à un temps d'arrivée au plus tard t_y .

Partie 2 : Traitement de la réception de la nouvelle demande de transport

- Des véhicules de transport sont sollicités pour satisfaire la demande de transport du client. Chaque véhicule vérifie sa disponibilité en se basant sur des connaissances historiques du trafic routier et fait une proposition.
- Suivant la meilleure proposition, un véhicule est choisi par un protocole de négociation pour transporter les passagers.
- Le véhicule choisi propose le temps de départ ($\geq t_x$) et le temps d'arrivée ($\leq t_y$) prévus au client.
- Les passagers et le véhicule s'entendent.

Partie 3 : Traitement de changement de l'état du trafic

Un sous-système donne l'état de la fluidité du trafic qui peut varier plus ou moins fortement selon les moments sur chaque route en se basant sur des connaissances historiques. Il informe de plus les véhicules du temps nécessaire pour traverser une trajectoire à partir d'un temps initial de départ.

Partie 4 : Transport et circulation des véhicules avec les clients

- Un véhicule prend les passagers pour effectuer la trajectoire.
- Pendant le parcours, le véhicule est informé de façon continue par le système de la variation de sa trajectoire (l'arrivée d'une nouvelle demande, par exemple).
- Les passagers sont arrivés au point de destination au temps t' .

Fin

Dans ce qui suit, nous détaillons la modélisation et la résolution du *scénario dynamique déterministe*.

4 Structure de données des composants du système du TAD du scénario dynamique déterministe

Le système du TAD du *scénario dynamique déterministe* est constitué de quatre composants fondamentaux : (1) l'infrastructure de la ville ; (2) la demande de transport du client ; (3) le véhicule et (4) la trajectoire. Dans ce qui suit, nous allons détailler la structure de données de ces quatre composants et présentons le problème du TAD ainsi que son objectif.

4.1 Infrastructure dynamique déterministe

L'infrastructure du système du TAD du *scénario dynamique déterministe* symbolise un réseau routier d'une ville avec des vitesses de voyage dynamiques déterministes. Ces vitesses sont dynamiques car elles varient en fonction du temps et du trafic et sont déterministes car elles sont connues à priori selon les connaissances historiques du trafic. La variation de la vitesse sur une route résulte principalement de l'évolution de la situation dynamique du trafic de cette route. Dans ce scénario, nous traitons les situations dynamiques déterministes du trafic. Ces situations sont déterminées par les connaissances historiques du trafic. Ces connaissances concernent l'effet des événements prévisibles sur le trafic routier. Un événement prévisible est tout événement qui se répète dans le même intervalle du temps d'un même jour et affecte de la même manière la situation du trafic d'une route. Les heures de pointe et la présence des

zones bâties dans l'environnement de la route comme une école, une université, un centre commercial, un théâtre, un marché, etc., sont des exemples des événements prévisibles.

Alors, l'infrastructure du *scénario dynamique déterministe* est représentée par un graphe $G = (P, R)$ composé d'un ensemble de nœuds routiers $P = \{p_1, \dots, p_n\}$ désignant l'ensemble des lieux de la ville et un ensemble d'arcs $R = \{r_1, \dots, r_n\}$ désignant les routes de la ville. La figure 32 représente l'exemple de ce type d'infrastructure.

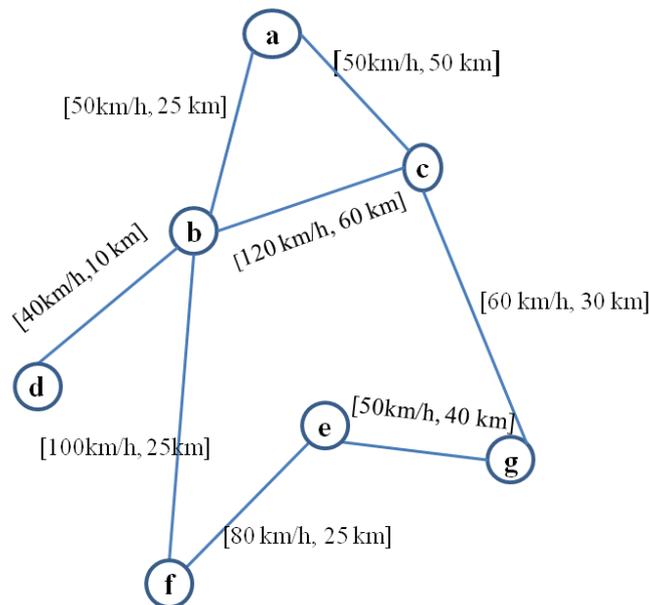


Figure 32 : Infrastructure dynamique déterministe

Chaque route $r_i = p_i \xrightarrow{(v_i, d_i, L_i)} p_j$ relie deux places p_i et p_j de la ville. Elle est caractérisée par v_i la vitesse limite de circulation sur cette route, d_i la longueur de la route et une liste L_i appelée *liste des situations de trafic prévues*.

Rôle et construction de la liste des situations de trafic prévues

La situation du trafic représente la fluidité des véhicules sur la route. Cette situation peut être :

- fluide pour symboliser une circulation de véhicules aisée sur la route ;
- dense pour symboliser une circulation ralentie des véhicules sur la route. La route est semi-concentrée ;
- saturée pour symboliser une circulation très ralentie des véhicules sur la route. La route est engorgée (bloquée).

La liste L_i d'une route r_i contient les situations habituelles quotidiennes du trafic de la route pour chaque jour de l'année. Ces situations sont estimées après une étude et une analyse des connaissances historiques du trafic de la route selon les jours de l'année. Le rôle de la liste est

de refléter les situations habituelles du trafic de la route dans le système du TAD en fonction du temps et du jour de l'année. Elle est utilisée pour estimer et prédire à partir d'un temps t , le temps de voyage requis pour traverser la route r_i .

La méthode de construction de la liste est faite en fonction des catégories du jour et des intervalles de temps. En se basant sur les connaissances historiques du trafic de la route, nous catégorisons les jours de l'année et nous classons ceux qui se ressemblent dans la situation du trafic dans la même catégorie. Puis, nous discrétisons les 24 heures d'un jour de chaque catégorie en des intervalles de temps hétérogènes et nous assignons une situation du trafic à chacun de ses intervalles. Dans la liste L_i , nous mémorisons les situations de trafic selon les catégories du jour et les intervalles de temps. Nous représentons l'exemple de la liste L_i pour la route r_i par le tab 10. Chaque ligne de la table est définie par une catégorie du jour, un intervalle de temps et une situation du trafic. Nous remarquons que les jours de l'année de la route r_i sont classés en deux catégories : "Cat1" et "Cat2". Chaque catégorie a ses propres situations du trafic pendant des intervalles de temps couvrant les 24 heures d'une journée.

Catégories du jour	Intervalles de temps	Situations du trafic
"Cat1"	[07h00, 09h00]	dense
	[09h00, 16h00]	fluide
	[16h00, 19h00]	dense
	[19h00, 07h00]	fluide
"Cat2"	[07h00, 18h00]	fluide
	[18h00, 20h00]	saturée
	[20h00, 07h00]	fluide

Tab 10 : Exemple d'une liste des situations de trafic prévues

Une valeur de pondération est assignée à chaque type de situation. A titre d'exemple, pour une situation fluide, la valeur de pondération est 1, pour une situation dense la valeur est 0.5 et pour une situation saturée, la valeur est 0.2. Cette valeur joue le rôle de réduction de la vitesse limite de la route. Pour calculer la vitesse de voyage sur une route à un instant t , il faut multiplier la vitesse limite de la route par la valeur de pondération correspondant à la situation du trafic de la route pendant l'intervalle de temps auquel t appartient. Nous notons dans le tab 10 que pour un jour appartenant à la catégorie "Cat1", entre les deux intervalles de temps [07h00, 09h00] et [16h00, 19h00], le trafic est dense. Alors, la vitesse limite de la route doit être multipliée par la valeur de pondération 0,5 pour diminuer de moitié la vitesse durant ces

tranches de temps. Tandis qu'elle reste constante pendant les autres intervalles où la situation du trafic est fluide.

Acquisitions des connaissances historiques

Notre modèle est basé sur les connaissances historiques du trafic routier de la ville. La collecte cohérente de ces connaissances est primordiale pour assurer l'exactitude de ce modèle. Cette mission ne fait pas partie de notre étude. Ces connaissances doivent être collectées par des personnes dotées d'une expérience appropriée et aptes à de longues journées de collecte des données pendant différentes périodes. D'ailleurs, en raison de l'émergence de la technologie, ces connaissances peuvent être recueillies par différentes techniques mentionnées dans le chapitre 2¹⁷.

4.2 Demande de transport

La demande de transport d'un client est définie par :

$$D = (dep, t_{dep}) \xrightarrow{n_{passagers}} (arr, t_{arr}) \text{ où}$$

- dep et arr sont les lieux de départ et d'arrivée ;
- t_{dep} représente le temps de départ au plus tôt ;
- t_{arr} représente le temps d'arrivée au plus tard ;
- $n_{passagers}$ représente le nombre de passagers à transporter.

Pour mieux comprendre les différents paramètres, voici l'exemple d'une demande de transport D :

$$D = ('A', 08h00) \xrightarrow{2} ('D', 09h00)$$

Cette demande de transport signifie que deux passagers veulent quitter le lieu "A" au plus tôt à 08h00 et arriver au lieu "D" au plus tard à 09h00.

4.3 Véhicule

Dans le *scénario dynamique déterministe*, les passagers d'une demande de transport sont transportés par un seul véhicule. La définition du véhicule est donnée dans le chapitre 3¹⁸.

¹⁷ Voir chapitre 2, paragraphe 6.1.1

¹⁸ Voir chapitre 3, paragraphe 4.3

4.4 Trajectoire

La trajectoire est définie dans le système du TAD du *scénario dynamique déterministe* par :

$$trajectoire = \left((dep, t_{dep}) \rightarrow \dots (x_i, t_{x_i}) \rightarrow \dots (arr, t_{arr}) \right) \text{ où :}$$

- $chemin = (dep \rightarrow \dots x_i \rightarrow \dots arr)$ est le chemin de la trajectoire. Il est composé de la liste des lieux et des routes traversés pour atteindre le lieu d'arrivée arr à partir du lieu de départ dep . Les lieux d'un chemin sont classés en deux types : un lieu principal et un lieu transitoire. Les lieux de départ et d'arrivée d'une demande sont des lieux principaux. Tandis que les lieux reliant deux lieux principaux sont des lieux transitoires ;
- t_{dep} est le temps de départ du nœud dep , t_{x_i} est le temps de départ prévu du nœud x_i (ou le temps de début du parcours de la route $x_i \xrightarrow{(t_i, d_i)} x_{i+1}$) et t_{arr} est le temps d'arrivée prévu au nœud arr . Ces temps sont calculés en prenant en considération la situation habituelle du trafic à partir du temps de départ t_{dep} . Donc, ils sont calculés en se basant sur la *liste des situations de trafic prévues* de chaque route du chemin.

Un chemin évalué à partir d'un temps initial t_0 est un chemin dont la durée de parcours de chacune de ses routes est calculée. Une trajectoire est un chemin évalué.

Notons qu'une trajectoire est menée par un seul véhicule du lieu de départ jusqu'au lieu d'arrivée.

4.5 Problème du TAD

Soit une infrastructure dynamique déterministe $G = (P, R)$ et un ensemble de véhicules $V = \{v_1, \dots, v_n\}$. Le problème du TAD est défini par une demande de client :

$$D = (dep, t_{dep}) \xrightarrow{n_{passagers}} (arr, t_{arr})$$

Une trajectoire $traj = \left((dep, t_{dep'}) \rightarrow \dots (x_i, t_{x_i}) \rightarrow \dots (arr, t_{arr'}) \right)$ menée par un véhicule v_i est une solution valide de la demande D si :

- $t_{dep'} \geq t_{dep}$ et $t_{arr'} \leq t_{arr}$;
- $t_{x_{i+1}} > t_{x_i}$ avec $i \in [1, \dots, n-1]$;
- $pVide(v_i)_{[t_{dep'}, t_{arr'}]} \geq n_{passagers}$ où $pVide(v_i)_{[t_{dep'}, t_{arr'}]}$ représente le nombre de places vides de v_i durant $[t_{dep'}, t_{arr'}]$.

Dans ce cas, nous pouvons affirmer que v avec sa trajectoire $traj$ satisfait la demande D .

Nous remarquons que le temps de départ t_{dep} et le temps d'arrivée t_{arr} de la demande sont deux contraintes temporelles considérées comme des contraintes fortes et elles doivent être respectées par le véhicule qui va servir la demande. Donc, la possibilité qu'une demande soit refusée est très probable lorsque tous les véhicules du système ne réussissent pas à respecter ces contraintes temporelles.

4.6 Objectif du système du TAD

Comme le véhicule dans le problème du TAD est un facteur essentiel, coûteux et nuisible à l'environnement, l'objectif du système du TAD du *scénario dynamique déterministe* est de minimiser le nombre de véhicules en circulation. Pour réaliser cet objectif, la solution minimisant ce nombre est choisie comme solution optimale parmi les solutions satisfaisant une demande de transport. La solution optimale est représentée par une trajectoire menée par le véhicule possédant le nombre de places vides le plus réduit durant la période du service de la demande. Si deux solutions minimisent de la même manière le nombre de places vides, la solution optimale sera celle qui minimise la distance totale parcourue. A cette fin, nous mesurons pour chaque solution satisfaisant la demande deux critères d'optimisation :

- le minimum des nombres de places vides du véhicule durant la période du service de la demande ;
- la distance totale parcourue par le véhicule pour servir la demande.

Soit $V = \{v_1, \dots, v_n\}$ l'ensemble de véhicules satisfaisant une demande D et $Traj = \{traj_1, \dots, traj_n\}$ l'ensemble de trajectoires menées par ces véhicules pour servir D . $v_i \in V$ est optimale en nombre de véhicules en circulation si et seulement si : $\forall v_j \in V \setminus \{v_i\}$, $\min(pVide(v_i)_{[t_{dep'_i}, t_{arr'_i}]}) \leq \min(pVide(v_j)_{[t_{dep'_j}, t_{arr'_j}]})$ où $t_{dep'_i}$ est le temps de début du parcours du $traj_i$ et $t_{arr'_i}$ est le temps de fin du parcours $traj_i$. Alors, v_i est le véhicule qui minimise le nombre de véhicules en circulation.

$v_i \in V$ est optimale en distance totale parcourue si et seulement si : $\forall v_j \in V \setminus \{v_i\}$, $distance(v_i) \leq distance(v_j)$ où $distance(v_i)$ est la distance totale parcourue par v_i pour servir la demande D . Alors, v_i est le véhicule qui optimise la distance totale parcourue.

Ainsi, un véhicule v_i avec sa trajectoire $traj_i$ est une solution pour une demande D , si elle satisfait D et si elle est optimale en nombre de véhicules en circulation. S'il existe un autre

véhicule v_j qui satisfait D et est optimale en nombre de véhicules en circulation, alors, celui qui est optimal en distance totale parcourue sera la solution à la demande D .

5 Modélisation multi-agents massifs du problème du TAD

L'utilité d'utiliser les systèmes multi-agents pour modéliser et résoudre des problèmes complexes comme le problème du transport à la demande a été expliquée dans le chapitre 3¹⁹. Dans cette section, nous modélisons le problème du TAD du *scénario dynamique déterministe* comme un système multi-agent massif décentralisé basé sur des connaissances historiques du trafic routier (Malas et al. 2016a). Nous expliquons le pourquoi de l'agentification massive du problème du TAD de ce scénario et nous définissons les différents types d'agents.

5.1 Pourquoi l'agentification massive du TAD ?

Un système multi-agent massif est complexe, dynamique (Campagne et al. 2004) et caractérisé par un nombre d'agents dépassant le millier (Cardon 2002; Cardon 2005). De même, le problème du TAD est un problème complexe et dynamique. Voire, son infrastructure, qui représente le réseau routier d'une ville, est constituée de plus d'un millier de routes. Chacune de ces routes sera associée à un agent Route. Par conséquent, le problème du TAD a des milliers d'agents. Pour ces raisons, nous avons modélisé ce problème via un système multi-agent massif.

5.2 Architecture générale et types d'agents

Quatre types d'agents interagissent ensemble afin de trouver une solution pour le *scénario dynamique déterministe* du TAD comme le montre la figure 33. Ces agents sont :

Agent Client : il agit comme une interface entre les clients et le système du TAD. Il envoie chaque nouvelle demande de transport à l'ensemble des véhicules du système et reçoit, s'il existe, le meilleur véhicule qui l'exécutera. Comme illustré sur la figure 33, le cycle de vie d'une demande de transport commence par une demande produite par un client.

Agent Véhicule : il est associé à chaque véhicule dans le système. Son rôle est de vérifier s'il est disponible de transporter les passagers d'une demande d'un lieu de départ à un lieu d'arrivée durant les temps espérés par les passagers. S'il est disponible, il entrera dans une

¹⁹ Voir chapitre 3, paragraphe 5.

négociation avec d'autres agents Véhicule capable de la servir dans le but de choisir le bon agent qui va la réaliser comme illustrée dans la figure 33.

Agent Pathfinder : il accompagne un agent Véhicule. Son rôle est de guider l'agent Véhicule en lui offrant une trajectoire reliant une paire de nœuds.

Agent Route : il est associé à chaque route de l'infrastructure. Il utilise la *liste des situations de trafic prévues* de sa route pour calculer le temps nécessaire pour la traverser à partir d'un moment précis. Il y en a autant d'agents Route que de routes.

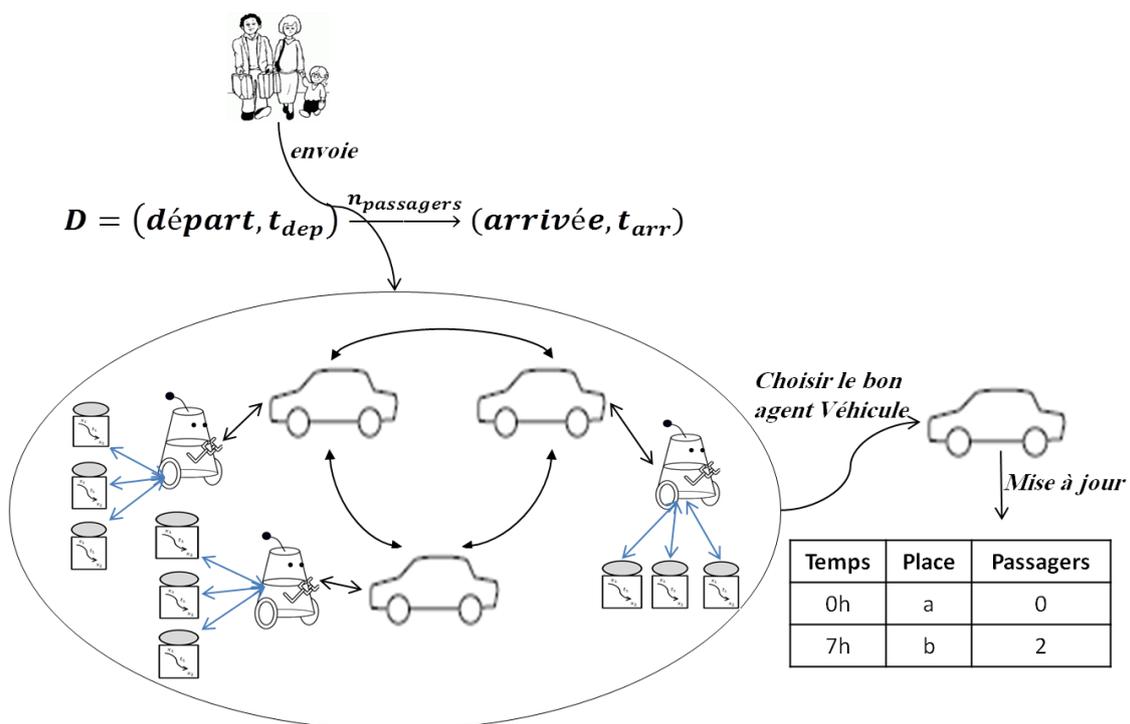


Figure 33 : Architecture SMA du scénario dynamique déterministe

L'architecture du système multi-agent est illustrée dans la figure 33. Cette figure montre un agent Client qui diffuse une demande de transport D à tous les agents Véhicule. Chaque agent Véhicule communique avec son agent Pathfinder pour lui demander une trajectoire. L'agent Pathfinder calcule un chemin et contacte des agents Route pour l'évaluer. Ensuite, l'ensemble des agents Véhicule disponibles pour servir la demande négocient pour choisir le bon agent parmi eux. Le bon agent, c'est celui qui propose la solution optimale pour la demande D . Enfin, l'agent choisi met à jour sa table de planification pour servir la demande.

Dans ce qui suit, la manière dont les agents coopèrent entre eux pour traiter une nouvelle demande est expliquée en détail.

6 Stratégie de coopération entre les agents pour traiter une demande de transport

Lorsqu'une nouvelle demande de transport est lancée, l'agent Client crée une liste vide pour y stocker les agents Véhicule qui seront capables de la servir. Il diffuse également cette demande à l'ensemble des agents Véhicule du système du TAD. Les agents Véhicule, Pathfinder et Route correspondants coopèrent par la méthode d'envoi de message afin de trouver une trajectoire menée par un véhicule qui va servir la demande. Dans ce qui suit, nous décrivons les différentes contraintes à considérer par l'agent Véhicule pour pouvoir vérifier s'il est capable de servir une demande et la méthode de vérification de ces contraintes.

6.1 Description des contraintes temporelles des agents Véhicule

Un agent Véhicule peut servir une nouvelle demande de transport si :

- le nombre de places vides du véhicule auquel il est associé, entre le temps de départ et le temps d'arrivée de la demande, est plus grand ou égal au nombre de passagers de la demande ;
- l'insertion de la demande dans la table de planification du véhicule respecte le temps de départ au plus tôt et le temps d'arrivée au plus tard de cette demande et ne viole pas les temps de départ ou les temps d'arrivée des demandes précédentes lesquelles il s'est engagé à servir.

Soit une demande de transport $D = (dep, t_{dep}) \xrightarrow{n_{passagers}} (arr, t_{arr})$ diffusée par l'agent Client aux agents Véhicule. Chaque agent Véhicule doit répondre à l'agent Client s'il peut l'exécuter. En se référant à sa table de planification, l'agent Véhicule vérifie si le nombre de sièges vides entre t_{dep} et t_{arr} est supérieur ou égal à $n_{passagers}$. Puis, il calcule les intervalles de temps $ts_1 = [ta_1, tb_1]$ où il va insérer le temps de départ de la demande et $ts_2 = [ta_2, tb_2]$ où il va insérer le temps d'arrivée de la demande. Ensuite, il vérifie, en coopérant avec son agent Pathfinder, les quatre conditions suivantes :

$$t_{a_1} + t_{chemin(a_1 \rightarrow dep)} \geq t_{dep} \quad (1)$$

$$t_{dep} + t_{chemin(dep \rightarrow b_1)} \leq t_{b_1} \quad (2)$$

$$t_{a_2} + t_{chemin(a_2 \rightarrow arr)} \leq t_{arr} \quad (3)$$

$$t_{arr} + t_{chemin(arr \rightarrow b_2)} \leq t_{b_2} \quad (4)$$

La condition (1) vérifie que le véhicule arrive à un temps plus grand ou égal au temps de départ au plus tôt à condition qu'il y ait des passagers à bord. Si le véhicule est vide, cette condition n'a aucun sens.

La condition (2) vérifie que les passagers sont ramassés sans violer le temps de départ ou d'arrivée des passagers de la demande suivante.

La condition (3) vérifie que les passagers arrivent à un temps ne dépassant pas le temps d'arrivée au plus tard.

La condition (4) vérifie que les passagers sont livrés sans violer le temps de départ ou d'arrivée de la demande qui la suit.

Dans le cas où il n'y aurait qu'un seul intervalle $ts_1 = [ta_1, tb_1]$ qui doit contenir les temps de départ et d'arrivée de la demande, il est suffisant de vérifier les deux conditions suivantes :

$$t_{a_1} + t_{chemin(a_1 \rightarrow dep)} \geq t_{dep} \quad (5) \text{ à condition qu'il y ait des passagers à bord}$$

$$t_{arr} + t_{chemin(arr \rightarrow b_1)} \leq t_{b_1} \quad (4)$$

Ces conditions confirment le respect des temps de départ et d'arrivée de la nouvelle demande et des anciennes lesquelles l'agent Véhicule est engagé à servir.

6.2 Vérification des contraintes temporelles

Chaque agent Véhicule commence à vérifier successivement les contraintes décrites dans le paragraphe précédent. Pour chacune des contraintes, l'agent Véhicule contacte son agent Pathfinder pour lui trouver une trajectoire (ou un chemin évalué) entre la paire de lieux de cette contrainte à partir du temps de départ du lieu initial t_0 correspondant. L'agent Pathfinder propose le chemin le plus court en termes de distance entre la paire des lieux en appliquant l'algorithme de Dijkstra sur l'infrastructure dynamique. Pour évaluer ce chemin, il contacte, à tour de rôle, les agents Route associés à chaque route du chemin. Il demande au premier agent Route Ar_1 associé à la première route r_1 du chemin de calculer la durée nécessaire t_1 pour traverser sa route à partir du temps t_0 . Ar_1 calcule la durée t_1 et l'envoie à l'agent Pathfinder. Ce dernier demande au deuxième agent Route Ar_2 associé à la deuxième route r_2 du chemin de calculer la durée t_2 pour traverser sa route à partir du $t_1 + t_0$. Et ainsi de suite, jusqu'à ce que le tour du dernier agent Route Ar_n arrive. L'agent Pathfinder demande à Ar_n de calculer la durée t_n pour traverser la route r_n à partir du temps $\sum_{i=0}^{n-1} t_i$. Chaque agent Route contacté se base sur la *liste des situations de trafic prévues* sur sa route et applique la méthode

expliquée dans la section 6.3 pour évaluer le temps nécessaire pour traverser sa route. Enfin, l'agent Pathfinder compose la trajectoire en fonction du chemin et des réponses des agents Route et l'envoie à son agent Véhicule. L'agent Véhicule teste si la trajectoire satisfait la contrainte à vérifier. Dans le cas positif, il continue la vérification des autres contraintes. Autrement, il sera indisponible pour servir la demande.

L'agent Véhicule, qui respecte toutes les contraintes, calcule le coût d'exécution de la demande représenté par deux valeurs. La première valeur indique le minimum des nombres de places vides de son véhicule durant la période du service de la demande et la seconde indique la distance totale parcourue par le véhicule pour servir la demande. Il informe aussi l'agent Client de sa disponibilité à servir la demande. L'agent Client l'ajoute à la liste créée au début.

La figure 34 montre comment les différents agents coopèrent lorsqu'ils reçoivent une nouvelle demande afin de chercher une trajectoire et vérifier chaque contrainte.

