

N° d'ordre : 37

**Ecole Centrale de Lille
Université des Sciences et Technologies de Lille**

THESE

présentée en vue d'obtenir le grade de

DOCTEUR

Spécialité : Automatique et Informatique Industrielle

par

Kamel ZIDI

Doctorat délivré conjointement par l'Université des Sciences et
Technologies de Lille et l'Ecole Centrale de Lille

Système Interactif d'Aide au Déplacement Multimodal (SIADM)

Soutenu publiquement le 13 décembre 2006 devant le jury :

| | | |
|-----------------------|------------------------|---|
| Président : | Daniel JOLLY | Professeur, Université d'Artois |
| Rapporteurs : | Patrick SIARRY | Professeur, Université de Paris 12 |
| | Christian TAHON | Professeur, Université de Valenciennes |
| Examineur : | Khaled GHEDIRA | Professeur, ENSI Université Manouba (Tunisie) |
| Directeur : | Slim HAMMADI | Professeur, Ecole Centrale de Lille |
| Co-directeur : | Pierre BORNE | Professeur, Ecole Centrale de Lille |

Thèse préparée dans le laboratoire LAGIS à l'Ecole Centrale de Lille et l'Université des Sciences et
Technologies de Lille.

Résumé

L'objectif de notre travail est la réalisation d'un système interactif d'aide aux déplacements, en mode normal, et en mode dégradé de fonctionnement du réseau de transport en commun. Ce système vise par ailleurs à minimiser le temps d'attente des voyageurs, en mode dégradé, dans les pôles d'échanges et à leur assurer, dans la mesure du possible, la continuité des déplacements dans les réseaux multimodaux. Il s'agit donc d'améliorer la qualité du service rendu aux voyageurs et les maintenir informés. Une grande partie du travail de cette thèse concerne la conception, le développement et la validation des approches qui permettent de donner des solutions optimales ou quasi optimales, pour un réseau de transport normal et perturbé. Ces approches utilisent une méthode multicritère de recherche d'itinéraire qui s'appuie sur une hybridation entre un algorithme de Dijkstra modifié et un algorithme génétique, pour générer une population de chemins minimums. L'algorithme de Dijkstra modifié nous donne un ensemble de solutions servant de population initiale pour l'algorithme génétique.

La modélisation du réseau de transport est représentée par une architecture multi-zones. Cette architecture nous montre l'aspect distribué du système, les interactions et les relations qui peuvent avoir lieu entre les différentes zones. Nous présentons dans ce travail un Système Multi-Agent d'Aide au Déplacement, SMAAD. Les agents de ce système utilisent le module d'optimisation développé dans la première partie. Notre travail est réalisé dans le cadre du projet « VIATIC-MOBILITE », qui est le projet 6 du pôle de compétitivité I-Trans.

Mots-clés

Optimisation multi-objectif, théorie des graphes, algorithme génétique, système multi-agent, transport multimodal, perturbation ...

Abstract

The objective of this work is the realization of a system allowing to assist the travellers, and to facilitate their movement in normal and degraded functioning of the transport network. This system aims to minimize the waiting time of the travellers, in degraded mode, at exchanges stations and to assure them, as well as possible, the continuity of their journey in the multimodal transport networks. So it improves the quality of the service returned to the travellers in order to inform them. A first part of the work in this thesis concerns conception, development and validation of our approach which allows giving optimal or almost optimal solutions for a normal and disrupted transport system. This approach uses a multi-objective method of search for optimal route which leans on a hybridization between a modified Dijkstra algorithm and a genetic algorithm. The modified Dijkstra algorithm gives us a set as solutions serving of initial population for the genetic algorithm.

The modelling of the transport system is represented by multi-zones architecture. This architecture shows us the distributed aspect of the system, and the interactions and the relations which can take place among various zones. We present in this work a Multi-agent system of Help to the Movement. These agents use the module of optimization developed in the first part. Our work is realized within the framework of the "VIATIC-MOBILITE" project, which is the project 6 of the I-Trans Competitiveness cluster.

Key-words

Multi-objective optimization, graphs theory, genetic algorithm, multi-agent system, multimodal transport, Disturbance ...

*A mes chers parents et grand-mère,
à mes frères et sœurs,
à toute ma grande famille
et à la mémoire de mariem.*

Remerciements

La première personne qui me vient à l'esprit et que je tiens à remercier profondément est le Professeur Slim HAMMADI, Professeur à l'Ecole Centrale de Lille et directeur de cette thèse, pour l'énorme soutien scientifique et moral qu'il a su m'adresser pendant ces trois années au sein du LAGIS. Je le remercie aussi pour ses qualités humaines et scientifiques et de m'avoir toujours encouragée à aller de l'avant.

Je tiens également à exprimer ma profonde gratitude au Professeur Pierre BORNE, Professeur et directeur scientifique à l'Ecole Centrale de Lille, et co-directeur de cette thèse, pour sa rigueur scientifique et son aide précieuse et pour m'avoir accueillie au sein de l'équipe ID du LAGIS.

Que Professeur Daniel JOLLI, Professeur à l'université d'Artois et Directeur du laboratoire de Génie Informatique et d'Automatique de l'Artois, reçoive mes plus sincères remerciements pour l'honneur qu'il m'a fait en présidant ce jury.

Professeur Christian TAHON, Professeur à Université de Valenciennes, et Professeur Patrick SIARRY, Professeur à Université de Paris 12, m'ont fait l'honneur d'accepter le rôle de rapporteur de cette thèse malgré un emploi du temps surchargé en cette période de fin d'année. Je les en remercie, leurs différentes suggestions, leurs précieux conseils et remarques m'ont permis d'améliorer la qualité de la version finale de ce document.

Pour m'avoir fait l'honneur d'accepter d'être examinateurs de cette thèse et de participer à ce jury, un grand merci s'adresse à Professeur Khaled GHEDIRA, Professeur et directeur de l'Ecole Nationale des Sciences de l'Informatique à Tunis, ENSI.

Ces remerciements ne seraient pas complets si je n'y associais pas toutes les personnes qui ont contribué de près ou de loin à la réalisation de cette thèse, en particulier, tout le personnel du LAGIS et de l'Ecole Centrale de Lille que j'ai côtoyé durant ma thèse, pour leur bonne humeur et leur disponibilité.

Je n'oublie pas d'adresser mes vifs remerciements à mes Collègues : M^{ed} Mahmoud Ould Sidi, Rayan Kammarti, Inaam Ben Khaled, M^{ed} Amine et Sana Kammoun, Besma Glaa, Nacime Ihdaden etc. Aussi mes sincères remerciements à Françoise dont les nombreuses relectures ont permis que cette thèse soit présentable et écrite dans un français correct.

Finalement, je ne peux qu'être infiniment reconnaissant envers mes parents pour leur soutien indescriptible, leur patience, leur confiance et leurs nombreux sacrifices. Je leur dédie avec plaisir ce travail ainsi qu'à ma grand-mère Baha, mes deux frères Tarek et Salah, mes sœurs Radhiya, Souleyma, Zakia et Chagra sans oublier mon oncle Thabet et mon grand frère Mongi ainsi que toute ma grande famille. Qu'ils sachent que je suis conscient de ce que je leurs dois et j'espère être un jour en mesure de leur prouver mon affection !

Table des matières

| | |
|--|-----------|
| Table des matières | 1 |
| Table des Figures..... | 4 |
| Introduction générale..... | 6 |
| Chapitre I..... | 11 |
| Les problèmes de transport multimodaux centrés autour des services et des clients de transport..... | 11 |
| I.1 Introduction | 11 |
| I.2 Multimodalité..... | 11 |
| I.3 Problème de recherche en transport pour aider à la régulation | 12 |
| I.3.1 Planification du trafic | 13 |
| I.3.2 Régulation du trafic | 14 |
| I.4 Problématique de recherche en transport pour aider les clients | 15 |
| I.5 Les applications existantes | 17 |
| I.6 Les applications en projet..... | 21 |
| I.7 Les limites des systèmes existants | 24 |
| I.8 Conclusion..... | 24 |
| Chapitre II | 25 |
| Etat de l’art en modélisation et optimisation dans le domaine du transport..... | 25 |
| II.1 Introduction | 25 |
| II.2 Modélisation graphique..... | 25 |
| II.3 Introduction du problème NP-complet..... | 28 |
| II.4 Optimisation combinatoire..... | 29 |
| II.5 Optimisation multicritère | 30 |
| II.6 Heuristiques..... | 33 |
| II.7 Les métaheuristiques | 34 |
| II.7.1 Le recuit simulé..... | 35 |
| II.7.2 Méthode Tabou | 35 |
| II.7.3 Algorithme de colonies de fourmis (ACF)..... | 36 |
| II.8 Les Algorithmes évolutionnaires | 37 |
| II.8.1 Principe général..... | 37 |
| II.8.2 Architecture des algorithmes évolutionnaires | 39 |
| II.8.3 Mise en œuvre des algorithmes évolutionnaires | 39 |
| II.8.3.1 Choix d’un codage | 40 |
| II.8.3.2 Opérateur de sélection | 41 |
| II.8.3.3 Opérateur de croisement..... | 42 |
| II.8.3.4 Opérateur de mutation | 44 |
| II.8.3.5 Caractéristiques des algorithmes évolutionnaires | 45 |
| II.8.4 Les algorithmes évolutionnaires et le domaine du transport..... | 47 |
| II.9 Les systèmes multi-Agents (SMA) | 50 |
| II.9.1 Les systèmes multi-agents et l’Intelligence Artificielle..... | 51 |
| II.9.2 Caractéristiques des agents..... | 52 |
| II.9.2.1 Notion d’agent..... | 52 |
| II.9.2.2 Raisonnement individuel des agents | 53 |
| II.9.3 Caractéristiques des systèmes multi-agents | 54 |

| | | |
|--|--|-----|
| II.9.3.1 | Société d'agents..... | 54 |
| II.9.3.2 | Organisations dans les systèmes multi-agents | 55 |
| II.10 | Applications des SMA | 57 |
| II.10.1 | Applications générales | 57 |
| II.10.2 | Applications des SMA dans le domaine du transport | 58 |
| II.11 | Conclusion..... | 61 |
| Chapitre III..... | 62 | |
| Approche de résolution pour un Système Interactif d'Aide au Déplacement Multimodal (SIADM)..... | 62 | |
| III.1 | Introduction | 62 |
| III.2 | Complexité du problème | 63 |
| III.3 | Normalisation des critères | 64 |
| III.4 | Calcul de plus court chemin | 65 |
| III.5 | Algorithme de Dijkstra..... | 67 |
| III.5.1 | Algorithme de Dijkstra classique | 67 |
| III.5.2 | Algorithme de Dijkstra Amélioré..... | 71 |
| III.6 | Graphe dynamique et multimodal | 74 |
| III.7 | Les problèmes et les limites des algorithmes de la théorie des graphes | 77 |
| III.8 | L'application des algorithmes génétiques dans le SIADM | 78 |
| III.8.1 | Codage..... | 80 |
| III.8.2 | Initialisation de la population | 81 |
| III.8.3 | Les opérateurs | 82 |
| III.9 | Hybridation entre l'algorithme de Dijkstra et un AG | 87 |
| III.10 | Architecture du système d'aide au déplacement | 89 |
| III.10.1 | Modélisation globale du système | 90 |
| III.10.2 | Architecture multi-zones | 90 |
| III.10.3 | Le Système Distribuer | 92 |
| III.11 | Système Multi-Agent d'Aide au Déplacement (SMAAD) | 92 |
| III.11.1 | Architecture | 93 |
| III.11.2 | Les agents de SMAAD..... | 94 |
| III.11.2.1 | Agent « SYSTEME »..... | 94 |
| III.11.2.2 | Agent « RESPONSABLE ZONE » | 95 |
| III.11.2.3 | Agent « CALCUL CHEMIN » | 97 |
| III.11.2.4 | Agent « GESTIONNAIRE D'INFORMATION »..... | 98 |
| III.11.2.5 | Agent « SUPERVISEUR » | 98 |
| III.11.2.6 | Agent « PERTURBATION »..... | 98 |
| III.11.3 | Structures organisationnelles..... | 99 |
| III.11.3.1 | Coordination et coopération | 99 |
| III.11.3.2 | Communication | 100 |
| III.11.3.3 | Protocole de communication | 101 |
| III.12 | Conclusion..... | 102 |
| Chapitre IV | 104 | |
| Applications et résultats..... | 104 | |
| IV.1 | Introduction | 104 |
| IV.2 | Outil informatique utilisé | 104 |
| IV.3 | Validation des algorithmes d'optimisation | 105 |
| IV.4 | Système multi-agent..... | 114 |
| IV.4.1 | Plate-forme multi-agent : MadKit | 115 |

| | | |
|---|--|------------|
| IV.4.1.1 | Qu'est-ce que MadKit ? | 115 |
| IV.4.1.2 | Modèle organisationnel : Agents/Groupes/Rôles (AGR)..... | 116 |
| IV.4.2 | Plate-forme multi-agent : JADE..... | 119 |
| IV.5 | Conclusion..... | 122 |
| Conclusion et perspectives..... | | 123 |
| Références bibliographique | | 126 |

Table des Figures

| | | |
|----------------|--|----|
| Figure I.1: | Processus classique de planification. | 14 |
| Figure I.2: | Statistique des voyageurs sur l'ensemble du réseau SNCF [Bulletin 05]. ... | 16 |
| Figure I.3: | Statistique des voyageurs sur le réseau de TGV SNCF [Bulletin 05]. | 16 |
| Figure I.4: | Statistique des voyageurs sur l'ensemble du réseau RATP [Bulletin 05]. ... | 16 |
| Figure I.5: | Architecture d'Infobus à Metz. | 19 |
| Figure II.1: | Exemple de graphe (en pointillé, le plus court chemin entre A et G). | 27 |
| Figure II.2: | Exemple de réseau de transport multimodal. | 28 |
| Figure II.3: | Exemple d'optimisation à deux objectifs. | 33 |
| Figure II.4: | Architecture générale d'un algorithme évolutionnaire. | 40 |
| Figure II.5: | Exemple de codage d'un chromosome. | 41 |
| Figure II.6: | Exemple d'un croisement à un point. | 43 |
| Figure II.7: | Exemple de croisement à deux points. | 43 |
| Figure II.8: | Exemple de croisement uniforme. | 44 |
| Figure II.9: | Exemple de mutation. | 44 |
| Figure II.10: | Exemple de structure d'un algorithme évolutionnaire. | 45 |
| Figure II.11: | Exemple de codage de l'ordonnancement des atterrissages. | 47 |
| Figure II.12: | Exemple de codage pour un problème de voyageur de commerce. | 48 |
| Figure II.13: | Représentation d'un système multi-agent. | 55 |
| Figure III.1: | Fonction de normalisation des critères. | 65 |
| Figure III.2: | Algorithme de Dijkstra classique. | 69 |
| Figure III.3: | Sous-chemin d'un plus court chemin. | 70 |
| Figure III.4: | Exemple de réseau pour le mode k [site RATP]. | 71 |
| Figure III.5: | Algorithme de Dijkstra modifié. | 73 |
| Figure III.6: | Un exemple de graphe. | 73 |
| Figure III.7: | Exemple de réseau nœud de départ dynamique. | 75 |
| Figure III.8: | Exemple de réseau nœud d'arrivée dynamique. | 75 |
| Figure III.9: | Exemple de réseau avec deux nœuds dynamiques. | 76 |
| Figure III.10: | Exemple de réseau avec un nœud dynamique. | 76 |
| Figure III.11: | Exemple d'horaires pour un nœud pendant un jour. | 77 |
| Figure III.12: | Codage d'une solution. | 80 |
| Figure III.13: | Changement des modes de transport. | 81 |

| | | |
|----------------|--|-----|
| Figure III.14: | Changement d'horaires de passage. | 81 |
| Figure III.15: | Croisement à deux points. | 84 |
| Figure III.16: | Croisement à un point | 84 |
| Figure III.17: | Exemple de mutation..... | 85 |
| Figure III.18: | Structure globale de l'algorithme génétique. | 88 |
| Figure III.19: | Hybridation entre algorithme de Dijkstra et un AG..... | 88 |
| Figure III.20: | Architecture multi-zones | 91 |
| Figure III.21: | Architecture du SMA. | 94 |
| Figure III.22: | Comportement de l'agent SYSTEME..... | 96 |
| Figure III.23: | Comportement de l'agent RESPONSABLE ZONE. | 97 |
| Figure III.24: | Diagramme de séquences d'interaction entre agents. | 99 |
| Figure III.25: | Echange des messages entre les agents. | 101 |
| Figure III.26: | Format de l'en-tête du message échangé entre les agents..... | 102 |
| Figure III.27: | Format de corps du message échangé entre les agents..... | 102 |
| Figure IV.1: | Exemple de l'interface utilisateur. | 107 |
| Figure IV.2: | Comparaison entre AG et ACF pour 4 scénarii de perturbation. | 111 |
| Figure IV.3: | Probabilité de trouver une solution en cas de perturbation sur les modes. 112 | |
| Figure IV.4: | Probabilité de trouver une solution en cas de perturbation en zone..... | 113 |
| Figure IV.5: | Comparaison de la convergence entre les deux méthodes d'optimisation. 114 | |
| Figure IV.6: | Hiérarchie par niveau du SMAAD réalisé. | 117 |
| Figure IV.7: | Image de « Communicator » et « SiteAgent ». | 117 |
| Figure IV.8: | Interface de l'agent « SYSTEME »..... | 118 |
| Figure IV.9: | Exemple de communication entre deux agents. | 119 |
| Figure IV.10: | Exemple des conteneurs sur la même machine sous JADE..... | 120 |
| Figure IV.11: | Exemple des communications et échanges de messages sous JADE..... | 121 |

Introduction générale

En parallèle avec l'augmentation considérable du nombre de citoyens partout dans le monde, en particulier dans les grandes villes, la croissance de l'usage des automobiles pose des problèmes aigus, tant sociaux qu'économiques et environnementaux. Aux côtés des problèmes d'augmentation des congestions, de dégradation de la qualité de l'air aux échelles locale et régionale, d'augmentation des émissions de gaz à effet de serre, il faut aussi mentionner la ségrégation spatiale qui accompagne la dépendance accrue à l'automobile ainsi que le mouvement d'étalement des territoires urbains. C'est surtout au sein des villes que ces questions, vraisemblablement peu compatibles avec les objectifs d'un développement durable, sont les plus sensibles. De plus, le nombre de modes de transport ne cesse de s'accroître et les enjeux auxquels ils sont confrontés sont donc de taille. Dans ce cadre, les pouvoirs publics et les exploitants de transport cherchent à développer des solutions qui permettent d'assurer une meilleure mobilité, ainsi qu'une croissance plus équilibrée du transport.

Le développement des technologies de l'information (Web, GSM, etc.) a conduit à un développement des systèmes d'information sur les transports et les déplacements. La fonction essentielle d'un « système d'information multimodal » est de fournir à l'utilisateur des transports toute l'information nécessaire à la réalisation de son voyage. Ces informations permettent de réduire l'incertitude des usagers sur les itinéraires et de leur fournir l'information sur les modes de déplacement envisageables, la durée et le coût de ces déplacements selon le mode utilisé. Ce système vise, si c'est possible, à orienter le comportement des usagers au bénéfice d'une utilisation optimale des infrastructures et d'une priorité aux transports collectifs.

Toutefois, pour les autorités publiques, si elle est bien maîtrisée et de bonne qualité, l'information multimodale peut être un outil efficace d'orientation du comportement des usagers au bénéfice du développement des transports collectifs, plus généralement, de la complémentarité la plus optimale entre les modes de transport. Les parcours dans la ville peuvent changer tous les jours, l'occasionnel remplace la routine. La tendance est à une mobilité qui utilise indifféremment les offres de transport individuel ou collectif. Donc, favoriser la multimodalité consiste à améliorer le service offert afin d'aider les individus dans leurs déplacements et à développer la complémentarité entre les modes afin d'optimiser la gestion du système de transport.