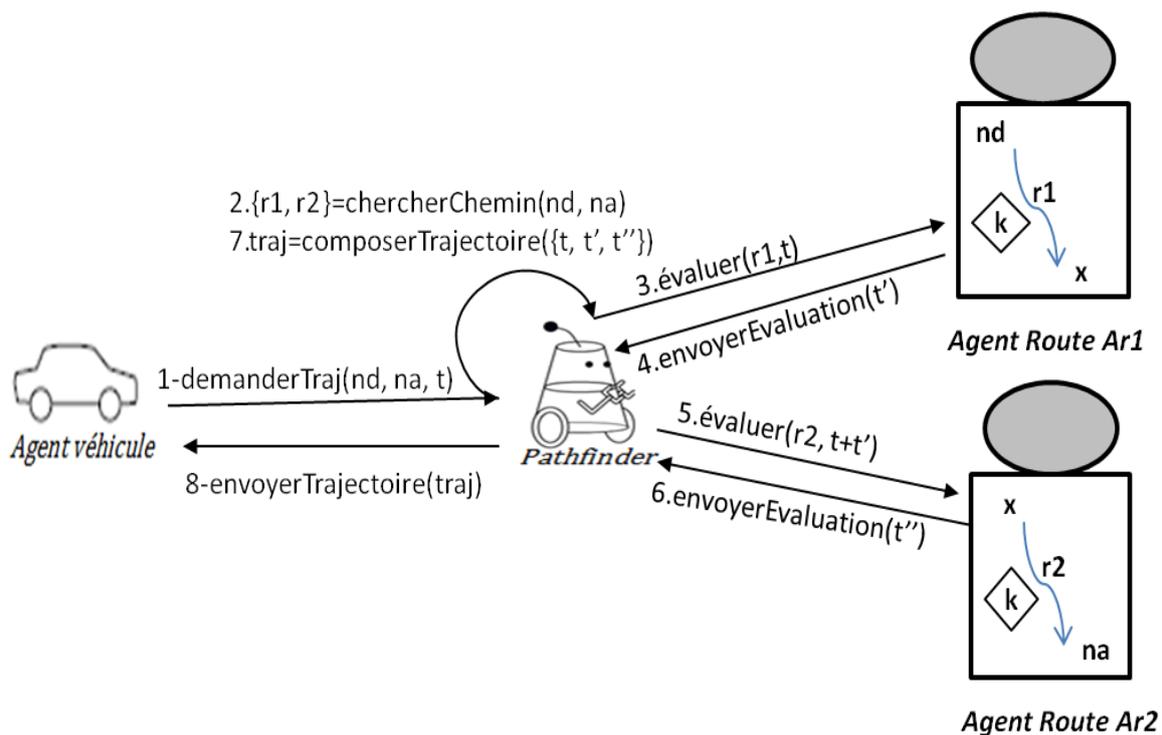


Figure 34 : Chercher une trajectoire entre deux lieux

6.3 Méthode d'évaluation d'une route

Pour évaluer le temps nécessaire pour traverser une route du chemin à partir d'un temps initial, l'agent Route associé à la route, utilise la procédure de calcul du temps de voyage

proposée par (Ichoua, Gendreau, et Potvin 2003). Cette procédure respecte la propriété FIFO²⁰ disant que deux véhicules qui se déplacent sur le même chemin reliant un lieu de départ avec un lieu d'arrivée, doivent arriver au lieu d'arrivée dans le même ordre dans lequel ils ont quitté le lieu de départ. L'algorithme 8 présente cette procédure de calcul. Supposons que le temps de départ de la route est t_0 où $t_0 \in T = [t_{min}, t_{max}]$. Soient d la distance à parcourir de la route et v_{ijT} la vitesse calculée selon la situation du trafic pendant l'intervalle T en se référant à la *liste des situations de trafic prévues* de la route.

Algorithme 8 : Algorithme de calcul du temps requis pour traverser une route r_i par un agent Route Ar_i à partir d'un temps initial t_0

$t = t_0$; $d = d_{ij}$; $T = \text{getIntervalle}(t)$;

$T = [t_{min}, t_{max}]$

$t' = t + d/v_{ijT}$

tant que ($t' > t_{max}$) **faire**

$d = d - v_{ijT} (t_{max} - t)$

$t = t_{max}$

$T = T + 1$

$t' = t + d/v_{ijT}$

fin

retourner ($t' - t_0$)

Dans la section suivante, deux méthodes de négociation seront proposées afin de choisir le bon véhicule qui va exécuter la demande.

7 Stratégies de négociation entre les agents Véhicule

Comme nous l'avons déjà expliqué, l'agent Client crée une liste qui va contenir tous les agents Véhicule disponibles pour servir une nouvelle demande.

Après avoir reçu toutes les réponses des agents Véhicule concernant une nouvelle demande, l'agent Client distribue la liste contenant les agents Véhicule disponibles à chaque agent Véhicule présent dans cette liste. A cet instant, le processus de négociation débute. Les agents

²⁰ La propriété FIFO doit être respectée dans un problème considérant la variation de la vitesse de voyage en fonction du temps.

Véhicule qui reçoivent la liste commencent à négocier entre eux. Le but de cette négociation est de choisir le bon agent Véhicule qui transportera les passagers de la demande. Le bon agent Véhicule choisi est celui qui minimise le nombre de véhicules en circulation. Dans le cas où deux agents Véhicule minimisent de la même manière ce paramètre, alors celui qui minimise la distance totale parcourue pour servir la demande est choisi.

Nous proposons deux stratégies de négociation (Malas et al. 2016b) et nous les comparons avec une troisième proposée dans la littérature (Bertelle et al. 2009). La première stratégie est appelée "*envoyer aux suivants*" (envoyer à ceux qui suivent) et la deuxième est appelée "*envoyer au suivant*" (envoyer à celui qui suit). Le processus de négociation est réalisé par adaptation d'une de ces deux stratégies.

7.1 Première stratégie de négociation : "*envoyer aux suivants*"

La première stratégie de négociation "*envoyer aux suivants*" commence par le premier agent Véhicule présent dans la liste diffusée par l'agent Client et ainsi de suite jusqu'à atteindre le dernier agent. A tour de rôle, chaque agent Véhicule diffuse un message aux agents qui le suivent dans la liste. Ce message contient le bon agent Véhicule, le minimum des nombres de places vides de son véhicule durant la période du service de la demande et la distance totale parcourue par le véhicule pour servir la demande (le premier agent Véhicule dans la liste diffuse ses deux propres valeurs). À chaque fois qu'un agent Véhicule reçoit un message d'un autre, il déclenche la méthode de comparaison. Cette méthode consiste à ce que cet agent compare les valeurs contenues dans le message avec ses propres valeurs. Il commence à comparer les premières valeurs représentant le minimum des nombres de places vides et retient l'agent Véhicule qui propose la valeur minimale. Dans le cas où les deux agents présentent la même valeur, il passe à comparer les deuxièmes valeurs représentant la distance totale parcourue. Il retient celui qui propose la distance minimale. Au cas où les deux agents présentent les mêmes distances, l'agent Véhicule garde ses propres valeurs.

En résumé, à chaque fois qu'un agent Véhicule reçoit un message d'un autre, il compare ses valeurs avec les valeurs reçues et maintient le bon agent Véhicule avec ses deux valeurs proposées. Lorsque son tour arrive, il diffuse son message aux agents Véhicule suivants. Lorsque le tour du dernier agent Véhicule de la liste arrive, selon les informations retenues, il déclare le bon agent Véhicule. Enfin, ce dernier informe l'agent Client de son engagement de transporter les passagers de la demande et lui envoie un message contenant l'heure de départ et l'heure d'arrivée prévues.

L'algorithme 9 présente la première méthode de négociation.

Algorithme 9 : Algorithme de la première méthode de négociation "envoyer aux suivants"

L : liste des agents Véhicule disponibles à servir une demande

N : dimension(L)

pour i allant de 0 à $N-1$ **faire**

$Av = L[i]$

pour j allant de $i + 1$ à $N-1$ **faire**

$Av.diffuse(Av.msg, L[j])$

$L[j].msg = L[j].compare(msgRecu)$

fin

fin

$Av.declareBonAgent(Av.msg)$

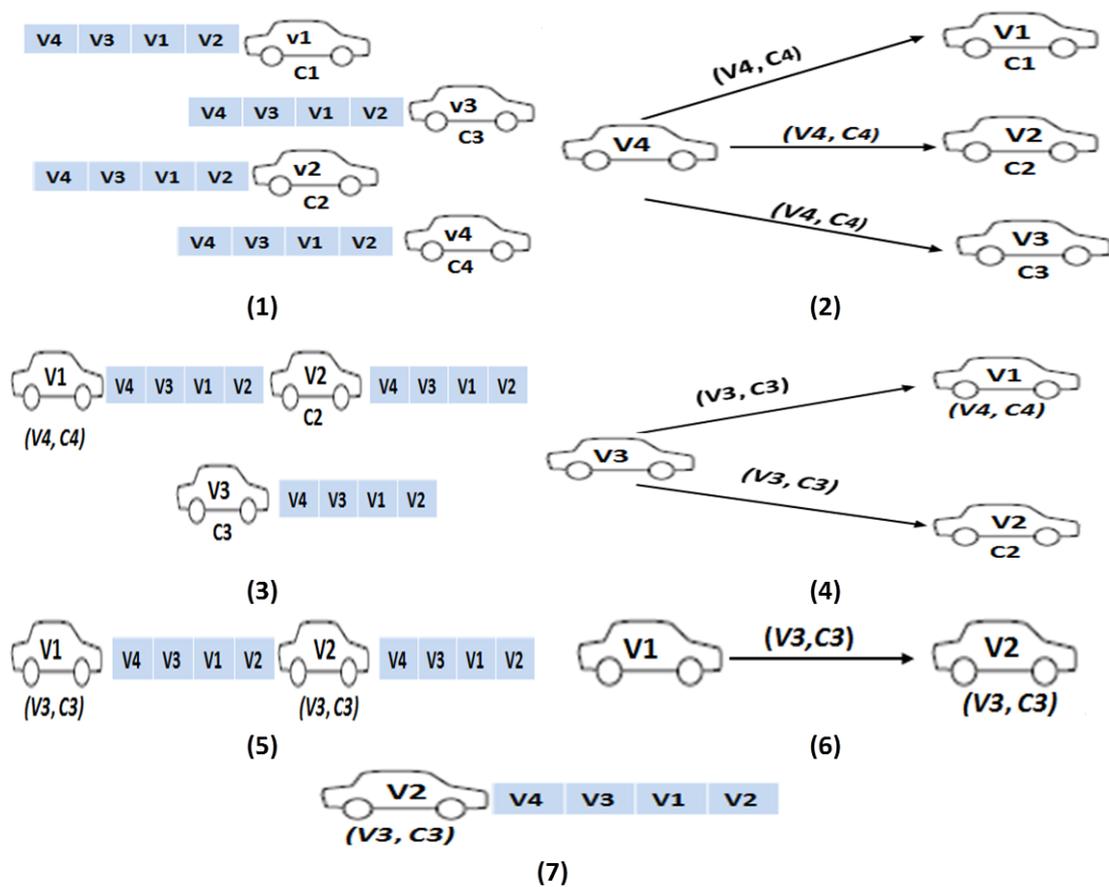


Figure 35 : La méthode de négociation "envoyer aux suivants"

La figure 35 explique sur un exemple la première stratégie de négociation proposée "*envoyer aux suivants*". Supposons que la négociation se déroule entre quatre agents Véhicule v_1 , v_2 , v_3 et v_4 . Pour i allant de 1 à 4, v_i présente c_i comme nombre de véhicules en circulation. Supposons que $c_3 < c_2 < c_4 < c_1$. L'image numéro (1) représente les quatre agents capables d'exécuter une demande avec la liste des agents Véhicule diffusée par l'agent Client. Dans l'image (2), le premier agent Véhicule dans la liste v_4 diffuse le message aux agents Véhicule suivants : v_3, v_2, v_1 . Nous remarquons dans l'image (3) que v_1 maintient v_4 avec sa valeur c_4 et les autres gardent leur propre valeur parce que $c_3 < c_2 < c_4 < c_1$. Dans l'image (4), le deuxième agent Véhicule de la liste v_3 diffuse son message à ceux qui le suivent v_1 et v_2 . Dans l'image (5), v_1 et v_2 gardent l'agent Véhicule v_3 avec sa valeur proposée c_3 . Dans l'image (6), v_1 envoie son message au dernier agent Véhicule dans la liste v_2 . L'image (7) montre que v_2 maintient v_3 avec sa valeur. Finalement, v_2 déclare v_3 comme le bon agent Véhicule.

7.2 Deuxième méthode de négociation : "*envoyer au suivant*"

La deuxième stratégie de négociation commence par le premier agent Véhicule de la liste et ainsi de suite jusqu'à atteindre le dernier. A tour de rôle, chaque agent diffuse un message à celui qui le suit dans la liste. Ce message contient le bon agent Véhicule et ses deux valeurs proposées : le minimum des nombres de places vides de son véhicule durant la période du service de la demande et la distance totale parcourue par le véhicule pour servir la demande (le premier agent Véhicule dans la liste diffuse ses deux propres valeurs). L'agent Véhicule suivant qui reçoit le message, compare les valeurs reçues avec ses valeurs proposées. L'opération de comparaison est la même que celle adaptée dans la première stratégie de négociation. Elle a comme but de réaliser l'objectif du système du TAD. Alors, l'agent Véhicule commence à comparer les premières valeurs représentant le minimum des nombres de places vides des deux véhicules durant la période du service de la demande et maintient celui qui propose la valeur minimale. Dans le cas où ces deux valeurs sont égales, l'agent Véhicule compare les deux dernières valeurs représentant la distance totale parcourue et il maintient celui qui propose la distance minimale. Ensuite, il diffuse le message à l'agent qui le suit et l'opération de comparaison recommence jusqu'à que le tour du dernier agent Véhicule arrive. A ce moment, ce dernier détermine le bon agent. Enfin, le bon agent informe l'agent Client que c'est lui qui va transporter les passagers en déterminant les heures de départ et d'arrivée prévues. L'algorithme 10 présente la deuxième méthode de négociation.

Algorithme 10 : Algorithme de la deuxième méthode de négociation "*envoyer au suivant*"

L : liste des véhicules disponibles à servir une demande

N : dimension(L)

pour i allant de 0 à $N-2$ **faire**

$Av = L[i]$

$Av' = L[i+1]$

Av .diffuse(Av .msg, Av')

Av' .msg = Av' .compare(Av .msg)

fin

La figure 36 explique sur un exemple la deuxième méthode de négociation proposée "*envoyer au suivant*". Supposons que la négociation se déroule entre quatre agents Véhicule v_1 , v_2 , v_3 et v_4 . Pour i allant de 1 à 4, v_i présente c_i comme le nombre de véhicules en circulation. Supposons que $c_3 < c_2 < c_4 < c_1$. A tour de rôle et en commençant par le premier agent Véhicule de la liste, v_4 diffuse son message à l'agent suivant v_3 . Nous remarquons que v_3 maintient sa propre valeur c_3 parce que $c_3 < c_2 < c_4 < c_1$. Ensuite, le tour de v_3 arrive. Il envoie son message à celui qui le suit v_1 . Nous remarquons que v_1 garde la valeur de v_3 . Après, v_1 envoie son message au dernier agent Véhicule de la liste v_2 . Ce dernier maintient à nouveau la valeur de v_3 . Finalement, v_2 déclare v_3 comme étant le bon agent Véhicule.

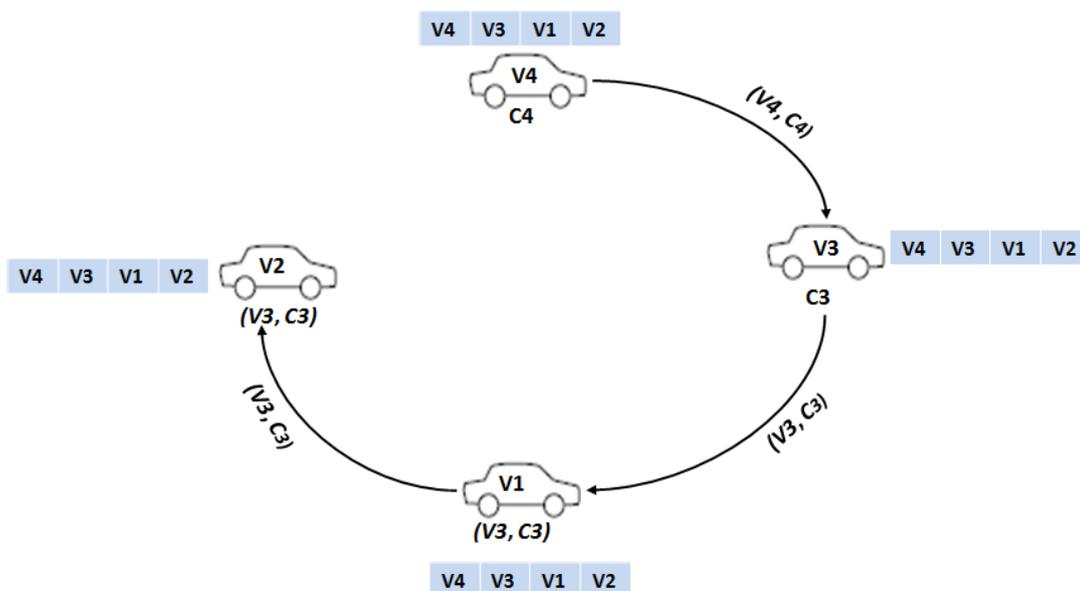


Figure 36 : La méthode de négociation "*envoyer au suivant*"

7.3 Résultats théoriques et expérimentaux

Dans cette section, nous comparons les deux stratégies de négociation présentées ci-dessus avec celle proposée dans (Bertelle et al. 2009). Trois critères de comparaison sont étudiés théoriquement pour chaque stratégie : la complexité, le nombre de messages échangés et la tolérance aux pannes (la fiabilité). Enfin, nous effectuons des expérimentations pour valider les résultats théoriques concernant le nombre de messages échangés entre les agents Véhicule de chaque stratégie.

7.3.1 Résultats théoriques

L'architecture de chacune des trois stratégies est décentralisée. Ainsi, les traitements sont distribués. Le seul désavantage de la distribution du traitement est que le nombre de messages échangés augmente considérablement. Dans cette section, nous étudions théoriquement le nombre de messages échangés entre les agents Véhicule de ces trois stratégies de négociation.

Supposons qu'il y a n agents Véhicule disponibles à servir une demande. Nous intitulons la stratégie de négociation proposée dans (Bertelle et al. 2009) "*envoyer à tous*". Dans cette stratégie, chaque agent Véhicule calcule le coût de servir une nouvelle demande et le diffuse à tous les autres agents Véhicule. Cela implique que, chacun envoie $n-1$ messages. Par conséquent, les n agents Véhicule envoient $n * (n-1)$ messages.

Dans la stratégie "*envoyer aux suivants*", chaque agent Véhicule envoie un message à chaque agent qui le suit dans la liste. Alors, le premier agent Véhicule envoie $n-1$ messages, le deuxième envoie $n-2$ messages, et le $(n-1)^{\text{ème}}$ envoie un seul message au dernier agent. Nous notons que le nombre total de messages échangés de cette stratégie est égale à la somme de i allant de 1 à $n-1$: $\sum_{i=1}^{n-1} i = n * (n - 1)/2$.

Dans la stratégie "*envoyer au suivant*", chaque agent Véhicule envoie un seul message à l'agent suivant, sauf le dernier. Donc pour n véhicules, il y a $n-1$ messages échangés entre eux.

Le tab 11 récapitule les résultats concernant la complexité et le nombre de messages échangés pour les trois stratégies de négociation. Nous notons que la première stratégie est de complexité $O(n^2)$ et de $n(n - 1)$ messages échangés. Tandis que la stratégie "*envoyer aux suivants*" est de complexité $O(n^2)$ mais le nombre de messages échangés est réduit à la moitié par rapport à la première stratégie : $n(n - 1)/2$. Concernant la troisième stratégie "*envoyer*

au suivant", elle est de complexité $O(n)$ et un nombre de messages très réduit qui est égal à $n - 1$ messages échangés.

Types de stratégie	Complexité	Messages échangés
"envoyer à tous"	$O(n^2)$	$n(n - 1)$
"envoyer aux suivants"	$O(n^2)$	$n(n - 1)/2$
"envoyer au suivant"	$O(n)$	$n - 1$

Tab 11 : Résultats théoriques de la comparaison

7.3.2 Fiabilité des stratégies de négociation

La tolérance aux pannes représente la capacité du système à résister aux pannes ou aux défaillances d'un ou de plusieurs agents Véhicule. L'agent Véhicule est considéré en panne ou défaillant lorsque la voiture tombe en panne ou lorsqu'elle perd son moyen de communication (le système de communication est défaillant). Nous traitons le cas où l'agent Véhicule confronte des problèmes avant d'envoyer son message.

Dans la stratégie "envoyer à tous", les valeurs proposées par chaque agent Véhicule sont distribuées à tous les autres agents. Ainsi, chacun de ses agents a toujours la bonne information concernant le bon agent Véhicule. Dans le cas où un de ses agents tombe en panne, les agents restants continuent à se négocier en ignorant la défaillance de cet agent Véhicule. La bonne information est toujours gardée avec les autres agents. Ainsi, cette stratégie est fiable.

Dans la stratégie "envoyer aux suivants", à tour de rôle, chaque agent Véhicule distribue l'information relative au bon agent vers tous les agents qui le suivent dans la liste. Ainsi, chaque agent participant à la négociation aura toujours la bonne réponse (le bon véhicule avec le moindre coût). Supposons que le tour arrive à un agent en panne. Au bout d'un certain temps (timeout), l'agent Véhicule qui le suit dans la liste, va détecter qu'il y a un problème. Alors, cet agent prend le relais et continue la négociation en ignorant l'échec de l'agent Véhicule qui le précède. Donc, l'information concernant le bon agent Véhicule est sauvegardée. Recommencer la négociation depuis le début n'est plus à considérer. Par conséquent, cette stratégie est tolérante aux pannes.

La stratégie "envoyer au suivant" est la moins tolérante aux pannes. Si un agent Véhicule impliqué dans la négociation tombe en panne ou perd son moyen de communication, alors l'agent qui le suit dans la liste n'a pas et n'obtiendra aucune information concernant les agents Véhicule qui le précèdent dans la liste. Dans ce cas, le processus de négociation est

interrompu parce que l'information concernant le bon agent Véhicule est perdue. La négociation doit être redémarrée dès le début.

Nous remarquons que la deuxième stratégie présente un nombre de messages échangés plus petit que celui de la première. En outre, elle est plus tolérante aux pannes que la troisième. Alors, elle fait un compromis entre le nombre de messages échangés et la tolérance aux pannes.

Or, le choix de la méthode de négociation dépend de la fiabilité du système. Si le système est fiable, la troisième méthode de négociation sera le plus convenable car elle présente le minimum de nombre de messages échangés. En revanche, si le système n'est pas fiable et peut défaillir (tomber en panne) à cause des problèmes de communication, la deuxième méthode sera plus convenable. Elle garde toujours la meilleure solution même si un ou plusieurs agents Véhicule tombent en panne. Pour cela, nous adoptons la méthode "*envoyer aux suivants*" comme méthode de négociation adaptée dans nos approches.

7.3.3 Résultats expérimentaux

Afin d'étudier expérimentalement le nombre de messages échangés entre les agents Véhicule dans chaque stratégie de négociation, nous utilisons la même plate-forme et les mêmes paramètres d'expérimentations décrits dans le chapitre 3 dans le cas de l'infrastructure réelle²¹. De 500 demandes de transport créées et sauvegardées dans un fichier XML, nous utilisons 400 demandes et 100 agents Véhicule. Nous exécutons ensuite le système pour mesurer le nombre de véhicules parmi les 100 disponibles à servir chaque demande de transport. Les résultats montrent que 10 à 40 véhicules sont disponibles pour servir 17 % des demandes, entre 40 et 80 véhicules sont disponibles pour servir 47 % des demandes et 36 % des demandes peuvent être servies par 80 à 100 véhicules.

Pour chaque stratégie de négociation, nous mesurons par des expérimentations le nombre de messages échangés entre les agents Véhicule disponibles à servir une demande. Nous étudions cinq types de demandes. Le premier type est une demande qui peut être servie par 20 agents Véhicule, le deuxième type est une demande que 40 agents Véhicule sont disponibles pour la servir, le troisième type est une demande qui peut être servie par 60 agents, le quatrième est une demande qui peut être servie par 80 agents Véhicule et le dernier est une demande qui peut être servie par 100 agents Véhicule.

²¹ Voir chapitre 3, paragraphe 7.2.2

Les résultats des expérimentations sont présentés dans la figure 37. Elle montre le nombre de messages échangés résultant de la négociation entre un nombre fixe des agents Véhicule. Nous notons que ces résultats confirment les résultats théoriques présentés dans le tab 11.

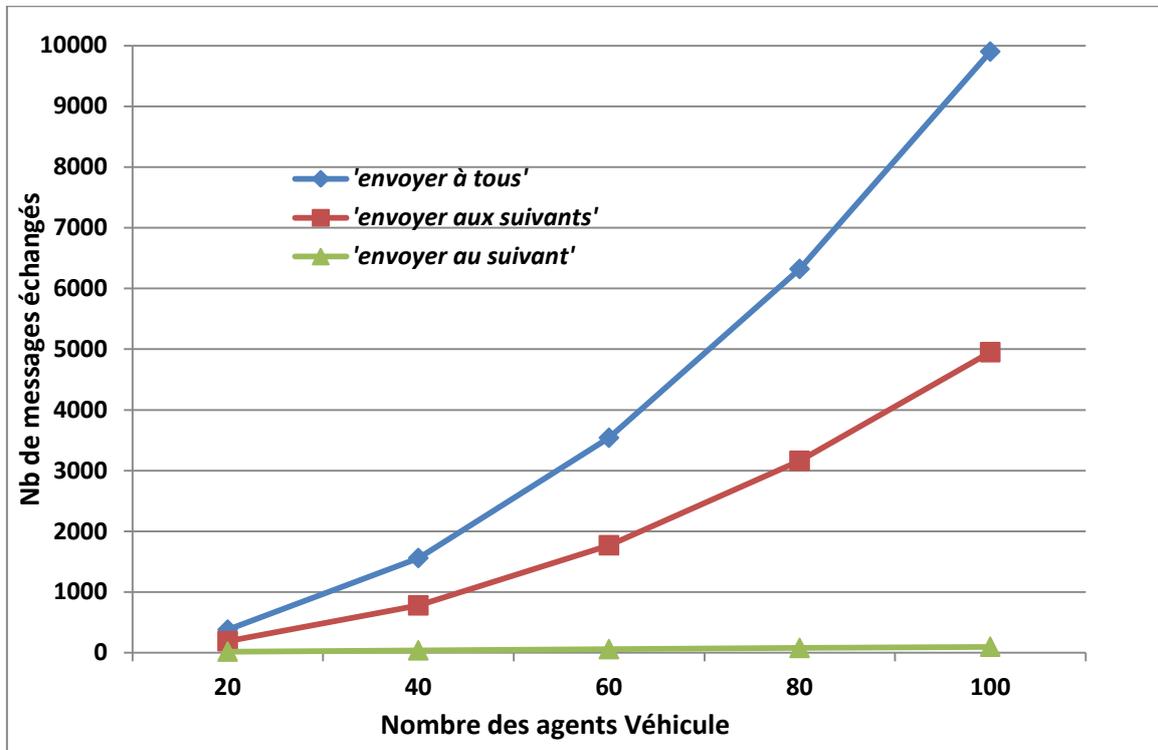


Figure 37 : Nombre de messages échangés en fonction du nombre des agents Véhicule

8 Expérimentations

Afin de tester la nouvelle approche proposée et montrer la nécessité de considérer la dépendance des vitesses de voyage du temps, nous utilisons la même plate-forme et les mêmes paramètres des expérimentations faites dans le chapitre précédent²² pour le cas de l'infrastructure réelle. Dans les expériences, nous mesurons le nombre de demandes (ou clients) servies dans les deux modèles suivants : le premier modèle considère les vitesses de voyage fixes et indépendantes du temps. Le deuxième considère les vitesses de voyage dépendantes du temps. Ensuite, nous mesurons le nombre de clients insatisfaits parmi ceux servis si leurs demandes sont exécutées dans le premier modèle. Nous utilisons les 500 demandes créées et nous distinguons cinq cas. Le premier cas comprend 100 demandes, le deuxième est composé de 200, le troisième est constitué de 300, la quatrième contient 400

²² Voir chapitre 3, paragraphe 7.2.2

demandes et le dernier comprend 500. Chaque cas contient les demandes du cas précédent. Les résultats des expériences sont présentés dans le graphe de la figure 38. Ce dernier montre le nombre de demandes servies parmi un nombre fixe de demandes exécutées dans les deux modèles et le nombre de clients insatisfaits à cause de l'exécution de leurs demandes dans le premier modèle.

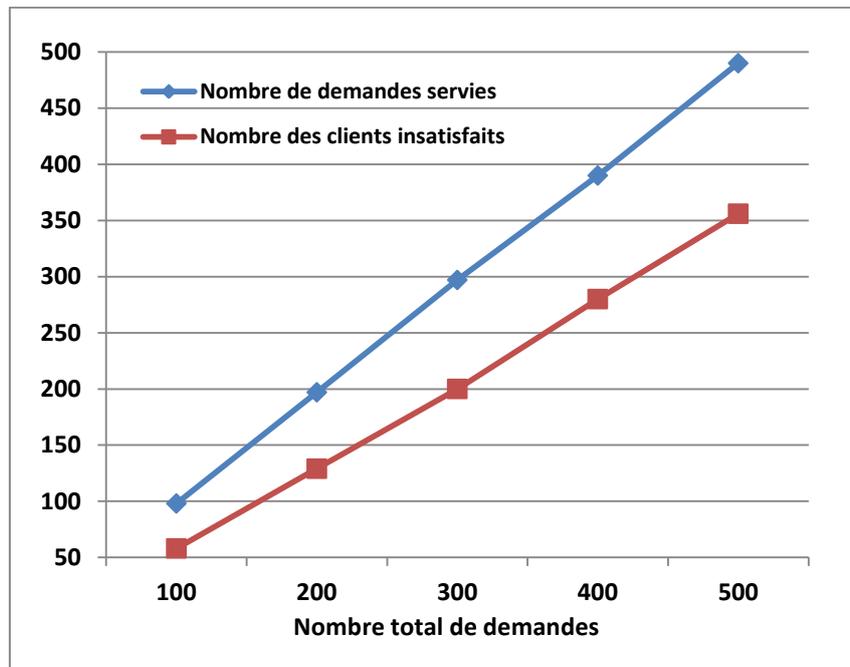


Figure 38 : Nombre de demandes servies et nombre de clients insatisfaits

Les résultats montrent que les deux modèles donnent le même nombre de demandes servies à partir d'un nombre fixe. Mais nous notons qu'il y a plus que 50 % de clients insatisfaits parmi ceux servis lorsque leurs demandes sont exécutées dans le premier modèle. Cela est dû à la violation de la fenêtre de temps de leurs demandes. Nous constatons que la considération de la relation entre la vitesse de voyage et le temps est indispensable et aboutit à augmenter la satisfaction des clients suite à la qualité de service.

9 Conclusion

Dans ce chapitre, nous avons procédé à une agentification massive du problème du TAD dans une ville caractérisée par des vitesses de voyages qui varient selon le temps. Cette approche utilise les connaissances historiques du trafic de la ville pour estimer le temps nécessaire pour traverser les routes d'une trajectoire à partir d'un temps donné. L'approche vise à respecter les temps de départ et d'arrivée exigés par les clients et à minimiser le nombre des véhicules en circulation et leurs distances totales parcourues.

De plus, nous avons présenté deux stratégies de négociation afin de choisir le meilleur agent Véhicule qui va s'occuper de servir une nouvelle demande. Ces deux stratégies sont comparées avec une troisième de la littérature. Une de ses stratégies est choisie comme méthode de négociation adaptée dans nos approches. Ce choix est fait parce que cette stratégie fait un compromis entre le nombre de messages échangés entre les agents et la tolérance aux pannes.

Les résultats expérimentaux montrent plus que 50 % des clients sont insatisfaits dans le cas où la vitesse de voyage est considérée constante. Donc, ils confirment la nécessité de prendre en compte la variation des vitesses de voyage selon le temps afin d'augmenter la qualité de service et par la suite la satisfaction des clients.

Or, les connaissances historiques du trafic sont insuffisantes pour refléter la réalité de la situation du trafic surtout en cas d'apparition d'un événement imprévu comme un accident. De plus, nous devons penser à une méthode permettant de réduire le nombre de messages échangés entre les agents. Comme par exemple, fusionner les agents Route ayant des caractéristiques communes dans un seul agent. Finalement, ce très grand nombre d'agents Route doit être régulé et mis à jour en cas de situation du trafic imprévue. Pour toutes ces raisons, nous allons résoudre, dans le chapitre suivant, le problème du TAD dynamique tout en considérant la situation réelle du trafic dans le calcul des trajectoires des véhicules. Nous allons étendre l'approche de ce chapitre pour inclure des informations de trafic en temps réel dans l'objectif d'obtenir une solution juste, satisfaisante et actualisée selon la situation réelle du trafic.

Chapitre 5. Approche multi-agent massif auto-organisé pour le transport à la demande dans une ville

1 Introduction

Dans le chapitre précédent, nous avons montré que les vitesses de voyage sur les routes d'une ville dépendent de la situation habituelle du trafic routier et que plus que 50 % des clients seront insatisfaits si ces vitesses sont considérées constantes. Pour surmonter ce problème, nous avons associé un agent Route à chaque route géographique. Cet agent a comme rôle d'évaluer la durée nécessaire pour traverser sa route à partir d'un temps initial en se basant sur des connaissances historiques du trafic. Or, les connaissances historiques ne suffisent pas pour refléter la situation réelle du trafic. Elles doivent être accompagnées par des informations du trafic en temps réel. Dans l'étude faite par (Kim, Lewis, et White 2005), la prise en compte des informations du trafic permet une réduction de 12 % de l'utilisation de véhicules. 8.4 % de la réduction est due à des informations historiques et 3.6 % est due à des informations du trafic en temps réel.

Ce chapitre traite le problème du transport à la demande (TAD) dans une ville tout en :

- minimisant le nombre d'agents Route de l'approche précédente pour minimiser les échanges entre eux et avec eux ;
- considérant la situation réelle avec la situation habituelle de trafic. Cela afin que le système TAD soit capable de donner une solution basée sur des informations fiables. De cette manière, les clients seront satisfaits.

Pour réaliser ces deux objectifs, nous améliorons l'approche proposée dans le chapitre 4. Nous proposons une approche multi-agent massif auto-organisé pour modéliser et résoudre le problème du TAD avec un réseau routier dynamique stochastique. L'agent Route sera associé à des routes ayant la même situation du trafic et la même vitesse limite durant un intervalle de temps commun. Il peut se décomposer lorsque ses routes possèdent des situations différentes du trafic et fusionner avec un autre agent Route voisin pour en devenir un seul au cas où les routes des deux agents présentent la même situation du trafic. De cette manière, le nombre d'agents Route sera optimisé.

La question posée est : comment refléter la situation réelle du trafic surtout en cas d'apparition d'un événement imprévu tel qu'un accident sur une route ? Comment refléter l'impact spatial (les routes affectées) et temporel (la durée d'affectation) de cet événement sur la situation habituelle du trafic ? A cet effet, nous examinons le mécanisme selon lequel le système du TAD réagit à ce type de problème. Nous intégrons un nouvel agent, appelé agent Régulateur, au modèle d'agents proposé dans le chapitre précédent. Cet agent est associé à des routes ayant la même situation du trafic durant un intervalle de temps commun. Les agents Route et Régulateur sont dynamiques. Ils s'auto-organisent en ligne en se basant d'une part sur des connaissances historiques relatives au trafic et à l'impact des événements imprévus sur le trafic et d'autre part sur des informations du trafic en temps réel. Le but de cette auto-organisation est de représenter la vraie organisation du trafic à son échelle dans le système du TAD.

En résumé, ce système serait capable :

- d'estimer la vitesse de voyage sur chaque route à tout instant présent et futur ;
- d'estimer la durée exacte des trajectoires des véhicules ;
- d'anticiper l'impact des événements imprévus sur la situation du trafic ;
- d'actualiser la durée des trajectoires suite à des situations du trafic inhabituelles causées par des événements imprévus ;
- d'éviter les véhicules de passer par des routes congestionnées en les déroutant ;
- de produire des informations exactes et des résultats fiables.

Ce chapitre sera organisé comme suit : dans la deuxième section, nous présentons l'idée *d'un scénario dynamique stochastique* du problème du TAD. Dans la troisième, nous introduisons l'algorithme général pour la résolution de ce scénario. La section 4 présente la structure de données des principaux composants du système du TAD de ce scénario. La section 5 explique la modélisation multi-agent massif du scénario en définissant les agents et leurs rôles. Puis, nous donnons dans la section 6 un exemple introductif. Nous passons à expliquer dans la section 7 le traitement d'une demande de transport. Dans une huitième section, nous examinons l'auto-organisation du système du TAD pour traiter l'évolution de la situation dynamique du trafic. La section 9 explique la gestion des connaissances historiques. La section 10 présente les expérimentations faites pour tester l'approche proposée. La dernière section conclut le chapitre.

2 Troisième scénario du problème du TAD : *scénario dynamique stochastique*

Le troisième scénario du problème du TAD est appelé *scénario dynamique stochastique*. L'idée principale de ce scénario est la suivante : des clients sont présents dans des différents endroits d'une ville. Chacun déclenche une demande de transport en indiquant le lieu de départ, le lieu d'arrivée, le temps de départ au plus tôt, le temps d'arrivée au plus tard et le nombre de passagers à transporter.

La ville est caractérisée par des routes avec des vitesses de voyage dynamiques stochastiques. Elles sont dynamiques car elles varient en fonction du temps selon le trafic et stochastiques car elles dépendent des informations du trafic en temps réel. Les connaissances historiques du trafic ne sont pas suffisantes pour refléter la situation réelle du trafic. Des événements imprévus (par exemple, un accident) peuvent changer le rythme habituel du trafic.

L'objectif du système du TAD est de chercher pour chaque demande de transport le bon véhicule avec la trajectoire qui va suivre pour transporter les passagers tout en :

- respectant les contraintes du client (le temps de départ au plus tôt et le temps d'arrivée au plus tard) et du véhicule (le nombre de places vides et les contraintes temporelles concernant leurs engagements avec d'autres clients) ;
- minimisant le nombre de véhicules en circulation puis la distance totale parcourue ;
- considérant la situation réelle du trafic. Pour que le système reflète la situation réelle du trafic, il se base d'une part sur les connaissances historiques relatives au trafic et à l'impact des événements imprévus sur le trafic et d'autre part sur des informations de trafic captées en temps réel. D'abord, il évalue le chemin du véhicule en se basant sur les connaissances historiques du trafic de la ville. Ultérieurement, il actualisera cette évaluation suite à l'apparition d'un événement imprévu en se basant sur des informations en temps réel et sur des connaissances historiques sur l'impact de cet événement sur le trafic.

3 Algorithme général de la résolution du scénario *dynamique stochastique*

L'algorithme général qui décrit le système du TAD du scénario *dynamique stochastique* est le suivant :

Début

Partie 1 : Formulation d'une nouvelle demande de transport

Un sous-système prend les informations de la demande d'un client connecté pour aller d'une position x à un temps de départ au plus tôt t_x et pour arriver à la position y à un temps d'arrivée au plus tard t_y .

Partie 2 : Traitement de la réception de la nouvelle demande de transport

- Des véhicules sont sollicités pour satisfaire la demande de transport du client. Chaque véhicule fait une proposition (selon différents critères) en se basant sur les connaissances historiques du trafic routier.
- Suivant la meilleure proposition, un véhicule est choisi par le système selon un protocole de négociation pour transporter le client.
- Le véhicule choisi propose le temps de départ ($\geq t_x$) et le temps d'arrivée ($\leq t_y$) prévus au client.
- Les passagers et le véhicule s'entendent.

Partie 3 : Traitement de changement de la situation du trafic

Un sous-système donne en continu la situation réelle de la fluidité du trafic sur chaque route. Cette situation peut varier plus ou moins fortement selon les moments et informe les véhicules de la situation de leurs trajectoires en cours.

Partie 4 : Transport et circulation des véhicules avec les clients

- Un véhicule prend les passagers pour effectuer la trajectoire.
- La trajectoire du véhicule est évaluée de façon continue indiquant toutes les perturbations possibles.
- Pendant le parcours, le véhicule est informé de façon continue par le système de la variation de sa trajectoire à cause :
 - de l'arrivée d'une nouvelle demande ;
 - d'une variation imprévue de la situation du trafic.
- Les passagers sont arrivés au point de destination au temps t .

Fin

4 Structure de données des composants du système du TAD

Les composants du système du TAD du scénario dynamique stochastique sont les mêmes que les scénarios précédents : l'infrastructure de la ville, la demande de transport, le véhicule et la trajectoire. L'infrastructure qui représente un réseau routier dynamique stochastique est le seul élément qui a changé de structure. Dans ce qui suit, nous détaillons la structure de données de cette infrastructure.

4.1 Infrastructure dynamique incertaine

L'infrastructure du système du TAD du scénario *dynamique stochastique* symbolise un réseau routier d'une ville avec des vitesses de voyage dynamiques stochastiques. Elles sont dynamiques car elles varient avec le temps et stochastiques car elles ne sont pas connues par avance. Elles sont affectées par l'évolution de la situation dynamique du trafic routier de la ville. Dans ce scénario, nous distinguons deux types de situation dynamique : une situation dynamique déterministe et une situation dynamique stochastique. La situation déterministe est connue par avance. Elle est déterminée par les connaissances historiques du trafic habituel. Ce type a été considéré dans le chapitre précédent. Par contre, la situation stochastique ne peut pas être prédite. Elle est causée par des événements imprévus et connus par des informations du trafic captées en temps réel. Ces informations dérivent des techniques présentées dans le chapitre 2²³.

Nous représentons l'infrastructure du *scénario dynamique stochastique* par un graphe $G = (P, R)$ composé d'un ensemble de nœuds $P = \{p_1, \dots, p_n\}$ désignant l'ensemble de lieux de la ville et un ensemble d'arcs $R = \{r_1, \dots, r_n\}$ désignant les routes de la ville.

Dans ce qui suit, la représentation des nœuds et des routes de l'infrastructure dans le système du TAD est exhibée. Avant tout, nous définissons un événement imprévu et sa représentation dans le système.

4.1.1 Événement imprévu

Un événement imprévu apparaît soudainement sur une route et affecte directement la situation du trafic. Il survient à n'importe quel instant au cours de la circulation des véhicules et cause une variation imprévue dans la situation habituelle du trafic. Un accident, une panne d'un véhicule, une manifestation, les conditions météorologiques et tout élément qui varie le

²³ Pour plus de détails, voir chapitre 2, paragraphe 6.1.1.

rythme habituel du trafic sont considérés comme un événement imprévu. L'émergence d'un événement imprévu sur l'une des routes est connue par des informations en temps réel. Ces informations peuvent dériver des systèmes de détection automatique d'incidents et des bouchons (comme les capteurs vidéo). Ces systèmes s'occupent du traitement et de l'analyse des images prises par les caméras et les vidéos situées sur les routes.

Chaque événement imprévu est défini dans le système du TAD par les cinq paramètres suivants :

- le type de l'événement (accident, manifestation, etc.) ;
- la localisation où l'événement a lieu ;
- le temps d'apparition de l'événement ;
- la sévérité de l'événement. Un événement peut être simple, sévère ou très sévère.

4.1.2 Nœud

Un nœud représente un lieu de la ville où deux ou plusieurs routes se rencontrent. Il est défini par :

- un nom ;
- une localisation défini par des coordonnées ;
- une liste des routes entrantes ;
- une liste des routes sortantes.

4.1.3 Route

Une route est un arc reliant deux nœuds. Elle est définie par deux types de paramètres: les paramètres concernant la route et ceux concernant la situation du trafic de la route.

Les paramètres concernant la route sont :

- un type (autoroute, route nationale, route départementale, route locale) ;
- un nom ;
- une longueur en km ;
- un sens de circulation défini par son nœud initial et son nœud final. A titre d'exemple, l'arc reliant les deux nœuds A et B dans le sens allant de A vers B forme une route. L'autre sens allant de B vers A forme une autre route ;
- une vitesse limite de voyage en km/h ;
- une liste de routes voisines entrantes ayant le nœud initial de la route comme nœud d'intersection ;

- une liste de routes voisines contigües sortantes ayant le nœud final de la route comme nœud d'intersection.

Les paramètres concernant la situation du trafic de la route sont :

- une *liste des situations de trafic prévues*²⁴. Cette liste reflète les situations habituelles quotidiennes du trafic de la route. Elle est construite en se basant sur des connaissances historiques relatives au trafic de la route ;
- une situation du trafic en temps réel. Cette situation résulte du traitement et de l'analyse des informations captées en temps réel par des capteurs de collecte des données sur le trafic²⁵. Elle est utilisée pour confirmer la situation prévue ;
- une *liste d'impact des événements imprévus*. Cette liste reflète l'impact d'apparition d'un événement imprévu sur la route. Un événement imprévu a un impact spatial (les routes affectées) et temporel (la durée d'affectation) sur le trafic routier. Cet impact dépend de cinq paramètres : le type, le temps d'apparition, la localisation et la sévérité de l'événement et la catégorie du jour. Selon ces paramètres, la *liste d'impact des événements imprévus* est construite en se basant sur des connaissances historiques relatives à l'impact des événements imprévus sur la situation habituelle du trafic de la route et des routes voisines. Le système du TAD se base sur la liste pour prévoir la situation future du trafic de la route et de ses routes voisines suite à l'apparition d'un événement imprévu sur la route. Chaque élément de la liste est défini par les paramètres suivants :
 - le type de l'événement ;
 - l'intervalle du temps durant lequel l'événement a eu lieu ;
 - la sévérité de l'événement ;
 - la catégorie du jour dans lequel l'événement imprévu a eu lieu ;
 - les routes affectées par l'événement. Chaque route est affectée de manière différente qu'une autre. Cette affectation est indiquée par :
 - le temps de début d'affectation de la route ;
 - la durée d'affectation ;
 - la situation du trafic de la route durant la durée d'affectation.

²⁴ Pour plus de détails, voir chapitre 4, paragraphe 4.1.

²⁵ Voir chapitre 2, paragraphe 6.1.1

Types	Sévérité	Catégories du jour	Intervalles de temps	Impacts temporels et spatiaux
Accident de voiture	Très sévère	"Cat1"	[7h00, 10h00]	R_1 (t, 40 mn, saturée), R_1 (t+ 40 mn, 20 mn, dense), R_2 (t+5 mn, 30 mn, saturée), R_2 (t+35 mn, 15 mn, dense), R_3 (t+10 mn, 20 mn, saturée), R_3 (t+30 mn, 5 mn, dense), R_4 (t+20 mn, 10 mn, dense)
			[10h00, 14h00]	R_1 (t, 20 mn, saturée), R_1 (t+ 20 mn, 10 mn, dense), R_2 (t+10 mn, 10 mn, dense), R_3 (t+15 mn, 10 mn, dense)
			[14h00, 18h00]	R_1 (t, 40 mn, saturée), R_1 (t+ 40 mn, 20 mn, dense), R_2 (t+5 mn, 30 mn, saturée), R_2 (t+35 mn, 15 mn, dense), R_3 (t+10 mn, 20 mn, saturée), R_3 (t+30 mn, 5 mn, dense)
			[18h00, 07h00]	R_1 (t, 10 mn, dense), R_2 (t+5 mn, 5 mn, dense)
		"Cat2"	[07h00, 14h00]	R_1 (t, 15 mn, dense), R_2 (t+5 mn, 10 mn, dense)
			[14h00, 07h00]	Aucune affectation
		Sévère	"Cat1"	[07h00, 10h00]
	[10h00, 14h00]			R_1 (t, 10 mn, dense), R_2 (t+5 mn, 5 mn, dense), R_3 (t+10 mn, 5 mn, dense)
	[14h00, 18h00]			R_1 (t, 30 mn, dense), R_2 (t+5 mn, 20 mn, dense), R_3 (t+25 mn, 15 mn, dense)
	Voiture en panne

Tab 12 : Liste d'impact des événements imprévus d'une route R_1

Le tab 12 représente l'exemple d'une *liste d'impact des événements imprévus* d'une route R_1 . Chaque ligne de la table contient les routes affectées avec le temps de début d'affectation, la durée d'affectation en minute et la situation du trafic de chaque route durant l'affectation selon une catégorie du jour et un intervalle de temps. Supposons qu'un accident de voiture très sévère a eu lieu à $t=11h00$ sur la route R_1 dans un jour de catégorie "Cat1". L'impact spatial et temporel de cet accident sur le trafic routier selon le tab 12 est la suivante : la situation du trafic de la route R_1 sera saturée de 11h00 jusqu'à 11h20. Elle deviendra dense de 11h20 jusqu'à 11h30. La situation du trafic de la route R_2 sera dense entre 11h10 et 11h20 et celle de la route R_3 deviendra dense entre 11h15 et 11h25.

La *liste des situations de trafic prévues* et la *liste d'impact des événements imprévus* sont construites après une étude des connaissances historiques relatives au trafic de la route et à l'impact de chaque type d'événement sur la situation du trafic de la route et de ses voisines. La collection de ces types de connaissances ne fait pas partie de ce travail, mais elle est très essentielle pour assurer l'exactitude de ce modèle et réaliser son but. En raison de l'émergence de la technologie, les données historiques peuvent être collectées par différentes techniques²⁶.

4.1.4 Domaine géographique

Un domaine géographique est défini par une liste de routes contiguës possédant la même situation du trafic pendant un intervalle de temps commun I . Le domaine peut avoir pendant l'intervalle I des domaines voisins. Chaque domaine voisin possède au moins un nœud commun avec lui et a une situation du trafic différente de la sienne.

4.2 Demande de transport

La représentation d'une demande de transport dans le système TAD est donnée dans le chapitre 4, paragraphe 4.2.

4.3 Véhicule

La définition d'un véhicule est expliquée dans le chapitre 4, paragraphe 4.3.

4.4 Trajectoire

La définition d'une trajectoire est expliquée dans le chapitre 4, paragraphe 4.4.

²⁶ Voir chapitre 2, paragraphe 6.1.1.

4.5 Problème du TAD

La définition du problème du TAD est expliquée dans le chapitre 4, paragraphe 4.5.

4.6 Objectif du système du TAD

L'objectif du système du TAD est expliqué dans le chapitre 4, paragraphe 4.6.

5 Modélisation multi-agent massifs du problème du TAD

Nous modélisons le problème du TAD de ce scénario comme un système multi-agent massif auto-organisé décentralisé. Nous avons expliqué dans le chapitre 4²⁷ que le problème du TAD a une dimension importante car il est composé d'un très grand nombre d'éléments (des clients, des véhicules, des chemins, des routes et des domaines géographiques avec des situations du trafic différentes). Pour cette raison, nous avons décidé d'agentifier massivement les divers composants du problème pour réaliser un système auto-organisé donnant les meilleures informations à tout instant et résolvant le problème du TAD dans des villes caractérisées par des vitesses de voyage dynamiques stochastiques.

5.1 Types et rôles d'agents

Cinq types d'agents sont identifiés. Ils sont groupés en deux catégories : les agents statiques qui groupent l'agent Client, l'agent Véhicule et l'agent Pathfinder et les agents dynamiques qui groupent l'agent Route et l'agent Régulateur.

Dans ce paragraphe, nous décrivons le rôle de chacun de ces agents et ses caractéristiques.

5.1.1 Agents statiques

Dans la catégorie agents statiques, nous distinguons trois types d'agents :

Un agent Client :

Il joue le rôle d'interface entre les clients et le système du TAD. Il reçoit les demandes et il les envoie à l'ensemble des agents Véhicule du système. Dès que la réponse est prête, il la reçoit.

Un agent Véhicule :

Il est associé à chaque véhicule réel du système du TAD. Son rôle principal consiste à vérifier s'il est capable de transporter les passagers d'une demande du lieu de départ au lieu d'arrivée

²⁷ Voir chapitre 4, paragraphe 5.1

aux temps désirés par le client. Et cela en respectant sa table de planification et sans violer les contraintes temporelles : (1) des demandes courantes qui sont en cours pour le véhicule et (2) de celles futures que le véhicule sera engagé à servir ultérieurement. Si l'agent Véhicule est capable de servir la demande, il entrera dans une négociation avec d'autres agents Véhicule capables de la servir dans le but de choisir le bon agent qui va la réaliser (Malas et al. 2016b).

Un agent Pathfinder :

Il est relié à un agent Véhicule. Son rôle principal est de piloter l'agent Véhicule en lui proposant des trajectoires à suivre. Il se base sur l'algorithme de Dijkstra et sur les listes des chemins usuels des conducteurs²⁸ pour proposer le chemin des trajectoires.

5.1.2 Agents dynamiques

Dans la catégorie agents dynamiques, nous distinguons deux types d'agents : l'agent Route et l'agent Régulateur. Ils sont dynamiques parce que leur nombre et leurs caractéristiques varient en fonction du temps selon la situation du trafic. Dans le paragraphe suivant, nous allons définir le rôle des agents dynamiques.

Un agent Route :

Il est associé à un ensemble de routes successives contiguës ayant la même vitesse limite et la même situation prévue du trafic durant un intervalle de temps commun I . De cette manière, le nombre des agents Route dans le système du TAD est plus optimisé que celui qui est dans le chapitre 4 (le nombre des agents Route sera plus petit que celui des routes). Ce qui permet de réduire le temps de calcul ainsi que le nombre de messages échangés entre les agents Route et entre les autres types d'agents.

Un agent Route possède durant l'intervalle de temps I les paramètres suivants :

- une liste des routes auxquelles il est associé ;
- une situation prévue du trafic. Cette situation représente la situation prévue de chacune de ses routes pendant I selon la *liste des situations de trafic prévues* ;
- une valeur de pondération correspondant à la situation prévue du trafic. Le but de cette valeur est de diminuer la vitesse limite de ses routes selon cette situation. Par exemple, elle est égale à "1" lorsque la situation du trafic est fluide, "0.5" lorsque la situation du trafic est dense et "0.2" lorsqu'elle est saturée ;

²⁸ La notion de listes des chemins usuels est expliquée dans la section 7.4

- une vitesse de voyage prévue sur ses routes. Cette vitesse est calculée en multipliant la vitesse limite de ses routes par la valeur de pondération correspondant à la situation du trafic ;
- une couleur assignée à chaque situation du trafic. Cette couleur est verte lorsque la situation du trafic est fluide, orange lorsqu'elle est dense et rouge lorsqu'elle est saturée. Le choix des couleurs symbolisant la situation du trafic est réalisé en respectant les codes couleurs utilisés dans la ville de Rouen pour refléter l'état du trafic sur une carte²⁹ ;
- une liste des agents Route voisins. Un agent Route voisin est associé à des routes contiguës ayant une vitesse limite ou une situation du trafic différente de celle de l'agent Route et il possède au moins un nœud commun avec lui.

Le rôle de cet agent est de s'auto-organiser pour refléter la situation réelle du trafic des routes auxquelles il est associé pendant l'intervalle courant I. Pour cela, il :

- compare régulièrement sa situation prévue du trafic avec la situation réelle de chacune de ses routes. Cela afin de confirmer la situation prévue de ses routes ou détecter l'apparition d'un événement imprévu sur une des routes ;
- définit suite à la détection d'un événement imprévu, le type, le temps d'apparition, la localisation et la sévérité de cet événement ;
- déclenche une opération de décomposition pour se décomposer en plusieurs agents Route lorsqu'il détecte une hétérogénéité de situation du trafic entre ses routes ;
- déclenche une opération de fusion pour fusionner avec un voisin ayant la même situation du trafic et la même vitesse limite pour en devenir un seul agent.

Dans l'objectif de minimiser le nombre de messages échangés entre les agents Pathfinder et les agents Route de l'approche proposée dans le chapitre précédent et de prendre compte de l'évolution de la situation dynamique du trafic et de l'impact de tout événement stochastique sur la fluidité des routes, nous introduisons un autre agent dynamique appelé agent Régulateur.

Un agent Régulateur :

²⁹ <http://trafic.rouen.fr/trafic>

Il est associé à un domaine géographique pendant un intervalle de temps I' . Il possède durant l'intervalle I' les paramètres suivants :

- une liste contenant les routes du domaine auquel il est associé. Selon la définition du domaine, ces routes sont successives contiguës ayant la même situation du trafic au cours de l'intervalle de temps I' ;
- une situation prévue du trafic. Cette situation représente la situation du trafic de chacune de ses routes durant l'intervalle I' selon leur *liste des situations de trafic prévues*. Cette situation peut être fluide, dense ou saturée ;
- une valeur de pondération correspondant à la situation prévue du trafic. Cette valeur est multipliée par les vitesses limites de ses routes pour calculer la vitesse de voyage prévue sur ces routes pendant I' . A titre d'exemple, la valeur de pondération est égale à "1" lorsque le trafic est fluide, "0.5" lorsque le trafic est dense et "0.2" lorsqu'il est saturé ;
- une couleur assignée à chaque situation du trafic. Cette couleur est verte en cas de trafic fluide, orange en cas de trafic dense et rouge en cas de trafic saturé. Le choix des couleurs symbolisant la situation du trafic, est réalisé en respectant les codes couleurs utilisés dans la ville de Rouen pour refléter l'état du trafic sur une carte³⁰ ;
- une liste des agents Régulateur voisins associés à chaque domaine voisin ;
- une liste des agents Route associés aux routes appartenant à son domaine géographique ;
- une liste des agents Pathfinder dont la trajectoire du véhicule va passer par l'une des routes du domaine auquel il est associé.