Au vu notamment des progrès réalisés à l'étranger dans plusieurs pays, le besoin de développer les services d'information multimodale en France a été souligné dès l'année 2000 par le ministère de l'équipement et l'action fédérative sur l'intermodalité des voyageurs, qui s'est traduite concrètement, à la fin de l'année 2001, par le lancement d'une Plate-forme de Recherche et d'Expérimentale pour le Développement de l'Information Multimodale (PREDIM). Au cours de la thèse, nous avons participé à la préparation, l'étude et la proposition de projets d'innovation dans le dossier I-Trans. Désormais reconnu pôle de compétitivité, le pôle I-Trans devra répondre aux enjeux posés par l'intensification des échanges : l'intermodalité, ou mise en synergie des différents modes de transport, et l'interopérabilité, ou compatibilité des multiples systèmes utilisés. Notre travail est réalisé dans le cadre du projet « VIATIC-MOBILITE », qui est le projet 6 du pôle I-Trans.

En France, les systèmes existants sont en général des sites Web pour chaque exploitant tels que RATP, TRANSPOLE, SNCF, etc. Par exemple, le système multimodal de Marseille Lepilote donne des informations multimodales, sans les associer ensemble dans l'étude d'un déplacement propre à un client. A l'étranger, le système d'information multimodale le plus connu est celui des allemands DELFI qui donne des solutions multimodales sans assurer le suivi des déplacements en cas de perturbation.

L'optimisation des itinéraires repose sur la mise à disposition du voyageur d'une information multimodale de bout en bout. Il s'agit donc de faciliter l'accès à des données très dispersées sur une offre elle-même très variée, appartenant à des acteurs de tailles et d'activités très diverses, ce qui implique la mise en place des fonctions d'annuaire, de gestion des accès, et éventuellement la mise en place des systèmes d'organisation entre les utilisateurs, les distributeurs et les producteurs des données.

En général, une perturbation concerne un ensemble de stations dans une zone ou bien un ensemble de modes de transport. Afin de réagir en cas de perturbation, pour un système d'aide au déplacement, la gestion en temps réel est très importante. Dans ce cas, le système gère la préparation du voyage et l'accompagnement du voyageur tout au long du déplacement. En outre, la mise à jour des informations et la prise en compte des perturbations constituent un enjeu important d'un point de vue opérationnel, car les usagers ont besoin d'informations fiables avant et pendant le voyage. Pour cette raison, un déplacement représente plusieurs cas possibles. En effet, un utilisateur sollicite l'information nécessaire à un déplacement entre deux points, de départ 'D' et d'arrivée 'A'. Mais ces deux points peuvent bouger

dynamiquement dans les cas de perturbations. Autrement dit, l'utilisateur peut changer son point de départ ou bien sa destination avant le début du déplacement. De même, au cours du déplacement, une perturbation éventuelle dans le réseau de transport peut entraîner des changements en un ou plusieurs points situés dans le chemin de déplacement.

Un des grands enjeux de la multimodalité consiste à tirer parti de tous ces développements et de la demande croissante du public en reliant la multitude de services d'information. Une approche plus globale et décentralisée permettra la création des services plus ouverts et plus répandus, capables par exemple de renseigner les usagers sur leur trajet quelle que soit leur destination et le moyen de transport choisi. Cette approche doit notamment permettre une recherche d'itinéraire entre deux endroits sur l'ensemble du territoire sans que l'utilisateur ait à consulter plusieurs sources d'information. D'autre part, elle doit laisser le marché des systèmes d'information et de recherche d'itinéraire ouvert aux systèmes existants, ainsi qu'à des nouveaux partenaires ou fournisseurs d'information.

Le voyageur souhaite accéder, via des supports multiples, à une information fiable, multimodale, personnalisée et intégrant les perturbations. Avant son déplacement, il souhaite connaître les itinéraires, les horaires et les tarifs alors que, pendant son déplacement, il veut être informé des perturbations et des moyens alternatifs mis à sa disposition lorsque l'incident se prolonge.

L'objectif de nos travaux de recherche est de réaliser un système permettant d'aider les voyageurs et de faciliter leur déplacement en mode normal et en mode dégradé de fonctionnement du réseau. Ce système vise par ailleurs à minimiser le temps d'attente des voyageurs, en mode dégradé, dans les pôles d'échanges et à leur assurer, dans la mesure du possible, la continuité des déplacements dans les réseaux multimodaux. Il s'agit donc d'améliorer la qualité du service rendu aux voyageurs et de les maintenir informés.

La réalisation d'un tel système consiste généralement à créer une base unique centralisée contenant l'ensemble des données nécessaires aux déplacements. Cette méthode s'applique très bien dans le cas d'un seul mode de transport, mais devient contraignante quand plusieurs transporteurs ou fournisseurs de services interviennent. Dans ce cas, un certain nombre de problèmes se présentent, notamment, les problèmes d'agrément, les problèmes de responsabilité, les problèmes de rationalité et les problèmes de fiabilité des bases de données.

De nombreuses difficultés de développement de l'information multimodale ont été identifiées, notamment en France par l'INRETS et ITS France [Uster 01]. La plupart sont non techniques. Il y a globalement trois types de raisons pour lesquelles un acteur sollicité pourrait refuser de participer à la mise en place d'un système (et surtout de fournir des données) : soit c'est trop cher, soit cela n'a jamais été fait jusqu'à présent et il pourrait y avoir un risque technique (de non faisabilité), soit cela ne répond pas à sa stratégie institutionnelle ou commerciale (typiquement en matière de diffusion d'information).

L'information multimodale constitue l'un des maillons essentiels de cette problématique. Il s'agit de fournir toute information utile et pertinente sur les différents modes de déplacement (métro, tramway, bus, etc.), afin, d'une part, d'améliorer le confort et l'efficacité des trajets à un niveau individuel, et d'autre part de favoriser l'usage multimodal et raisonné des différents modes de transport à l'échelle collective.

Les acteurs français de l'information multimodale développent des solutions spécifiques sans rendre public leurs algorithmes, ni même souvent les spécifications d'interface, et ne s'appuient pas beaucoup sur les compétences des autres acteurs ni sur les résultats obtenus à l'étranger.

En conclusion, l'information multimodale est un thème actuel de recherche. De nombreuses études s'effectuent de manière à mieux concevoir des systèmes d'information ; certaines portent notamment sur les besoins des utilisateurs. Cependant, il est actuellement difficile de trouver un système fournissant à la fois la possibilité de planifier son itinéraire en tenant compte de tous les modes de transport possibles, des critères fournis par l'utilisateur et accompagnant celui-ci tout au long de son déplacement [Danflous 00].

Le présent document est organisé de la manière suivante :

Dans la première partie, nous étudierons les problèmes de transport en commun et les systèmes existants dans l'aide au déplacement. Ce domaine est en pleine évolution, ce qui explique le nombre de projets en cours de réalisation.

Une grande partie du travail de cette thèse concerne la conception, le développement et la validation de nos approches, qui permettent de donner des solutions optimales, pour un réseau de transport normal ou perturbé. Dans le premier chapitre, nous étudions les problèmes à résoudre, ainsi que les systèmes d'aide au déplacement existants et les projets en cours. En dégagant les avantages et les inconvénients de ces systèmes, pour établir un état de l'art des algorithmes, méthodes et approches existants dans le chapitre deux. Dans la partie suivante,

nous modélisons notre système ainsi que les approches utilisées afin de résoudre les problèmes de déplacement, qui fait l'objet de la première partie du chapitre trois. Après la résolution du problème d'optimisation des déplacements, nous présentons la réalisation du Système Interactif d'Aide au Déplacement Multimodal (SIADM) par un système multi-agent dans la suite de ce chapitre. Le dernier chapitre est consacré à la validation des méthodes d'optimisation et du système multi-agent par des scénarii d'application sur des zones de réseau urbain de l'agglomération lilloise.

Chapitre I

Les problèmes de transport multimodaux centrés autour des services et des clients de transport

I.1 Introduction

La notion de multimodalité dans les réseaux de transport collectif fait l'objet de différentes thématiques de recherche liées à l'aménagement des réseaux, à la gestion conjointe des différents modes de transport, à l'information multimodale et à la planification et la régulation du trafic. La première partie de ce chapitre est consacrée ainsi à cette notion. Les systèmes informatiques dans le domaine du transport sont nombreux. Ces systèmes sont divisés en deux catégories, les systèmes pour les exploitants et les systèmes pour les clients. Nous trouvons des systèmes d'aide à la décision côté régulateur (SAD), des systèmes d'aide à l'exploitation (SAE) pour le côté exploitant et des systèmes d'information (SI) pour le côté client.

Dans ce chapitre, nous présentons les objectifs et l'intérêt des systèmes pour le côté exploitant, dans la première partie. En seconde partie, nous étudions des systèmes existants du côté client en France, ainsi que les systèmes européens. La troisième partie est consacrée aux projets en cours de réalisation pour les systèmes multimodaux d'information, d'aide au déplacement et les projets auxquels nous participons.

I.2 Multimodalité

Pour commencer, il est important de donner des définitions de ce que l'on entend par « multimodalité » et « information multimodale » :

- **La multimodalité** : C'est le recours à plusieurs modes de transport pour satisfaire des besoins de déplacements. Il s'agit donc de l'offre de plusieurs moyens de transport pour un déplacement entre une origine et une destination. Elle constitue donc un constat technique d'une situation sur le terrain.

- ***L'intermodalité*** : C'est un principe d'organisation et d'articulation de l'offre de transport, visant à coordonner plusieurs systèmes modaux par une gestion et un aménagement spécifique des interfaces entre les différents réseaux. Il s'agit donc d'un enchaînement de plusieurs moyens de transport pour un déplacement entre une origine et une destination.
- ***L'information multimodale*** : Elle sert à renseigner tout « utilisateur » d'un quelconque mode de transport (route, fer, air, mer, fleuve, vélo...) sur les modalités et conditions d'un déplacement ou d'un voyage qu'il souhaite effectuer ou qu'il effectue. Cette notion d'utilisateur doit être prise au sens large : il peut s'agir d'un voyageur « utilisateur potentiel » de l'un des modes de transport, mais aussi de tout organisme (public ou privé) intéressé pour ses besoins propres par l'exploitation de cette information (agence de voyage, office de tourisme, agence d'urbanisme, collectivité locale, localisation de voiture, opérateurs télécoms, fournisseurs de service à valeur ajoutée...).
- ***Les services d'information multimodale*** : Ils sont diffusés par différents canaux et supports. Ils fournissent des fonctions aussi bien de calcul d'itinéraires sur le Web ou sur le minitel, que des affichages ou diffusions d'horaires (guichet, Web, téléphone, bornes, panneaux lumineux...), que des conditions de circulation transmises par radio ou sur le Web. Ils diffusent des informations sur les réservations, sur les tarifs, sur la disponibilité de places de parking, sur des temps d'attente, et ce aussi bien « en théorie » qu'en temps réel.

I.3 Problème de recherche en transport pour aider à la régulation

Un réseau de transport multimodal est caractérisé, dans le cas des systèmes de transport des personnes et non des marchandises, par la présence simultanée de différents modes de transport tels que les bus, les métros et les tramways. La liaison entre ces modes est établie au niveau de pôles d'échange ou de nœuds de correspondance, où il y a des échanges de voyageurs entre deux ou plusieurs véhicules à différents horaires de la journée.

Dans un réseau multimodal, la qualité de l'offre est principalement perçue par les voyageurs lors des correspondances entre les différentes lignes d'un même mode de transport ou entre les différents modes.

Plusieurs questions liées à la nature des perturbations et aux manœuvres de régulation peuvent se poser lors de la gestion en temps réel du trafic d'un réseau de transport urbain. Comment et quand peut-on détecter une perturbation? Quel est son impact sur le réseau? Comment peut-

on mesurer l'importance de l'incident? Nécessite-t-il une régulation? Si oui, existe-t-il des mesures qui lui sont adaptées? Quelle mesure faut-il choisir? Selon quel(s) critère(s) faut-il faire ce choix? Et enfin, comment peut-on évaluer son impact pour pouvoir décider?

Le régulateur doit constamment faire face à toutes ces questions en temps réel. Tout en étant submergé d'informations provenant du Système d'Aide à l'Exploitation (SAE), il doit prendre, en un temps limité, des décisions immédiates pour traiter les incidents qui peuvent aussi apparaître simultanément [Fayech 02a]. Des travaux de recherche dans notre équipe sont en cours qui traitent ces questions dans le cadre des projets SART et SART-suite, soutenus par la Région et le GRRT.

I.3.1 Planification du trafic

Par rapport au cas monomodal, l'affectation et la planification du trafic dans un réseau multimodal sont plus difficiles, étant donnée la complexité des déplacements. En effet, ce processus réalise l'ordonnancement des véhicules, afin de leur affecter des courses et aussi l'ordonnancement du personnel, afin de leur affecter des services. Ainsi, cette problématique de la planification fait l'objet de diverses études, qui relèvent du domaine de l'ordonnancement et des tournées de véhicules.

L'objectif de ce processus est d'établir une offre prévisionnelle de transport qui s'ajusterait à la demande. Le processus classique est illustré par la figure I.1 [Huissman 01]. La configuration initiale des lignes du réseau ainsi que leurs fréquences sont fixées comme données d'entrée. Les durées des trajets entre les différents points du réseau sont aussi considérées comme connues. Des tableaux d'horaires sont alors établis et résultent en des « courses », qui correspondent aux temps et lieux d'arrivée et de départ des véhicules.

Ensuite, l'affectation de ces courses aux véhicules consiste en une tâche d'ordonnancement statique, à l'issue de laquelle on a des « blocs de véhicules » (ensemble des courses affectées à chaque véhicule). Par ailleurs, l'ordonnancement du personnel est aussi réalisé lors de la planification pour fixer les fiches horaires journalières de service des conducteurs et des régulateurs, ainsi que les différentes relèves.

Quelques outils informatiques traitant ce type de problèmes sont utilisés par les compagnies de transport par les bus, notamment HASTUS, qui a été développé dans les années 80 [Rousseau 85].

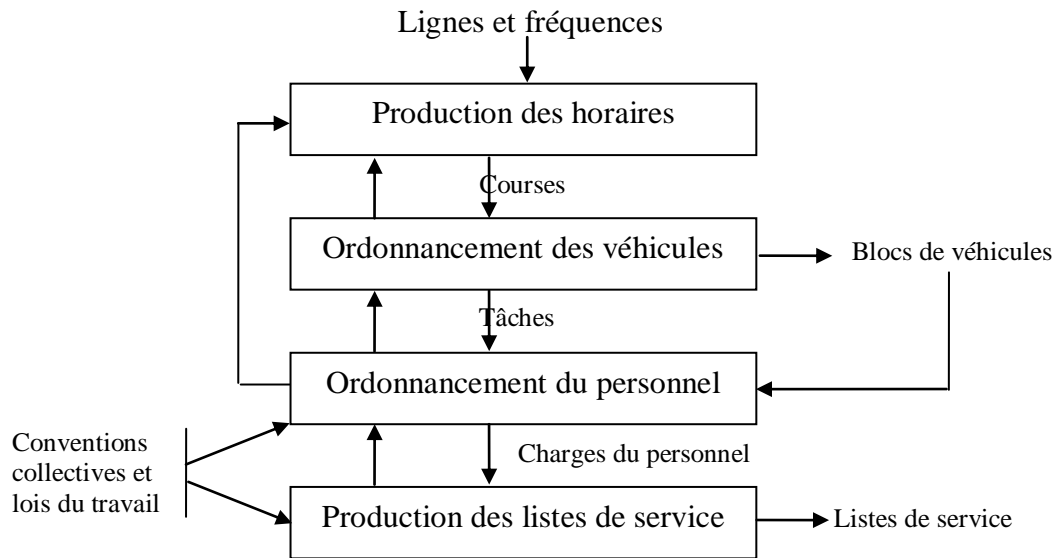


Figure I.1: Processus classique de planification.

I.3.2 Régulation du trafic

Le processus de planification est effectué en général une fois par an et ne s'appuie que sur des estimations de la demande et des durées des trajets entre les différents arrêts du réseau. Par conséquent, la moindre variation, en temps réel, de ces paramètres, est susceptible de contraindre les véhicules à des retards, ce qui provoque des attentes des passagers encore plus longues. Ainsi, la gestion du trafic dans un réseau de transport ne s'arrête pas à l'étape de la planification, qui se fait en avance. Un processus de régulation est réalisé en temps réel afin de gérer les écarts entre les horaires planifiés ou théoriques et les horaires réels de passage, ce qui constitue un autre axe de recherche dans le domaine du transport.

La régulation du trafic d'un réseau de transport collectif peut être définie comme étant une adéquation maximale des horaires théoriques annoncés aux voyageurs aux conditions réelles d'exploitation. En effet, certains incidents peuvent apparaître, obligeant les clients à attendre plus longtemps aux stations ou à bord des véhicules, ce qui dégrade la qualité du service offert aux usagers. Les régulateurs doivent donc prendre des actions de régulation immédiates pour résorber ces perturbations, qui peuvent aussi apparaître simultanément. Pour ce faire, ils doivent réaliser des tâches différentes et difficiles : l'acquisition de l'information, l'analyse de l'impact de chaque perturbation sur l'état actuel du réseau, l'élaboration des décisions réalisables, l'évaluation de celles-ci et la mise en œuvre de l'action retenue. Ces différentes tâches qu'effectuent les régulateurs montrent la complexité d'un poste de régulation d'un réseau de transport. En outre, la nature complexe des situations décisionnelles et des enjeux

liés aux décisions prises prouve combien il peut être précieux, pour le régulateur, de bénéficier du support d'un outil pouvant modéliser et formaliser les critères et l'aider à différencier les décisions candidates, en lui proposant un classement de celles-ci, selon un ordre reflétant les préférences explicitées. Ce travail représente un axe de recherche mené par un chercheur M.M. Ould Sidi dans notre équipe. L'objectif du système d'aide à la décision que M.M. Ould Sidi développe consiste, à partir des informations fournies par le SAE (Système d'Aide à l'Exploitation), à élaborer, à évaluer et à proposer des solutions efficaces aux régulateurs, en tenant compte de leurs préférences. Ainsi, la tâche du régulateur sera allégée, ce qui améliorera la qualité de régulation. Dans son travail, M.M. Ould Sidi traite essentiellement les deux problèmes de construction et d'évaluation des décisions [Ould Sidi_a 05]. Il utilise une méthode d'optimisation utilisant la théorie de la logique floue et les algorithmes évolutionnaires pour optimiser les solutions de régulation et une méthode d'évaluation des décisions basée sur l'application de l'intégrale de Choquet, comme opérateur d'agrégation [Ould Sidi_b 05], pour évaluer les décisions.

B. Fayech [Fayech 02] a présenté dans sa thèse, au sein de notre équipe, un Système d'Aide à la Décision pour la régulation, qui repose sur :

- une modélisation hybride basée sur les graphes pour la représentation des itinéraires, et sur les agents, pour la représentation spatiale et temporelle du réseau ;
- une approche Agent pour le diagnostic et la régulation, en cas de situations familières, et une approche évolutionnaire pour la régulation temporelle ou spatio-temporelle, en cas de situations non familières.