Le rôle d'un agent Régulateur durant l'intervalle de temps courant I' est de :

- s'auto-organiser régulièrement pour refléter la situation réelle du trafic du domaine auquel il est associé pendant l'intervalle courant I' ;
- estimer la durée nécessaire pour traverser les routes de son domaine à partir d'un temps t . Et cela, en se référant à sa valeur de pondération et à la *liste des situations de trafic prévues* de ses routes ;
- prédire l'impact temporel et spatial d'un événement imprévu qui a eu lieu sur une route de son domaine en se référant à la *liste d'impact des événements imprévus* de cette route ;

³⁰ <http://trafic.rouen.fr/trafic>

- mettre à jour la durée estimée pour traverser les routes de son domaine affectées par un événement imprévu ;
- déclencher une opération de décomposition pour se décomposer en plusieurs agents Régulateur lorsqu'il détecte une hétérogénéité de situation du trafic entre les routes de son domaine ;
- déclencher une opération de fusion pour fusionner avec d'autres agents Régulateur voisins ayant la même situation du trafic pour en devenir un seul.

Le tab 13 ci-dessous récapitule les couleurs des agents, les valeurs de pondération et les vitesses correspondant aux différentes situations du trafic.

Situations du trafic	Couleurs de l'agent	Valeurs de pondération	Vitesses (v_l est la vitesse limite)
fluide	verte 	1	$v_l * 0.5 < V \leq v_l$
dense	orange 	0.5	$v_l * 0.2 < V \leq v_l * 0.5$
saturée	rouge 	0.2	$V \leq v_l * 0.2$

Tab 13 : Couleur et valeur de pondération des situations du trafic

Ces cinq agents interagissent ensemble afin de trouver une solution fiable pour chaque demande de transport. La figure 39 représente l'interaction entre les agents afin de traiter une demande de transport et de trouver le véhicule correspondant pour la servir. Cette figure montre un agent Client qui diffuse une demande de transport D à tous les agents Véhicule. Ces agents doivent vérifier leur disponibilité à servir la demande. Pour cela, chaque agent Véhicule communique avec son agent Pathfinder pour lui demander une trajectoire. L'agent Pathfinder cherche un chemin et contacte des agents Régulateur pour l'évaluer. Ensuite, l'ensemble des agents Véhicule disponibles pour servir la demande se négocient pour choisir le bon agent entre eux. Le bon agent c'est celui qui propose la solution optimale pour la demande D . Enfin, l'agent choisi met à jour sa table de planification pour servir la demande.

La figure 40 montre l'interaction de l'agent Pathfinder avec l'agent Régulateur pour évaluer un chemin et donner à l'agent Véhicule une évaluation précise de ses trajectoires. L'agent Régulateur communique avec ses voisins et avec des agents Route afin de refléter la situation réelle du trafic.

Le traitement d'une demande de transport et de l'évolution du trafic routier est détaillé respectivement dans les sections 7 et 8.

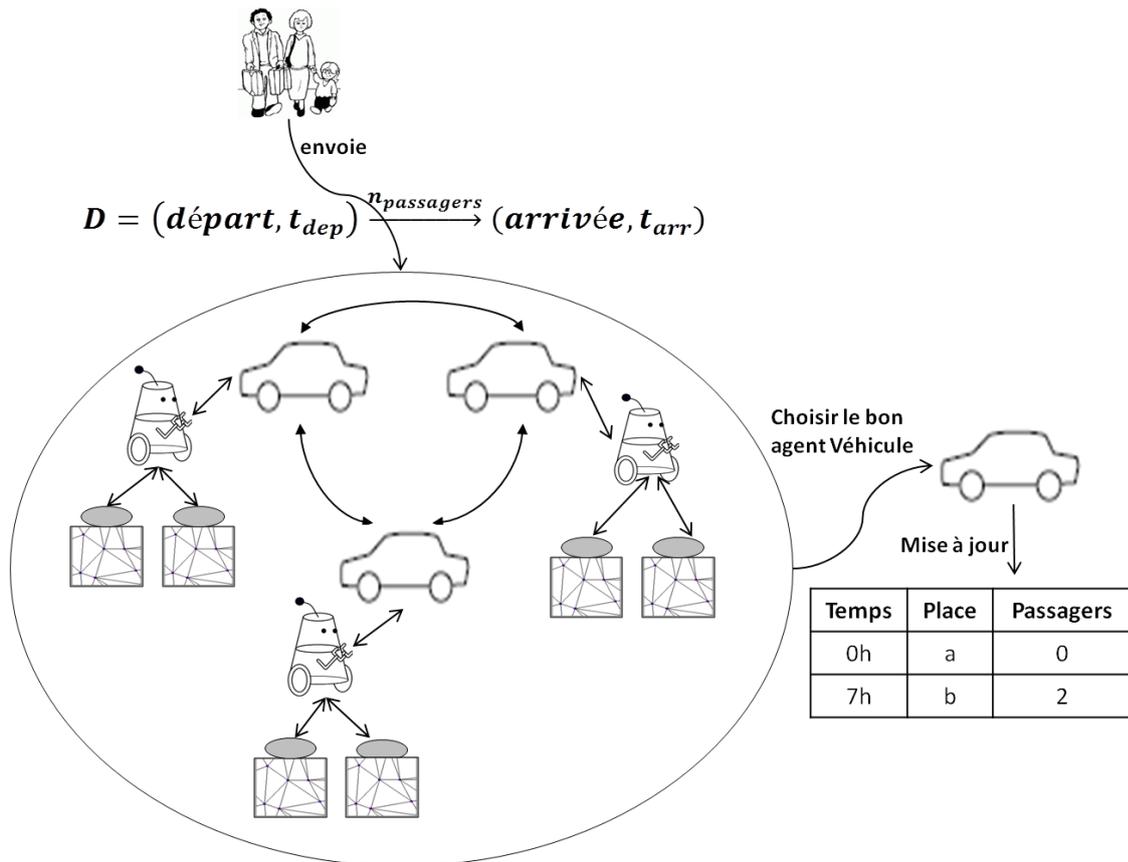


Figure 39 : Interactions des agents pour traiter une demande de transport

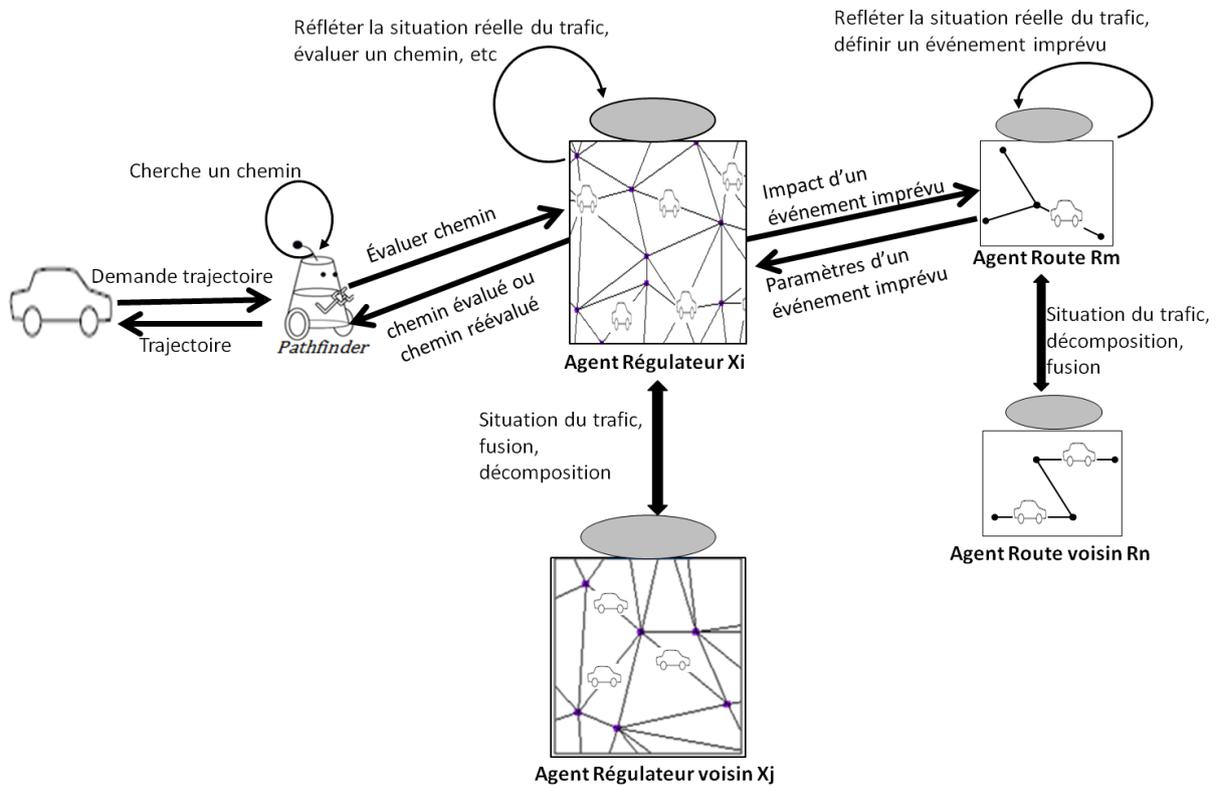


Figure 40 : Interactions des agents pour refléter la situation réelle du trafic

La modélisation du problème du TAD par un système multi-agent massif facilite le passage vers le monde réel. En d'autres termes, son application dans le monde réel sera plus simple qu'un autre système classique. A titre d'exemple, le chauffeur d'une voiture est équipé d'un smartphone. Ce téléphone mobile peut représenter les deux agents : agent Véhicule par l'utilisation du service de localisation et agent Pathfinder par l'utilisation de Google Map. Par ailleurs, la communication des agents Route et Régulateur avec des capteurs de surveillance du trafic routier devient facile avec le développement de l'Internet des objets ou IoT : Internet of Things (Al-Fuqaha et al. 2015; Whitmore, Agarwal, et Xu 2015; Torğul, Şağbanşua, et Balo 2016). Les routes peuvent être munies de capteurs eux-mêmes connectés à l'Internet et qui envoient via l'Internet des informations sur le trafic aux agents Route et Régulateur pour mieux gérer le trafic.

Dès lors, il serait intéressant d'expliquer les différentes interactions qui se déroulent entre ces agents afin de traiter une demande de transport et l'évolution de la situation dynamique du trafic. Tout d'abord, nous présentons un exemple introductif.

6 Exemple introductif simple

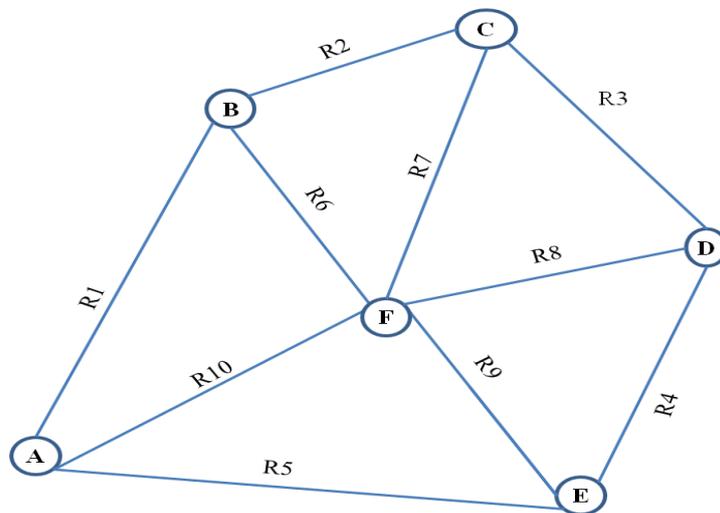


Figure 41 : Exemple d'une infrastructure simple

Supposons une infrastructure composée de 6 nœuds routiers et 10 routes connectées entre elles comme le montre la figure 41. Chaque route R_i a ses propres paramètres : $R_i(d_i, v_i, l_i)$ où d_i est la distance de la route R_i , v_i sa vitesse et l_i sa liste des situations de trafic prévues. Pour simplifier l'exemple, nous considérons que $v_1 = v_2 = v_3 = v_4 = v_5$, $v_6 = v_7 = v_8 =$

$v_9 = v_{10}$, $l_1 = l_2 = l_3 = l_6 = l_7 = l_{10}$ et $l_8 = l_9$. Les tab 14, tab 15, tab 16 et tab 17 représentent respectivement les listes l_1 , l_4 , l_5 , et l_8 .

Catégories	Intervalles de temps	Situations du trafic
"Cat1"	[07h00, 09h00]	dense
	[09h00, 14h00]	fluide
	[14h00, 16h00]	dense
	[16h00, 07h00]	fluide
"Cat2"	Toute la journée	fluide

Tab 14 : Représentation de la liste des situations de trafic prévues L1

Catégories	Intervalles de temps	Situations du trafic
Toute l'année	[07h00, 12h00]	fluide
	[12h00, 16h00]	dense
	[16h00, 07h00]	fluide

Tab 15 : Représentation de la liste des situations de trafic prévues L4

Catégories	Intervalles de temps	Situations du trafic
Toute l'année	Tout le temps	fluide

Tab 16 : Représentation de la liste des situations de trafic prévues L5

Catégories	Intervalles de temps	Situations du trafic
"Cat1"	[07h00, 09h00]	saturée
	[09h00, 14h00]	dense
	[14h00, 16h00]	saturée
	[16h00, 20h00]	dense
	[20h00, 07h00]	fluide

Tab 17 : Représentation de la liste des situations de trafic prévues L8

Nous notons que les situations du trafic de cette infrastructure varient selon la catégorie du jour et l'intervalle du temps. Nous présentons, dans ce qui suit, les agents Route et Régulateur associés aux routes de l'infrastructure et leur situation du trafic durant un jour de catégorie "Cat1". Supposons que le système se déclenche à 08h00 :

- A 08h00, les routes de l'infrastructure sont associées à 4 agents Route et 3 agents Régulateur comme le montrent le tab 18 et le tab 19. La figure 42 et la figure 43 montrent respectivement l'infrastructure couverte par les 4 agents Route et par les 3 agents Régulateur.

Agents Route	Routes associées	Intervalles de temps	Situations du trafic
Ar_1	R_1, R_2, R_3	[08h00, 09h00]	dense
Ar_2	R_6, R_7, R_{10}	[08h00, 09h00]	dense
Ar_3	R_4, R_5	[08h00, 12h00]	fluide
Ar_4	R_8 et R_9	[08h00, 09h00]	saturée

Tab 18 : Agents Route associés aux routes de l'infrastructure à partir de 08h00

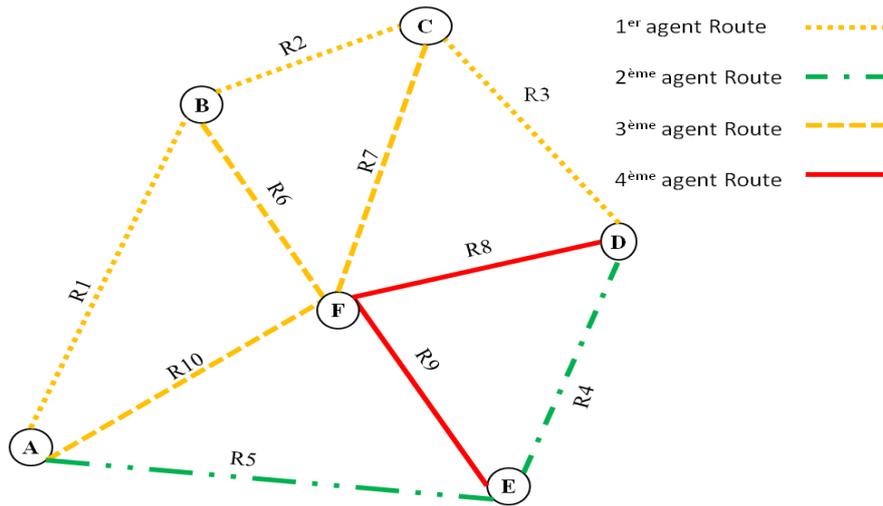


Figure 42 : Agents Route couvrant l'infrastructure à 08h00

Agents Régulateur	Agents Route	Routes associées	Intervalles de temps	Situations du trafic
$Areg_1$	Ar_1 et Ar_2	$R_1, R_2, R_3, R_6,$ R_7 et R_{10}	[08h00, 09h00]	dense
$Areg_2$	Ar_3	R_4 et R_5	[08h00, 12h00]	fluide
$Areg_3$	Ar_4	R_8 et R_9	[08h00, 09h00]	saturée

Tab 19 : Agents Régulateur associés aux routes de l'infrastructure à partir de 08h00

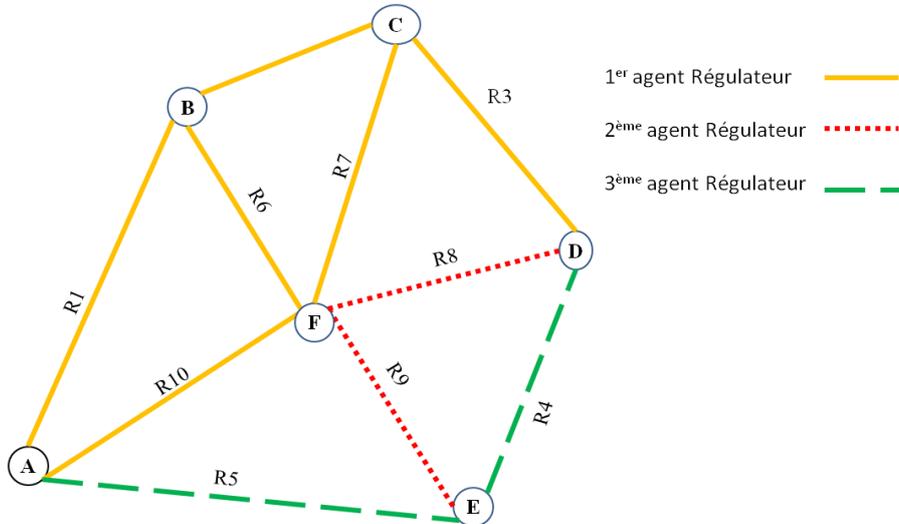


Figure 43 : Agents Régulateur couvrant l'infrastructure à 08h00

- A 09h00, les deux agents Route Ar_1 et Ar_3 fusionnent entre eux pour en devenir un seul parce que leurs routes présentent la même situation du trafic et la même vitesse limite. 3 agents Routes et 2 agents Régulateur s'associent aux routes de l'infrastructure comme le montrent le tab 20 et le tab 21. La figure 44 montre l'infrastructure couverte par les 3 agents Route.

Agents Route	Routes associées	Intervalles de temps	Situations du trafic
Ar_1	R_1, R_2, R_3, R_4 et R_5	[09h00, 12h00]	fluide
Ar_2	R_6, R_7 et R_{10}	[09h00, 14h00]	fluide
Ar_3	R_8 et R_9	[09h00, 14h00]	dense

Tab 20 : Agents Route associés aux routes de l'infrastructure à partir de 09h00

Agents Régulateur	Agents Route	Routes associées	Intervalles de temps	Situations du trafic
$Areg_1$	Ar_1 et Ar_2	$R_1, R_2, R_3, R_4, R_5, R_6, R_7$ et R_{10}	[09h00, 12h00]	fluide
$Areg_2$	Ar_3	R_8 et R_9	[09h00, 14h00]	dense

Tab 21 : Agents Régulateur associés aux routes de l'infrastructure à partir de 09h00

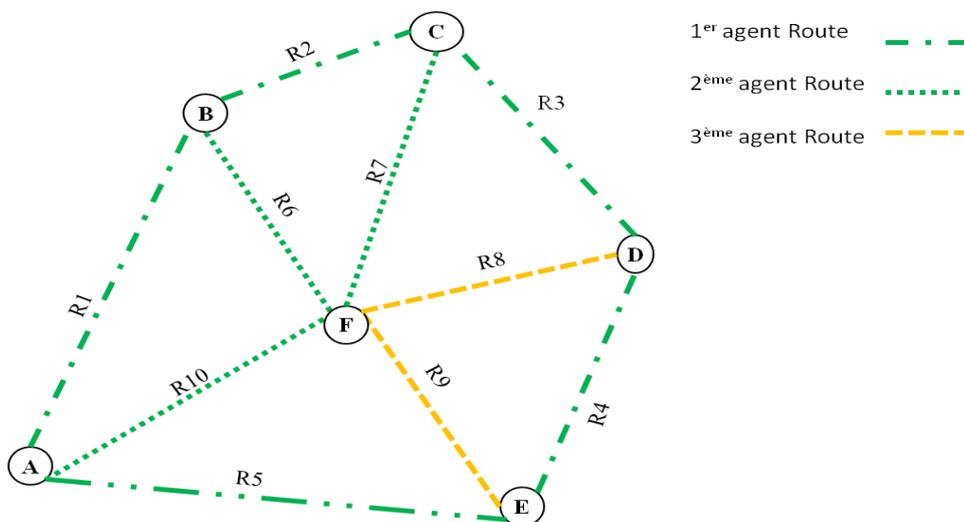


Figure 44 : Agents Route couvrant l'infrastructure à 09h00

- A 12h00, les deux agents Route Ar_2 et Ar_3 ne subissent aucune modification. En revanche, Ar_1 se décompose en deux autres : Ar_1 avec une situation du trafic fluide et Ar_4 avec une situation dense. Alors, 4 agents Routes et 2 agents Régulateur s'associent aux routes de l'infrastructure comme le montrent le tab 22 et le tab 23.

Agents Route	Routes associées	Intervalles de temps	Situations du trafic
Ar_1	R_1, R_2, R_3 et R_5	[12h00, 14h00]	fluide
Ar_2	R_6, R_7 et R_{10}	[12h00, 14h00]	fluide
Ar_3	R_8 et R_9	[12h00, 14h00]	dense
Ar_4	R_4	[12h00, 16h00]	dense

Tab 22 : Agents Route associés aux routes de l'infrastructure à partir de 12h00

Agents Régulateur	Agents Route	Routes associées	Intervalles de temps	Situations du trafic
$Areg_1$	Ar_1 et Ar_2	$R_1, R_2, R_3, R_5,$ R_6, R_7 et R_{10}	[12h00, 14h00]	fluide
$Areg_2$	Ar_3 et Ar_4	R_4, R_8 et R_9	[12h00, 14h00]	dense

Tab 23 : Agents Régulateur associés aux routes de l'infrastructure à partir de 12h00

- A 14h00, les routes de l'infrastructure sont associés à 4 agents Routes et 3 agents Régulateur comme le montrent le tab 24 et tab 25.

Agents Route	Routes associées	Intervalles de temps	Situations du trafic
Ar_1	R_1, R_2, R_3 et R_4	[14h00, 16h00]	dense
Ar_2	R_6, R_7 et R_{10}	[14h00, 16h00]	dense
Ar_3	R_8 et R_9	[14h00, 16h00]	saturée
Ar_4	R_5	[14h00, 14h00]	fluide

Tab 24 : Agents Route associés aux routes de l'infrastructure à partir de 14h00

Agents Régulateur	Agents Route	Routes associées	Intervalles de temps	Situations du trafic
$Areg_1$	Ar_1 et Ar_2	$R_1, R_2, R_3, R_4,$ R_6, R_7 et R_{10}	[14h00, 16h00]	dense
$Areg_2$	Ar_3	R_8 et R_9	[14h00, 16h00]	saturée
$Areg_3$	Ar_4	R_5	[14h00, 14h00]	fluide

Tab 25 : Agents Régulateur associés aux routes de l'infrastructure à partir de 14h00

- A 16h00, les routes de l'infrastructure sont associées à 3 agents Routes et 2 agents Régulateur comme le montrent le tab 26 et le tab 27.

Agents Route	Routes associées	Intervalles de temps	Situations du trafic
Ar_1	R_1, R_2, R_3, R_4 et R_5	[16h00, 07h00]	fluide
Ar_2	R_6, R_7 et R_{10}	[16h00, 07h00]	fluide
Ar_3	R_8 et R_9	[16h00, 20h00]	dense

Tab 26 : Agents Route associés aux routes de l'infrastructure à partir de 16h00

Agents Régulateur	Agents Route	Routes associées	Intervalles de temps	Situations du trafic
$Areg_1$	Ar_1 et Ar_2	$R_1, R_2, R_3, R_4,$ R_5, R_6, R_7 et R_{10}	[16h00, 07h00]	fluide
$Areg_2$	Ar_3	R_8 et R_9	[16h00, 20h00]	dense

Tab 27 : Agents Régulateur associés aux routes de l'infrastructure à partir de 16h00

- A 20h00, les routes de l'infrastructure sont associées à deux agents Routes et un seul agent Régulateur comme le montrent le tab 28 et le tab 29. La présence d'un seul agent Régulateur signifie que toutes les routes de l'infrastructure ont la même situation du trafic.

Agents Route	Routes associées	Intervalles de temps	Situations du trafic
Ar_1	R_1, R_2, R_3, R_4 et R_5	[20h00, 07h00]	fluide
Ar_2	R_6, R_7, R_8, R_9 et R_{10}	[20h00, 07h00]	fluide

Tab 28 : Agents Route associés aux routes de l'infrastructure à partir de 20h00

Agents Régulateur	Agents Route	Routes associées	Intervalles de temps	Situations du trafic
$Areg_1$	Ar_1 et Ar_2	$R_1, R_2, R_3, R_4,$ $R_5, R_6, R_7, R_8,$ R_9 et R_{10}	[20h00, 07h00]	fluide

Tab 29 : Agents Régulateur associés aux routes de l'infrastructure à partir de 20h00

Cet exemple montre que les agents Routes et Régulateur sont des agents dynamiques. Ils s'auto-organisent tout le temps selon la situation du trafic. Cette auto-organisation se révèle par la variation de leur situation du trafic, de leur nombre et des routes auxquelles ils sont associés.

7 Traitement d'une demande de transport

Dans cette section, nous détaillons l'interaction entre les différents agents (Client, Véhicule, Pathfinder et Régulateur) pour traiter une nouvelle demande de transport.

Lorsqu'une demande de transport arrive, l'agent Client crée une liste vide afin de mémoriser les agents Véhicule disponibles pour la servir. En parallèle, il diffuse la demande à l'ensemble des agents Véhicule du système et attend leur réponse.

Un agent Véhicule peut exécuter la demande s'il vérifie un ensemble de contraintes expliquées dans le chapitre 4³¹. Ces contraintes se résument dans les deux points suivants :

- le véhicule a le nombre de places vides, entre le temps de départ au plus tôt et le temps d'arrivée au plus tard de la demande, plus grand ou égale que le nombre de passagers à transporter ;
- l'insertion de la trajectoire, qui permet de transporter les passagers aux temps de départ et d'arrivée adéquats, dans la table de planification du véhicule respecte les contraintes temporelles de la demande courante et des demandes précédentes lesquelles l'agent Véhicule s'est engagé à servir.

La méthode de vérification de la seconde contrainte diffère de celle expliquée précédemment. Dans le chapitre précédent, un agent Route est associé à une route. Son rôle est d'évaluer le temps nécessaire à traverser sa route à partir d'un temps initial en se basant sur la *liste des situations de trafic prévues* de cette route. Alors, pour évaluer un chemin, il faut contacter chaque agent Route associé à chaque route du chemin. Dans ce chapitre, l'agent Régulateur est associé à un ensemble de routes contigües ayant la même situation du trafic durant un intervalle du temps. Au cas où un chemin devrait être évalué, il faut contacter les agents Régulateur associés aux routes du chemin. Cette méthode diminue le nombre des agents contactés, ainsi que le nombre de messages échangés avec eux.

La section suivante sera consacrée à détailler la méthode de vérification des contraintes temporelles pour une nouvelle demande.

7.1 Vérification des contraintes temporelles

Après la vérification de la première contrainte confirmant qu'il y a suffisamment de places libres pour transporter les passagers, l'agent Véhicule passe à la vérification de chacune des contraintes temporelles expliquées dans le chapitre 4³¹.

Pour chaque contrainte, l'agent Véhicule demande de son agent Pathfinder de lui trouver une trajectoire entre la paire des nœuds de cette contrainte à partir du temps de départ du nœud initial t . L'agent Pathfinder propose comme chemin primaire pour la trajectoire le chemin usuel que le conducteur du véhicule a l'habitude de prendre entre ces nœuds au temps t . Il

³¹ Voir chapitre 4, paragraphe 6.1

cherche ce chemin de la *liste de chemins usuels*³² du conducteur selon la catégorie du jour et l'intervalle de temps auquel le temps t appartient.

Pour estimer la durée nécessaire pour traverser le chemin usuel, l'agent Pathfinder cherche les agents Régulateur associés aux routes du chemin. Il les contacte à tour de rôle en commençant par celui associé à la première route. Puis, il lui demande d'estimer, à partir du temps t , la durée nécessaire pour traverser sa propre partie des routes. Le premier agent Régulateur évalue la durée³³ en prenant en considération la situation du trafic de chaque route durant la période de son parcours. Ensuite, il envoie la durée calculée à l'agent Pathfinder. Ce dernier continue à contacter les agents Régulateur restants. Il demande de chacun d'eux, à tour de rôle, d'évaluer la durée pour parcourir ses routes à partir d'un temps t' égal au temps t plus la durée calculée par chaque agent Régulateur antérieur. La figure 45 montre comment les agents coopèrent pour trouver une trajectoire entre deux nœuds à l'agent Véhicule.

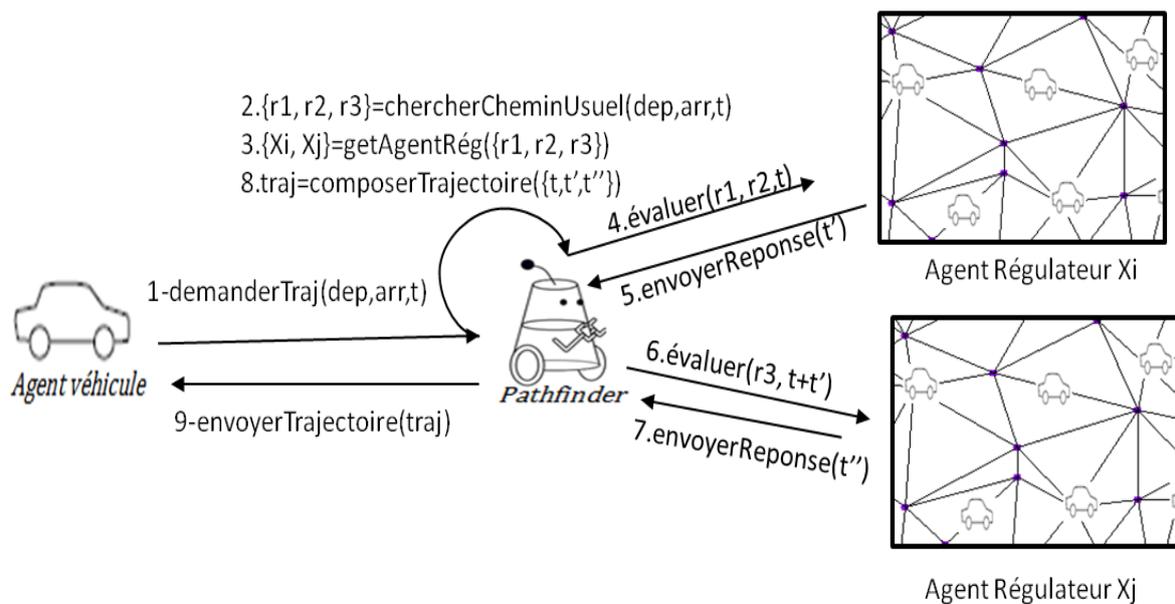


Figure 45 : Chercher une trajectoire entre deux nœuds

Une fois que tous les agents Régulateur ont estimé leurs routes et envoyé leurs réponses à l'agent Pathfinder, ce dernier compose la trajectoire en fonction des réponses reçues et l'envoie à l'agent Véhicule. Ce dernier teste si le temps pour traverser la trajectoire répond à la contrainte courante qu'il vérifie. Dans le cas positif, l'agent Véhicule continue à vérifier les autres contraintes. Dans le cas négatif, il réclame à son agent Pathfinder une autre trajectoire reliant la paire des nœuds. A ce moment, l'agent Pathfinder propose un chemin secondaire

³² Voir section 7.4

³³ La méthode adaptée pour évaluer la durée est expliquée dans la section 7.3

alternatif³⁴. Puis, il l'envoie de nouveau aux agents Régulateur associés aux routes pour l'évaluer. Et ainsi de suite, jusqu'à trouver une trajectoire convenable ou échouer. Si aucun chemin (usuel et alternatif) n'est convenable, l'agent Véhicule sera indisponible pour servir la demande.

Chaque agent Véhicule qui respecte toutes les contraintes calcule deux valeurs. La première valeur représente le minimum des nombres de places vides de son véhicule durant la période du service de la demande et la seconde représente la distance totale parcourue par le véhicule pour servir la demande. De même, il informe l'agent Client de sa disponibilité à servir la demande. Ceci l'ajoute à la liste créée au début.

7.2 Négociation entre les agents Véhicule

A la fin de la phase de vérification des contraintes, l'agent Client diffuse sa liste à chaque agent Véhicule qui y est présent. Les agents Véhicule recevant la liste commencent à négocier les uns avec les autres afin de choisir le bon agent entre eux qui va servir la demande. Les valeurs de deux paramètres de nombre de véhicules en circulation et de la distance totale traversée, indiquent qui va être le bon agent Véhicule. Ce dernier sera celui qui propose la solution optimale³⁵ de la demande qui minimise le nombre de véhicules en circulation. Dans le cas où deux agents Véhicule ont le même nombre de véhicules en circulation, celui proposant la solution qui minimise la distance totale traversée sera le bon agent. La stratégie de négociation adaptée pour choisir le bon agent Véhicule est celle appelée "*envoyer aux suivants*"³⁶. A la fin de cette stratégie, le bon agent Véhicule choisi envoie une réponse contenant les temps de départ et d'arrivée prévus à l'agent Client. En parallèle, il insère la trajectoire (les lieux principaux, les lieux transitoires³⁷ et le temps d'arrivée prévu à ces lieux) de la demande dans sa table de planification.

7.3 Méthode d'évaluation d'une route

Par définition, un agent Régulateur associé à un domaine géographique a une situation du trafic bien définie durant un intervalle de temps courant I' et une valeur de pondération correspondant à cette situation. Pour évaluer la durée pour parcourir une route appartenant à son domaine à partir d'un temps t , il vérifie si t appartient à son intervalle courant I' . Dans le

³⁴ Les caractéristiques du chemin alternatif sont expliquées dans la section 7.4

³⁵ Voir chapitre 4, paragraphe 4.6

³⁶ Voir chapitre 4, paragraphe 7.1

³⁷ Voir chapitre 4, paragraphe 4.4

cas positif, il multiplie la vitesse limite de cette route par sa propre valeur de pondération. Autrement, si t appartient à $(I'+1)$ qui suit I' , l'agent Régulateur se base sur *la liste des situations de trafic prévues* de la route pour déterminer sa situation du trafic au temps t , puis, il multiplie la vitesse limite de la route par la valeur de pondération correspondant à cette situation.

L'algorithme 11 décrit la procédure de calcul de la durée du parcours d'une route r à partir d'un temps initial t_0 par un agent Régulateur. Il est inspiré de l'algorithme 8 proposé par (Ichoua, Gendreau, et Potvin 2003). Nous supposons que d_r est la distance de la route r , v_{L_r} est sa vitesse limite, t_0 appartient à un intervalle de temps I , $valPond_{rI}$ la valeur de pondération correspondant à la situation du trafic de la route r durant I . Si t_0 appartient à l'intervalle courant de l'agent Régulateur I' , alors $valPond_{rI} = valPond_{ARégI'}$, où $valPond_{ARégI'}$ est la valeur de pondération de l'agent Régulateur durant l'intervalle de temps courant I' .

Algorithme 11 : Algorithme de la procédure de calcul du temps du parcours d'une route à partir d'un temps initial t_0

$t = t_0$; $d = d_r$; $I = \text{getIntervalle}(t)$; $I = [t_{min}, t_{max}]$; $t' = t + d / v_{rI}$ où $(v_{rI} = v_{L_r} * valPond_{rI})$

tant que $(t' > t_{max})$ **faire**

$d = d - v_{rI} (t_{max} - t)$

$t = t_{max}$

$I = I + 1$

$t' = t + d / v_{rI}$

fin

retourner $(t' - t_0)$

Supposons que la route à estimer est R_1 de l'exemple introductif et le temps initial de son parcours est $t = 08h50$. L'agent Régulateur $Areg_1$ est associé à la route R_1 pendant l'intervalle $I_1 = [08h00, 09h00]$ contenant t et sa situation est dense. Alors, sa valeur de pondération est 0.5 pour la période $[08h00, 9h00]$. Si la route R_1 n'est pas toute traversée pendant cet intervalle, alors $Areg_1$ doit se référer à la *liste des situations de trafic prévues* l_1 de R_1 , pendant l'intervalle $I' = [09h00, 14h00]$ qui suit $[08h00, 09h00]$. Durant cet intervalle, la situation devient fluide et la valeur de pondération est égale à 1.

7.4 Listes de chemins usuels

Chaque conducteur dans le système du TAD a une liste des chemins usuels construite et gérée par les agents Pathfinder. Cette liste contient les chemins usuels que le conducteur préfère (ou a l'habitude) de prendre pour aller d'un lieu à un autre selon la catégorie du jour et à un intervalle de temps. Le tab 30 représente l'exemple de la liste des chemins usuels d'un conducteur. Nous remarquons qu'à côté de chaque chemin usuel, il y a des chemins alternatifs. Ces chemins ont comme rôle de remplacer le chemin usuel lorsque ce dernier ne convient pas à la demande du client.

Le but des listes de chemins usuels est de respecter les désirs des conducteurs, de proposer les chemins les plus utilisés et convenables aux demandes de transport et d'éviter les véhicules de suivre les mêmes trajectoires. Ce qui aide à minimiser la possibilité d'avoir du trafic sur les routes. A titre d'exemple, si l'agent Pathfinder propose toujours le chemin le plus court en termes de temps, alors ce chemin pris par tous les véhicules ne restera jamais le chemin le plus court.

La construction de la liste des chemins usuels de chaque conducteur est expliquée dans le paragraphe suivant.

Catégories du jour	Nœuds de départ	Nœuds d'arrivée	Intervalles de temps	Chemins usuel	Chemins alternatifs
"Cat1"	"a"	"b"	[07h00, 09h00] [14h00, 16h00]	a→e→c→b (p1)	a→e→b (p4) a→d→f→c→b (p5)
			[09h00, 14h00] [16h00, 07h00]	a→d→f→c→b (p2)	a→e→c→b (p6)
"Cat2"	"a"	"b"	[07h00, 09h00] [14h00, 16h00]	a→e→b (p3)	a→e→c→b (p7)
			[09h00, 14h00] [16h00, 07h00]	a→e→c→b (p3)	a→e→b (p8)

Tab 30 : Exemple d'une liste des chemins usuels

7.4.1 Construction des listes de chemins usuels

La construction des listes de chemins usuels est inspirée du concept Road Time Table (Eglese, Maden, et Slater 2006). Chaque liste est relative à un conducteur. Elle est construite selon des catégories des jours et des intervalles de temps. Nous remarquons dans le tab 30 que pour chaque catégorie de jour, il y a un chemin usuel et des chemins alternatifs reliant une paire de nœuds selon un intervalle du temps.

La construction de la liste pour chaque conducteur est faite par l'agent Pathfinder. Lorsque l'agent Véhicule vérifie sa disponibilité de servir une demande de transport, il demande une trajectoire de son agent Pathfinder. Ce dernier applique l'algorithme de Dijkstra sur l'infrastructure pour chercher tous les chemins possibles de la trajectoire et il les propose au conducteur pour en choisir un. Si le chemin choisi par le conducteur ne convient pas à la demande du client, l'agent Pathfinder propose de nouveau les chemins restants au conducteur. Cette procédure est répétée jusqu'à que le conducteur choisit un chemin qui convient à la demande du client (en cas où aucun chemin ne convient à la demande du client, cela signifie que l'agent Véhicule est indisponible à servir la demande). Le chemin choisi et utilisé par l'agent Véhicule est considéré comme chemin usuel. L'agent Pathfinder le sauvegarde dans la liste des chemins usuels relatifs au conducteur en fonction de la catégorie du jour et d'un intervalle de temps. De plus, il sauvegarde les autres chemins non utilisés comme des chemins alternatifs. Si la même trajectoire est demandée une autre fois, l'agent Pathfinder propose tout de suite le chemin usuel.

Dans l'objectif de proposer les chemins désirés par les conducteurs et les plus convenables aux demandes des clients, l'agent Pathfinder associe à chaque chemin usuel et alternatif un poids. Ce poids est inspiré de la notion de phéromone (Dorigo, Maniezzo, et Colorni 1996; Valckenaers et al. 2007; Khichane, Albert, et Solnon 2009; Dahbi et al. 2014). Il est initialisé au départ à la valeur nulle. Au fur et à mesure de son utilisation, le chemin est ou bien renforcé ou bien affaibli pour simuler respectivement le renforcement et l'évaporation de la phéromone.

Le renforcement et l'affaiblissement des chemins sont expliqués dans les deux paragraphes suivants.

7.4.2 Renforcement d'un chemin

A chaque fois qu'un chemin usuel ou alternatif est utilisé par un agent Véhicule, il est renforcé. Le principe de renforcement consiste à ajouter au poids du chemin une constante k . Supposons qu'un chemin a un poids p_{ancien} . Après son utilisation, son poids sera renforcé par k et son nouveau poids p_{nouv} devient $p_{nouv} = p_{ancien} + k$. Le but de ce renforcement est d'élever le poids des chemins convenables les plus utilisés.

7.4.3 Affaiblissement d'un chemin

Chaque période T , les chemins usuels et alternatifs sauvegardés dans la liste des chemins usuels sont affaiblis. L'affaiblissement simule la forme d'évaporation de phéromone. Son principe consiste simplement à multiplier, chaque période T , le poids du chemin par $(1-\rho)$ où $0 \leq \rho \leq 1$. Donc, chaque période T , le nouveau poids p_{nouv} de chaque chemin devient $p_{nouv} = p_{ancien} * (1 - \rho)$. Le but de l'affaiblissement est d'atténuer le poids des chemins chaque période T .

7.4.4 Mise à jour des chemins usuels

L'agent Pathfinder considère le chemin avec le poids maximal comme chemin usuel. A chaque fois que le chemin usuel est utilisé, il sera renforcé. Au moment où un chemin usuel devient inapproprié aux demandes des clients à cause par exemple des perturbations dans la situation du trafic, l'agent Pathfinder propose le chemin alternatif choisi par le conducteur ou ayant le poids maximal. Le chemin alternatif est renforcé après chaque utilisation et le chemin usuel sera affaibli avec le temps. A l'instant où le poids d'un chemin alternatif dépasse celui du chemin usuel, l'agent Pathfinder met à jour le chemin usuel en le remplaçant par le chemin alternatif.

Les trois paramètres ρ , k et T permettent de définir la réactivité du système. Ils ont une très forte influence sur les processus de proposition des chemins par l'agent Pathfinder et sur l'actualisation des listes des chemins usuels. Plus T est petit et ρ est grand, plus la période de l'actualisation des chemins usuels est courte et plus les poids sont affaiblis. Ces paramètres dépendent de plusieurs facteurs comme la viabilité des routes, leur possibilité d'avoir des accidents, etc.

8 Auto-organisation des agents dynamiques pour traiter l'évolution du trafic

Nous distinguons maintenant deux types de situation dynamique du trafic routier :

- une situation dynamique déterministe qui représente la situation habituelle du trafic. Ce type est connu par avance. Il est déterminé dans notre système du TAD via *les listes des situations du trafic prévues* des routes ;
- une situation dynamique stochastique qui représente la situation réelle du trafic. Ce type ne peut pas être prédit en avant. Il est connu par des informations du trafic captées en temps réel. Il est causé par des événements imprévus (comme un accident).

L'impact d'un événement imprévu sur le trafic routier est prédit dans le système du TAD via les *listes d'impact des événements imprévus* des routes.

Les deux agents Route et Régulateur se reposent sur les connaissances historiques représentées par les deux listes citées précédemment et sur des informations en temps réel relatives au trafic pour s'auto-organiser spatio-temporellement et s'associer aux routes de l'infrastructure selon le trafic routier. Le mécanisme de l'auto-organisation des agents dynamiques se traduit par : (1) la variation de la situation du trafic de leurs routes pour refléter la situation réelle du trafic; (2) le déclenchement d'une opération de fusion qui se traduit par la fusion de deux ou plusieurs agents Route (resp. Régulateur) pour en devenir un seul agent ; (3) le déclenchement d'une opération de décomposition qui se traduit par la division d'un agent Route (resp. un agent Régulateur) en deux ou plusieurs agents.

Dans cette section, nous expliquons le traitement de l'évolution déterministe et stochastique de la situation dynamique du trafic par les agents Route et Régulateur. Enfin, nous détaillons les deux opérations de décomposition et de fusion de ces deux agents dynamiques.

8.1 Traitement de l'évolution déterministe de la situation dynamique du trafic

Dans ce paragraphe, nous expliquons comment les agents Route et Régulateur se comportent pour refléter les situations déterministes du trafic.

8.1.1 Détermination des agents dynamiques et de leurs intervalles de temps courants

A t_0 (temps de démarrage du système du TAD) et en se basant sur la *liste des situations de trafic prévues* sur chaque route, un agent Route Ar_i s'associe à chaque contiguïté de routes voisines ayant des vitesses limites et des situations prévues du trafic similaires pendant un intervalle de temps commun $I_i = [t_0, t_i]$. Cet agent Route aura la même situation prévue du trafic que ses routes durant son intervalle courant I_i . Egalement, un agent Régulateur $Areg_j$ s'associe à un domaine géographique pendant un intervalle courant $I_j = [t_0, t_j]$. Cet agent Régulateur aura la même situation prévue du trafic que les routes du domaine durant l'intervalle de temps I_j . Nous obtenons n agents Route et m agents Régulateurs couvrant l'infrastructure à partir du temps t_0 .

La borne supérieure t_i (resp. t_j) de l'intervalle de temps I_i de Ar_i (resp. I_j de $Areg_j$) représente le minimum des bornes supérieures des intervalles de temps contenant t_0 . Chacun

de ces intervalles appartient à un élément de la *liste des situations de trafic prévues* d'une route associé à Ar_i (resp. à $Areg_j$).

Dans l'exemple donné dans la section 6, le système se déclenche à $t_0=08h00$. L'agent Route Ar_1 s'associe à R_1 , R_2 et R_3 durant $I_1 = [08h00, 09h00]$, car ces trois routes ont la même situation et la même vitesse limite pendant I_1 . L'agent Régulateur $Areg_1$ s'associe à R_1 , R_2 , R_3 , R_6 , R_7 et R_{10} car ces routes ont la même situation du trafic pendant $I_1 = [08h00, 09h00]$. Nous remarquons qu'à partir de 09h00, la situation du trafic des routes R_1 , R_2 et R_3 devient fluide durant l'intervalle $[09h00, 14h00]$, celle de la route R_4 est fluide durant l'intervalle $[09h00, 12h00]$ et celle de la route R_5 est fluide durant toute la journée. Comme ces routes voisines ont la même vitesse limite et la même situation du trafic, un agent Route leur est associé durant l'intervalle $[09h00, 12h00]$. Cet intervalle résulte de l'intersection des trois intervalles $[09h00, 12h00]$, $[09h00, 14h00]$ et $[09h00, 09h00]$ de toute la journée.

8.1.2 Comportement des agents dynamiques durant les intervalles de temps courants

Durant l'intervalle I_i , l'agent Route Ar_i doit s'assurer pour chaque route que les deux situations du trafic prévues (résultantes de la *liste des situations prévues* de la route) et réelles (résultantes des informations du trafic captées en temps réel) sont similaires (homogènes). Pour cela, il compare de manière continue et régulière les deux situations de chacune de ses routes tout au long de l'intervalle I_i . Si elles sont homogènes, l'agent Route continue à refléter la situation prévue. Si elles ne le sont pas, ce cas est traité dans le paragraphe 8.2.

Durant l'intervalle I_j , l'agent Régulateur $Areg_j$ est en écoute de ses agents Route couvrants son domaine géographique. S'il ne reçoit d'eux aucun message urgent pendant I_j , il continuera à refléter la situation prévue dans son domaine durant I_j . Le cas inverse est traité dans le paragraphe 8.2

8.1.3 Comportement des agents dynamiques à la fin des intervalles de temps courants

A la fin des deux intervalles de temps courants I_i et I_j qui correspondent respectivement à un agent Route Ar_i et un agent Régulateur $Areg_j$, ces deux agents passent à déterminer respectivement les intervalles de temps suivants $I_i + 1$ et $I_j + 1$, leurs nouvelles routes et leur nouvelle situation du trafic.

8.1.3.1 Précision de l'intervalle de temps suivant $I_i + 1$ par l'agent Route

A la fin de l'intervalle de temps I_i , c'est-à-dire qu'à un temps $t = borneSup(I_i)$, l'agent Route doit s'associer de nouveau à des routes contigües ayant la même vitesse limite et la même situation du trafic pendant un intervalle de temps commun $I_i + 1$ qui suit I_i . De la *liste des situations de trafic* prévues de chacune de ses routes, il cherche l'élément contenant l'intervalle de temps qui suit I_i . En d'autres termes, il cherche l'élément contenant l'intervalle dont la borne inférieure égale à la borne supérieure de I_i . Il en résulte deux cas :

- les éléments trouvés ont des situations du trafic similaires pendant un intervalle de temps commun $I_i + 1$. Dans ce cas, l'agent Route garde le même ensemble de routes mais avec une nouvelle situation du trafic durant $I_i + 1$. L'intervalle suivant $I_i + 1$ est calculé de la même manière que I_i . Sa borne inférieure est égale à la borne supérieure de I_i et sa borne supérieure est le minimum des bornes supérieures des intervalles de temps appartenant aux éléments trouvés. Il met à jour sa situation du trafic, sa valeur de pondération et sa couleur selon cette nouvelle situation. Il envoie également un message à chaque agent Route voisin pour les informer de sa nouvelle situation. Si un de ses agents voisins a la même situation, il le contacte pour se fusionner avec lui. C'est le cas de l'agent Route Ar_2 dans l'exemple introductif donné. Pendant l'intervalle initial $[08h00, 09h00]$, il était associé à R_6 , R_7 et R_{10} et le trafic était dense. Pendant l'intervalle $[09h00, 12h00]$ qui suit $[08h00, 09h00]$, il reste associé aux mêmes routes mais le trafic devient fluide ;
- les éléments trouvés ont des situations du trafic différentes. En d'autres termes, au moins une des routes de l'agent Route a une situation du trafic prévue différente de celle des autres durant un intervalle de temps $I_i + 1$ qui suit I_i . Ce qui contredit la définition de l'agent Route. Pour cela, ce dernier se décompose³⁸ en créant un ou plusieurs nouveaux agents Route selon la situation du trafic de ses routes. Chaque nouvel agent s'associe aux routes voisines ayant les mêmes situations du trafic durant un intervalle de temps commun. Ce nouvel agent calcule l'intervalle commun de la même manière que I_i . Sa borne inférieure est égale à la borne supérieure de I_i . Sa borne supérieure est le minimum des bornes supérieures des intervalles de temps appartenant aux éléments trouvés relatifs aux routes de ce nouvel agent. Ensuite, il informe ses voisins de sa nouvelle situation du trafic afin d'effectuer l'opération de

³⁸ L'opération de décomposition d'un agent Route est expliquée dans le paragraphe 8.3.1

fusion si c'est possible. Dans l'exemple introductif, c'est le cas de l'agent Route Ar_1 . Pendant la période [09h00, 12h00], il était associé à R_1, R_2, R_3, R_4 et R_5 et son trafic était fluide. Pendant [12h00, 14h00], il se décompose en Ar_1 associé à R_1, R_2, R_3 et R_5 , avec le même trafic et en Ar_4 associé à R_4 avec une situation du trafic dense.

8.1.3.2 Précision de l'intervalle de temps suivant $I_j + 1$ par l'agent Régulateur

Après l'intervalle de temps I_j , l'agent Régulateur doit déterminer de nouveau l'ensemble des routes ayant la même situation du trafic pendant un intervalle de temps $I_j + 1$ qui suit I_j . De la liste des situations du trafic de chacune de ses routes, il cherche l'élément contenant l'intervalle de temps qui suit I_j . Deux cas peuvent se présenter :

- les éléments trouvés ont des situations du trafic similaires durant l'intervalle de temps commun $I_j + 1$. Dans ce cas, l'agent Régulateur garde le même ensemble de routes mais avec une nouvelle situation du trafic durant $I_j + 1$. Ce dernier est calculé de la même manière que I_j . Sa borne inférieure est égale à la borne supérieure de I_j et sa borne supérieure est le minimum des bornes supérieures des intervalles de temps appartenant aux éléments trouvés. Il modifie sa situation du trafic, sa valeur de pondération et sa couleur selon cette nouvelle situation. Il envoie également un message à chaque agent Régulateur voisin pour l'informer de la nouvelle situation. Si un de ses agents voisins possède la même situation, il le contacte pour se fusionner. C'est le cas de l'agent Régulateur Are_{g2} dans l'exemple introductif. Entre la période [14h00, 16h00], il était associé à R_8 et R_9 et sa situation était saturée. Entre 16h00 et 20h00, il reste associé aux mêmes agents Route mais sa situation devient dense ;
- les éléments trouvés ont des situations du trafic différentes. C'est-à-dire qu'au moins une des routes de l'agent Régulateur a une situation du trafic différente des autres pendant l'intervalle de temps $I_j + 1$. Ainsi, cette route ne doit pas appartenir à cet agent Régulateur. Pour cela, ce dernier se décompose³⁹ en créant un ou plusieurs nouveaux agents Régulateur selon la situation du trafic de ses routes. Puis, il informe ses voisins de cette décomposition. Chaque nouvel agent créé s'associe aux routes voisines ayant les mêmes situations du trafic durant un intervalle de temps commun. Ce nouvel agent calcule son intervalle de temps courant de la même manière que I_j . Sa borne inférieure est égale la borne supérieure de I_j . Sa borne supérieure est le

³⁹ L'opération de décomposition d'un agent Régulateur est expliquée dans le paragraphe 8.3.3

minimum des bornes supérieures des intervalles de temps appartenant aux éléments trouvés relatifs aux routes de ce nouvel agent Régulateur. Enfin, il informe ses agents Régulateur voisins de sa nouvelle situation du trafic afin de fusionner si c'est possible.

8.2 Traitement de l'évolution stochastique de la situation dynamique de trafic

A un temps réel t appartenant à l'intervalle courant I_i d'un agent Route Ar_i , il peut arriver que la situation prévue de trafic pour une de ses routes diffère de sa situation réelle captée (deux situations hétérogènes). Par exemple, dans l'exemple donné dans le paragraphe 6, une route R_1 a une situation du trafic prévue fluide entre 10h00 et 12h00. Supposons que par une des techniques de collecte d'informations de trafic, la situation apparaît dense à 10h30. Dans ce cas, l'agent Route détecte l'apparition d'un événement imprévu sur cette route. D'après le traitement des informations captées en temps réel concernant le trafic de cette route (comme les images captées par des caméras ou par des satellites, etc.), l'agent Route définit les paramètres de l'événement suivants : le type, la localisation, le temps d'apparition et la sévérité de cet événement. Si c'est un événement de sévérité simple, alors aucune modification ne sera faite. Si c'est un événement sévère ou très sévère, il enverra un message urgent à l'agent Régulateur pour l'informer de l'apparition d'un événement imprévu important sur la route. Ce message contient les paramètres de cet événement : le type, la localisation, le temps d'apparition et la sévérité de l'événement. D'après ces paramètres et en se référant à la *liste d'impact des événements imprévus* de la route, l'agent Régulateur détermine l'effet spatial et temporel de cet événement sur l'infrastructure. L'effet spatial correspond aux routes affectées par cet événement et leur nouvelle situation du trafic. L'effet temporel correspond à la durée de la nouvelle situation pour chacune des routes affectées. Ensuite, il informe les agents Route associés aux routes affectées par l'impact futur en espace et en temps de cet événement. Et cela, en envoyant pour chacun d'eux, une liste urgente concernant chaque route affectée appelée *liste urgente de nouvelles situations inhabituelles* semblable à la *liste des situations de trafic prévues* et qui la remplace. Chaque agent Route, qui reçoit la liste urgente, se base sur cette liste pour calculer les situations prévues de ses routes durant les intervalles de temps contenus dans la liste urgente. Les agents dynamiques (Route et Régulateur) commencent à déterminer leur nouvelle situation du trafic, à se décomposer ou à se fusionner selon les listes urgentes. L'agent Route et l'agent Régulateur, qui se créent à cause de la liste urgente, sont appelés respectivement *agent Route urgent* et *agent Régulateur urgent*. Le tab 31 représente l'exemple d'une liste urgente créé pour la route

R_1 suite à l'accident. L'agent Régulateur associé à R_1 construit cette liste urgente en se référant à la *liste des situations de trafic prévues* de R_1 présentée par le tab 14 et la *liste d'impact des événements imprévus* de R_1 présentée par le tab 12.