I.4 Problématique de recherche en transport pour aider les clients

Face à l'augmentation du nombre et de la complexité des déplacements, les usagers souhaitent disposer d'une information fiable sur l'ensemble des modes de transport qui sont mis à leur disposition. Dans le même objectif, les exploitants, ainsi que les autorités, visent à améliorer les services à proposer aux clients des transports en commun. Dans ce sens, plusieurs projets ont vu le jour. Certains sont réalisés par les exploitants, d'autres par les régions ou bien par le ministère de transport public. Ces projets cherchent à étudier la faisabilité et la réalisation de systèmes qui améliorent les services offertes aux clients. Ces systèmes visent à réduire l'incertitude des voyageurs sur les itinéraires, les modes de déplacement envisageables, la durée et le coût de ces déplacements, et si possible à orienter le comportement des voyageurs

au bénéfice d'une utilisation optimale des infrastructures et d'une priorité aux transports collectifs [Danflous, 01] [Meskine, 01].

Actuellement le nombre des utilisateurs de transport en commun est en croissance continue [Bulletin 05] (Figures : I.2, I.3, et I.4). Cette croissance suit l'objectif des autorités et des communautés organisationnelles de transport. Ces organismes visent à améliorer les services autour des transports en commun, afin d'encourager les citoyens à remplacer leurs voitures privées par les modes de transport en commun.

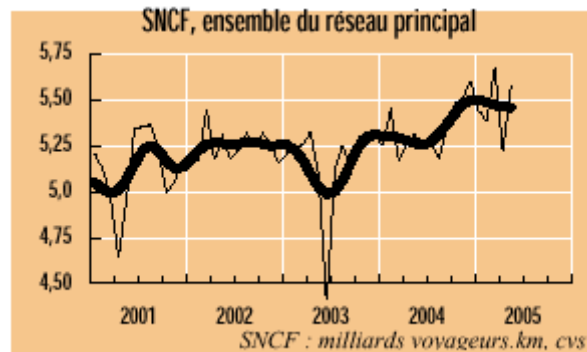


Figure I.2: Statistique des voyageurs sur l'ensemble du réseau SNCF [Bulletin 05].

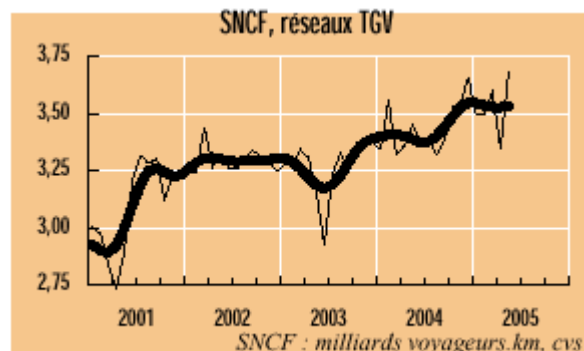


Figure I.3: Statistique des voyageurs sur le réseau de TGV SNCF [Bulletin 05].

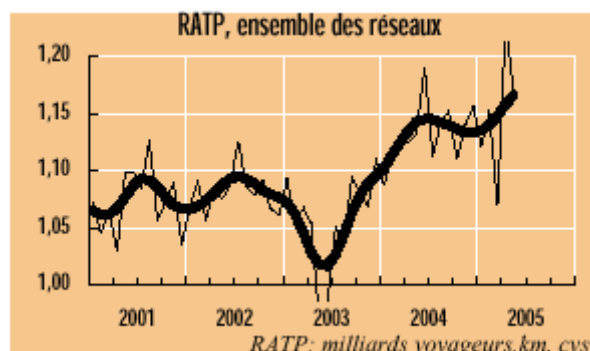


Figure I.4: Statistique des voyageurs sur l'ensemble du réseau RATP [Bulletin 05].

Toutefois, dans un déplacement multimodal, l'utilisateur est simultanément client de plusieurs opérateurs, puisqu'il empruntera successivement, dans son déplacement, plusieurs modes de transport. Il peut être utile qu'il connaisse au plus tôt l'existence d'une perturbation affectant

son trajet, afin de prendre les dispositions qui conviennent (une correspondance avant, par exemple). C'est donc dans ce contexte d'assurance de qualité de service en continuité de bout en bout (et non simplement pour une prestation isolée) que les opérateurs ont un besoin d'échanges d'informations d'un réseau à l'autre.

I.5 Les applications existantes

Actuellement, plusieurs régions et exploitants des transports en commun (TC) cherchent à développer l'information multimodale. Il existe différents supports pour diffuser l'information : bornes interactives dans les lieux publics, panneaux d'affichage électroniques en stations, PC ou terminaux reliés à Internet et le minitel. Depuis l'année 1997, des sites web d'informations TC commencent à apparaître sur Internet : SNCF et RATP, TCL à Lyon, TRANSPOLE à Lille, Lepilote à Marseille... Toutefois ces sites fournissent de l'information sur un seul réseau (à l'exception du site d'Ile-de-France, qui donne de l'information concernant la RATP et la SNCF).

a. lePILOTE

L'association lePilote a pour vocation de fournir l'information multimodale sur l'agglomération marseillaise et les Bouches-du-Rhône. Côté TC, le site web www.lepilote.com intègre l'information du trafic à Marseille et sur les autoroutes urbaines gérées par la direction départementale de l'équipement (DDE), l'information sur les horaires des bus départementaux (CarTreize), des TER (SNCF), des réseaux de Marseille, Berre/Vitrolles, Aubagne, Martigues, Gardanne, La Ciotat, ainsi que les réseaux nouvellement gérés par la Communauté Urbaine Marseille Provence Métropole : Allauch, la Côte Bleue (Carry, Chateaufort, Ensues, Sausset et le Rove), la Marcouline (Cassis, Carnoux-en-Provence et Roquefort-la-Bédoule). Le site web a été amélioré en 2004, par la mise en place d'un service vocal sur les perturbations VP/TC (véhicules particuliers et transport en commun) de l'agglomération.

Cette catégorie regroupe plusieurs exploitants d'un même secteur. Ceci permet à l'utilisateur d'obtenir un véritable plan de voyage, sans devoir vérifier auprès des autres exploitants l'existence d'un autre trajet. Le même système calcule les différentes combinaisons. Ce site a été développé dans le cadre du projet européen CONTACT, dont l'objectif était de permettre aux voyageurs urbains d'avoir l'accès à l'information multimodale de mobilité, à partir de leurs domiciles, leurs bureaux ou dans la rue, en utilisant une carte permettant d'obtenir et de payer des services liés à leurs déplacements.

b. SAE de Metz

Le projet Metz interactif vise à développer de nouveaux services aux habitants de l'agglomération messine. Un moyen d'avertir l'utilisateur d'un retard éventuel est celui qui a été expérimenté par la ville de Metz : le boîtier InfoBus (Figure I.5). Il s'agit d'un terminal de radio messagerie permettant d'informer son possesseur, en temps réel, sur le temps d'attente et les éventuelles perturbations sur le réseau de bus [Gen 99].

D'après ce mode de fonctionnement, on peut constater qu'il existe deux types de bases de données :

- Les bases de données « statiques », qui représentent une planification des conditions de circulation, et sont définies en général un an à l'avance (ce sont typiquement les horaires de passage que l'on trouve dans les arrêts). Pendant leur durée de validité, ces bases ne varient que pour tenir compte d'événements non prévisibles, comme des manifestations, des grèves, des travaux... Ces mises à jour se font alors plutôt par saisie manuelle.
- Les bases de données « dynamiques », qui sont remises à jour en temps réel, et ne concernent que la journée en cours. Les mises à jour sont effectuées automatiquement par les SAE, évoqués précédemment.

c. DELFI

Cette partie traite rapidement la manière dont d'autres pays européens ont envisagé, la mise en place d'une information de niveau national sur l'offre de transports collectifs. Le Système d'Information Multimodale (SIM) le plus connu est celui des allemands DELFI. Dans cette partie, nous précisons le contour de l'information multimodale et sa déclinaison en Allemagne, puis nous présenterons rapidement les divers aspects du projet DELFI, sans entrer dans les détails, ce qui pourra faire l'objet d'études complémentaires.

Depuis plusieurs années, l'information multimodale fait donc l'objet de réflexions et d'expérimentations. L'Allemagne semble parmi les plus avancés dans ce domaine. En se basant sur une étude réalisée par le CERTU, nous présenterons l'expérience de ce pays et plus particulièrement le projet DELFI.

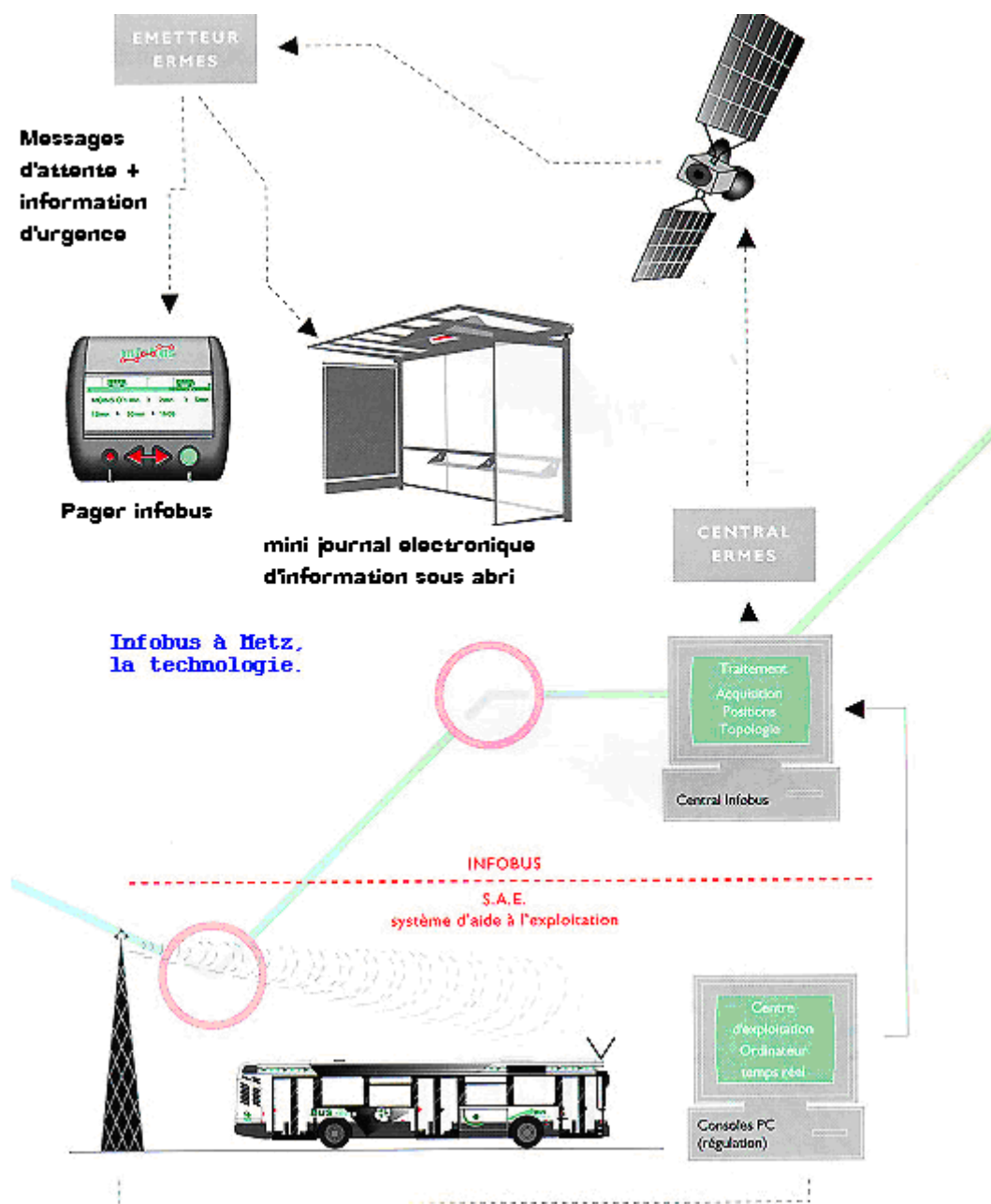


Figure I.5: Architecture d'Infobus à Metz.

Plusieurs concepteurs de SIM ont participé au développement de DELFI. Ces concepteurs ont déjà développé des systèmes pour des exploitants ou des régions tels que HAFAS, EFA ... Les systèmes peuvent informer sur un ou plusieurs modes de transport en commun. Dans tous les cas, ils ne le font que sur un périmètre géographique bien déterminé. Notamment, ils ne permettent pas une recherche d'itinéraire ou d'horaire en continu de porte à porte sur des parcours longue distance. En effet, si le serveur du réseau ferré peut, par exemple, informer sur les itinéraires entre deux villes, le voyageur devra interroger deux autres serveurs pour le calcul de ses itinéraires dans les villes de départ et d'arrivée.

C'est de cette limitation géographique, de la diversité des systèmes existants, ainsi que de l'organisation complexe des transports en Allemagne qu'est né le projet DELFI. Il a pour objectif de mettre en place un système d'information multimodale étendu à l'ensemble du territoire allemand. Il doit notamment permettre une recherche d'itinéraire de porte à porte sur l'ensemble du territoire sans que l'utilisateur ait à consulter plusieurs sources d'information. D'autre part, et c'est une deuxième condition au projet, celui-ci doit laisser le marché des systèmes d'information et de recherche d'itinéraires ouvert aux systèmes existants, ainsi qu'à de nouveaux partenaires, fournisseurs de systèmes ou fournisseurs d'information.

C'est ainsi qu'est née l'idée d'un système à architecture répartie ou distribuée. En proposant non pas de centraliser les données, mais de mettre en réseau les différentes sources d'information, le projet préservait la multitude des systèmes et fournisseurs d'information existants. D'un point de vue organisationnel, chaque fournisseur conserve son autonomie et la compétition est maintenue à la fois entre fournisseurs de systèmes d'information et fournisseurs de services.

Comme nous l'avons vu, le principe de DELFI est d'ajouter à des systèmes d'information existants une « sur-couche » logicielle et une interface, permettant à ces systèmes de coopérer à travers un réseau. Pour ce faire, DELFI utilise l'architecture CORBA. Sans entrer dans les détails, on dira que, contrairement à l'architecture traditionnelle client-serveur, CORBA (*Common Object Request Broker Architecture*) fonctionne dans un modèle à trois niveaux constitué de clients, de serveurs d'applications et de serveur de données.

Comme système d'information multimodal sur les horaires et de calcul d'itinéraires à l'échelon national, DELFI ne couvre qu'une partie de l'information multimodale, c'est-à-dire l'information "avant le voyage". Alors que l'information qui est nécessaire "pendant" la réalisation du déplacement est absente du projet.

Alors que DELFI est entré dans sa phase opérationnelle, cinq autres grands projets mènent des recherches dans le domaine de la mobilité urbaine et de la télématique appliquée aux transports. Parmi ceux-ci, les projets Wayflow à Francfort et Mobilist à Stuttgart ont tous les deux un volet d'information multimodale qui intègre notamment les déplacements en voiture particulière.

d. RATP

La RATP donne donc la priorité à son projet COPILOT qui est maintenant en phase de réalisation. Celui-ci répondra en grande partie aux objectifs proposés par le projet PREDIM.

L'information sur les perturbations des lignes de bus, métro et RER sera hiérarchisée en fonction de son importance, et stockée dans une base de données à disposition des exploitants. COPILOT assurera la mise en forme et la diffusion des messages au public, en fonction de standards préétablis sur les différents médias internes et externes.

D'autre part, il a été convenu de maintenir un échange régulier d'informations sur les études et réalisations menées par la RATP et le réseau de transport marseillais (Lepilote) sur ce sujet.

I.6 Les applications en projet

En ce qui concerne l'information multimodale, on ne peut pas citer des réalisations mais des projets en cours. Le plus important est celui de l'agglomération de Montpellier qui devrait associer la plupart des acteurs du transport dans la région. Ce système d'information aura pour objectif de favoriser l'intermodalité et une utilisation plus rationnelle de la voirie, grâce à une meilleure information des usagers, notamment en temps réel.

Au niveau départemental, on mentionnera les projets du syndicat mixte Hérault-transport, à savoir, dans un premier temps, la mise en place d'un site web d'information permettant la recherche d'itinéraire. Et, à l'horizon 2007, la création d'une centrale d'information.

Dans tous les cas, il manque encore sans doute une approche plus « publique » et plus « normalisée », facilitant l'intégration de ces sources d'information et permettant aux utilisateurs de s'y retrouver plus facilement.

Pour ce faire, il est nécessaire de normaliser les échanges de données concernant l'information aux voyageurs entre exploitants et Autorités Organisatrices de transport.

Le modèle de TRIDENT est un modèle d'échange de l'information pour les véhicules particuliers (VP) et le transport en commun (TC). Ces informations peuvent être théoriques ou prendre en compte des situations perturbées.

Après examen des travaux en cours au niveau européen, il est apparu que les spécifications d'échange élaborées dans le cadre du projet européen TRIDENT étaient la meilleure base de travail sur laquelle il faut s'appuyer pour la normalisation.

Ces spécifications d'échange couvrent l'information VP et TC, théorique et en situation perturbée ; elles s'appuient sur :

- un modèle d'échange entre fournisseur et client, avec possibilité d'abonnement ;

- un modèle de données orienté objets décrit en UML, reprenant pour ce qui concerne les transports publics, le modèle conceptuel de données Transmodel V4.1, normalisé au niveau européen ;
- un format d'échange sous forme des schémas XSD.

Les spécifications TRIDENT ont été implémentées dans plusieurs pays européens. En France, ils ont été mis en œuvre par la RATP pour son système d'information sur les situations perturbées SIPRE ; et également pour les échanges d'information au niveau francilien, en cours, entre les systèmes de la RATP et de l'AMIVIF.

a. CHOUETTE

CHOUETTE signifie « Création d'Horaires avec un Outil d'Echange de données TC selon le format TRIDENT Européen ». L'outil se limite dans un 1^{er} temps aux échanges d'information concernant la description « statique » de l'offre TC (réseau, horaires), et ne couvre donc pas les messages de perturbation en « temps réel ».

Les objectifs de CHOUETTE consistent à permettre :

- à tous les acteurs qui le souhaitent (industriels, exploitants, autorités organisatrices, bureaux d'études) de tester les spécifications TRIDENT sur des données réelles, et éventuellement de faire des propositions d'amélioration qui pourront être prises en compte pour la normalisation de ce format d'échange ;
- à des « petits exploitants », de transformer leurs fiches horaires au format TRIDENT, et éventuellement de développer automatiquement un mini-site Web comprenant la description des lignes et des heures de passage ;
- à des acteurs responsables de réseaux plus importants, et mieux outillés, de télécharger le logiciel eux-mêmes (exécutable et codes sources), afin de l'analyser et éventuellement de l'intégrer à leur chaîne d'information voyageurs.

b. PREDIM

Créée au sein du programme français de recherche et d'innovation dans les transports terrestres (PREDIT), la PREDIM constitue un dispositif national d'appui mutualisé au bénéfice des autorités de transport et, de manière générale, de tous les acteurs désireux d'améliorer, par un dispositif d'information adéquat, la complémentarité des différents modes de transport et de déplacement, tant individuels que collectifs.

Cette complémentarité repose sur la capacité des usagers à disposer d'une information de qualité, leur permettant d'orienter leur choix de transport vers le mode de déplacement le mieux adapté à leurs besoins. Elle offre ainsi des alternatives au déplacement automobile individuel.

Ces nouveaux services d'information multimodale supposent de croiser les données de plusieurs réseaux et services, de les organiser, puis de diffuser l'information par différents médias. Ils impliquent la bonne coordination de plusieurs acteurs (État, collectivités territoriales, opérateurs de transport, industriels, sociétés de services, chercheurs ...).