Intervalles de temps	Situations de trafic
[10h30, 10h50]	saturée
[10h50, 11h00]	dense
[11h, 14h00]	fluide

Tab 31 : Liste urgente pour la route R_1

Chaque *agent Régulateur urgent* cherche de la liste des agents Pathfinder ceux qui sont influencés par cet événement imprévu. Ce sont les agents dont la trajectoire de leur agent Véhicule va être perturbée parce qu'elle passe par des routes affectées durant la période d'affectation de l'événement. L'*agent Régulateur urgent* réévalue les routes perturbées de la trajectoire de ces agents et envoie à chacun d'eux la nouvelle évaluation.

À son tour, l'agent Pathfinder informe son agent Véhicule de la nouvelle évaluation des routes affectées. L'agent Véhicule vérifie si cette évaluation a un effet sur la fenêtre de temps d'au moins une de ses demandes. Si la nouvelle évaluation respecte les temps d'arrivée au plus tard de tous les clients, l'agent Véhicule continue à suivre sa trajectoire normalement sans aucune variation. Autrement, il décide de se dérouter pour éviter les routes affectées et demande une nouvelle trajectoire de son agent Pathfinder. Ce dernier propose comme chemin de trajectoire celui qui ne passe pas par des routes affectées. Cela aide à réduire autant que possible l'effet négatif de cet événement sur les fenêtres de temps des demandes de transport.

8.3 Opérations de décomposition et de fusion

Les deux agents Route et Régulateur sont des agents dynamiques. Ils se décomposent en plusieurs et fusionnent avec d'autres afin de couvrir le réseau routier selon la situation dynamique de trafic. Alors, leur nombre se modifie. Il augmente après une opération de décomposition et diminue après une opération de fusion.

8.3.1 Décomposition de l'agent Route

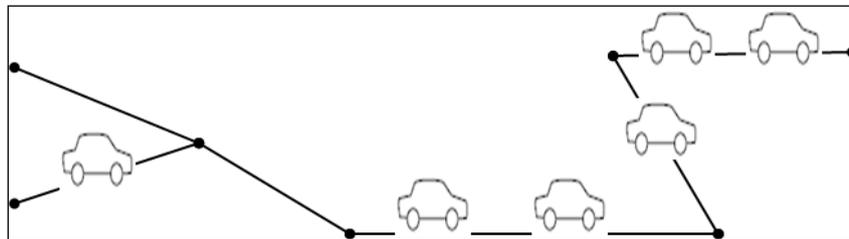
Durant son intervalle courant I_i , l'agent Route Ar_i se décompose dans les deux cas suivants :

- à un temps réel t appartenant à I , il détecte qu'au moins une de ses routes a une situation du trafic réelle différente de celle prévue. Cela à cause de l'apparition d'un événement imprévu sur la route ;

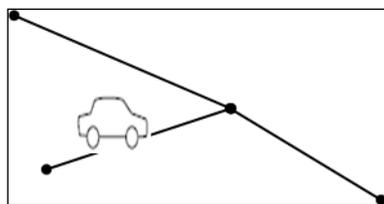
- à l'instant $t = \text{borneSup}(I_i)$ et en déterminant l'intervalle suivant $I_i + 1$, il détecte que ses routes n'ont pas toutes la même situation prévue de trafic.

Les deux cas impliquent que les routes associées à l'agent n'ont pas une situation du trafic similaire. Ce qui est contradictoire avec la définition d'un agent Route. C'est pour cette raison, il décide de se décomposer en deux ou plusieurs agents Route. Cela afin de reprendre l'homogénéité de la situation du trafic entre ses routes et associer un nouvel agent Route aux routes contigües ayant la même situation du trafic.

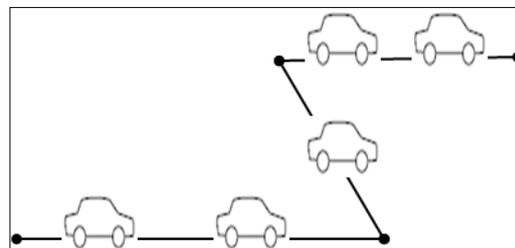
L'idée de l'opération de décomposition est la suivante : l'agent Route principal (qui va se décomposer) se réassocie à des routes contigües ayant la même situation du trafic durant un intervalle de temps commun. Puis, il crée un nouvel agent Route et l'associe aux routes contigües restantes ayant la même situation du trafic. Les nouveaux agents déterminent ensuite leur intervalle de temps courant comme expliqué dans la section 8.1.3.1, leurs nouvelles routes, leur situation du trafic, leurs agents Route voisins et leur agent Régulateur durant cet intervalle. La figure 46 montre l'exemple de décomposition d'un agent Route. Le premier graphe montre un agent Route ayant deux situations de trafic différentes et le second montre la décomposition de cet agent.



Un agent Route a une situation du trafic fluide sur un ensemble de ses routes et une situation dense sur un autre en même temps.



Agent Route 1- Traffic fluide



Agent Route 2 – Traffic dense

Figure 46 : Décomposition d'un agent Route

L'algorithme 12 décrit l'opération de décomposition des agents Route.

Algorithme 12 : Algorithme de la décomposition d'un agent Route

Ar_i : agent Route qui va se décomposer

l_r : liste des routes de Ar_i durant $I = [t_i, t_f]$

l_{nouv} : une liste de routes vide

r : une route

$r \in l_r$

situation = r .situation à partir de t_f

res := liste des nouvelles listes de routes

$res.add(l_r)$

pour (chaque $r' \in l_r \setminus r$) **faire**

si ($r'.situation == situation$) **alors**

continue ;

fin

si ($r'.situation != situation$) **alors**

$l_r.remove(r')$;

$l_{nouv} = res.get(r'.situation)$ // get la liste de res dont les routes ont une situation égale à $r'.situation$

si ($l_{nouv} != null$) **alors**

pour (chaque route $\in l_{nouv}$) **faire**

 ajouté = faux ;

si ($r' \in route.voisins()$) **alors**

$l_{nouv}.add(r')$

 ajouté = true ;

 break ;

fin

fin

fin

si ($l_{nouv} == null \ || \ ajouté == faux$) **alors**

 créer (l_{nouv})

$l_{nouv}.situation = r'.situation$

$l_{nouv}.add(r')$

$res.add(l_{nouv})$

fin

fin

fin

Associer à chaque liste de res un agent Route

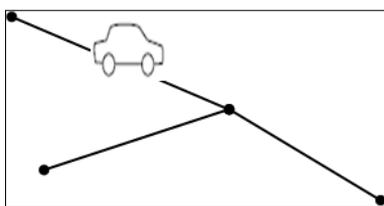
retourner res

8.3.2 Fusion des agents Route

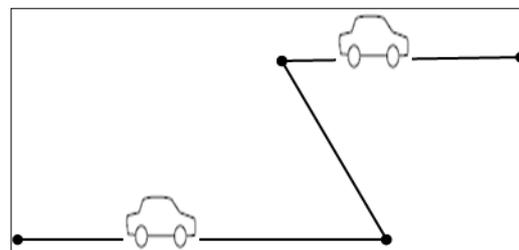
Un agent Route a des agents Route voisins de situation du trafic différente de la sienne. A chaque fois qu'un agent Route Ar_i est créé ou met à jour sa situation du trafic pour un intervalle de temps I_i , il envoie un message à ses voisins pour les informer de sa situation. Lorsqu'un de ses voisins $Ar_{i'}$ a la même vitesse limite et la même situation du trafic durant un intervalle de temps $I_{i'}$, il l'informe de sa situation et de l'intervalle de temps $I_{i'}$ de cette situation. A ce moment, l'agent Route Ar_i déclenche l'opération de fusion. Il en résulte que ces deux agents Route deviennent un seul agent Ar_i ayant la situation du trafic commune durant l'intervalle de temps $I_i = I_i \cap I_{i'} = [t_{min}, t_{max}]$ où $t_{min} = \max\{\text{borneInf}(I_i), \text{borneInf}(I_{i'})\}$ et $t_{max} = \min\{\text{borneSup}(I_i), \text{borneSup}(I_{i'})\}$.

Dans l'exemple introductif, c'est le cas des deux agents Route Ar_1 et Ar_3 . Durant l'intervalle [08h00, 09h00], Ar_1 est associé aux routes R_1, R_2 et R_3 et sa situation est dense. Ar_3 est associé à la route R_4 et R_5 pendant [08h00, 12h00] et sa situation est fluide. A 09h00, la situation de l'agent Ar_1 devient fluide. Ar_1 envoie un message à son voisin Ar_3 pour l'informer du changement de sa situation. Comme Ar_3 a la même situation que Ar_1 à partir de 09h00, ces deux agents se fusionnent pour former un seul agent Route avec une situation fluide durant l'intervalle commun [09h00, 12h00].

La figure 47 montre l'exemple de fusion de deux agents Route. Les deux premiers graphes montrent deux agents Route voisins ayant deux situations de trafic similaire. La troisième montre que ces deux agents se fusionnent dans un seul agent.



Agent Route1- trafic fluide



Agent Route 2 – Trafic fluide

Deux agents Routes voisins ont la même situation du trafic pendant le même temps

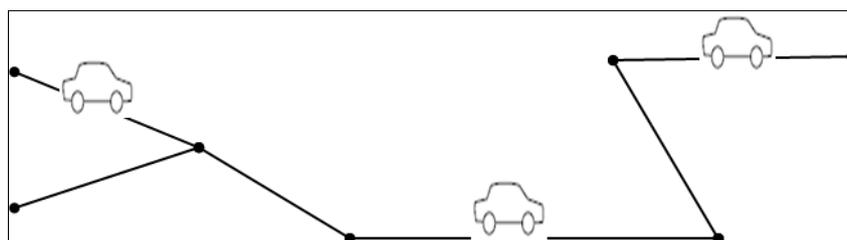


Figure 47 : Fusion de deux agents Route

L'algorithme 13 décrit l'opération de fusion entre deux agents Route.

Algorithme 13 : Algorithme de l'opération de fusion

Ar_i : agent Route

voisins : liste des agents Route voisins de Ar_i

pour (chaque $v \in$ voisins) **faire**

Ar_i .send(Ar_i .situation, v)

si (Ar_i .situation == v .situation && Ar_i .vitesseLimite == v .vitesseLimite) **alors**

Ar_i .routes = Ar_i .routes \cup v .routes

Ar_i .agentsVoisins = Ar_i .agentsVoisins \cup v .agentsVoisins \setminus ($Ar_i \cup v$)

$v = \text{null}$

fin

fin

retourne Ar_i

8.3.3 Décomposition d'un agent Régulateur

Un agent Régulateur $Areg_j$ est associé à un domaine géographique avec une situation du trafic homogène sur ses routes pendant un intervalle de temps I_j . Cet agent se décompose dans les deux cas suivants :

- pendant l'intervalle de temps I_j , il reçoit un message urgent d'un de ses agents Route l'informant d'apparition d'un événement imprévu important sur une de ses routes. Dans ce cas, il en résulte que la situation prévue de la route n'est pas la même que la situation réelle ;
- à l'instant $t = \text{borneSup}(I_j)$, il détecte que les routes de son domaine géographique n'ont pas la même situation du trafic.

Les deux cas impliquent que les routes du domaine n'ont pas toutes la même situation du trafic. En d'autres termes, l'homogénéité de la situation du trafic entre les routes de l'agent n'est plus gardée. Ce qui est contradictoire avec la définition d'un agent Régulateur. Dans ce cas, il décide de se décomposer en deux ou plusieurs agents Régulateur. Cela afin de reprendre l'homogénéité de la situation du trafic sur son domaine géographique et associer un nouvel agent Régulateur aux routes contigües ayant la même situation du trafic.

L'idée de l'opération de décomposition est la suivante : l'agent Régulateur principal (qui va se décomposer) se réassocie aux routes contigües de son domaine ayant la même situation du

trafic durant un intervalle de temps commun. Après, il crée un nouvel agent Régulateur et l'associe aux routes contiguës restantes du domaine ayant la même situation du trafic durant un intervalle de temps commun. Les nouveaux agents déterminent ensuite leur intervalle de temps courant comme expliqué dans la section 8.1.3.2, leur nouveau domaine, leur situation du trafic et leurs agents Régulateur voisins durant cet intervalle.

8.3.4 Fusion des agents Régulateur

Un agent Régulateur a des agents Régulateur voisins avec une situation du trafic différente de la sienne. A chaque fois qu'un agent Régulateur $Areg_j$ est créé ou met à jour la situation du trafic de son domaine pour un intervalle de temps I_j , il envoie un message à ses voisins pour les informer de sa situation. Lorsqu'un de ses voisins $Areg_{j'}$ a la même situation du trafic durant un intervalle de temps $I_{j'}$, il l'informe qu'il a la même situation en lui précisant l'intervalle de temps I_j de cette situation. A ce moment, l'agent Régulateur $Areg_j$ déclenche l'opération de fusion. Il en résulte que ces deux agents Régulateur deviennent un seul agent $Areg_j$ ayant la situation du trafic commune durant l'intervalle de temps $I_j = I_j \cap I_{j'} = [t_{min}, t_{max}]$ où $t_{min} = \max\{\text{borneInf}(I_j), \text{borneInf}(I_{j'})\}$ et $t_{max} = \min\{\text{borneSup}(I_j), \text{borneSup}(I_{j'})\}$.

9 Gestion des connaissances

Dans notre système du TAD, chaque route de l'infrastructure a *une liste des situations de trafic prévues* et *une liste d'impact des événements imprévus*. La première liste résulte de l'analyse des connaissances historiques de trafic. La deuxième liste résulte de l'analyse des connaissances historiques de l'impact de chaque type d'événement imprévu sur le trafic du réseau routier. Ces deux listes doivent être gérées par les agents Route et Régulateur afin que le système soit le plus proche de la réalité et donne de bons résultats avec moins de calcul. La première liste peut être actualisée par l'agent Route par exemple lorsque ce dernier compare la situation prévue d'une de ses routes avec la situation réelle et détecte que les deux situations sont différentes. Si cette différence se répète fréquemment de la même manière pendant une période T_1 pour le même intervalle de temps et la même catégorie du jour, il faut que l'agent Route mette à jour la *liste des situations de trafic prévues* de la route en tenant compte de la nouvelle situation.

Par ailleurs, la deuxième liste peut être mise à jour par l'agent Régulateur. A titre d'exemple, après l'apparition d'un événement imprévu sur une de ses routes, l'agent Régulateur envoie

un message contenant une liste des situations urgentes pour chaque route affectée par cet événement. Chaque route considère la liste urgente au détriment de *la liste des situations de trafic prévues*. Il peut arriver qu'un agent Route détecte une situation réelle différente de la situation urgente de trafic envoyé par l'agent Régulateur. En d'autres termes, l'impact réel de l'événement imprévu diffère de ce qu'il est attendu selon *la liste d'impact des événements imprévus*. Dans ce cas, l'agent Route informe son agent Régulateur qu'il a été mal averti et lui communique la situation réelle de la route suite à l'apparition de cet événement. Si cette situation se répète fréquemment pour une période T_2 , il fallait que l'agent Régulateur mette la deuxième liste à jour.

Il peut aussi arriver qu'un événement imprévu inconnu dans le système du TAD apparaisse sur le réseau routier. Dans ce cas, les agents Route et Régulateur doivent apprendre à chercher des similitudes à cet événement de leurs connaissances historiques afin de synthétiser son impact sur le trafic du réseau routier.

La procédure de gestion des connaissances et de l'apprentissage des agents n'est pas étudiée dans cette thèse.

10 Expérimentations

Cette section sera consacrée à la présentation des expérimentations réalisées pour valider l'approche multi-agent massif auto-organisé proposée dans ce chapitre. Ces expérimentations mesurent :

- le nombre de demandes servies parmi un ensemble en considérant la situation du trafic ;
- le nombre de véhicules utilisés pour servir ces demandes ;
- le nombre de clients insatisfaits si les événements imprévus tel un accident ne sont pas pris en considération ;
- le nombre de clients qui deviennent satisfaits après la prise en considération par le système des événements imprévus ;
- le nombre des agents Route et Régulateur associés au réseau routier dynamique stochastique du Tripoli à divers temps ;
- le nombre d'agents contactés afin d'évaluer des chemins dans l'approche proposée dans le chapitre 4 et celui dans l'approche proposée dans ce chapitre.

Nous passons dans la section suivante à expliquer les paramètres des expérimentations.

10.1 Paramètres des expérimentations

La plateforme utilisée pour réaliser nos expérimentations est toujours la même : Mason. L'approche proposée se base sur les systèmes multi-agents massifs. Elle comporte plus d'un millier d'agents. Pour cette raison, nous avons choisi d'utiliser Mason dès le début comme plateforme multi-agent car elle peut supporter un très grand nombre d'agents (Luke 2011).

Le réseau routier utilisé est celui de la ville de Tripoli représentée dans le chapitre 3⁴⁰. Les informations concernant le type, le nom, la vitesse limite et la longueur de chaque route sont récupérées de la municipalité tripolitaine. La *liste des situations de trafic prévues* et la *liste d'impact des événements imprévus* de chaque route sont générées en se basant sur des statistiques relatives au trafic de la ville données par la municipalité et en utilisant la méthode de Monte Carlo (Eckhardt 1987). Nous utilisons les 500 demandes de transport générées dans le chapitre 3 et sauvegardées dans un fichier XML. De même, nous disposons 200 véhicules pour servir les demandes.

10.2 Expérimentations, résultats et analyses

Afin de tester l'approche proposée, nous avons réalisé trois expérimentations expliquées dans les sous-sections suivantes.

10.2.1 Mesure du nombre de clients insatisfaits

Afin de montrer la nécessité de considérer la situation réelle de trafic routier surtout en cas d'apparition d'un événement imprévu (par exemple, un accident) sur une route, nous générons quatre accidents routiers de type sévère et nous effectuons une expérimentation qui mesure pour chaque accident :

- le nombre de clients (demandes) servis durant la période d'impact de l'accident ;
- le nombre de clients affectés par l'accident. Ce sont des clients qui seront insatisfaits si le système ne considère pas l'accident ;
- le nombre de clients satisfaits parmi ceux qui sont affectés par l'accident après que le système considère l'accident.

Nous supposons que pendant une journée, quatre accidents routiers ont eu lieu à Tripoli. Les routes, où les accidents ont eu lieu, sont choisies en se basant sur des statistiques données par

⁴⁰ Voir chapitre 3, paragraphe 7.2.2

la municipalit  concernant les routes les plus sujettes   avoir des accidents et les routes voisines qui peuvent  tre affect es par ces accidents. Le premier accident est g n r    07h30 sur une route R_1 choisie par l'application de la m thode de Monte Carlo sur les routes les plus sujettes   avoir des accidents entre 07h00 et 9h00. Le deuxi me accident est g n r  sur la m me route   12h30. La route du troisi me accident g n r    20h30 est choisie parmi les routes les plus sujettes   avoir des accidents entre 19h00 et 21h00. Dans cette exp rimentation, nous consid rons l'impact de l'accident sur la route et ses voisines. Nous distinguons cinq cas : le premier cas comprend 100 demandes, le deuxi me est compos  de 200, le troisi me est constitu  de 300, la quatri me en contient 400 et le dernier comprend 500. Chaque cas contient les demandes du cas pr c dent. Le tab 32 montre les r sultats obtenus de chaque accident pour les cinq cas.

Nombre de clients	Nombre de clients servis	Nombre de v�hicules utilis�s	Nombre de clients servis durant l'accident#1	Nombre de clients affect�s par l'accident#1	Nombre de clients satisfaits	Nombre de clients servis durant l'accident#2	Nombre de clients affect�s par l'accident#2	Nombre de clients satisfaits	Nombre de clients servis durant l'accident#3	Nombre de clients affect�s par l'accident#3	Nb de clients satisfaits
50	49	14	4	0	0	2	0	0	9	1	1
100	98	28	17	1	1	2	0	0	11	1	1
150	148	41	23	2	2	3	0	0	19	2	1
200	196	56	31	6	5	3	0	0	23	2	1
250	244	70	43	6	5	3	0	0	26	2	1
300	293	82	52	9	6	3	0	0	28	2	1
350	342	91	58	10	7	5	1	1	34	3	2
400	390	103	74	13	9	5	1	1	35	3	2
450	437	107	82	13	9	6	2	2	47	3	2
500	487	118	96	15	11	6	2	2	51	4	2

Tab 32 : Impact des trois accidents sur les demandes de transport

La figure 48, la figure 49 et la figure 50 montrent par des histogrammes respectivement les r sultats pour l'accident #1, l'accident #2 et l'accident #3.

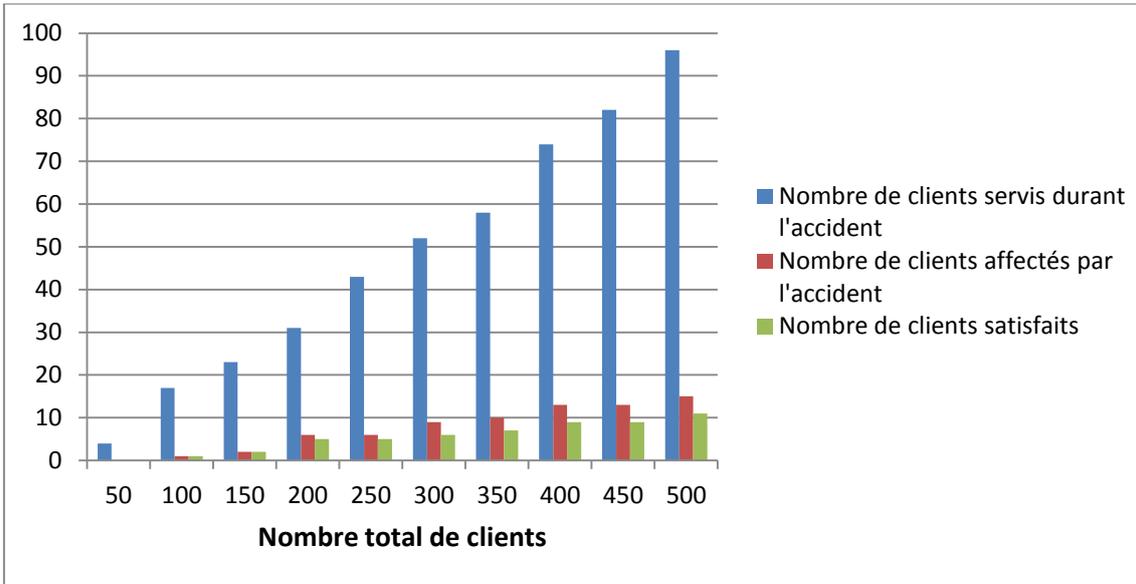


Figure 48 : Impact de l'accident#1 sur les clients

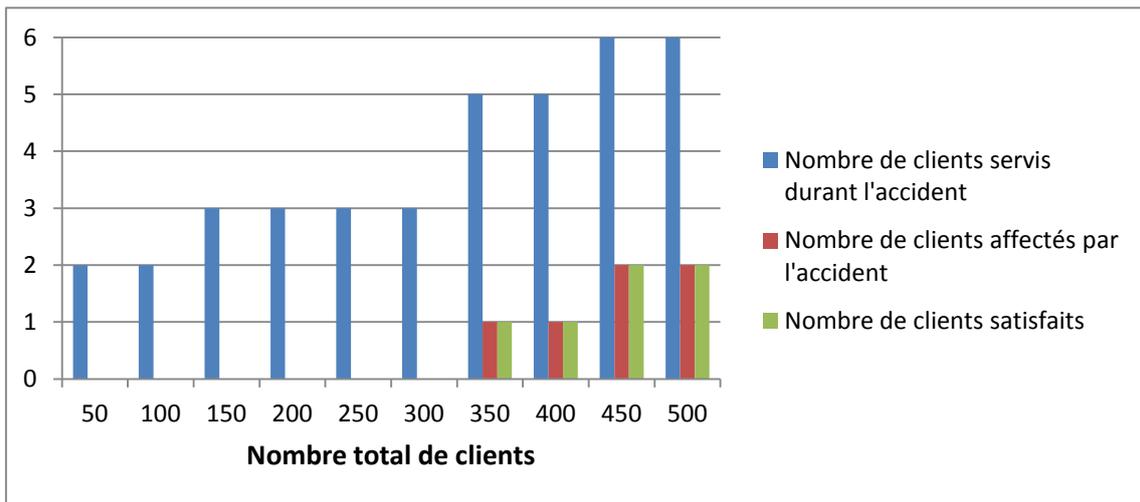


Figure 49 : Impact de l'accident#2 sur les clients

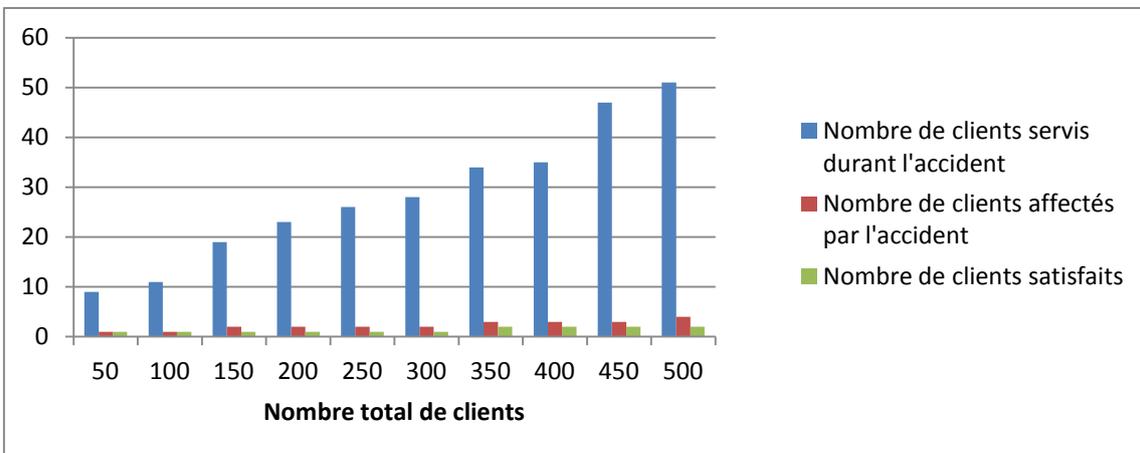


Figure 50 : Impact de l'accident#3 sur les clients

Le quatrième accident est généré à 08h00 (heure de pointe) sur une route R_x caractérisée par une viabilité forte pendant une heure de pointe. Cette route est choisie en appliquant la distribution uniforme sur un ensemble contenant les routes les plus viables à Tripoli. Le tab 33 montre les résultats obtenus pour cet accident. Ces résultats sont représentés par un histogramme dans la figure 51.

Nombre de clients	Nombre de clients servis	Nombre de clients servis durant l'accident#4	Nombre de clients affectés par l'accident#4	Nombre de clients satisfaits
50	49	2	0	0
100	98	16	5	4
150	148	24	8	4
200	196	38	15	9
250	244	53	19	12
300	293	67	25	16
350	342	73	27	18
400	390	89	32	22
450	437	98	36	25
500	487	116	41	30

Tab 33 : Impact de l'accident#4 sur les clients

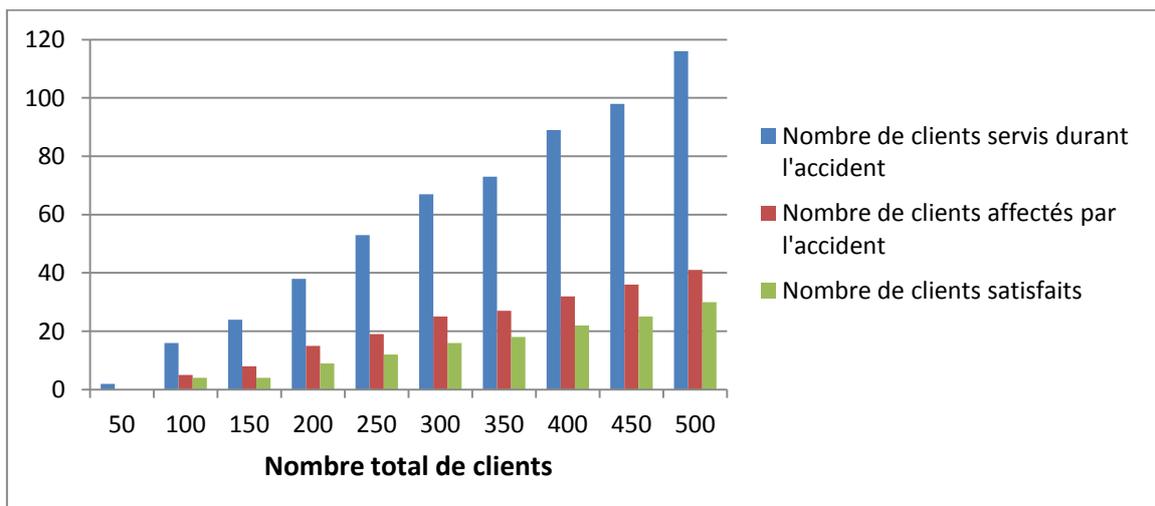


Figure 51 : Nombre de clients insatisfaits à cause de l'accident#4

Le nombre de clients affectés par les accidents dépend de :

- la viabilité de la route où l'accident a eu lieu ;
- la localisation et le temps d'apparition de l'accident ;
- la sévérité spatio-temporelle de l'accident ;
- le nombre de clients servis pendant la période de l'affectation de l'accident.