La PREDIM permet :

- de capitaliser et échanger les informations et les expériences en France et à l'étranger,
- de proposer des concepts de services d'information multimodale concrètement applicables,
- d'expérimenter de nouveaux services, produits ou composants, jusqu'au développement d'un prototype diffusable industriellement,
- d'évaluer des systèmes ou services d'information multimodale, issus de projets proposés dans le cadre de la PREDIM ou non,
- de développer la réflexion des différents intervenants de l'information multimodale et favoriser l'émergence de nouveaux acteurs et régulateurs.

Elle s'adresse donc à la fois :

- aux décideurs, en leur permettant de contribuer à la mise en place de nouvelles organisations et modèles économiques, tant au niveau local que national,
- aux porteurs de projets, en leur offrant des possibilités de support et un réseau d'échange d'informations,
- aux techniciens impliqués dans des projets, à la recherche d'informations et d'échanges techniques avec leurs homologues, et souhaitant faire connaître leurs idées et leurs réalisations,
- aux chercheurs souhaitant utiliser des outils et méthodologies multidisciplinaires pour la mise en œuvre et l'évaluation des services,

c. Pôle de compétitivité I-Trans (le projet VIATIC)

Le projet VIATIC-MOBILITE a pour objectif de promouvoir les transports collectifs, par la mise en œuvre opérationnelle des services d'aide à la mobilité basés sur les nouvelles

technologies d'information et de communication. Ces services ont pour objectif de proposer de l'information d'accompagnement au cours des déplacements.

I.7 Les limites des systèmes existants

En conclusion, l'information multimodale est un thème actuel et innovant de la recherche. De nombreuses études s'effectuent de manière à mieux concevoir des systèmes d'information; certaines portent notamment sur les besoins des utilisateurs. Cependant, il est actuellement difficile de trouver un système fournissant à la fois la possibilité de planifier son itinéraire en tenant compte de tous les modes de transport possibles et des critères fournis par l'utilisateur, et accompagnant l'utilisateur tout au long de son déplacement. Les systèmes existants sont en général mono-exploitant (mono-mode) ou, dans les meilleurs des cas, multimodaux, mais ils n'assurent pas l'accompagnement pendant les déplacements.

I.8 Conclusion

Face à la nécessité d'améliorer les services annexes au service de transport en commun, afin d'encourager les citoyens à les choisir en priorité par rapport au véhicule privé, les pouvoirs publics et les exploitants de transport public ont lancé plusieurs projets concernant les systèmes d'information et d'aide au déplacement multimodal. L'une des fonctionnalités fondamentale de ces systèmes est l'optimisation multicritère des chemins multimodaux. Nous étudierons la complexité de ces problèmes d'optimisation, ainsi que les algorithmes nécessaires pour les résoudre, dans les chapitres suivants.

Chapitre II

Etat de l'art en modélisation et optimisation dans le domaine du transport

II.1 Introduction

L'objectif d'un système d'aide au déplacement est d'optimiser et planifier un déplacement entre deux points. Ce service doit être maintenu dans le cas d'utilisation d'un réseau de transport en commun en mode normal ou perturbé. Afin de bien étudier les problèmes à résoudre, ainsi que les méthodes à utiliser, nous avons besoin d'une bonne modélisation du réseau de transport. Un tel réseau est représenté par un graphe orienté. Les nœuds du graphe modélisent les stations et les arcs modélisent les tracés suivis par chaque mode entre deux stations. La théorie des graphes est néanmoins une théorie autonome, dans la mesure où elle a su développer des outils et concepts, qui lui sont propres, pour résoudre les problèmes qui se présentent. Ces instruments fondamentaux ont de multiples applications dans des domaines très divers.

Ce chapitre est organisé en trois parties. La première partie est consacrée à la modélisation des réseaux de transport en commun par des graphes, afin de montrer la complexité du problème d'aide au déplacement. La deuxième partie du chapitre présente un état de l'art sur les problèmes *NP-difficiles* et les méthodes d'optimisation existantes pour les résoudre. Dans cette partie, nous étudions plus en détail les algorithmes évolutionnaires et leur mise en œuvre par les algorithmes génétiques, que nous utiliserons dans le chapitre suivant. Nous consacrons la troisième partie de ce chapitre à la caractérisation des agents et des SMA. En passant par les méthodologies de conception des SMA, nous notons les plus importantes applications des approches orientées agents, notamment dans le domaine du transport.

II.2 Modélisation graphique

Dans la première étape de la modélisation, nous considérons les données liées au problème d'optimisation dans un réseau de transport en question, les arrêts, les lignes du réseau, les

véhicules et les distances inter-arrêts, etc. Afin d'illustrer les différents déplacements, les réseaux de transport sont souvent modélisés par des graphes. Les représentations par des graphes diffèrent cependant d'un modèle à un autre. Elles s'avèrent indispensables dans l'appréhension du comportement des entités formant le réseau, et donc dans la prise de décision en cas de perturbation. Nous présentons ainsi dans la première partie de ce chapitre différentes modélisations, basées sur les graphes, qui ont été attribuées aux réseaux de transport.

Les graphes sont actuellement l'outil privilégié pour modéliser des ensembles structurés complexes. Leurs applications sont très nombreuses : modélisation de l'évolution d'un système dans le temps (en économie, en automatique), réseaux divers (électriques, routiers, ou d'adduction d'eau), décomposition en tâches d'un projet (en informatique, dans le bâtiment et les travaux publics), liens entre informations dans les bases de données, etc... [Lacomme 03].

Nous présentons quelques définitions qui vont être utiles par la suite.

Définition 1 :

Un graphe $G=(X,U)$ est déterminé par les données suivantes :

- Un ensemble X dont les éléments sont appelés sommets ou nœuds. Si $N=\text{card}(X)$ est le nombre de sommets (de nœuds), on dit que le graphe G est d'ordre N .
- Un ensemble U dont les éléments sont des couples ordonnés de sommets appelés arcs. Si $u=(i,j)$ est un arc de G . i est l'extrémité initiale de u et j est l'extrémité terminale de u . on notera $\text{card}(U)=M$.

Définition 2 :

Considérons un graphe orienté valué $G=(X,U,W)$. X désigne un ensemble de N sommets (ou nœuds) et U un ensemble de M arcs. $W(i,j)$, aussi noté W_{ij} , est la valuation (aussi appelée poids ou coût) de l'arc (i,j) , par exemple une distance, un coût de transport, ou un temps de parcours. Dans la figure suivante (Figure II.1) l'ensemble X est égal $\{A,B,C,D,E,F,G\}$.

Définition 3 :

Un multigraphe est un couple $G=(S,A)$, dans lequel S est un ensemble de sommets, et A une famille d'arcs $A=(a_1,a_2,\dots,a_m)$ [Lévy 94]. Cette définition permet de traiter des graphes dont plusieurs arcs auraient la même origine et la même extrémité, d'où le nom de multigraphes.

Trois types de données sont nécessaires pour décrire l'offre de transport : le tracé des lignes, les coordonnées des points d'arrêt et les horaires théoriques de passage. Les stations sont appelées nœuds et les trajets parcourus par un moyen de transport entre deux stations successives sont appelés arcs. A ces arcs peuvent être associés des poids ; on parle alors d'arcs pondérés. Pour simplifier, nous mettons, dans la figure suivante, des pondérations simples (une seule valeur). Alors que, dans un réseau de transport, ces paramètres sont nombreux. Sur un arc on peut noter le temps de parcours, le prix, etc., bien que les deux premières données soient faciles à représenter dans les graphes, les horaires de passage eux sont plus difficiles.

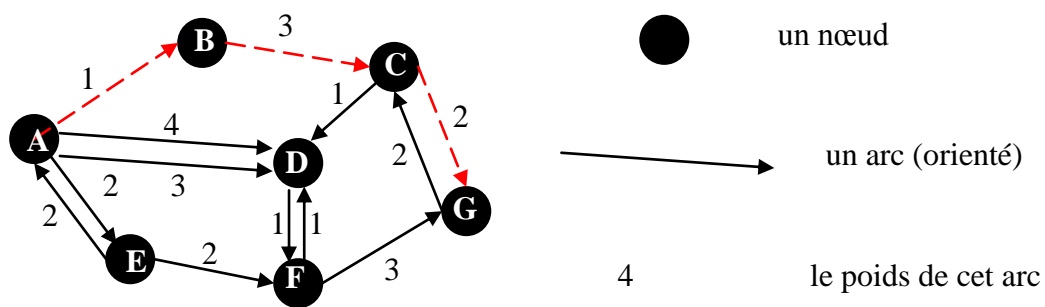


Figure II.1: Exemple de graphe (en pointillé, le plus court chemin entre A et G).

Dans notre cas, chaque station (de métro, bus, train, ...) est un nœud, et à chaque fois qu'un véhicule de transport en commun va d'un nœud à un autre, on dit qu'il crée un arc entre les deux nœuds. Les arcs sont donc très nombreux dans le cas de transports urbains. La densité des graphes n'est pas la seule difficulté devant les algorithmes de recherche de plus court chemin. Il existe en outre des moyens de transport, comme le métro, qui n'ont pas d'heure de départ fixe mais simplement une fréquence de passage qui varie en fonction de l'heure de la journée. Or, dans un itinéraire, les arcs doivent s'enchaîner correctement, c'est-à-dire qu'il faut arriver en un nœud, avant de pouvoir en repartir. Il nous faut donc connaître systématiquement les heures de départ de chaque arc. Nous avons donc besoin d'une fonction qui calcule une heure de départ de tels moyens de transport, étant donnée l'heure d'arrivée de l'arc précédent. Le problème de changement d'horaire périodique est accompagné par le changement d'autres paramètres, tels que le prix. Pour la pondération, chaque paramètre est calculé par une fonction qui permet d'affecter les poids des arcs en fonction des paramètres de l'utilisateur, l'importance du confort, du temps et du prix du trajet. Nous expliquerons ces fonctions dans le chapitre suivant.

A chaque mode de transport on associe un réseau qui est représenté par un graphe comme le montre la Figure II.2. Dans ce graphe, nous ne pouvons pas mettre tous les paramètres (tels que les heures de passage par chaque nœud), d'où la nécessité d'ajouter des outils qui gèrent les informations de base pour le calcul de ces paramètres. Entre deux ou plusieurs modes se trouvent des nœuds d'intersections, appelés nœuds de correspondance. Par exemple, les nœuds 1,2 et 3 entre les modes M_1 , M_2 et le mode M_3 (Figure II.2).

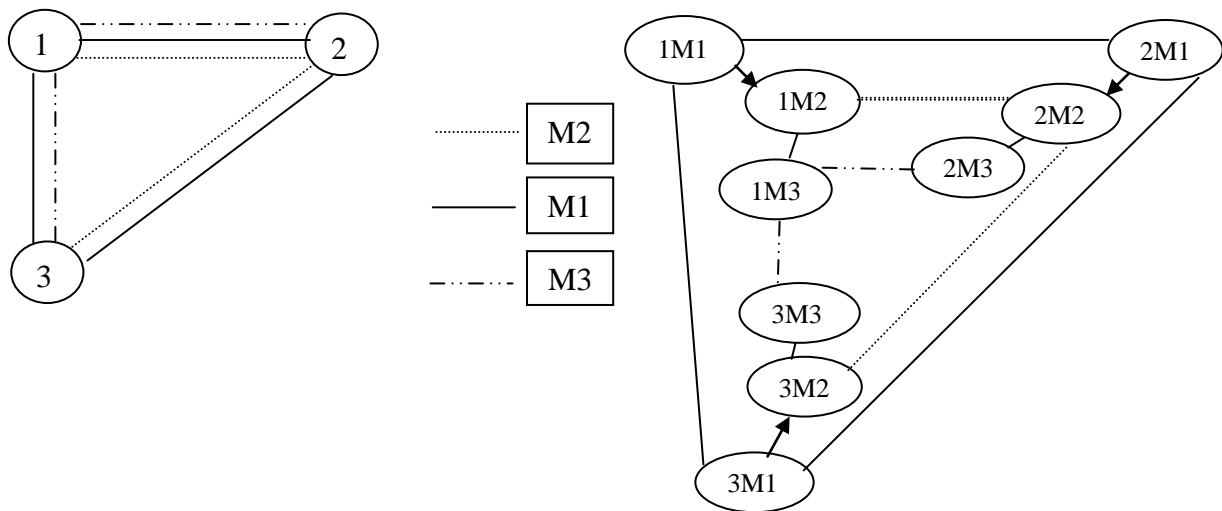


Figure II.2: Exemple de réseau de transport multimodal.

Le graphe du réseau multimodal résultant de la superposition de ces graphes est un multigraphe. Ce dernier présente de nouvelles caractéristiques. Ces caractéristiques sont transformées en contraintes pendant les résolutions du problème de plus court chemin. Parmi ces contraintes, on trouve la gestion de la multimodalité dans les nœuds de correspondance. L'utilisateur de transport en commun souhaite minimiser les temps d'attente en station et surtout assurer les correspondances.

II.3 Introduction du problème NP-complet

Les méthodes d'optimisation des problèmes mono-objectif ou multiobjectif sont très variées. Certains des problèmes d'optimisation peuvent être résolus par des méthodes exactes simples, qui permettent de trouver à coup sûr la (ou les) meilleure(s) solution(s) ou solution(s) optimale(s). Mais la plupart des problèmes étudiés en optimisation appartiennent à la classe des problèmes NP-difficiles [Basseur 05]. Cette classe rassemble des problèmes pour lesquels "on ne connaît pas d'algorithme exact rapide. Dont la résolution exacte n'est pas possible en

un temps de calcul proportionnel à N^n , où N désigne le nombre de paramètres inconnus du problème, et n est un entier” [Collette 02]. Le problème de voyageur de commerce (*Traveling Salesman Problem*), est le plus connu dans la classe des problèmes NP-complets. Car son énoncé est très simple (minimiser la longueur d’un circuit hamiltonien dans un graphe pondéré) [Dréo 03]. Il existe une quantité innombrable de problèmes de ce type dans la littérature. D’une manière générale, la résolution de problèmes multiobjectif est très complexe, et combine parfois plusieurs objectifs, dont chacun est NP-difficile.

Les problèmes les plus difficiles de la classe NP définissent la classe des problèmes *NP-complets* : un problème de NP est *NP-complet* s’il est au moins aussi difficile à résoudre que n’importe quel autre problème de NP, i.e., si n’importe quel autre problème de NP peut être transformé en ce problème par une procédure polynomiale [Solnon 05].

Deux grandes classes de méthodes générales existent pour résoudre des problèmes d’optimisation combinatoire NP-difficiles : les heuristiques et les méthodes arborescentes. Nous étudierons ces méthodes afin de choisir le mécanisme adéquat pour traiter les problèmes de déplacement difficiles.

Plusieurs problèmes issus de domaines divers consistent à maximiser (ou minimiser) une fonction linéaire, sous des contraintes linéaires avec des variables bivalentes. Ces problèmes, dits d’optimisation combinatoire, sont généralement NP-difficiles.

II.4 Optimisation combinatoire

L’optimisation combinatoire est une discipline combinant diverses techniques des mathématiques discrètes et de l’informatique afin de résoudre des problèmes d’optimisation dont la structure sous-jacente est discrète (généralement un graphe).

Un problème d’optimisation combinatoire est un problème qui consiste à maximiser (ou minimiser) une certaine fonction sur un ensemble fini d’éléments. Un tel problème peut être présenté de la manière suivante : étant donné un ensemble fini $E = \{e_1, \dots, e_n\}$, une famille F de sous-ensembles de E et un système de poids $(\omega(e_1), \dots, \omega(e_n))$ associé aux éléments de E , trouver un ensemble F de poids $\omega(F) = \sum_{e_i \in F} \omega(e_i)$ maximum (ou minimum). Ici, la famille F est l’ensemble des solutions du problème, elle permet de représenter diverses structures combinatoires, comme par exemple des chemins, des cycles, des arbres, etc, dans les graphes. Ce genre de problème apparaît dans des domaines divers, tels que l’industrie, le

transport, les télécommunications, ... Bien qu'il soit fini, l'ensemble de solutions d'un problème d'optimisation combinatoire (F) peut avoir un nombre très grand de solutions. Une méthode énumérative qui consisterait à énumérer les solutions du problème ne peut donc être envisagée, même si la puissance des calculateurs augmentait considérablement. En conséquence, d'autres outils plus performants ont été développés pour approcher ce type de problèmes, comme la programmation linéaire, et les méthodes de la recherche opérationnelle... L'optimisation combinatoire se trouve ainsi au carrefour de plusieurs disciplines telles que :

- la combinatoire,
- l'algèbre linéaire,
- la programmation linéaire,
- la programmation en nombres entiers,
- la théorie des graphes,
- les polyèdres combinatoires,
- la complexité des algorithmes,

Les problèmes d'optimisation combinatoire peuvent être formulés comme des programmes linéaires en nombres entiers (généralement en 0-1). Ils sont généralement *NP*-difficiles. Plusieurs méthodes ont été introduites durant les cinq dernières décennies pour résoudre les problèmes d'optimisation combinatoire. Parmi ces techniques, l'approche dite polyédrale s'est révélée particulièrement efficace.

Pour un certain type de problèmes d'optimisation combinatoire, on ne connaît pas d'algorithmes efficaces de résolution. Ces problèmes ont la propriété que, s'il existe un algorithme polynomial pour un d'entre eux, alors il en existerait un pour chacun d'eux. Pour ces problèmes dits *NP*-difficiles il y a donc peu d'espoir de pouvoir trouver une méthode efficace.

II.5 Optimisation multicritère

La résolution d'un problème d'optimisation consiste à explorer un espace de recherche afin de maximiser (ou minimiser) une fonction donnée. Les complexités (en taille ou en structure) relatives de l'espace de recherche et de la fonction à maximiser conduisent à utiliser des méthodes de résolution radicalement différentes. En première approximation, on peut dire

qu'une méthode déterministe est adaptée à un espace de recherche petit et complexe et qu'un espace de recherche grand nécessite plutôt une méthode de recherche stochastique (recuit simulé, algorithme génétique,...).

Résoudre un problème d'optimisation consiste à trouver la ou les meilleures solutions vérifiant un ensemble d'objectifs et contraintes définis par l'utilisateur.

Un problème d'optimisation consiste à rechercher la valeur maximale ou minimale, appelée optimum global, d'une fonction $F : S \rightarrow \mathbb{R}$. F est appelée la fonction objectif. Cette fonction est appelée fonction de coût, fonction d'adéquation ou « *fitness* ». Comme maximiser une fonction F est équivalent à minimiser la fonction $-F$, nous considérons dans ce manuscrit que les fonctions doivent être minimisées.

Dans un problème d'optimisation multiobjectif, on ne cherche pas à optimiser une fonction objectif unique mais un vecteur de fonctions. Dans notre problème, l'utilisateur de transport en commun souhaite en général l'optimisation de plusieurs critères, tels que le coût, le temps, le confort, etc...

Actuellement il existe plusieurs méthodes d'optimisation. Ces méthodes varient selon leurs complexités et leurs champs d'application. En plus des méthodes exactes il existe d'autres classes pour les méthodes d'optimisation. Dans la partie suivante, nous étudierons ces classes.

Dans un problème d'optimisation multicritère, il n'y a pas une solution optimale unique, mais un ensemble de solutions potentielles, car en général aucune solution n'est la meilleure vis-à-vis de tous les critères simultanément ; on identifie alors un ensemble de solutions non dominées, qui définissent un front optimal de Pareto [Marco 99]. Un individu i domine un autre j , si et seulement si, pour toute critère x la fonction objectif $F_i(x) < F_j(x)$.

Depuis 1993, un grand nombre de méthodes multiobjectif utilisent la dominance de Pareto pour rechercher les solutions d'un problème. Fonseca et Fleming ont proposé la méthode MOGA (*Multiple Objective Genetic Algorithm*) dans laquelle chaque individu de la population est rangé en fonction du nombre d'individus qui le dominant [Fonseca 93]. Ensuite, ils utilisent une fonction de notation permettant de prendre en compte le rang de l'individu et le nombre d'individus ayant le même rang.

Dans NSGA (*Non Dominated Sorting Genetic Algorithm*), la méthode proposée par Srivinas et Deb [Srivinas 93], le calcul de la *fitness* s'effectue en séparant la population en plusieurs groupes en fonction du degré de dominance au sens de Pareto de chaque individu. L'algorithme se déroule ensuite comme un algorithme génétique classique.

NPGA (*Niched Pareto Genetic Algorithm*), la méthode proposée par Horn et Napfliotis, utilise un tournoi basé sur la notion de dominance de Pareto [Horn 93]. Elle compare deux individus pris au hasard avec une sous-population de petite taille également choisie au hasard. Si un seul de ces deux individus domine la sous-population, il est alors positionné dans la population suivante. Dans les autres cas, une méthode dite « sharing » ou partage est appliquée pour sélectionner l'individu [Berro 04].

Dans une deuxième version de NSGA [Deb 00], l'auteur tente de résoudre toutes les critiques relevées sur NSGA I : complexité, non élitisme et utilisation du partage « sharing ». Pour diminuer la complexité de calcul de NSGA, Deb propose une modification de la procédure de tri de la population en plusieurs frontières. L'autre critique sur NSGA est l'utilisation du « sharing », méthode qui exige le réglage d'un ou plusieurs paramètre(s) et qui est également grosse consommatrice de calculs [Basseur 05]. Dans NSGA II, Deb remplace cette fonction par une méthode de diversification, dite « crowding » [Basseur 05], en attribuant deux caractéristiques à chaque individu : le rang de non dominance et la distance de « crowding ». Pour répondre à la critique de non élitisme, Deb utilise une sélection par tournoi, modifie la procédure de passage entre deux générations et définit une notion de préférence entre deux solutions, en fonction des caractéristiques des individus.

Pour les solutions des problèmes multiobjectif, la relation d'ordre n'est pas totale (une solution peut être meilleure qu'une autre sur certains objectifs et moins bonne sur les autres). On parle alors de solutions de compromis [Basseur 05].

Plusieurs approches différentes ont été mises au point pour transformer les problèmes multiobjectif en problèmes mono-objectif : les méthodes d'agrégation, les méthodes avec vecteur cible, et les méthodes ε -contraintes [Basseur 05].

La transformation en un seul objectif est l'une des premières méthodes employées pour résoudre les problèmes multiobjectif. C'est aussi l'une des plus largement employées [Garrett 99]. La transformation en problème mono-objectif se réalise en définissant une fonction de coût unique, comme étant la somme pondérée des différentes fonctions objectifs du problème initial :

$$f(x) = \sum_{i=1}^n \alpha_i f_i(x)$$

En définissant le problème de cette manière, la représentation du domaine réalisable de $f(x)$ devient linéaire [Basseur 05]. Donc, pour une agrégation donnée, il n'existe qu'une seule solution optimale pour le problème (la solution s dans la Figure II.3).

Généralement, les approches proposées résolvent le problème en utilisant différentes valeurs pour les poids α_i . Ainsi, on peut découvrir l'ensemble des solutions Pareto supportées du problème initial (Figure II.3).

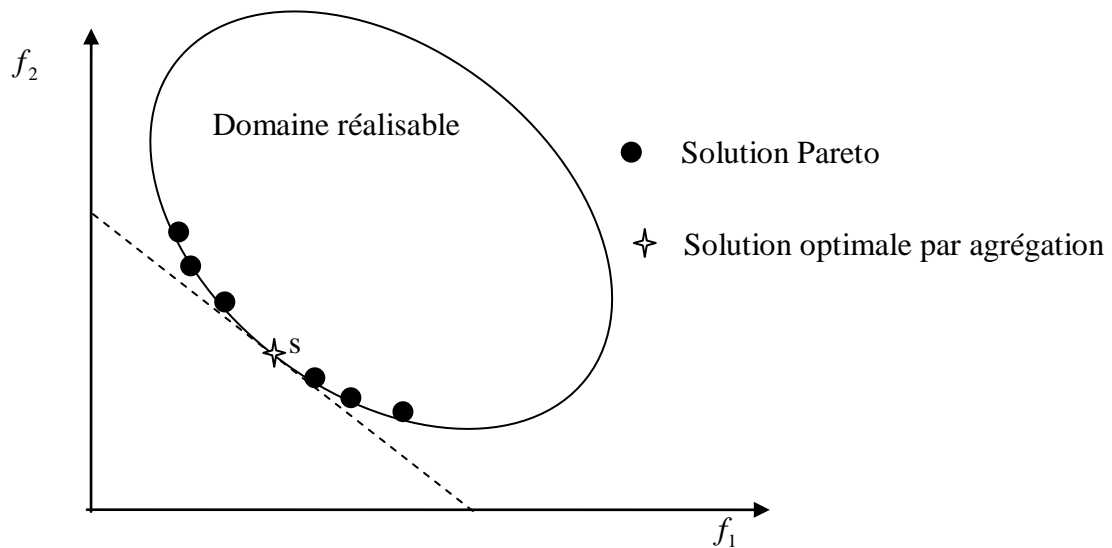


Figure II.3: Exemple d'optimisation à deux objectifs.

Appliquées à des problèmes *NP-difficiles*, les méthodes arborescentes (*branch and bound methods*) restent bien sûr exponentielles, mais leur complexité en moyenne est bien plus faible que celle d'une énumération complète. Elles peuvent donc pallier le manque d'algorithmes polynomiaux pour des problèmes de taille moyenne. Pour des problèmes de grande taille, leur durée d'exécution devient prohibitive, et il faut se tourner vers des heuristiques.

II.6 Heuristiques

Une méthode approchée ou heuristique (*heuristic, approximation method*), pour un problème d'optimisation combinatoire, est un algorithme qui a pour but de trouver une solution réalisable, tenant compte de critères d'optimisation et des contraintes, mais sans garantie d'optimalité. On oppose les méthodes approchées aux méthodes exactes, qui trouvent toujours l'optimum, mais leur inconvénient est le temps de résolution.

Il existe un très grand nombre d'heuristiques selon les problèmes à traiter, et il est aisé d'en inventer. Ces méthodes sont classées en trois classes [Hao 99].

a) **Méthodes constructives**, construisant une seule solution par une suite de choix partiels et définitifs, c'est-à-dire sans retours en arrière. On les appelle **méthodes gloutonnes** (*greedy*) quand elles cherchent à chaque itération à faire le choix le plus avantageux [Hao 99].

b) **Recherches locales** (*local search*). On part d'une solution initiale et, par transformations successives, on construit une suite de solutions de coûts décroissants. Le processus s'arrête quand on ne peut plus améliorer la solution courante [Hao 99].

c) **Métaheuristiques** Une recherche locale peut être piégée dans un minimum local. Les métaheuristiques peuvent échapper à ces minima en construisant aussi une suite de solutions, mais dans laquelle la fonction objectif peut temporairement augmenter. Ces méthodes, qui comprennent notamment la méthode du recuit simulé, les algorithmes génétiques, la méthode de recherche Tabou, les algorithmes de colonies de fourmis, etc [Hao 99] sont apparues, à partir de l'année 1980, avec une ambition commune : résoudre au mieux les problèmes d'optimisation difficile [Siarry 02]. Ces méthodes ont en commun les caractéristiques suivantes :

- Elles sont, au moins pour partie, stochastiques : cette approche permet de faire face à l'explosion combinatoire des possibilités ;
- Elles sont inspirées par des analogies : avec la physique (recuit simulé par exemple), avec la biologie (comme les algorithmes génétiques) ou avec l'éthologie (les colonies de fourmis) ;
- Elles sont capables de guider, dans une tâche particulière, une autre méthode de recherche spécialisée (par exemple, une autre heuristique, ou une méthode d'exploration locale).

Ces méthodes ont aussi les mêmes inconvénients : les difficultés de réglage des paramètres mêmes de la méthode, et le temps de calcul élevé. Dans le cadre de notre travail de thèse, nous avons étudié les métaheuristiques plus en détail, afin de choisir la bonne approche pour résoudre notre problème.

II.7 Les métaheuristiques

Les métaheuristiques sont des techniques d'amélioration progressive d'une première solution. Elles incluent notamment le recuit simulé, les méthodes Tabou, et les algorithmes

génétiques. Elles permettent de traiter des problèmes de grande taille, tout en obtenant des solutions excellentes, souvent optimales [Lacomme 05].

P. Siarry et d'autres chercheurs ont remarqué que, la plupart du temps, les méthodes d'optimisation multiobjectif et les métaheuristiques associées peuvent être complètement découplées. Seuls les algorithmes génétiques engendrent des méthodes d'optimisation multiobjectif qui ne peuvent pas être découplées [Siarry 02]. Ces méthodes peuvent être classées sur deux classes : les métaheuristiques "de voisinage", qui améliorent une seule solution en recherchant dans leur voisins (recherche tabou, recuit simulé...), et les métaheuristiques "distribuées", qui traitent simultanément toute une population de solutions (algorithme génétique...)[Siarry 02].

Dans la partie suivante, nous donnons une petite introduction sur le recuit simulé, la méthode tabous et les algorithmes de colonies de fourmis. Par la suite, nous expliquons un peu plus en détail les algorithmes évolutionnaires.

II.7.1 Le recuit simulé

Cette méthode de recherche a été proposée par des chercheurs d'IBM qui étudiaient les verres de spin. Par analogie avec des phénomènes physiques, on utilise un processus métallurgique (le recuit) pour trouver un minimum [Dréo 03].

La méthode de recuit simulé, appliquée aux problèmes d'optimisation, considère une solution initiale et recherche dans son voisinage une autre solution de façon aléatoire. L'originalité de cette méthode est qu'il est possible de se diriger vers une solution voisine de moins bonne qualité avec une probabilité non nulle. Ce technique permet de sortir des optima locaux [Dréo 03]. Au début de l'algorithme, un paramètre T , apparenté à la température, est déterminé et décroît tout au long de l'algorithme pour tendre vers 0. La probabilité P d'acceptation des mauvaises solutions dépend de la valeur du paramètre T [Nicola 04]. Le recuit simulé a été appliqué pour résoudre plusieurs problèmes. Aussi, il existe plusieurs variantes de cette méthode, tel que la diffusion simulé, recuit microcanonique, méthode de seuil...[Dréo 03].

II.7.2 Méthode Tabou

La méthode Tabou (*tabu search*) a été inventée par Glover [Goldberg 89] [Goldberg 91]. Le principe de cette méthode est à chaque itération le voisinage de la solution courante est examiné. L'algorithme enregistre la meilleur solution parmi les voisins, même si elle est moins bonne que la solution courante. L'acceptation des solutions moins performant que la

solution courante permet d'éviter de tomber dans un optimum local. Pour échapper de tourner dans un cercle entre plusieurs solutions, l'algorithme interdit le passage par des solutions récemment visitées. En pratique la méthode stocke dans une liste taboue T les attributs des dernières solutions visitées. Dans l'itération suivante, la meilleure nouvelle solution voisine enlève la solution la plus ancienne dans la liste (algorithme suivant). Dans d'autres cas, la méthode mémorise les mouvements réalisés plutôt que les solutions. Ensuite, on interdit les mouvements inverses. Cette technique est rapide et consomme peu de mémoire.

```

Initialiser la liste T (T vide)
S0 point de départ
N : nombre d'itération maximal
I=0
Sc=S0 (solution courante)
Pour i<N répéter
E= voisinage (Sc)/T (les voisins de Sc sans les élément de T)
soit y ∈ E et y meilleur solution dans E
T=T+{y}
Sc=y
I=I+1
Fin de répéter

```

Algorithme : Recherche Tabou

II.7.3 Algorithme de colonies de fourmis (ACF)

Les études réalisées par les éthologistes ont montré que certains comportements collectifs des insectes sociaux étaient auto-organisés [Deneubourg 89]. L'auto-organisation caractérise des processus au cours desquels des structures émergent au niveau collectif, à partir d'une multitude d'interactions simples entre insectes, sans être codées explicitement au niveau individuel. En marchant du nid à la source de nourriture et vice-versa (ce qui dans un premier temps se fait essentiellement de façon aléatoire), les fourmis déposent au passage sur le sol une substance volatile odorante appelée phéromones. Cette substance permet de créer une piste chimique, sur laquelle les fourmis s'y retrouvent. En effet, d'autres fourmis peuvent détecter les phéromones grâce à des capteurs sur leurs antennes.

Les éthologistes ont montré aussi que les fourmis étaient capables de sélectionner le plus court chemin pour aller du nid à une source de nourriture grâce au dépôt et au suivi de pistes de phéromone.

Les fourmis déposent de la phéromone à l'aller vers la source de nourriture et au retour vers le nid. Au départ, le choix est aléatoire mais la branche courte devient vite la plus marquée car

les fourmis qui l'empruntent arrivent plus vite au nid et auront statistiquement plus de chance de l'emprunter lorsqu'elles retourneront vers la source de nourriture.

Dans une itération de l'algorithme ACF, chaque agent (fourmi) construit une solution d'après des décisions basées sur les quantités de phéromone. Ces traces sont mises à jour en examinant les critères heuristiques des solutions obtenues. Elles sont renforcées pour les décisions ayant donné de meilleures solutions et diminuées pour les autres. On répète cette itération générale jusqu'à la réalisation d'un critère d'arrêt, comme un nombre maximum [Lacomme_b 03]. Ce mécanisme permet d'améliorer progressivement les solutions au cours des itérations.

II.8 Les Algorithmes évolutionnaires

II.8.1 Principe général

Les Algorithmes Evolutionnaires (AE) sont des métaheuristiques basées sur des métaphores biologiques inspirées des mécanismes d'évolution darwinienne. En vue d'imiter les processus d'évolution observés dans la nature, la première adaptation des premiers AE, à travers les algorithmes génétiques, aux problèmes d'optimisation combinatoire, a été réalisée par Holland, dans les années 70 [Holland 92].

Les algorithmes génétiques (AG) ont été ensuite développés par d'autres chercheurs comme Goldberg, Davis et Michalewicz [Goldberg 89] [Michalewicz 94]. Les AG constituent certainement l'exemple le plus connu (ou populaire) des algorithmes évolutionnaires [Dréo 03]. Depuis, d'autres variantes de ces algorithmes ont été aussi développées, telles que les algorithmes à stratégie d'évolution [Schwefel 97], la programmation génétique et la programmation évolutionnaire [Bounsaythip 98].

Les AE sont des algorithmes itératifs de recherche globale, fondés sur une analogie avec le monde biologique. En effet, pour un problème donné, une solution est un individu et un ensemble de solutions correspond à une population d'individus. Chaque individu peut être appelée chromosome, et chaque chromosome est constitué d'un ensemble de caractéristiques, appelés les gènes. Dans le codage binaire, un gène vaut soit 0 soit 1. L'ensemble des gènes d'un individu est son génotype et l'ensemble du patrimoine génétique d'une espèce est le génome. Les différentes versions d'un même gène sont appelées allèles.

Le principe des AE est de reproduire l'évolution naturelle d'individus [Dréo 03], génération après génération, en respectant les lois de l'hérédité et le concept de sélection naturelle, autrement dit, la survie de l'individu le plus fort ou le mieux adapté à l'environnement. Cette évolution se fait avec un échange structuré, mais aussi aléatoire, d'informations entre les individus, en vue de reproduire de nouveaux éléments, qui sont meilleurs, au sens du critère choisi, c'est-à-dire, au sens de la fonction objectif à optimiser [Goldberg 89].

Pour un problème d'optimisation donné, un individu représente un point de l'espace de recherche (appelé aussi espace d'état). L'exploration désigne les processus visant à récolter de l'information sur le problème optimisé. L'exploitation (ou l'intensification) vise à utiliser l'information déjà récoltée pour définir et parcourir les zones intéressantes de l'espace de recherche. Pour chaque individu on associe la valeur du critère à optimiser (son adaptation). On génère ensuite de façon itérative (Figure II.4) des populations d'individus, sur lesquelles on applique des processus de sélection, de croisement et de mutation. La sélection a pour but de favoriser les meilleurs éléments de la population, le croisement et la mutation assurent une exploitation et une exploration efficace de l'espace de recherche. On commence par générer une population d'individus. Pour passer d'une génération K à la génération $K+1$, les trois opérations suivantes sont répétées pour tous les éléments de la population K . Des couples de parents (P_1 et P_2) sont sélectionnés en fonction de leurs adaptations. L'opérateur de croisement leur est appliqué avec une probabilité P_c (généralement autour de 0,6) et on génère des couples d'enfants (E_1 et E_2). D'autres éléments sont sélectionnés en fonction de leur adaptation. L'opérateur de mutation leur est appliqué avec la probabilité P_m (P_m est généralement inférieure à P_c)

L'algorithme vise alors à chercher la meilleure combinaison des individus de la population selon la fonction d'adéquation choisie. On crée ainsi, à chaque itération, ou génération, une nouvelle population, qui est généralement formée des meilleurs éléments. La création des nouvelles populations se fait par combinaison de certains individus à l'aide d'opérateurs de croisement et aussi par modification pseudo-aléatoire de leurs codages génétiques via l'opérateur de mutation. Au fur et à mesure des générations, les individus vont tendre, en général, vers l'optimum de la fonction de coût [Mesghouni 99].

II.8.2 Architecture des algorithmes évolutionnaires

Un algorithme évolutionnaire est un algorithme itératif qui possède une architecture bien définie. Il procède par différentes étapes en vue de résoudre un problème d'optimisation. Comme illustré par la figure II.4, il s'agit en effet des étapes de [Schwefel 97] :

- **Genèse** : première phase de l'algorithme, dans laquelle la population initiale est construite d'une manière aléatoire ou à travers des résultats issus d'autres techniques d'optimisation.
- **Evaluation** : consiste à calculer la valeur de la fonction de coût pour chaque individu.
- **Sélection** : le choix des éléments les plus adaptés pour la formation de la nouvelle génération.
- **Recombinaison et mutation** : phase de reproduction, dans laquelle une nouvelle population est construite à partir des individus sélectionnés, via des opérateurs de croisement et de mutation.
- **Arrêt** : il s'agit d'un test de l'efficacité de l'algorithme, à travers une valeur de la fonction objectif à atteindre, le nombre d'itérations ou le temps d'exécution. La solution courante est prise quand ce test est vérifié ; sinon, l'algorithme passe à l'itération suivante, qui commence à partir de l'étape d'évaluation.

Nous détaillons les principales étapes réalisées par les algorithmes évolutionnaires (AE) dans la section suivante.

II.8.3 Mise en œuvre des algorithmes évolutionnaires

La mise en œuvre d'un algorithme évolutionnaire nécessite plusieurs étapes de conception et de choix de certains paramètres. En effet, il est important de choisir le codage des solutions, l'opérateur de sélection et le processus de renouvellement de la population, via les opérateurs de croisement et de mutation.

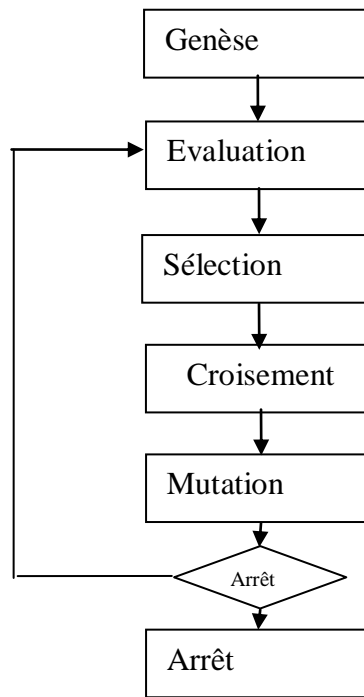


Figure II.4: Architecture générale d'un algorithme évolutionnaire.