Nous remarquons que :

- le nombre de clients insatisfaits croît lorsque le nombre de clients servis pendant la période de l'accident croît ;
- un accident affecte de 2 % à 39 % des clients ;
- de 50 % à 100 % des clients affectés sont satisfaits car les accidents sont pris en considération par le système.

La deuxième expérimentation sera présentée dans ce qui suit.

10.2.2 Couverture du réseau routier du Tripoli par des agents Route et Régulateur

La construction des agents Route dépend de la vitesse limite et de la situation du trafic des routes. Par contre, la construction des agents Régulateur dépend seulement de la situation du trafic des routes. Dans cette expérimentation, nous mesurons le nombre d'agents Route et Régulateur associés aux routes de Tripoli selon la situation du trafic routier. Ce nombre est mesuré chaque 30 minutes en débutant de 07h00 le matin jusqu' à 24h00 le minuit. Nous choisissons en plus une route en appliquant la distribution uniforme sur les routes les plus probables d'avoir un accident. Puis, nous générons un accident sur la route choisie et nous mesurons la modification de ce nombre d'agents à cause de cet accident. Le tab 34 montre les résultats de cette expérimentation. La figure 52 et la figure 53 représentent par des graphiques les résultats du tab 34. Ces graphiques montrent que :

- à minuit où le trafic routier est fluide partout, la ville de Tripoli, composée de 315 routes, est couverte par 14 agents Route et un seul agent Régulateur ;
- le nombre d'agents Régulateur est plus petit que celui d'agents Route ;
- le nombre d'agents augmente pendant les heures de pointe entre [07h00,09h00] et [14h00,16h00] et diminue pendant les autres intervalles de temps ;
- le nombre d'agents augmente pendant les accidents.

Temps de mesure	Nombre d'agents Route	Nombre d'agents Régulateur	Nombre d'agents Route suite à un accident	Nombre d'agents Régulateur suite à un accident
07h00	34	4	40	6
07h30	35	3	42	5
08h00	31	5	38	8
08h30	27	7	33	7
09h00	28	6	36	7
09h30	23	6	27	6
10h00	25	5	29	6
10h30	27	6	33	7
11h00	20	5	28	7
11h30	29	5	35	8
12h00	27	6	34	6
12h30	27	4	33	5
13h00	27	5	37	8
13h30	27	4	30	5
14h00	31	6	36	8
14h30	41	4	46	5
15h00	38	4	41	5
15h30	43	3	46	4
16h00	30	7	38	6
16h30	28	5	32	5
17h00	28	4	31	6
17h30	26	4	31	5
18h00	26	7	32	8
18h30	27	6	30	7
19h00	28	3	34	6
19h30	26	5	32	6
20h00	22	6	30	7
20h30	23	6	34	7
21h00	18	5	35	6
21h30	18	3	23	4
22h00	16	2	20	3
22h30	15	2	21	3
23h00	14	1	23	2
23h30	14	1	18	2

Tab 34 : Nombre d'agents Route et Régulateur associés aux routes du Tripoli pendant une journée

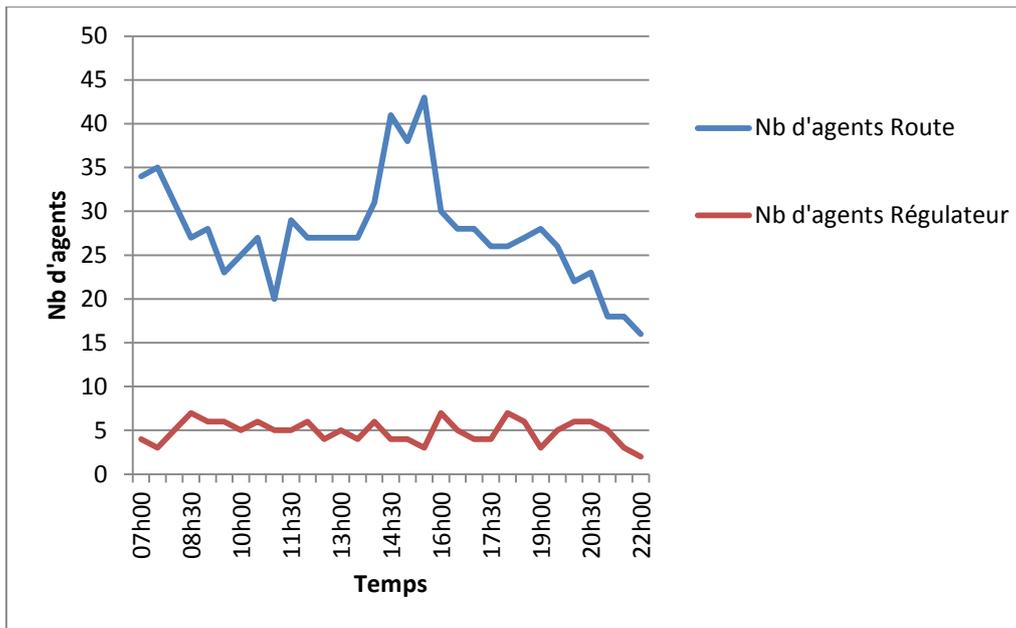


Figure 52 : Nombre d'agents Route et Régulateur associés aux routes du Tripoli pendant une journée

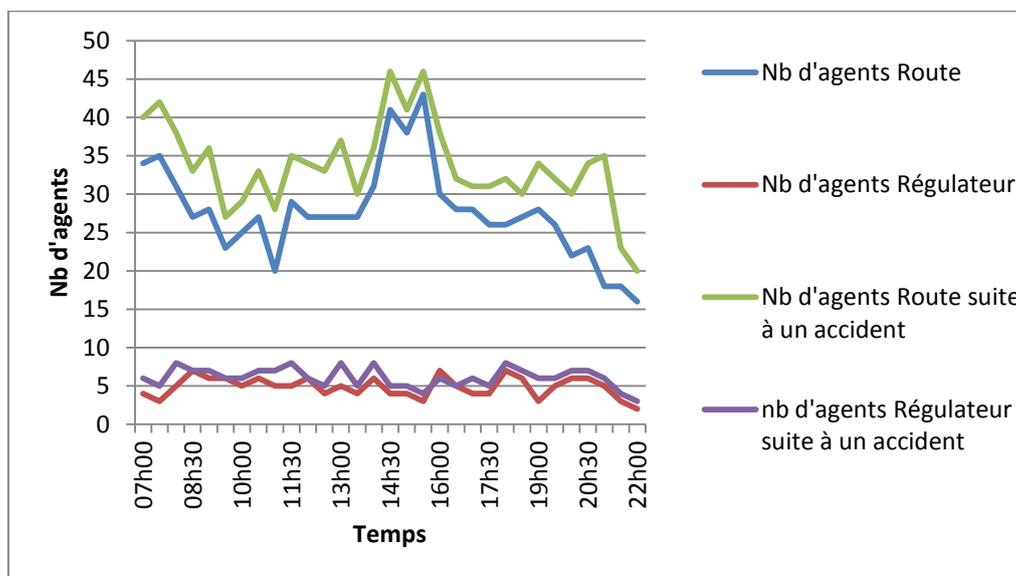


Figure 53 : Variation du nombre d'agents Route et Régulateur suite à un accident

10.2.3 Mesure du nombre d'agents Régulateur contactés

Dans l'approche proposée dans le chapitre 4, un agent Route est associé à chaque route. L'agent Pathfinder contacte les agents Route associés aux routes qui composent un chemin pour l'évaluer. Dans cette approche, l'agent Pathfinder contacte les agents Régulateur associés aux routes du chemin pour l'évaluer. Dans cette expérimentation, nous mesurons le nombre d'agents Route et celui d'agents Régulateur contactés respectivement dans l'approche du chapitre 4 et celle du chapitre présent. Ces agents sont contactés afin d'évaluer un chemin pour chaque demande de transport. Le tab 35 donne les résultats de l'expérimentation.

Nb de demandes	Nb d'agents Route contactés	Nb d'agents Régulateur contactés
100	544	211
200	1096	440
300	1591	593
400	2149	821
500	2688	947

Tab 35 : Comparaison entre le nombre d'agents Route et celui d'agents Régulateur

La figure 54 représente les résultats du tab 35. Elle montre que le nombre d'agents Route contactés dans le chapitre 4 est toujours plus grand que celui des agents Régulateur. L'approche de ce chapitre économise plus que 60 % des agents contactés.

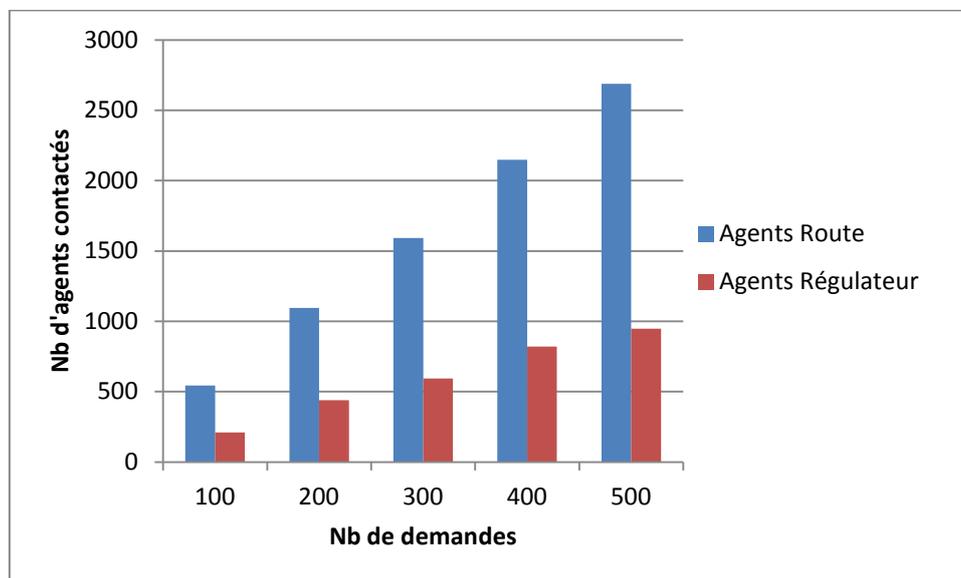


Figure 54 : Différence entre le nombre des agents Routes et Régulateur contactés

11 Conclusion

Dans ce chapitre, nous avons proposé une approche multi-agent massif auto-organisé pour le problème du TAD. Cette approche sert à modéliser et à résoudre ce problème dans un réseau routier avec des vitesses de voyages dynamiques stochastiques. Ces vitesses évoluent en fonction du temps selon la situation réelle de trafic. Nous avons agentifié le réseau routier du système du TAD en associant à ses routes des agents Route et Régulateur dynamiques. Ces agents changent en nombres et en caractéristiques afin de refléter la situation réelle de trafic. Enfin, l'organisation de trafic à son échelle est présentée dans le système du TAD. Cela en s'appuyant sur des connaissances historiques relatives au trafic et à l'impact des événements imprévus (comme un accident) sur le trafic et des informations de trafic en temps réel. Les

informations sur la fluidité du trafic sont examinées afin que le système donne les durées exactes des trajectoires, évite les incidents sur les routes par des détournements calculés et donne des solutions fiables pour que les clients soient satisfaits.

Conclusion générale et perspectives

1 Conclusion

Dans ce manuscrit, nous avons modélisé et résolu le problème du transport à la demande (TAD) par les systèmes multi-agents. Les principales idées de nos contributions sont les suivantes :

- la proposition d'une approche multi-agent décentralisée basée sur l'algorithme A* pour résoudre, de manière optimale, le problème du TAD dans un réseau routier statique. Ce réseau se caractérise par des vitesses de voyages constantes. Nous avons adapté l'algorithme A* pour chercher, pour chaque demande de transport, la trajectoire la plus rapide menée par les véhicules les moins chers ;
- la proposition d'une approche multi-agent massif pour le problème du TAD dans une ville avec un réseau routier dynamique déterministe. Ce réseau se caractérise par des vitesses de voyages dépendantes de la situation habituelle de trafic. L'approche se base sur des connaissances historiques de trafic routier pour calculer selon le temps, les vitesses de voyage et la durée des trajets des véhicules. Pour chaque demande de transport, un véhicule est choisi pour transporter les passagers tout en respectant des contraintes exigées par les clients et les véhicules et tout en optimisant le nombre de véhicules en circulation et la distance totale parcourue. Le bon véhicule résulte d'une négociation se déroulant entre les véhicules capables de servir la demande. Deux méthodes de négociation ont été proposées. Outre l'étude théorique, nous effectuons des études expérimentales pour comparer les méthodes proposées avec une troisième de la littérature. Les résultats expérimentaux montrent que le nombre de clients insatisfaits est supérieur à 50 % si l'évolution déterministe de la situation dynamique de trafic n'est pas considérée ;
- la proposition d'une approche multi-agent massif auto-organisée pour le problème du TAD dans une ville avec un réseau routier dynamique stochastique. Ce réseau se caractérise par des vitesses de voyages dépendantes de la situation réelle de trafic. Comme les connaissances historiques ne suffisent pas pour refléter la situation réelle de trafic au cas où un événement imprévu a eu lieu tel un accident, nous étendons l'approche proposée précédemment. Tout d'abord, l'approche prévoit l'état du trafic en se basant sur des connaissances historiques de trafic. Dans le cas où le système

détecte une situation inhabituelle de trafic, il se base sur des informations réelles (comme le traitement et l'analyse des images prises par des capteurs visuels) pour déterminer la nature de l'événement qui a causé cette situation. Ensuite, il se base sur des connaissances historiques relatives à l'impact de cet événement sur le trafic pour déterminer son impact spatio-temporel présent et futur sur la fluidité des véhicules. Ainsi, le système sera capable de définir la situation actuelle et à venir du trafic selon ses connaissances. Le système multi-agent massif s'auto-organise pour refléter la situation réelle de trafic. Cette auto-organisation se révèle par l'évolution des agents dynamiques pour couvrir des routes dynamiques incertaines selon la situation du trafic. Le système adapte les trajectoires des véhicules selon l'évolution stochastique du trafic afin de servir les clients et les satisfaire. Les résultats expérimentaux montrent que jusqu'à 39 % des clients seront insatisfaits si les accidents routier ne sont pas considérés durant le traitement de leurs demandes. Autrement, 50 % à 100 % de ces clients sont satisfaits.

2 Perspectives

2.1 Enrichissement des demandes de transport

La définition d'une demande de transport peut être étendue pour contenir plus de paramètres reflétant les envies du client et ses préférences. Par exemple : le choix du chemin (le chemin le moins cher, le chemin le plus court en termes de distance ou de temps, etc.), le nombre de passagers à bord, etc. De plus, le client et le système peuvent négocier afin de satisfaire au mieux les préférences et les besoins des passagers.

2.2 Correspondances avec d'autres véhicules du système

Dans les SMA massifs que nous proposons dans cette thèse et qui modélisent le problème du transport à la demande dans une ville, les passagers d'une demande de transport sont transportés par le même véhicule du point de départ au point d'arrivée. Nous envisageons de considérer la possibilité que plusieurs véhicules collaborent pour servir la même demande et transporter les passagers. Le trajet de la demande doit donc être divisé entre ces véhicules et des points de relais doivent être définis. Nous pouvons penser aux deux solutions suivantes :

- ou bien décomposer la demande de sorte à être semblable à un ensemble de demandes et chacune de ces demandes est servie par un véhicule ;

- ou bien agréger plusieurs parties des trajets d'un certain nombre de demandes pour être servies par le même véhicule.

2.3 Utilisation des modes de transports publics

Dans le même concept de l'idée précédente sur la décomposition d'une demande de transport pour être desservie par plusieurs véhicules, les systèmes de TAD peuvent utiliser les transports public (comme le bus, les trains, les métros, etc.) comme des correspondances. Dans ce cas, un langage commun doit exister entre le système du TAD et les autres systèmes de transports pour qu'ils puissent échanger des informations. Cette fois-ci, la correspondance sera avec des véhicules internes et externes au système du TAD.

2.4 Application aux autres problèmes de transport

La même modélisation peut être appliquée pour résoudre tout type de problème du transport routier rencontré dans la vie courante, comme la collecte et la distribution de marchandises, le ramassage de déchets, le ramassage scolaire, transport de fonds, etc. Durant leurs tournées, les véhicules peuvent éviter les zones encombrées.

2.5 Régulation autonome du trafic routier

Ce modèle d'agentification peut être étendu afin d'avoir un système autonome de régulation du trafic routier dans une ville. Ce système gère le trafic dans l'objectif d'éviter la création de congestion routière et résoudre le problème d'encombres et d'anomalies par des détournements calculés auparavant. Il doit d'une part gérer les feux de signalisation et, d'autre part, anticiper les actions des conducteurs sur la fluidité de leur trajet pour participer encore à la régulation du trafic. Un tel système sera un régulateur autonome qui a comme objectif de fluidifier au maximum la circulation selon les heures de la journée et selon les jours de la semaine.

2.6 Gestion des cas d'urgences

Nos propositions peuvent être approfondies et appliquées dans le thème de la gestion des cas d'urgence. Les flottes de véhicules et leurs trajets doivent bien être gérés de manière à accéder aux lieux d'urgence en temps très court et cela en prenant les routes les plus courtes et les moins congestionnées.

Bibliographie

- Abdulrab, Habib, Eduard Babkin, et Sergey Satunin. 2010. "A hybrid multi-layered approach to demand responsive transport systems modeling". *5th International Conference on System of Systems Engineering (SoSE)*, 1-6.
- Al-Fuqaha, Ala, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, et Moussa Ayyash. 2015. "Internet of things : A survey on enabling technologies, protocols, and applications". *IEEE Communications Surveys Tutorials*, 17 (4): 2347-76.
- Andreatta, G., M. Casula, C. De Francesco, et L. De Giovanni. 2016. "A branch-and-price based heuristic for the stochastic vehicle routing problem with hard time windows". *Electronic Notes in Discrete Mathematics* 52: 325-32.
- Atahran, Ahmed. 2012. "Etude et résolution d'un problème de transport à la demande multicritère". Thèse de doctorat, Université François Rabelais de Tours, France.
- Balakrishnan, Nagraj. 1993. "Simple Heuristics for the Vehicle Routing Problem with Soft Time Windows". *Journal of the Operational Research Society*, 279-287.
- Balbo, Flavien. 2000. "ESAC: un modèle d'interaction multi-agent utilisant l'environnement comme support actif de communication: application à la gestion des transports". Thèse de doctorat, Université Paris-Dauphine, Paris, France.
- Balseiro, S. R., Irene Loiseau, et Juan Ramonet. 2011. "An Ant Colony algorithm hybridized with insertion heuristics for the Time Dependent Vehicle Routing Problem with Time Windows". *Computers & Operations Research* 38 (6): 954-66.
- Bearse, Peter, Shiferaw Gurmu, Carol Rapaport, et Steven Stern. 2004. "Paratransit demand of disabled people". *Transportation Research Part B: Methodological* 38 (9): 809-31.
- Bellmore, Mandell, et George L. Nemhauser. 1968. "The traveling salesman problem: a survey". *Operations Research* 16 (3): 538-558.
- Berhan, Eshetie, Birhanu Beshah, Daniel Kitaw, et Ajith Abraham. 2014. "Stochastic Vehicle Routing Problem: A Literature Survey". *Journal of Information & Knowledge Management* 13 (3): 1-12.
- Bertelle, Cyrille, Michel Nabaa, Damien Olivier, et Pierrick Tranouez. 2009. "A Decentralised Approach for the Transportation On Demand Problem". *From System Complexity to Emergent Properties*, 281-289. Springer.

- Boissier, Olivier, et Yves Demazeau. 1994. "ASIC: An architecture for social and individual control and its application to computer vision". *European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, 135–149. Springer.
- Boissier, Olivier, Sylvain Gitton, et Pierre Glize. 2004. "Caractéristiques des systèmes et des applications". *Systèmes Multi-agents, Observatoire Français des Techniques Avancées (OFTA)*, chapitre 1, p. 25-54, Éditions TEC & DOC.
- Bonabeau, Eric, Marco Dorigo, et Guy Theraulaz. 1999. *Swarm intelligence: from natural to artificial systems*. New York, NY: Oxford University Press.
- Botti, V., Federico Barber, Alfons Crespo, Eva Onaindia, Ana García-Fornes, Ismael Ripoll, D. Gallardo, et Luis Hernández. 1995. "A temporal blackboard for a multi-agent environment". *Data & knowledge engineering* 15 (3): 189–211.
- Bourdon, François. 1997. "The interactional semantics of knowledge". *International Joint Conference on Artificial Intelligence*, Poster Session, p. 15, Nagoya (Japan).
- . 1998. "Systèmes d'information ouverts: sémantique interactionnelle des connaissances et systèmes multi-agents". Habilitation à diriger des recherches, Université de Caen.
- Bouzgarrou, Mohamed Ekbal. 1998. "Parallélisation de la méthode du "Branch and Cut" pour résoudre le problème du voyageur de commerce". Thèse de doctorat, Institut National Polytechnique de Grenoble (INPG).
- Bratman, Michael. 1987. *Intention, plans, and practical reason*. Cambridge, Massachusetts: Harvard University.
- Briot, Jean-Pierre. 2001. *Principes et architecture des systèmes multi-agents*. Hermès Science.
- Brooks, Rodney. A. 1991. "Intelligence without reason, computers and thought lecture". *Proceedings of IJCAI-91, Sydney, Australia* 32: 37–38.
- Bruglieri, Maurizio, Alberto Colorni, Federico Lia, et Alessandro Luè. 2014. "A Multi-objective Time-dependent Route Planner: A Real World Application to Milano City". *Transportation Research Procedia* 3: 460–469.
- Bussmann, Stefan, et Jorg Muller. 1992. "A negotiation framework for co-operating agents". *Proceedings of CKBS-SIG*, 1–17.
- Campagne, Jean-Charles, Alain Cardon, Etienne Collomb, et Toyooki Nishida. 2004. "Massive Multi-Agent Systems Control". *Formal Approaches to Agent-Based Systems*, édité par Michael G. Hinchey, James L. Rash, Walter F. Truszkowski, et

- Christopher A. Rouff, 275-80. *Lecture Notes in Computer Science*, vol 3228. Springer, Berlin, Heidelberg.
- Campbell, Ann M., Michel Gendreau, et Barrett W. Thomas. 2011. "The Orienteering Problem with Stochastic Travel and Service Times". *Annals of Operations Research* 186 (1): 61-81.
- Camps, Valérie, et Marie-Pierre Gleizes. 1995. "Principes et évaluation d'une méthode d'auto-organisation". *Troisièmes JFIADSM*, St Baldoph (Savoie), 15-17.
- . 1996. "Cooperative and mobile agents to find relevant information in a distributed resources network". *Workshop on Artificial Intelligence-based tools to help W3 users*, Fifth international conference on World Wide Web, Paris.
- Cardon, Alain. 2002. "Control and Behavior of a Massive Multi-Agent System". *Innovative Concepts for Agent-Based Systems*, édité par Walt Truszkowski, Mike Hinchey, et Chris Rouff, 46-60. *Lecture Notes in Computer Science*, vol. 2564. Springer, Berlin, Heidelberg.
- . 2005. "Design and Behavior of a Massive Organization of Agents". *Design of Intelligent Multi-Agent Systems*, 133-90. *Studies in Fuzziness and Soft Computing*. Springer, Berlin, Heidelberg.
- Cardon, Alain, et Mhamed Itmi. 2016. *Les nouveaux systèmes autonomes*. Génies mécaniques et mécaniques des solides. ISTE editions.
- Carotenuto, Pasquale, Daniele Monacelli, Giuseppe Raponi, et Marco Turco. 2012. "A dynamic simulation model of a flexible transport services for people in congested area". *Procedia-Social and Behavioral Sciences* 54: 357-364.
- Carotenuto, Pasquale, Leonardo Paradisi, et Giovanni Storchi. 2014. "Flexible Transport Service for Passengers". *Transportation Research Procedia* 3: 442-51.
- Castex, Elodie. 2007. "Le Transport A la Demande (TAD) en France : de l'état des lieux à l'anticipation. Modélisation des caractéristiques fonctionnelles des TAD pour développer les modes flexibles de demain". Thèse de doctorat, Université d'Avignon.
- Cattaruzza, Diego, Nabil Absi, Dominique Feillet, et Jesús González-Feliu. 2015. "Vehicle Routing Problems for City Logistics". *EURO Journal on Transportation and Logistics*: 1-29.
- Černý, V. 1985. "Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm". *Journal of Optimization Theory and Applications* 45 (1): 41-51.

- Chang, Yaw, et Lin Chen. 2007. "Solve the vehicle routing problem with time windows via a genetic algorithm". *Discrete Continuous Dynamical Systems*: 240–249.
- Chen, Chun-Hsien, et Luis G. Occeña. 2000. "Knowledge decomposition for a product design blackboard expert system". *Artificial Intelligence in Engineering* 14 (1): 71–82.
- Ciancarini, Paolo. 1996. "Coordination models and languages as software integrators". *ACM Computing Surveys (CSUR)* 28 (2): 300–302.
- Clausen, Jens. 1999. "Branch and bound algorithms-principles and examples". *Department of Computer Science, University of Copenhagen*, 1–30.
- Colomi, Alberto, Marco Dorigo, Vittorio Maniezzo, et others. 1991. "Distributed optimization by ant colonies". *Proceedings of the first European conference on artificial life*: 134–142. Paris, France.
- Commission of the European Communities. 2009. "A sustainable future for transport: Towards an integrated, technology-led and user friendly system". Technical Report COM, pp 279.
- Cordeau, Jean-François, Guy Desaulniers, Jacques Desrosiers, Marius Solomon, et François Soumis. 2002. *"The VRP with time windows"*. P. Toth and D. Vigo (eds.), *The Vehicle Routing Problem*, SIAM Monographs on Discrete Mathematics and Applications.
- Cordeau, Jean-François, et Gilbert Laporte. 2003. "A tabu search heuristic for the static multi-vehicle dial-a-ride problem". *Transportation Research Part B: Methodological* 37 (6): 579-94.
- . 2007. "The Dial-a-Ride Problem: Models and Algorithms". *Annals of Operations Research* 153 (1): 29-46.
- Cordeau, Jean-François, Gilbert Laporte, Jean-Yves Potvin, et Martin WP Savelsbergh. 2007. "Transportation on demand". *Transportation, handbooks in operations research and management science* 14: 429–466.
- Cordeau, Jean-François. 2006. "A Branch-and-Cut Algorithm for the Dial-a-Ride Problem". *Oper. Res.* 54 (3): 573–586.
- Crainic, Teodor Gabriel, Nicoletta Ricciardi, et Giovanni Storchi. 2009a. "Models for evaluating and planning city logistics systems". *Transportation science* 43 (4): 432–454.
- . 2009b. "Models for Evaluating and Planning City Logistics Systems". *Transportation Science* 43 (4): 432–454.