II.8.3.1 Choix d'un codage

Premièrement, il faut représenter les différents états possibles de la variable dont on cherche la valeur optimale sous une forme utilisable par un AE, c'est-à-dire par le codage approprié des solutions sous forme de chromosomes ou génotypes. Cela permet d'établir une connexion entre les valeurs de la variable et les individus de la population, de manière à imiter la connexion qui existe en biologie entre le génotype et le phénotype [Vallée 03]. Il existe principalement deux types de codage : le codage binaire (représentation sous forme de chaîne binaire) et le codage réel (représentation directe des valeurs réelles de la variable).

Le codage peut être direct, quand il y a une correspondance bijective entre l'ensemble des chromosomes et celui des solutions qui leur correspond. Il peut être indirect, quand un générateur doit être utilisé pour définir la solution associée à un chromosome. Aussi, il peut être mixte, lorsqu'il combine les deux codages [Aloulou 02].

Exemple : Supposons la problématique d'affectation des arrêts à un véhicule V dans un problème de régulation de trafic ou dans un problème de tournée de véhicule. Une solution non complète serait un ensemble d'affectations d'arrêts à ce véhicule sans le choix de leur ordre. Dans la figure II.5, nous avons représenté un chromosome possible décrivant les

variables de passage a_i aux stations s_i . Dans cet exemple, a_i est nul si V ne passe pas par s_i et est égal à 1 sinon. La taille du chromosome (nombre de gènes) est celui des arrêts.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|

Figure II.5: Exemple de codage d'un chromosome.

Longtemps, les algorithmes génétiques ont utilisé un codage binaire. Dans ce cas, quelle que soit leur nature, les éléments de l'ensemble des individus, sur lequel est menée l'optimisation, sont codés par des chaînes de bits. L'intérêt principal du codage binaire est de permettre l'utilisation d'opérateurs de croisement et de mutation simples.

Ce type de codage présente cependant des inconvénients :

- deux éléments voisins en terme de distance de Hamming (nombre de bits différents) ne codent pas nécessairement deux éléments proches dans l'espace de recherche. Cet inconvénient peut être évité en utilisant un codage de Gray (un codage tel que deux entiers consécutifs ne diffèrent que d'un bit), ou d'autres codages adaptés au problème. Cependant, un autre problème se pose alors, qui est celui qui concerne le choix du codage qui influe sur le comportement des opérateurs.
- Pour des problèmes d'optimisation dans des espaces de grande dimension, le codage binaire peut rapidement devenir mauvais. Dans notre problème, l'ensemble des stations ou des arcs est important, par suite le codage binaire n'est pas adéquat. Généralement, chaque variable est représentée par une partie de la chaîne de bits et la structure du problème n'est pas bien reflétée.

II.8.3.2 Opérateur de sélection

La sélection consiste à choisir les individus de la population courante qui vont survivre et se reproduire. Elle est réalisée en fonction de la valeur de la fonction de coût qui évalue les solutions. L'opérateur de sélection joue ainsi un rôle primordial dans la détermination de la performance des nouvelles générations et donc dans l'amélioration de la qualité des solutions.

Plusieurs techniques de sélection sont utilisées par les chercheurs ; elles peuvent être déterministes ou stochastiques [Bounsaythip 98] :

- La sélection déterministe consiste, par exemple, à garder les meilleurs individus au sens de leurs coûts et à rejeter le reste, ce qui implique leur classement ou « ranking ».

- Une sélection stochastique peut être réalisée par la technique de la roulette pondérée [Goldberg, 89], où chaque individu occupe une surface de la roue proportionnelle à sa valeur de la fonction de coût. En supposant que f_i est la valeur de la fonction de coût associée au i -ème individu, la probabilité de sélection de ce dernier est en fait égale à $\frac{f_i}{f_s}$, avec $f_s = \sum_j f_j$.

Selon cette méthode, chaque chromosome sera dupliqué dans une nouvelle population proportionnellement à sa performance. On effectue, en quelque sorte, autant de tirages avec remise qu'il y a d'éléments dans la population.

L'inconvénient de la sélection réside dans le choix exclusif des meilleurs individus de la population, au détriment de la diversité des solutions. L'algorithme risque ainsi de converger prématurément. Pour avoir une bonne exploration de l'espace de recherche, des opérateurs de croisement et de mutation sont appliqués aux individus sélectionnés, pour en créer des nouveaux.

Il existe d'autres méthodes de sélection, la plus connue étant celle du tournois (*tournament selection*) : on tire deux individus aléatoirement dans la population et on reproduit le meilleur des deux dans la nouvelle population. On applique cette procédure jusqu'à ce que la nouvelle population soit complète. Cette méthode donne de bons résultats. Toutefois, aussi importante que soit la phase de sélection, elle ne crée pas de nouveaux individus dans la population. La création des nouveaux individus est le rôle des opérateurs de croisement et de mutation.

II.8.3.3 Opérateur de croisement

Le croisement est une étape de recombinaison essentielle de l'algorithme évolutionnaire car il permet l'exploration de l'espace de recherche. Une fois la population intermédiaire déterminée, les individus sont aléatoirement répartis en couples. Les chromosomes sont alors copiés et recombinaison de façon à former, en général, deux descendants possédant des caractéristiques issues des deux parents. On forme ainsi la génération suivante.

L'opérateur de croisement opère avec une probabilité p_c , fixée selon le problème concerné. Plus ce taux est élevé, plus il y a de nouvelles structures qui apparaissent dans la population. Mais, s'il est trop élevé, les bonnes solutions risquent d'être modifiées trop vite par rapport à l'amélioration que peut apporter la sélection. D'autre part, si le taux de croisement est très faible, la recherche risque de stagner, à cause du faible taux d'exploration.

Parmi les méthodes de croisement les plus utilisées on peut souligner trois opérateurs: le croisement à un point, le croisement multi-points et le croisement uniforme :

- **Croisement à un point** : il s'agit de choisir, au hasard, un point de croisement pour chaque couple de chromosomes et d'effectuer un échange des ensembles d'allèles se trouvant de part et d'autre de ce point entre les deux parents (Figure II.6).

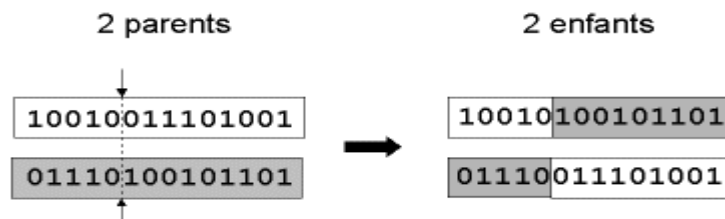


Figure II.6: Exemple d'un croisement à un point.

On peut étendre ce principe en découpant le chromosome non pas en 2 sous-chaînes, mais en 3, 4, etc. [Bridges 91].

- **Croisement multi-point** : dans ce cas, plusieurs points de croisement sont sélectionnés et il y a un échange des différentes parties d'allèles cernées par ces points, entre les parents. La figure II.7 illustre un croisement à deux points.

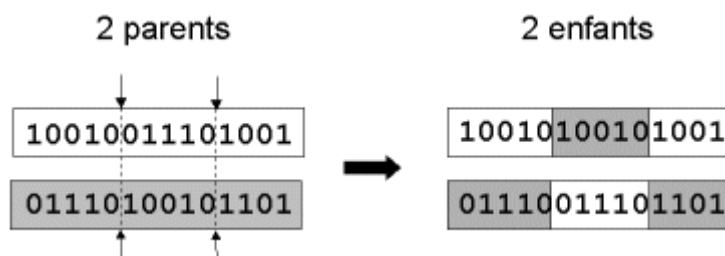


Figure II.7: Exemple de croisement à deux points.

- **Croisement uniforme** : il opère à l'aide d'un masque qui représente les tirages aléatoires, pour décider de la transmission de la valeur de l'allèle à l'un ou l'autre des descendants. Si, à la même position que l'allèle, la valeur du masque est égale à 1, l'allèle du parent 1 passe à celui de l'enfant 1 et l'allèle du parent 2 passe à l'enfant 2. Sinon, c'est l'inverse qui se produit (figure II.8).

D'autres types de croisement, plus spécifiques au problème traité, peuvent bien entendu être utilisés dans le cadre d'un algorithme génétique [Durand 96]. L'efficacité du croisement dépend souvent de son adaptation au problème.

| | | | | | | | | | | |
|----------|---|---|---|---|---|---|---|---|---|---|
| Parent 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| Parent 2 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| Masque | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| Enfant 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| Enfant 2 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |

Figure II.8: Exemple de croisement uniforme.

II.8.3.4 Opérateur de mutation

La mutation est définie comme étant la modification aléatoire de la valeur d'un allèle dans un chromosome. La figure II.9 illustre un exemple de mutation appliquée à la position 9 d'un chromosome binaire. Elle joue le rôle de bruit, empêche l'évolution de se figer et garantit que l'optimum global peut être atteint. Cet opérateur évite donc une convergence prématurée vers les optimums locaux. Il est appliqué avec une probabilité fixée, p_m . Le taux de mutation rend la recherche trop aléatoire s'il est trop élevé. Par ailleurs, s'il est trop faible, la recherche risque de stagner.

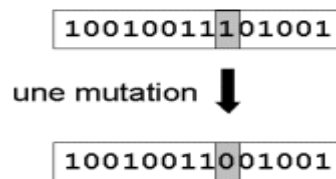


Figure II.9: Exemple de mutation.

Il existe d'autres façons d'effectuer des mutations :

- transposition de deux allèles consécutifs,
- transposition d'allèles dans un chromosome,
- inversion de l'ordre des allèles présents entre deux coupes.

Il est aussi possible d'associer une probabilité de mutation différente à chaque gène, selon le principe de l'auto-adaptation, où chaque variable est soumise au processus d'évolution. L'individu possède ainsi un second chromosome codant ces probabilités. En général, cet opérateur permet l'exploitation des individus non explorés par le croisement : si, par exemple, la solution optimale doit avoir la valeur 1 à la position 9 du chromosome, alors que dans la population initiale tous les individus ont 0 dans cette position, l'opérateur de croisement n'arrive jamais à explorer cette solution.

En conclusion, une fois que la performance de chaque individu dans la population actuelle est évaluée, les mécanismes évolutifs entrent en jeu pour procéder à l'exploration de l'espace des valeurs. Les opérateurs de reproduction (ou opérateurs génétiques) cherchent à imiter ces mécanismes. Les générations se succèdent, jusqu'à l'obtention des solutions satisfaisantes le critère d'arrêt de l'algorithme.

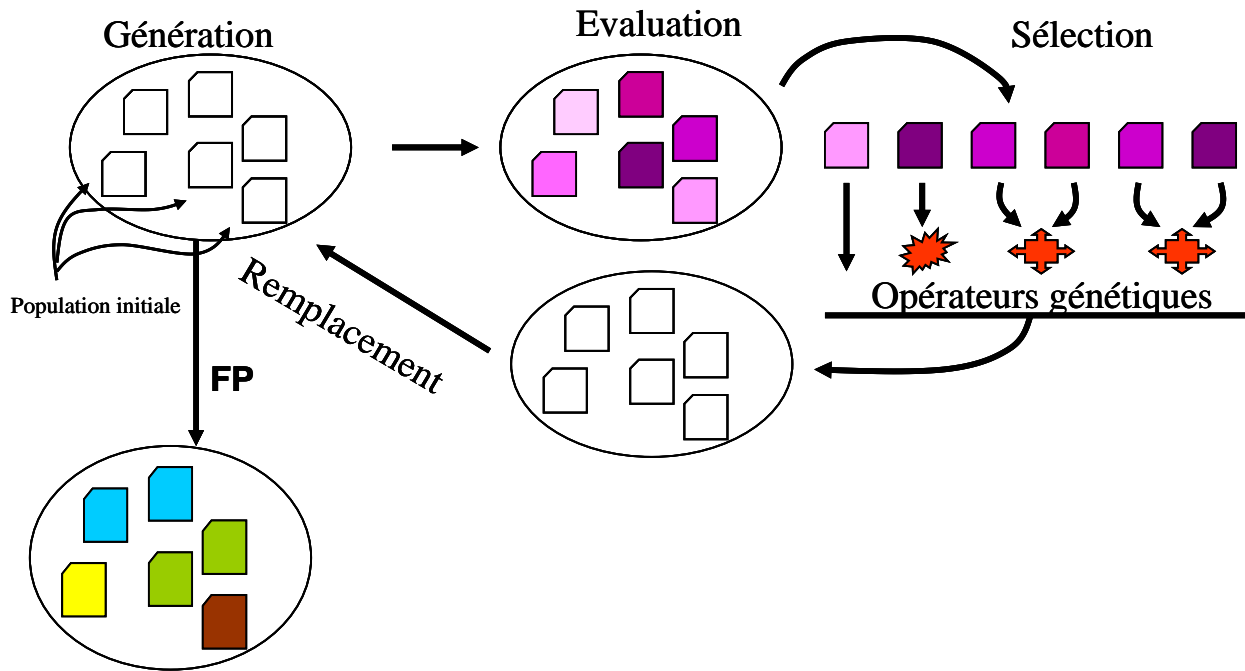


Figure II.10: Exemple de structure d'un algorithme évolutif.

II.8.3.5 Caractéristiques des algorithmes évolutifs

Les AE sont capables de s'adapter à n'importe quel espace de recherche. Ils demandent une mesure de la qualité de la solution et nécessitent la définition de l'espace par un codage et des opérateurs qui lui permettent de le parcourir efficacement.

Le principal avantage des AE par rapport aux autres techniques d'optimisation (énumératives, « hill-climbing », etc.) consiste en une combinaison de :

- l'exploration de l'espace de recherche, basée sur des paramètres aléatoires, grâce à une recherche parallèle,
- l'exploitation des meilleures solutions disponibles à un moment donné.

Néanmoins, comme pour les méthodes par voisinage, ces algorithmes ont des inconvénients. Le choix des codages et des opérateurs les plus adéquats est lié aux spécificités du problème.

En outre, il est nécessaire d'effectuer plusieurs expérimentations pour ajuster les paramètres de l'algorithme (taille de la population, probabilités de croisement et de mutation, nombre de générations, etc.).

Quelle doit être la taille de la population ? Une population trop petite évoluera probablement vers un optimum local peu intéressant. Une population trop grande sera inutile car le temps de convergence sera excessif. La taille de la population doit être choisie de façon à réaliser un bon compromis entre le temps de calcul et la qualité du résultat. Les algorithmes évolutionnaires souffrent, d'une manière générale, de la présence de plusieurs paramètres qui déterminent leur efficacité. Ces paramètres influent sur la vitesse de convergence, la capacité d'exploration, le temps de recherche...[Basseur 05]. Nous allons maintenant discuter rapidement du rôle de ces paramètres.

La taille de la population, N , et la longueur L du codage de chaque individu : si N est trop grand, ou si L est trop longue, le temps de calcul de l'algorithme peut s'avérer très important ; si N est trop petit, il peut converger très rapidement vers un mauvais chromosome.

La probabilité de croisement, P_c , dépend en général de la forme de la fonction de performance. Son choix est bien souvent heuristique (tout comme pour P_m). Plus elle est élevée, plus la population subit des changements importants. Les valeurs généralement admises sont comprises entre 0,5 et 0,9.

La probabilité de mutation P_m . Ce taux est généralement faible, puisqu'un taux élevé risque de conduire à une solution sous-optimale en perturbant celle qui est optimale. Plutôt que de réduire P_m , une autre façon d'éviter que les meilleurs individus soient altérés est d'utiliser *l'élitisme* : ainsi, peut-on choisir, par exemple, de recopier à l'identique les 5% meilleurs de la population à chaque génération, l'opérateur de reproduction ne jouant alors que sur les 95% restants.

Il est important de comprendre que le fonctionnement d'un tel algorithme ne garantit nullement la réussite. Nous sommes en présence d'un système stochastique et la probabilité existe qu'une population génétique soit trop éloignée de la solution, ou par exemple, qu'une convergence trop rapide bloque le processus d'évolution. Ces algorithmes n'en sont pas moins extrêmement performants, leur utilisation se développe dans des domaines aussi divers que la prévision boursière, l'ordonnancement des systèmes de production ou la programmation des robots d'assemblage dans l'industrie automobile.

II.8.4 Les algorithmes évolutionnaires et le domaine du transport

Grâce à leurs différents avantages liés à la résolution des problèmes d'optimisation combinatoire, les AE ont connu un réel développement dans le domaine de l'ordonnancement en général et dans l'ordonnancement des systèmes de production en particulier. En effet, [Pierreval 03] présente une étude des différentes approches évolutionnaires dans l'organisation de la production cellulaire et la conception des ateliers de production et des systèmes d'assemblage. [Caux 94] présente également les applications des algorithmes génétiques aux problèmes d'ordonnancement. Aussi, ces algorithmes ont été utilisés pour la résolution des problèmes du voyageur de commerce, de tournées de véhicules et aussi de contrôle du trafic aérien.

En ce qui concerne le contrôle du trafic aérien, [Hansen, 03] et [Delahaye 96] abordent ce domaine par des approches évolutionnaires. En effet, [Delahaye 96] traite, à travers des approches génétiques, trois problématiques liées à :

- La résolution des conflits des trajectoires entre les avions par le respect des trajectoires optimales en vue de garder une distance de séparation conforme à une norme fixée.
- La division du réseau aérien, considéré comme un espace bidimensionnel, en des secteurs équilibrés en termes de charge de travail liée au nombre d'avions et de conflits, tout en minimisant le travail de coordination.
- L'affectation du trafic entre les secteurs, en minimisant les augmentations des distances parcourues et la charge de travail des responsables des secteurs.

[Ciesielski 98] et [Stevens 95] présentent une approche évolutionnaire pour l'ordonnancement des temps d'atterrissage des avions sur un nombre limité de pistes. Selon [Ciesielski 98], le codage correspondant illustre, pour chaque avion, la piste sur laquelle il devrait atterrir et le temps de l'atterrissage (en nombre de périodes de 30 secondes) suivant l'heure courante. La figure II.11 présente un exemple pour ce codage. Nous pouvons alors déduire que l'avion 1 va atterrir à 12:00 sur la piste 0, que l'avion 2 va atterrir à 12:03 sur la piste 1, que l'horaire d'atterrissage de l'avion 3 est de 12:04:30 sur la piste 0, etc.

| 1 | 2 | 3 | 4 | ... | | | | | |
|---|---|---|---|-----|---|----|---|-----|-----|
| 0 | 0 | 6 | 1 | 9 | 0 | 12 | 0 | ... | ... |

Figure II.11: Exemple de codage de l'ordonnancement des atterrissages.

L'ordonnement des horaires d'atterrissage utilisé par [Ciesielski 98] est effectué en temps réel.

Le problème du voyageur de commerce représente un voyageur de commerce qui doit passer par un certain nombre de villes avec un minimum de temps de parcours total. Sa tournée doit être planifiée de manière à passer une seule fois par chaque ville. Potvin [Potvin 96] aborde ce problème par une approche génétique en codant les solutions par des chromosomes contenant la séquence ordonnée des passages aux différentes villes. La Figure II.12 présente un exemple d'un tel codage pour un problème à 9 villes. Potvin présente les différents opérateurs de croisement spécifiques respectant les contraintes du problème.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | 5 | 3 | 2 | 6 | 4 | 8 | 7 | 9 |
|---|---|---|---|---|---|---|---|---|

Figure II.12: Exemple de codage pour un problème de voyageur de commerce.