- Cubillos, Claudio, Franco Guidi-Polanco, et Claudio Demartini. 2004. "Multi-agent infrastructure for distributed planning of demand-responsive passenger transportation service". *IEEE International Conference on Systems, Man and Cybernetics*: 2013-17.
- Cubillos, Claudio, Nibaldo Rodriguez, et Broderick Crawford. 2007. "A Study on Genetic Algorithms for the DARP Problem". *Bio-Inspired Modeling of Cognitive Tasks*, édité par José Mira et José R. Álvarez, 498-507. Lecture Notes in Computer Science 4527. Springer, Berlin, Heidelberg.
- Cubillos, Claudio, Enrique Urra, et Nibaldo Rodríguez. 2009. "Application of Genetic Algorithms for the DARPTW Problem". *International Journal of Computers Communications & Control* 4 (2): 127.
- Dahbi, Aziz, Najib El Kamoun, Abdelhak Aqqal, et Asmaa El Hannani. 2014. "Application d'une approche inspirée des colonies de fourmis pour la recommandation des chemins d'apprentissage dans un cours en ligne: modèle et expérience". *Revue internationale des technologies en pédagogie universitaire/International Journal of Technologies in Higher Education* 11 (2): 6-18.
- Dantzig, George B., et John H. Ramser. 1959. "The truck dispatching problem". *Management science* 6 (1): 80-91.
- Delaunay, Boris. 1934. "Sur la sphere vide". *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk* 7: 793-800.
- DeLoach, Scott A. 2005. "Engineering Organization-Based Multiagent Systems". *Software Engineering for Multi-Agent Systems IV*, édité par Alessandro Garcia, Ricardo Choren, Carlos Lucena, Paolo Giorgini, Tom Holvoet, et Alexander Romanovsky, 109-25. Lecture Notes in Computer Science 3914. Springer, Berlin, Heidelberg.
- Serugendo, Giovanna Di Marzo, Marie-Pierre Gleizes, et Anthony Karageorgos. 2005. "Self-organization in multi-agent systems". *The Knowledge Engineering Review* 20 (2): 165-189.
- Dijkstra, E.W. 1959. "A Note on Two Problems in Connexion with Graphs", *Numerische Mathematik* 1, p. 269-271.
- Dorigo, Marco, Vittorio Maniezzo, et Alberto Colorni. 1996. "Ant system: optimization by a colony of cooperating agents". *IEEE Transactions on Systems, Man, and Cybernetics, Part B* 26 (1): 29-41.
- DREO, Johann, Alain Petrowski, Patrick Siarry, et Eric Taillard. 2003. *Métaheuristiques pour l'optimisation difficile*. EYROLLES.

- Drogoul, Alexis. 1993. "De la simulation multi-agent à la résolution collective de problèmes".
Thèse de doctorat, Université Paris VI.
- Duan, Zhengyu, Shichao Sun, Shuo Sun, et Weifeng Li. 2015. "Stochastic Time-Dependent Vehicle Routing Problem: Mathematical Models and Ant Colony Algorithm".
Advances in Mechanical Engineering 7 (11): 1-16.
- Eckhardt, Roger. 1987. "Stan ulam, john von neumann, and the monte carlo method". *Los Alamos Science* 15 (30): 131-136.
- Eglese, Richard. W. 1990. "Simulated annealing: A tool for operational research". *European Journal of Operational Research* 46 (3): 271-81.
- Eglese, Richard, Will Maden, et Alan Slater. 2006. "A Road timetableTM to Aid Vehicle Routing and Scheduling". *Comput. Oper. Res.* 33 (12): 3508–3519.
- Ehmke, Jan Fabian, et Dirk Christian Mattfeld. 2010. "Data Allocation and Application for Time-Dependent Vehicle Routing in City Logistics". *European Transport \ Trasporti Europei* n. 46 (2010): 24-35.
- . 2011. "Integration of Information and Optimization Models for Vehicle Routing in Urban Areas". *Procedia - Social and Behavioral Sciences* 20: 110-119.
- Ehmke, Jan Fabian, Stephan Meisel, et Dirk Christian Mattfeld. 2012. "Floating car based travel times for city logistics". *Transportation Research Part C: Emerging Technologies* 21 (1): 338-52.
- Ehmke, Jan Fabian, André Steinert, et Dirk Christian Mattfeld. 2012. "Advanced routing for city logistics service providers based on time-dependent travel times". *Journal of Computational Science, City Logistics*, 3 (4): 193-205.
- El Falou, Mohamad, Mhamed Itmi, Salah El Falou, et Alain Cardon. 2014a. "On demand transport system's approach as a multi-agent planning problem". *IEEE International Conference on Advanced Logistics and Transport (ICALT)*, 53–58.
- El Falou, Mohamad, Anas Malass, Mhamed Itmi, Salah El Falou, et Alain Cardon. 2014b. "An application of adapted A* decentralized approach for on demand transportation problem". *IEEE 26th International Conference on Tools with Artificial Intelligence*, 930–936.
- Ferber, Jacques. 1995. *Les systèmes multi-agents: vers une intelligence collective*. InterEditions.

- Ferber, Jacques, Tiberiu Stratulat, et John Tranier. 2009. "Towards an integral approach of organizations in multi-agent systems". *Handbook of Research on MultiAgent Systems: Semantics and Dynamics of Organizational Models*, 51–75.
- Finin, Tim, Richard Fritzon, Don McKay, et Robin McEntire. 1994. "KQML as an agent communication language". *Proceedings of the third international conference on Information and knowledge management*, 456–463. ACM Press.
- Fitzgerald, James, Donna Shaunesey, et Steven Stern. 2000. "The effect of education programs on paratransit demand of people with disabilities". *Transportation Research Part A: Policy and Practice* 34 (4): 261-85.
- Fleischmann, Bernhard, Martin Gietz, et Stefan Gnutzmann. 2004. "Time-Varying Travel Times in Vehicle Routing". *Transportation Science* 38 (2): 160–173.
- Foisel, Rémy, Vincent Chevrier, et Jean-Paul Haton. 1996. "De l'organisation d'une société à sa réorganisation". Journée thématique du PRC-IA sur les systèmes multi-agents, Toulouse, 121–128.
- Fox, Mark S. 1981. "An organizational view of distributed systems". *IEEE Transactions on Systems, Man, and Cybernetics* 1 (11): 70–80.
- Garaix, Thierry. 2007. "Étude et résolution exacte de problèmes de transport à la demande avec qualité de service". Thèse de doctorat, Université d'Avignon.
- Gendreau, Michel, Gilbert Laporte, Christophe Musaraganyi, et Éric D. Taillard. 1999. "A tabu search heuristic for the heterogeneous fleet vehicle routing problem". *Computers & Operations Research* 26 (12): 1153-73.
- Gendreau, Michel, François Guertin, Jean-Yves Potvin, et Éric D. Taillard. 1999. "Parallel Tabu Search for Real-Time Vehicle Routing and Dispatching". *Transportation Science* 33 (4): 381-90.
- Georgeff, Michael P., et Amy L. Lansky. 1987. "Reactive reasoning and planning". In *AAAI*, 87 : 677–682.
- Gherboudj, Amira. 2013. "Méthodes de résolution de problèmes difficiles académiques". Thèse de doctorat, Université de Biskra.
- Glover, Fred. 1986. "Applications of Integer Programming Future paths for integer programming and links to artificial intelligence". *Computers & Operations Research* 13 (5): 533-49.

- Golden, Bruce L., Subramanian Raghavan, et Edward A. Wasil, eds. 2008. *The Vehicle Routing Problem: Latest Advances and New Challenges*. Vol. 43. Springer Science & Business Media.
- Goldengorin, Boris, et Panos M. Pardalos. 2012. *Data Correcting Approaches in Combinatorial Optimization*. Springer Science & Business Media.
- Grassé, Plerre-P. 1959. "La reconstruction du nid et les coordinations interindividuelles chez *Bellicositermes natalensis* et *Cubitermes* sp. la théorie de la stigmergie: Essai d'interprétation du comportement des termites constructeurs". *Insectes sociaux* 6 (1): 41–80.
- Groß, Patrick-Oliver, Marlin W. Ulmer, Jan F. Ehmke, et Dirk C. Mattfeld. 2015. "Exploiting travel time information for reliable routing in city logistics". *Transportation Research Procedia* 10: 652-661.
- Grosso, Sergio, Jonathan Higgins, Jenny Mageean, et John D. Nelson. 2002. "Demand responsive transport: towards best practice in rural applications". *Proceedings of the aet european transport conference*, homerton college, Cambridge.
- Guerriero, Francesca, Ferdinando Pezzella, Ornella Pisacane, et Luigi Trollini. 2014. "Multi-objective Optimization in Dial-a-ride Public Transportation". *Transportation Research Procedia* 3: 299-308.
- Hadeli, Karuna, Paul Valckenaers, Constantin Zamfirescu, Hendrik Van Brussel, Bart Saint Germain, Tom Hoelvoet, et Elke Steegmans. 2003. "Self-Organising in Multi-Agent Coordination and Control Using Stigmergy". *Engineering Self-Organising Systems*, édité par Giovanna Di Marzo Serugendo, Anthony Karageorgos, Omer F. Rana, et Franco Zambonelli, 105-23. Lecture Notes in Computer Science 2977. Springer, Berlin, Heidelberg.
- Hadeli, Paul Valckenaers, Martin Kollingbaum, et Hendrik Van Brussel. 2004. "Multi-agent coordination and control using stigmergy". *Computers in Industry* 53 (1): 75-96.
- Harbaoui Dridi, Imen, Ryan Kammarti, Pierre Borne, et Mekki Ksouri. 2009. "Genetic algorithm for multicriteria optimization of a multi-pickup and delivery problem with time windows". *13th INCOM IFAC symposium*, 5 pages. Moscow, Russia.
- . 2008. "Un Algorithme génétique pour le problème de ramassage et de livraison avec fenêtres de temps à plusieurs véhicules". Manuscrit auteur, publié dans CIFA, Roumanie 2008.

- . 2010. "Multi-objective Optimization For The Dynamic Multi-Pickup and Delivery Problem with Time Windows". In *META '2010*, Tunisie, Djerba.
- Hart, Peter E., Nils J. Nilsson, et Bertram Raphael. 1968. "A formal basis for the heuristic determination of minimum cost paths". *IEEE transactions on Systems Science and Cybernetics* 4 (2): 100-107.
- Hashimoto, Hideki, Mutsunori Yagiura, et Toshihide Ibaraki. 2008. "An iterated local search algorithm for the time-dependent vehicle routing problem with time windows". *Discrete Optimization*, In Memory of George B. Dantzig, 5 (2): 434-56.
- Hayes-Roth, Barbara. 1985. "A blackboard architecture for control". *Artificial intelligence* 26 (3): 251-321.
- Hiard, Samuel. 2013. "Trimming the complexity of Ranking by Pairwise Comparison". Thèse de doctorat, Université de Liège, Liège, Belgique.
- Hill, Arthur V., et W. C. Benton. 1992. "Modelling Intra-City Time-Dependent Travel Speeds for Vehicle Scheduling Problems". *The Journal of the Operational Research Society* 43 (4): 343-51.
- Holland, John H. 1975. *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. Vol. viii. Oxford, England: U Michigan Press.
- Hosny, Manar I., et Christine L. Mumford. 2012. "Constructing initial solutions for the multiple vehicle pickup and delivery problem with time windows". *Journal of King Saud University-Computer and Information Sciences* 24 (1): 59-69.
- Hu, Xiangpei, et Lijun Sun. 2012. "Knowledge-based modeling for disruption management in urban distribution". *Expert Systems with Applications* 39 (1): 906-16.
- Ichoua, Soumia, Michel Gendreau, et Jean-Yves Potvin. 2003. "Vehicle dispatching with time-dependent travel times". *European Journal of Operational Research* 144 (2): 379-96.
- Iqbal, Sumaiya, M. Kaykobad, et M. Sohel Rahman. 2015. "Solving the multi-objective Vehicle Routing Problem with Soft Time Windows with the help of bees". *Swarm and Evolutionary Computation* 24: 50-64.
- Ishida, Toru, Les Gasser, et Makoto Yokoo. 1992. "Organization self-design of distributed production systems". *IEEE Transactions on Knowledge and Data Engineering* 4 (2): 123-134.

- Jaw, Jang-Jei, Amedeo R. Odoni, Harilaos N. Psaraftis, et Nigel H. M. Wilson. 1986. "A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows". *Transportation Research Part B: Methodological* 20 (3): 243-57.
- Jennings, Nicholas R., Katia Sycara, et Michael Wooldridge. 1998. "A Roadmap of Agent Research and Development". *Autonomous Agents and Multi-Agent Systems* 1 (1): 7-38.
- Jih, Wan-Rong, et J. Yung-Jen Hsu. 1999. "Dynamic vehicle routing using hybrid genetic algorithms". *IEEE International Conference on Robotics and Automation* 1: 453-458.
- Jin, Xu, Habib Abdulrab, et Mhamed Itmi. 2008. "A multi-agent based model for urban demand-responsive passenger transport services". *IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, 3668-75.
- Josselin, D. 2002. "Les services de transport à la demande dans leur marché et leur cadre institutionnel – étude de faisabilité d'un re-positionnement socio- économique de ces marchés". Synthèse, THEMA et Kéolis, 10 p.
- Kallehauge, Brian. 2008. "Formulations and exact algorithms for the vehicle routing problem with time windows". *Computers & Operations Research*, Part Special Issue: Includes selected papers presented at the ECCO'04 European Conference on combinatorial Optimization, 35 (7): 2307-30.
- Kammarti, Ryan, Slim Hammadi, Pierre Borne, et Mekki Ksour. 2004. "A new hybrid evolutionary approach for the pickup and delivery problem with time windows". *IEEE International Conference on Systems, Man and Cybernetics*, 2: 1498-1503.
- . 2005. "Lower bounds in an hybrid evolutionary approach for the pickup and delivery problem with time windows". *IEEE International Conference on Systems, Man and Cybernetics*, 2: 1156-61.
- . 2006. "Solving the Real Time Dynamic Pickup and Delivery Problem with Time Windows with Hybrid Evolutionary Approach". *IMACS Multiconference on Computational Engineering in Systems Applications*, 1520-25.
- Karp, Richard M. 1972. "Reducibility among Combinatorial Problems". *Complexity of Computer Computations*, édité par Raymond E. Miller, James W. Thatcher, et Jean D. Bohlinger, 85-103. The IBM Research Symposia Series. Springer US.

- Guilbert, David. 2015. "Analyse et classification des signatures des véhicules provenant de capteurs magnétiques pour le développement des algorithmes « intelligents » de gestion du trafic". Thèse de doctorat, Université de Nantes.
- Khichane, Madjid, Patrick Albert, et Christine Solnon. 2009. "Un modèle réactif pour l'optimisation par colonies de fourmis: application à la satisfaction de contraintes". *Journées Francophones de Programmation par Contraintes (JFPC), Orléans*.
- Kim, Gitae, Yew-Soon Ong, Chen Kim Heng, Puay Siew Tan, et Nengsheng Allan Zhang. 2015. "City Vehicle Routing Problem (City VRP): A Review". *IEEE Transactions on Intelligent Transportation Systems* 16 (4): 1654-66.
- Kim, Seongmoon, Mark E. Lewis, et Chelsea C. White. 2005. "Optimal vehicle routing with real-time traffic information". *IEEE Transactions on Intelligent Transportation Systems* 6 (2): 178-88.
- Kirkpatrick, Scott. 1984. "Optimization by Simulated Annealing: Quantitative Studies". *Journal of Statistical Physics* 34 (5-6): 975-86.
- Kong, Linghe, Yang Yang, JiaLiang Lu, Wei Shu, et Wu Min-You. 2012. "Evaluation of Urban Vehicle Routing Algorithms". *International Journal of Digital Content Technology and its Applications* 6 (23): 790-99.
- Koskosidis, Yiannis A., Warren B. Powell, et Marius M. Solomon. 1992. "An Optimization-Based Heuristic for Vehicle Routing and Scheduling with Soft Time Window Constraints". *Transportation Science* 26 (2): 69-85.
- Kumar, Suresh Nanda. 2012. "A Survey on the Vehicle Routing Problem and Its Variants". *Intelligent Information Management* 4 (3): 66-74.
- Lai, David S. W., Ozgun Caliskan Demirag, et Janny M. Y. Leung. 2016. "A tabu search heuristic for the heterogeneous vehicle routing problem on a multigraph". *Transportation Research Part E: Logistics and Transportation Review* 86: 32-52.
- Laporte, Gilbert. 2009. "Fifty Years of Vehicle Routing". *Transportation Science* 43 (4): 408-16.
- Lau, Hoong Chuin, et Zhe Liang. 2002. "Pickup and delivery with time windows: algorithms and test case generation". *International Journal on Artificial Intelligence Tools* 11 (3): 455-72.
- Le Breton, Eric. 2001. "Le transport à la demande comme innovation institutionnelle". *Flux* 43 (1).

- Lecluyse, Christophe, Kenneth Sørensen, et Herbert Peremans. 2013. "A network-consistent time-dependent travel time layer for routing optimization problems". *European Journal of Operational Research* 226 (3): 395-413.
- Lee, Der-Tsai, et Arthur K. Lin. 1986. "Generalized Delaunay triangulation for planar graphs". *Discrete & Computational Geometry* 1 (3): 201–217.
- Lee, Der-Tsai, et Bruce J. Schachter. 1980. "Two algorithms for constructing a Delaunay triangulation". *International Journal of Computer & Information Sciences* 9 (3): 219–242.
- Lin, Yeqian, Wenquan Li, Feng Qiu, et He Xu. 2012. "Research on Optimization of Vehicle Routing Problem for Ride-Sharing Taxi". *Procedia - Social and Behavioral Sciences* 43: 494-502.
- Lipowski, Adam, et Dorota Lipowska. 2012. "Roulette-wheel selection via stochastic acceptance". *Physica A: Statistical Mechanics and its Applications* 391 (6): 2193–2196.
- Liu, Mengyang, Zhixing Luo, et Andrew Lim. 2015. "A Branch-and-Cut Algorithm for a Realistic Dial-a-Ride Problem". *Transportation Research Part B: Methodological* 81: 267-88.
- Lorini, Sandro, Jean-Yves Potvin, et Nicolas Zufferey. 2011. "Online vehicle routing and scheduling with dynamic travel times". *Computers & Operations Research* 38 (7): 1086-90.
- Lu, Quan, et Maged Dessouky. 2004. "An Exact Algorithm for the Multiple Vehicle Pickup and Delivery Problem". *Transportation Science* 38 (4): 503-14.
- . 2006. "A new insertion-based construction heuristic for solving the pickup and delivery problem with time windows". *European Journal of Operational Research* 175 (2): 672-87.
- Luke, Sean. 2009. *Essentials of Metaheuristics*. Lulu.
- . 2011. "Multiagent simulation and the MASON library". George Mason University.
- Maden, Will, Richard Eglese, et Dan Black. 2010. "Vehicle routing and scheduling with time-varying data: A case study". *Journal of the Operational Research Society* 61 (3): 515-22.
- Malandraki, Chryssi. 1989. "Time dependent vehicle routing problems: formulations, solution algorithms and computational experiments". Thèse de doctorat, Northwestern University, Evanston, III.

- Malas, Anas, Mohamad El Falou, Salah El Falou, Mhamed Itmi, et Alain Cardon. 2016a. "Multi-agent approach for on-demand transport problem in a city". *Third International Conference on Electrical, Electronics, Computer Engineering and their Applications (EECEA)*, 40-45.
- Malas, Anas, Salah El Falou, Mohamad El Falou, Mhamed Itmi, et Alain Cardon. 2016b. "Solving on-demand transport problem through negotiation". *Proceedings of the Summer Computer Simulation Conference, SummerSim*, Montreal, QC, Canada, 24-27. Society for Modeling and Simulation International (SCS).
- Marmion, Marie-Eleonore. 2011. "Recherche locale et optimisation combinatoire: de l'analyse structurelle d'un problème à la conception d'algorithmes efficaces". Thèse de doctorat, Université des Sciences et Technologie de Lille-Lille I.
- Metropolis, Nicholas, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, et Edward Teller. 1953. "Equation of State Calculations by Fast Computing Machines". *The Journal of Chemical Physics* 21 (6): 1087-92.
- Mguis, Fethi, Kamel Zidi, Khaled Ghedira, et Pierre Borne. 2012. "Modélisation d'un Système Multi-Agent pour la résolution d'un Problème de Tournées de Véhicule dans une situation d'urgence". *9th International Conference on Modeling, Optimization & SIMulation*, Bordeaux, France.
- Mitchell, John E. 2011. "Branch and cut". *Wiley Encyclopedia of Operations Research and Management Science*.
- Montemanni, Roberto, Luca M. Gambardella, Andrea E. Rizzoli, et Alberto V. Donati. 2003. "A new algorithm for a dynamic vehicle routing problem based on ant colony system". *Second international workshop on freight transportation and logistics*, vol. 1, no. 1, pp. 27-30.
- Nabaa, Michel, Bisma Zeddini, et Pierrick Tranouez. 2007. "Approche décentralisée pour résoudre le problème du transport à la demande". *Majestic 2007*, Caen, France.
- Nemhauser, George L. 2012. "Column generation for linear and integer programming". *Optimization Stories* 20: 64.
- Novoa, Clara, et Robert Storer. 2009. "An approximate dynamic programming approach for the vehicle routing problem with stochastic demands". *European Journal of Operational Research* 196 (2): 509-15.

- Occello, M., et Y. Demazeau. 1998. "Modelling decision making systems using agents for cooperation in a real time constraints". *3rd IFAC Symposium on Intelligent Autonomous Vehicles*, 1:51–56.
- Occello, Michel, Christof Baeijs, Yves Demazeau, et Jean-Luc Koning. 2002. "Mask: An aeio toolbox to develop multi-agent systems". *Knowledge Engineering and Agent Technology, IOS Series on Frontiers in AI and Applications* 63: 64.
- O'Reilly, Grant Blaise, et Elizabeth Ehlers. 2006. "Synthesizing Stigmergy for Multi Agent Systems". *Agent Computing and Multi-Agent Systems*, édité par Zhong-Zhi Shi et Ramakoti Sadananda, 34-45. Lecture Notes in Computer Science 4088. Springer, Berlin, Heidelberg.
- Pang, Wei, Kang-ping Wang, Chun-guang Zhou, et Long-jiang Dong. 2004. "Fuzzy discrete particle swarm optimization for solving traveling salesman problem". *The Fourth International Conference on Computer and Information Technology, CIT '04*, 796-800.
- Parragh, Sophie N., Karl F. Doerner, et Richard F. Hartl. 2007. "A survey on pickup and delivery problems Part II: Transportation between pickup and delivery locations". *Journal für Betriebswirtschaft*.
- Pillac, Victor, Michel Gendreau, Christelle Guéret, et Andrés L. Medaglia. 2013. "A review of dynamic vehicle routing problems". *European Journal of Operational Research* 225 (1): 1–11.
- Potvin, Jean-Yves, Ying Xu, et Ilham Benyahia. 2006. "Vehicle routing and scheduling with dynamic travel times". *Computers & Operations Research, Part Special Issue: Optimization Days 2003*, 33 (4): 1129-37.
- Psaraftis, Harilaos N. 1980. "A Dynamic Programming Solution to the Single Vehicle Many-to-Many Immediate Request Dial-a-Ride Problem". *Transportation Science* 14 (2): 130–154.
- . 1983. "An Exact Algorithm for the Single Vehicle Many-to-Many Dial-A-Ride Problem with Time Windows". *Transportation Science* 17 (3): 351-57.
- Ralphs, Ted K., Leonid Kopman, William R. Pulleyblank, et Leslie E. Trotter. 2003. "On the capacitated vehicle routing problem". *Mathematical programming* 94 (2-3): 343–359.
- Rao, Anand S., et Michael P. Georgeff. 1995. "BDI agents: From theory to practice". *ICMAS*, 95:312–319.

- Rego, César, et Catherine Roucairol. 1994. "Le problème de tournées de véhicules: Étude et résolution approchée". *Projet Reflecs: « rapport de recherche N°2197 »*, vol 125, pp 3-86.
- Ropke, Stefan, Jean-François Cordeau, et Gilbert Laporte. 2007. "Models and Branch-and-Cut Algorithms for Pickup and Delivery Problems with Time Windows". *Networks* 49 (4): 258-72.
- Savelsbergh, Martin WP, et Marc Sol. 1995. "The General Pickup and Delivery Problem". *Transportation Science* 29 (1): 17-29.
- Schilde, Michael, Karl F. Doerner, et Richard F. Hartl. 2014. "Integrating stochastic time-dependent travel speed in solution methods for the dynamic dial-a-ride problem". *European Journal of Operational Research* 238 (1): 18-30.
- Shah, Megha Mihir. 2012. "Artificial Intelligence: Vehicle Routing Problem and Multi Agent System". *IJCA Proceedings on Development of Reliable Information Systems, Techniques and Related Issues DRISTI* (1): 1-3.
- Shamos, Michael Ian, et Dan Hoey. 1975. "Closest-point problems". *IEEE 16th Annual Symposium on Foundations of Computer Science*, p. 151-162.
- Singh, M. P., A. Rao, et M. J. Wooldridge. 1998. "A formal specification of dMARS agent theories, architectures, and languages". *Intelligent agents IV* 1365: 155-176.
- Smith, R. 1980. "The contract net protocol: Highlevel communication and control in a distributed problem solver, 1980". *IEEE Trans. on Computers*, C 29: 12.
- Sniedovich, Moshe. 2010. *Dynamic programming: foundations and principles*. CRC press.
- Solomon, Marius M. 1987. "Algorithms for the vehicle routing and scheduling problems with time window constraints". *Operations research* 35 (2): 254-265.
- Taillard, Eric, Philippe Badeau, Michel Gendreau, François Guertin, et Jean-Yves Potvin. 1997. "A Tabu Search Heuristic for the Vehicle Routing Problem with Soft Time Windows". *Transportation Science* 31 (2): 170-86.
- Talbi, El-Ghazali. 2009. *Metaheuristics: from design to implementation*. Hoboken, N.J: John Wiley & Sons.
- Torğul, Belkız, Lütfü Şağbanşua, et Figen Balo Balo. 2016. "Internet of Things: A Survey". *International Journal of Applied Mathematics, Electronics and Computers* 4: 104-110.
- Toth, Paolo, et Daniele Vigo, éd. 2002. *The Vehicle Routing Problem*. SIAM Monographs on Discrete Mathematics and Applications. Philadelphia, Pa: Society for Industrial and Applied Mathematics.

- Unsal, Cem, et J. S. Bay. 1994. "Spatial self-organization in large populations of mobile robots". *International Symposium on Intelligent Control*, 249–254.
- Valckenaers, Paul, John Sauter, Carles Sierra, et Juan Antonio Rodriguez-Aguilar. 2007. "Applications and environments for multi-agent systems". *Autonomous Agents and Multi-Agent Systems* 14 (1): 61–85.
- Van Woensel, Tom, Laoucine Kerbache, Herbert Peremans, et Nico Vandaele. 2008. "Vehicle routing with dynamic travel times: A queueing approach". *European Journal of Operational Research* 186 (3): 990-1007.
- Voorneveld, Mark. 2003. "Characterization of Pareto dominance". *Operations Research Letters* 31 (1): 7-11.
- Wassan, Niaz, et Gábor Nagy. 2014. "Vehicle Routing Problem with Deliveries and Pickups: Modelling Issues and Meta-heuristics Solution Approaches". *International Journal of Transportation* 2 (1): 95-110.
- Whitmore, Andrew, Anurag Agarwal, et Li Da Xu. 2015. "The Internet of Things—A Survey of Topics and Trends". *Information Systems Frontiers* 17 (2): 261-74.
- Yaman, Hande. 2005. "Formulations and Valid Inequalities for the Heterogeneous Vehicle Routing Problem". *Mathematical Programming* 106 (2): 365-90.
- Zargayouna, Mahdi. 2005. "Une représentation spatio-temporelle de l'environnement pour le Transport A la Demande". Atelier: Représentation sur le temps et l'espace (RTE 2005), Plate-forme AFIA 2005, Nice, France, 62 p.
- . 2007. "Modèle et langage de coordination pour les systèmes multi-agents ouverts. Application au probleme du transporta la demande". Thèse de doctorat, Université de Paris-Dauphine, Paris (France).
- Zeddini, Bisma. 2009. "Modèles d'auto-organisation multi-agent pour le transport à la demande". Thèse de doctorat, Université du Havre, Le Havre (France).
- Zidi, Issam, Kamel Zidi, Khaled Mesghouni, et Khaled Ghedira. 2011. "A Multi-Agent System Based on the Multi-Objective Simulated Annealing Algorithm for the Static Dial a Ride Problem". *IFAC Proceedings Volumes* 44 (1): 2172-77.