S. Ngamchai [Ngamchai 00] présente un algorithme génétique pour la conception des lignes d'un réseau de bus avec localisation des nœuds de correspondance. En fait, Ngamchai présente un individu en tant que collection de chemins. Chaque chemin représente une route et le nombre de chemins dans un individu représente le nombre de véhicules. Aussi Fayeche a présenté un algorithme génétique pour la régulation des horaires de transport en cas de perturbation dans un système d'aide à la décision [Fayeche 01]. Dans cet algorithme, l'auteur utilise une matrice pour coder les solutions.

L'approche génétique proposée possède alors un algorithme de génération des routes. Ce dernier établit des routes avec la contrainte de desservir tous les nœuds en ignorant la demande, avec des techniques de séparation. Différents opérateurs génétiques (croisement, fusion, rupture, etc.) sont présentés et sont appliqués sur les individus de la population avec des probabilités qui leur sont attribuées.

Le problème générique de cette classe, appelé problème de construction de tournées de véhicules, suppose qu'une flotte de k véhicules identiques de même capacité b livrent un certain produit à n clients $i=1, \dots, n$ à partir d'un dépôt central noté 0. La demande du client i est notée d_i et le coût du transport du point i vers le point j est noté c_{ij} . Le problème de construction de tournées de véhicules consiste à déterminer K routes, une pour chaque véhicule. Une route est définie comme un tour commençant au dépôt, qui ensuite visite un

sous-ensemble de clients dans un ordre donné et puis retourne au dépôt. Chaque client doit être affecté à exactement une route et la demande totale des clients affectés à une même route ne peut pas dépasser la capacité du véhicule. Un ensemble de K routes est optimal si son coût total de transport est minimum.

Les méthodes exactes les plus efficaces pour résoudre ce problème sont de type « Branch and Cut » et résolvent des problèmes comportant une centaine de clients au plus. Pour des problèmes de taille plus importante, comme c'est presque toujours le cas dans la réalité, différentes méthodes de recherche locale, heuristiques et métaheuristiques ont montré leur efficacité. Les articles Golden and Assad [Golden 86], Bodin [Bodin 90] et Laporte [Laporte 95] présentent l'état de l'art en ce domaine.

Le problème du voyageur de commerce est le cas particulier du problème de construction de tournées de véhicules le plus « épuré » et le plus étudié. Il consiste à déterminer le tour de coût minimum, qui partant et retournant au dépôt, visite une et une seule fois chaque client [Labbé 97].

Zhu [Zhu 95] présente un codage génétique similaire pour un problème de tournée de véhicules, néanmoins, il n'effectue pas de séparation entre les routes, puisqu'il restreint la validité des solutions qui résultent des opérations de croisement. Cette séparation est déduite des chromosomes après un décodage spécifique.

Salim et Cai [Salim 97] abordent le problème NP-difficile d'ordonnement de trains de marchandise avec une approche basée sur un algorithme génétique. Le but de cette approche est de trouver les routes et les horaires de passage des trains aux différentes stations et surtout aux boucles de passage. En fait, pour éviter les collisions, un train doit s'arrêter au niveau de ces boucles de passage afin de laisser un autre passer. Le chromosome représentatif des solutions est décrit sous forme de matrice avec des valeurs binaires (0 ou 1) correspondant aux passages des trains aux boucles en question.

Deb et Chakroborty [Deb 98] présentent une approche génétique de planification des horaires d'un réseau de bus. L'approche a pour objectif de minimiser la durée totale d'attente des passagers. Les variables de décision illustrées dans le codage correspondent aux temps de parcours et aux temps d'arrêt.

D'autres approches, qui reposent sur une hybridation entre problèmes de satisfaction de contraintes (*Constraint Satisfaction Problems, CSP*) et algorithme génétique, ont été expérimentées. La solution proposée était alors d'intégrer le CSP aux opérateurs de

l'algorithme génétique afin que tous les individus de la population correspondent à des solutions admissibles du problème. Dans ce cadre, Küchenhoff [Küchenhoff 93] obtient des bons résultats pour le problème du TSP avec un opérateur de mutation effectuant une optimisation locale (à l'aide d'un CSP) et un opérateur de croisement générant des individus entièrement nouveaux en cas d'échec du CSP (bonne exploration de l'espace de recherche).

II.9 Les systèmes multi-Agents (SMA)

Les systèmes multi-agents, SMA, présentent des avantages potentiels dans la résolution des problèmes liés à des systèmes ouverts, distribués et complexes. En effet, il est raisonnable de traiter ces derniers par la décomposition modulaire et fonctionnelle, pour que des « agents » soient spécialisés dans la résolution d'un aspect particulier du problème. Afin de concevoir des systèmes multi-agent cohérents, il est nécessaire de [Sycara 98] :

- formuler, décrire, décomposer et allouer des problèmes et faire une synthèse des résultats parmi un groupe d'agents intelligents ;
- permettre aux agents de communiquer et interagir ;
- s'assurer que les agents prennent des décisions et effectuent des actions de manière cohérente.

Les SMA sont à l'intersection de plusieurs domaines scientifiques : informatique répartie et génie logiciel, intelligence artificielle, vie artificielle. Ils s'inspirent également d'études issues d'autres disciplines connexes notamment la sociologie, la psychologie sociale, les sciences cognitives et bien d'autres. C'est ainsi qu'on les trouve parfois à la base des :

- systèmes distribués [Fagin 95];
- interfaces hommes-machines [Kobsa 89];
- bases de données et bases de connaissances distribuées coopératives [Babin 97];
- systèmes pour la compréhension du langage naturel [Kaplan 98];
- protocoles de communication et réseaux de télécommunications [Nwana 99];
- programmation orientée agents et génie logiciel [Jennings 00];
- robotique cognitive et coopération entre robots [Lakemeyer 99];
- applications distribuées comme le web, l'Internet, le contrôle de trafic routier, le contrôle aérien, les réseaux d'énergie, etc. [Chaib-draa 96].

II.9.1 Les systèmes multi-agents et l'Intelligence Artificielle

Le thème des systèmes multi-agents (SMA), est actuellement un champ de recherche très actif. Cette discipline est à la connexion de plusieurs domaines, en particulier de l'Intelligence Artificielle, des systèmes informatiques distribués et du génie logiciel [Chaib-draa 94].

Les recherches en Intelligence Artificielle avaient pour objectif de développer des logiciels afin de simuler des capacités des êtres humains, telles que le raisonnement, le langage naturel de communication et l'apprentissage. Les chercheurs ont ainsi traité le développement de théories, de techniques et de systèmes pour étudier les propriétés du comportement et du raisonnement d'une entité cognitive unique. L'évolution de l'Intelligence Artificielle a provoqué plus d'intérêts vers des problèmes plus complexes, plus réalistes et de plus grande échelle, qui dépassent les capacités d'un agent individuel. En effet, la capacité d'un seul agent est limitée par ses connaissances, ses ressources de calcul et ses perspectives. Cette rationalité limitée a engendré la création d'organisations artificielles de résolution de problèmes. Les outils les plus puissants pour la gestion de la complexité sont la modularité et l'abstraction. Les systèmes multi-agents offrent la modularité.

Si le domaine d'un problème est particulièrement complexe, large ou imprévisible, la seule manière raisonnable de le traiter est de développer un nombre de composants fonctionnels, spécifiques et modulaires (agents), qui sont chacun spécialisés dans la résolution d'un aspect particulier du problème. En effet, les problèmes réels impliquent des systèmes ouverts distribués. Un système ouvert est celui dont la structure est capable de changer de façon dynamique. Les composants d'un tel système ne sont pas connus d'avance, peuvent changer au cours du temps et peuvent être constitués d'agents hétérogènes implémentés par des personnes différentes, à des instants différents et avec des techniques et outils différents. Quand des problèmes interdépendants surgissent, les agents doivent être capables d'interagir et de se coordonner pour gérer cette interdépendance, en se munissant de techniques basées sur la négociation ou la coopération, ce qui relève du domaine des systèmes multi-agents. Ainsi, il est d'autant plus clair que, pour être réussies, les recherches doivent accorder plus d'importance aux systèmes composés non pas d'un seul agent, mais de plusieurs. Pour résumer, [Ferber, 95] stipule que la nécessité de la distribution de l'activité et de l'intelligence peut être expliquée par les raisons suivantes :

- Les problèmes sont physiquement distribués ;
- Les problèmes sont fonctionnellement très distribués et hétérogènes ;
- Les réseaux imposent une vision distribuée ;

- La complexité des problèmes impose une vision locale ;
- Les systèmes doivent pouvoir s'adapter à des modifications de structure ou d'environnement ;
- Le génie logiciel va dans le sens d'une conception en terme d'unités autonomes en interactions.

Ainsi l'Intelligence Artificielle Distribuée, sous-domaine de l'Intelligence Artificielle, s'est développée depuis plus de 20 ans pour s'intéresser à une société d'agents interagissant, dans le but de résoudre un problème commun lié aux ordinateurs, aux personnes, aux capteurs, aux avions, aux robots, etc. Une telle société est nommée système multi-agents, SMA [Green 97].

II.9.2 Caractéristiques des agents

II.9.2.1 Notion d'agent

Plusieurs définitions attribuées à la notion d'agent existent dans la littérature. Nous avons cependant jugé utile de noter la définition du terme agent proposée par [Florez, 99], car elle est la plus explicite et basée sur d'autres définitions.

Définition 1 : Un agent est une entité interactive qui existe en tant que partie d'un environnement partagé par d'autres agents. C'est une entité conceptuelle, qui perçoit et agit avec initiative ou en réaction, dans un environnement où d'autres agents existent et interagissent les uns avec les autres, sur la base de connaissances partagées de communication et de représentation.

Dans [Florez, 99], afin de mieux saisir la notion d'agent, nous pouvons relever quelques propriétés qui peuvent être attribuées aux agents :

- Adaptabilité : capacité à apprendre et à s'améliorer avec l'expérience ;
- Autonomie : capacité d'agir seul, pour atteindre des buts ;
- Comportement collaboratif : capacité à travailler avec d'autres agents pour un objectif commun ;
- Capacité inférentielle : capacité d'agir avec des spécifications abstraites des tâches ;
- Capacité de communication au niveau des connaissances : capacité de communiquer avec les autres agents avec un langage comme celui des êtres humains ;
- Personnalité : capacité à manifester des attributs d'un caractère humain crédible ;
- Réactivité : capacité à détecter et réagir ;
- Continuité temporelle : persistance d'une identité et d'un état sur une longue période.

Définition 2 : Un agent est une entité autonome, réelle ou abstraite, qui est capable d’agir sur elle-même et sur son environnement, qui, dans un univers multi-agents, peut communiquer avec d’autres agents, et dont le comportement est la conséquence de ses observations, de ses connaissances et des interactions avec les autres agents [Ferber 95].

Il ressort de cette définition des propriétés clés, comme l’autonomie, l’action, la perception et la communication. D’autres propriétés peuvent être attribuées aux agents. Nous citons en particulier la réactivité, la rationalité, l’engagement et l’intention.

II.9.2.2 Raisonement individuel des agents

Le raisonnement individuel des agents peut améliorer la cohérence des systèmes multi-agents, car chaque agent réfléchit sur les effets non locaux que peuvent avoir les actions locales, essaie de prévoir le comportement des autres agents et explique où résoudre les conflits et les mauvaises interactions. Par ailleurs, faut-il concevoir les agents comme des entités « intelligentes », capables de résoudre les problèmes par eux-mêmes, ou bien faut-il les assimiler à des entités simples, réagissant directement aux variations de l’environnement ? La réponse à cette question a donné naissance à trois écoles de pensée relatives aux agents cognitifs, réactifs et hybrides.

a- Agent cognitif

La notion d’agent a été développée par le biais de la psychologie humaine avec laquelle le comportement humain est prévu et expliqué à travers l’attribution d’attitudes. Ainsi, les agents cognitifs disposent d’une base de connaissances et de plans explicites leur permettant d’atteindre leurs buts. Par ailleurs, des modèles d’agent ont été formalisés dans le sens des croyances, des désirs, des intentions, des objectifs, etc.

b- Agent réactif

Un agent peut être réactif, il n’a ainsi pas de représentation de son environnement. Il agit avec un comportement de stimulus, réponse et réagit à l’état présent de l’environnement dans lequel il est situé. Ce genre d’agents ne tient pas compte du passé et ne planifie pas le futur. A travers des interactions simples avec les autres, le comportement global complexe peut apparaître. Ainsi, les agents n’ont pas à réviser le modèle de leur environnement quand il change, ce qui présente un grand avantage et rend les systèmes réactifs plus rapides. Les

propriétés principales de ces systèmes sont la robustesse et la tolérance aux fautes. En effet, un groupe d'agents peut compléter une tâche quand l'un d'eux échoue.

Par contre, les agents réactifs ont un comportement myope, puisqu'ils ne prévoient pas l'effet des décisions locales sur le comportement global du système. En plus, il est difficile de gérer les agents pour accomplir des tâches complexes, car la relation entre les comportements individuels et le comportement global du système n'est pas bien saisie.

c- Agent hybride

La plupart des problèmes ont une architecture ni purement réactive, ni purement cognitive, ce qui n'est pas approprié, il faut une architecture hybride adaptée. Ainsi, un agent hybride possède des composants réactifs et aussi des composants cognitifs, pour garantir un raisonnement de qualité.

II.9.3 Caractéristiques des systèmes multi-agents

II.9.3.1 Société d'agents

Un système multi-agents est un système distribué composé d'un ensemble d'agents. Contrairement aux systèmes d'IA, qui simulent dans une certaine mesure les capacités du raisonnement humain, les SMA sont conçus et implantés idéalement comme un ensemble d'agents interagissant, le plus souvent, selon des modes de coopération, de concurrence ou de coexistence [Chaib-draa94], [Chaib-draa96].

Les systèmes multi-agents sont des réseaux d'agents (solveurs) faiblement couplés, qui coopèrent pour résoudre des problèmes qui dépassent les capacités ou les connaissances individuelles de chaque agent. Les agents sont autonomes et peuvent être de nature hétérogène. En effet, comme l'illustre la figure II.13, un système multi-agent est composé d'un environnement d'objets passifs qui sont manipulés par les agents, d'agents représentant les entités actives du système, d'interactions entre les agents et d'opérations qui permettent les perceptions et les différentes actions.

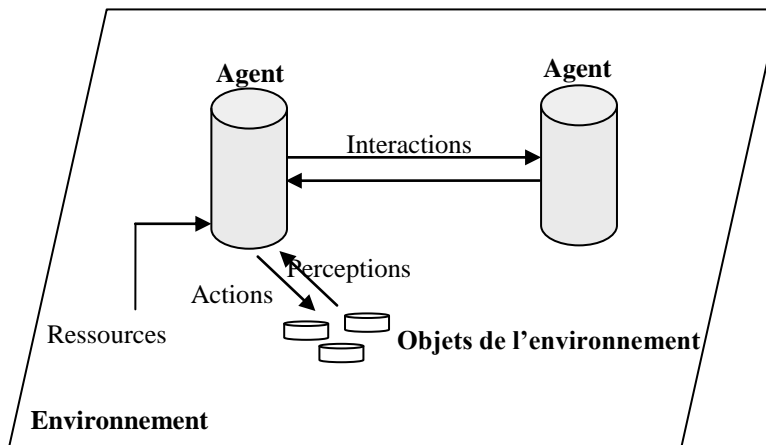


Figure II.13: Représentation d'un système multi-agent.

Ainsi, les agents d'un SMA, n'ayant qu'une vision locale de leur environnement, sont amenés à coopérer, et donc à communiquer, afin d'atteindre le but global du réseau. Aussi, il est nécessaire de définir la structure organisationnelle dans la société d'agents et d'y établir :

- la communication entre les agents,
- la coordination et la planification des tâches des agents,
- la négociation entre les agents pour la résolution des conflits.

En effet, dans tout type d'organisation, pour pouvoir résoudre avec cohérence les problèmes, les agents doivent communiquer entre eux, coordonner leurs activités et négocier dans les situations de conflits. Les conflits peuvent résulter d'une simple ressource limitée à des calculs plus complexes, où les agents tombent en désaccord, à cause des différences entre leurs domaines d'expertise. La coordination est nécessaire pour déterminer la structure organisationnelle d'un groupe d'agents et pour l'allocation des ressources et tâches. La négociation est nécessaire pour la détection et la résolution des conflits.

II.9.3.2 Organisations dans les systèmes multi-agents

a- Structures organisationnelles

Une organisation est constituée d'un groupe d'agents, d'un ensemble d'activités effectuées par les agents, d'un ensemble de connexions entre ces entités et d'un ensemble de buts ou de critères d'évaluation, avec lesquels les activités combinées des agents sont évaluées. Les structures organisationnelles imposent des contraintes sur la nature de la communication ou de la coordination.

b- Coordination

La coordination est un processus dans lequel les agents se sont engagés en vue d'assurer une communauté d'agents individuels agissant avec cohérence et harmonie. Les agents ont en fait besoin de la coordination pour empêcher les comportements chaotiques, pour être coordonnés de la même manière, puisqu'aucun agent ne possède une vue globale sur le système et parce qu'ils possèdent des capacités et expertises différentes, mais complémentaires.

Il est clair que la coordination s'avère plus facile dans les situations de routine que dans les situations non familières. En effet, dans la routine, les agents peuvent être parfaitement coordonnés, car on pourrait savoir ce qu'ils sont en train de faire et prévoir ce qu'ils vont faire. Les situations peuvent être de routine, familières ou non familières. Dans les situations familières, les agents peuvent être coordonnés selon des lois sociales, ce qui n'est pas possible dans les situations non familières.

La coordination comprend aussi l'allocation des tâches, qui consiste à affecter des responsabilités et des ressources nécessaires à la résolution de problèmes à un agent. Le créateur du système d'agents peut allouer toutes les tâches d'avance en engendrant ainsi une organisation de résolution des problèmes qui est non adaptable. Par contre, on peut avoir une allocation dynamique et flexible des tâches.

La planification fait également partie de la coordination. Pour un agent, elle constitue un processus de construction d'une séquence d'actions en tenant compte seulement des objectifs, des capacités et des contraintes environnementales. La planification a pour rôle d'éviter les conflits. Elle peut être centralisée ou distribuée.

c- Négociation

La négociation est un processus de communication d'un groupe d'agents, qui a pour objectif d'atteindre un accord mutuellement accepté dans une affaire donnée. Elle est donc considérée comme une méthode de coordination et de résolution de conflits. Le processus de négociation peut induire des échanges d'informations, des relaxations des buts initiaux, des concessions mutuelles, des mensonges ou des menaces. Elle relève ainsi principalement de la recherche d'un consensus. Elle peut être compétitive ou coopérative.

La négociation compétitive est utilisée dans les situations où des agents d'intérêts différents tentent de faire un choix de groupe avec des alternatives bien définies. Les agents sont néanmoins compétitifs et non coopératifs.

La négociation coopérative concerne les situations où les agents ont un but unique global envisagé pour le système. Il s'agit ainsi de systèmes distribués qui ont été conçus pour réaliser un seul objectif global. Ces agents sont aussi dits collaboratifs.

d- Communication

Les agents ont besoin de communiquer pour pouvoir interagir et échanger de l'information. Ils peuvent interagir soit en accomplissant des actions linguistiques (en communiquant entre eux), soit en accomplissant des actions non-linguistiques, qui modifient leur environnement. En communiquant, les agents peuvent échanger des informations et coordonner leurs activités. Dans les SMA, deux stratégies principales ont été utilisées pour supporter la communication entre agents : les agents peuvent échanger des messages directement, ou ils peuvent accéder à une base de données partagée (appelée tableau noir ou "blackboard") dans laquelle les informations sont postées. Les communications sont à la base des interactions et de l'organisation sociale d'un SMA [Chaib-draa 96].

Il existe plusieurs langages de communication, qui se basent sur des actes avec des locutions comme « demander » ou « commander ». Le plus connu parmi ces langages est le KQML, « Knowledge Query Manipulation Language » [Labrou 97], et FIPA-ACL (« Foundation for Intelligent Physical Agent-Agent Communication Language ») [FIPA 97] [FIPA 99]. Ces deux langages de communication entre agents, ont émergé des efforts de standardisation de la communauté des systèmes multi-agents [Jouvin 00].

FIPA ACL est syntaxiquement similaire à KQML, mis à part certains noms de primitives réservées. La particularité de FIPA – ACL consiste en un ensemble de messages avec une sémantique associée, c'est-à-dire les conditions que doivent respecter l'expéditeur et les effets attendus sur le destinataire.

II.10 Applications des SMA

II.10.1 Applications générales

Les SMA ont des applications diverses qui ont évolué avec le développement des outils technologiques, et surtout de l'Internet [Chaib-draa 96] :

- Applications industrielles : gestion de la production manufacturière [Kabachi 96], contrôle et commande des processus (accélérateur de particules), télécommunication

(gestion et contrôle des réseaux, transmission, etc.) et systèmes de transport (surveillance des véhicules automatisés (DVMT : « Distributed Vehicle Monitoring Task »), job shop flexible [Ennigrou 04], systèmes industriels flexibles [Berger 02], etc.

- Applications commerciales : gestion de l'information (Internet, filtrage, collecte), commerce électronique (agence de voyage), gestion des affaires, etc.
- Loisirs : Jeux, théâtre et cinéma interactifs.
- Applications médicales : orientation des patients, gestion des soins.

Le service de l'agence de voyage est un exemple classique dans le domaine des systèmes de commerce électronique, souvent pris comme étude de cas ([FIPA 97] [Linden 96], etc.).

Le système multi-agents de l'agence de voyages est constitué d'un ensemble d'agents reliés par l'intermédiaire de l'Internet et qui fournissent divers types de services :

- des agents représentant des compagnies aériennes : Air France, United Airlines, British Airways, etc. Ils fournissent des services de transport aérien ;
- des agents représentant des compagnies ferroviaires : SNCF par exemple ;
- des agents représentant des hôtels ou des compagnies de location de voitures.

II.10.2 Applications des SMA dans le domaine du transport

Parmi les premières applications développées à l'aide de la technologie multi-agent, on retrouve une application dans le contrôle du trafic aérien, une autre dans la surveillance de véhicules motorisés [Durfee 91] et les systèmes d'aide au pilotage des avions militaires (surveillance des capteurs, collecte, vérification et analyse des données, proposition de mesures correctives ou de plans optionnels au pilote pour qu'il puisse atteindre le but de sa mission) [Chaib-draa 96].

En ce qui concerne le contrôle du trafic aérien, [Ljungberg 92] propose un SMA, nommé OASIS (*Optimal Aircraft Sequencing using Intelligent Scheduling*), pour le contrôle du trafic aérien de la région de Sydney en Australie. Il a pour but de réduire la congestion du trafic aérien en maximisant l'utilisation des pistes en ordonnant les atterrissages. Ce système possède deux types d'agents : des agents « globaux », qui traitent les relations entre les avions, la coordination et le raisonnement (COORDINATEUR, SEQUENCEUR, VERIFICATEUR DE TRAJECTOIRE, MODELE DU VENT, INTERFACE UTILISATEUR) et des agents avion, responsables des calculs et du raisonnement associés à chaque avion (position, vitesse, heure d'atterrissage, etc.). Dans le contrôle de trafic aérien,

[Cammarata 83] a étudié des stratégies de coopération pour pouvoir résoudre les conflits entre les plans d'un groupe d'agents. Ces stratégies ont pu être ensuite appliquées dans le cadre d'un système de contrôle du trafic aérien. Dans ce système, chaque agent (représentant un avion) cherche à bâtir un plan de vol qui devrait le garder à une distance de sécurité des autres agents. Dans le cas où des agents se retrouveraient dans une situation conflictuelle, ils doivent alors choisir, parmi eux, un agent apte à élaborer un nouveau plan de vol sans engendrer de nouveaux conflits. Pour faire ce choix, Cammarata et ses collègues [Cammarata 83] ont opté pour deux stratégies qu'ils ont comparées entre elles. La première a consisté à choisir l'agent le mieux informé, ou l'agent le moins contraint, pour jouer le rôle de planificateur central et élaborer un nouveau plan de vol qui résoudrait le conflit. La deuxième a consisté à faire le partage suivant : il revient à l'agent le mieux informé d'élaborer un nouveau plan de vol et à l'agent le moins contraint de l'exécuter.

Fischer [Fischer 95] présente une plate-forme, « MARS », de simulation multi-agent pour le transport des marchandises. Elle décrit un scénario lié à des compagnies de transport géographiquement distribuées, qui gèrent des commandes arrivant en temps réel. Le système comprend deux types d'agents correspondant aux compagnies de transport et aux camions. Les agents COMPAGNIES doivent affecter les ordres aux camions ou coopérer avec les autres agents du même type pour qu'ils se chargent du ou des ordres. La coopération se base sur une négociation à l'aide d'une version évoluée du « Contract Net Protocole ». Le comportement et la cohérence du système sont évalués à partir des coûts associés aux ordres ordonnancés. Un scénario pour le transport des frets par trains est aussi présenté dans [Lind 98].

Chaib-draa [Chaib-draa 96] aborde le problème de gestion du trafic urbain par une approche agent. Les agents peuvent correspondre à des hommes (conducteurs, policiers, piétons) ou des machines (véhicules, feux de circulation, etc.) et doivent constamment ajuster leurs actions pour éviter les bouchons et les accidents. Les agents sont dotés alors de lois sociales pour que leurs fonctions soient basées sur des compétences, ce qui implique des activités de coordination rapide et sans effort entre eux. La performance des agents est guidée par des modèles stockés de procédures prédéfinies, qui vont de la perception ou de l'observation à l'action. Les situations non familières sont adaptées à des situations familières en utilisant des raisonnements basés sur des cas. Le modèle d'agent est ainsi basé sur trois niveaux : compétences (pour la routine), règles (situations familières) et connaissances (situations non familières).

Dans [Gruer 01], Gruer propose une approche multi-agent pour la modélisation et la simulation des systèmes de transport. Ce système est composé des agents bus, arrêt et section. Aussi [El Hmam 05] ont adopté le paradigme agent pour modéliser le flux de trafic au niveau microscopique. Par analogie avec les systèmes multi-agents, ils représentent un tronçon comme étant un environnement et les véhicules comme étant des agents. Ces derniers sont censés s'auto-organiser afin d'évoluer correctement (en évitant les accidents) dans leur environnement.

[Saussoil 00] a modélisé un réseau de transport urbain par un SMA dans le but d'assister le régulateur dans la régulation des correspondances. Un agent modélise une entité active du processus de transport. Il peut être un agent BUS, STATION, SECTION, DEPOT et INTERFACE.

Toujours dans le cadre de la régulation des correspondances, [Laïchour 01] présente un modèle multi-agent qui se base sur trois types d'agents :

- l'agent ACQUISITION, qui effectue la gestion des données relatives aux passages des bus aux arrêts de régulation ;
- l'agent CORRESPONDANCE, qui s'occupe de la détection et du diagnostic des perturbations au niveau des correspondances, ainsi que de la proposition de décisions ;
- l'agent SUPERVISEUR, qui a un rôle d'interface entre le régulateur et le système d'aide à la régulation des correspondances.

[Balbo 00] propose aussi un SMA d'aide à la décision pour la régulation d'un réseau de transport par bus. Ce système est basé sur un module d'interaction qui utilise l'environnement comme support de communication. Sa particularité réside dans son fonctionnement sans horaires préétablis pour les véhicules, mais suivant une gestion dynamique du planning. Il est composé d'agents :

- ARRET, qui assurent en conditions normales la gestion des horaires en temps réel. Ils définissent les horaires de passage, à travers la distance du parcours et le temps nécessaire, pour que les véhicules les suivent ;
- BUS, qui ont pour rôle d'exécuter les courses qui leur sont affectées et de proposer des mesures de régulation ;
- INCIDENT, qui sont responsables de la gestion des perturbations. Ils synthétisent les informations, afin d'évaluer la difficulté du problème et l'adéquation des solutions proposées par les agents BUS ;

- ZONE ARRET, qui assurent la gestion des informations concernant un problème, afin de les rendre disponibles aux agents INCIDENT. Ces agents calculent pour l'ensemble des arrêts concernés : la demande théorique, la difficulté de circulation, l'étude en mètres et en nombre d'arrêts et le temps gagné ou perdu par le véhicule précédent.

La surveillance de véhicules motorisés tournait, quant à elle, autour du système DVMT (*Distributed Vehicle Monitoring Task*) [Corkill 83]. Ce système avait comme tâche principale d'identifier quel type de véhicule circulait dans la zone où étaient placés ses capteurs. À partir de ces interprétations, il devait tenter d'établir une description des mouvements des véhicules dans la région qu'il supervisait. Comme bien d'autres systèmes multi-agents de l'époque, le système DVMT utilisait un « blackboard » pour coordonner les efforts de ses différents agents. Rappelons qu'un « blackboard » est simplement une structure de données partagées entre divers agents. Ces derniers peuvent la consulter pour obtenir des informations sur l'état actuel du problème ou y écrire la partie de la solution qu'ils ont obtenue.

II.11 Conclusion

Ce chapitre était consacré à la modélisation du réseau de transport en commun multimodal par des graphes. Cette modélisation a montré les caractéristiques du problème d'aide au déplacement. Afin de choisir une méthode performante pour résoudre ce problème, nous avons étudié les problèmes NP-difficiles, l'optimisation multiobjectif, ainsi que les approches existantes pour les résoudre. Bien que les méthodes d'optimisation pour les problèmes de classe NP-difficile soient nombreuses, nous avons approfondi notre recherche avec les algorithmes évolutionnaires. Aussi, nous avons présenté dans ce chapitre une introduction aux systèmes multi-agents et leurs caractéristiques. Notre objectif dans ce chapitre était de bien étudier les méthodes existantes afin de présenter notre approche dans le chapitre suivant. Cette approche consiste en l'utilisation de la théorie des graphes et des algorithmes génétiques pour optimiser des déplacements selon plusieurs critères. La deuxième partie concerne l'implémentation de notre système en se basant sur les systèmes multi-agents.

Chapitre III

Approche de résolution pour un Système Interactif d'Aide au Déplacement Multimodal (SIADM)

III.1 Introduction

Dans ce chapitre, nous présentons notre approche pour résoudre le problème d'aide au déplacement dans les réseaux de transport en commun multimodaux. En se basant sur la modélisation graphique réalisée dans le chapitre précédent, l'optimisation d'un déplacement dans un multigraphe exige l'utilisation de méthodes d'optimisation plus puissantes que l'algorithme de Dijkstra. En effet, les paramètres graphiques, les critères d'optimisation ainsi que les contraintes temporelles et la gestion des perturbations rendent notre problème *NP-difficile*.

Nous utilisons ici une méthode multicritère de recherche d'itinéraire qui s'appuie sur une hybridation entre un algorithme de Dijkstra modifié [Zidi_b 04] et un algorithme génétique, pour trouver une population de chemins minimums [Zidi 04]. L'algorithme de Dijkstra modifié nous donne un ensemble de solutions servant de population initiale pour l'algorithme génétique. Notre objectif est d'optimiser le déplacement du client suivant plusieurs critères et de lui proposer rapidement des solutions alternatives en cas de perturbation. Afin d'atteindre ce dernier but, nous contrôlons nos opérateurs génétiques pour augmenter la probabilité de trouver ces solutions alternatives dans la population finale.

Au début de ce chapitre, nous étudierons la complexité du problème d'aide au déplacement ainsi que les critères à optimiser. La partie suivante est consacrée à notre approche pour l'optimisation. Cette partie est dédiée, en premier lieu, à l'algorithme de Dijkstra et aux améliorations à lui ajouter, et en second lieu à l'algorithme génétique et ses caractéristiques.

Les réseaux de transport urbain formant des systèmes ouverts, distribués et complexes, leur gestion par des SMA est bien adaptée. Ainsi, nous présentons dans ce chapitre un Système Multi-Agent d'Aide au Déplacement, SMAAD, pour un réseau de transport multimodal.

III.2 Complexité du problème

Nous appelons graphe multimodal le plan des itinéraires de transport en commun multimodal. Généralement ces graphes sont très denses et contiennent beaucoup de nœuds de correspondance. Ces propriétés engendrent des échanges intermodaux dans ces nœuds de correspondance. Cet échange a un coût, qui peut être le temps d'attente, le prix ou le confort. Si nous considérons que chaque mode correspond à un élément, cet élément est représenté par son nœud de départ, son nœud d'arrivée et un tableau « Tcorrespondance ». Ce tableau représente les modes qui ont des nœuds de correspondance avec cet élément, ainsi que les coûts d'échange dans chaque nœud. Dans cette représentation, un déplacement dans un réseau multimodal est équivalent à un cheminement entre ces éléments. Généralement, l'utilisateur demande de l'aide au déplacement d'un point de départ vers un point de destination avec le retour à son point de départ. Nous cherchons les nœuds de correspondance entre les nœuds de départ et d'arrivée, ainsi ce déplacement est équivalent au cheminement entre plusieurs points, en passant par ces nœuds une et une seule fois. Dans la littérature, ce problème est équivalent à celui du voyageur de commerce.

| paramètres | Problème de voyageur de commerce | Notre problème |
|--------------------|--|--|
| Ensemble de noeuds | Les villes | Les nœuds de correspondance |
| contrainte | Passage une et une seule fois par chaque ville | -De même passage une seule fois par chaque nœuds -Contraintes temporelles |
| critères | Distance, temps, coût... | Distance, temps, coût... |
| solution | Circuit hamiltonien | Circuit hamiltonien |

Tableau III.1 : Analogie entre le problème de voyageur de commerce et notre problème

Suite à cette analogie entre le problème de voyageur de commerce et une instance de notre problème de déplacement, il est aisé de montrer que notre problème à plus de difficultés que celui de voyageur de commerce.

La complexité du problème de voyageur de commerce n'a pas facilité son étude et si on analyse tous les parcours possibles, pour n villes, le nombre de possibilités est de $(n-1)!$. Pour 6 villes nous avons 120 possibilités, pour 10 villes, plus de 362 000 et pour 60 villes plus de 10^{80} , soit le nombre d'atomes estimé dans l'univers. Ceci peut expliquer pourquoi le problème n'a pas été étudié sérieusement avant l'arrivée des ordinateurs dans les universités mais, depuis 1954, de nombreux chercheurs l'ont traité.

La complexité du problème d'aide au déplacement dans un réseau de transport multimodal est équivalente à celle du PVC multiple, car il y a plusieurs points de départ et plusieurs points d'arrivée, mais, dans l'aide au déplacement, nous avons d'autres problèmes à résoudre, comme la gestion des perturbations, le traitement en temps réel des requêtes et la gestion des demandes multiples. Ces problèmes justifient que la problématique étudiée dans cette thèse appartienne à la classe NP-difficile.

III.3 Normalisation des critères

Notre objectif est d'aider l'utilisateur de transport en commun à optimiser ses déplacements selon plusieurs critères. Dans notre application, nous optimisons trois critères : le coût, le temps et le confort. Ces critères d'optimisation ne sont pas dans le même intervalle de valeur. Afin que l'optimisation ne soit pas influencée par un de ces critères qui a des valeurs très grandes par rapport aux autres, nous avons ajouté une étape de normalisation des critères. Pour rendre les trois critères dans un même intervalle, nous avons utilisé la méthode de Reardon. Cette méthode de traitement multiobjectif utilise la logique floue [Siarry 02] [Reardon 97]. Avec cette méthode, on définit une fonction d'appartenance pour chaque critère, ce qui permet de donner une valeur $\mu(C)$ à chaque critère C dans l'intervalle $[0,1]$.

De cette manière, on a pu empêcher l'influence de la différence entre les unités de mesure des différentes fonctions objectifs, et aussi minimiser les effets dus aux différences de plages de variation entre les fonctions objectifs.

Parmi les trois critères que nous utilisons, le plus difficile est de déterminer les valeurs maximales (C_{\max}) et minimales (C_{\min}) des critères. Pour le premier critère, qui représente le prix, nous avons remarqué que les valeurs sont dépendantes des zones, ainsi que de la période. En général, dans la même zone, les exploitants de transport en commun définissent un prix commun donc le $C_{1\max}$ est le maximum des prix sur cette zone. Pour les déplacements entre plusieurs zones, $C_{1\max}$ est égale à $\text{Max}(C_1)$ et $C_{1\min} = \text{Min}(C_1)$. Pour simplifier les opérations

de calcul à faire avant l'optimisation, nous avons choisi d'utiliser un C_{1min} prédéfini. Pratiquement, C_{1min} est égal à la somme des billets les moins chers sur les zones à traverser. Pour chaque valeur de C_1 , nous calculons sa valeur correspondante $\mu(C_1)$, selon l'équation de la droite (Figure III.1). Nous appliquons le même principe pour les deux autres critères C_2 et C_3 .

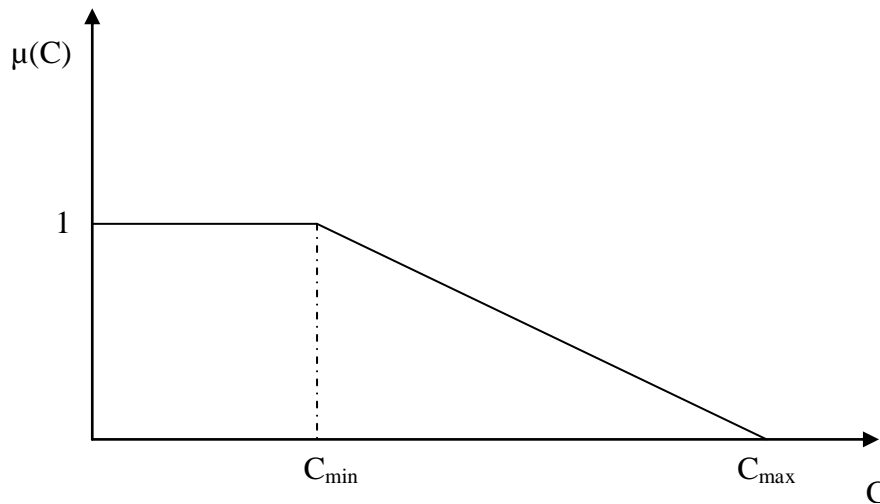


Figure III.1: Fonction de normalisation des critères

III.4 Calcul de plus court chemin

Les problèmes de chemins optimaux sont très fréquents dans les applications pratiques. Par exemple, on les rencontre dès qu'il s'agit d'acheminer un objet entre deux points d'un réseau, de façon à minimiser un coût, une durée, etc. Ils apparaissent aussi en sous-problèmes combinatoires, notamment les flots dans les graphes et les ordonnancements. Le nombre de ces applications a motivé très tôt la recherche d'algorithmes efficaces [Lacomme 03].

L'optimisation d'un voyage s'intéresse souvent à plusieurs critères, tels que le temps de parcours, le coût, le confort, etc. Les utilisateurs de transport en commun cherchent souvent à les optimiser ensemble. L'optimisation de ces critères est parfois difficile, parce que l'optimisation de l'un entraîne la dégradation des autres. Dans ce sens, nous proposons une fonction d'agrégation f qui calcule le coût de chaque arc du graphe du réseau. L'objectif est de minimiser le coût total du chemin qui représente la somme des coûts des arcs traversés. Les coefficients de pénalité sont calculés suivant le choix de l'utilisateur. Il a le choix de

favoriser un critère par rapport aux deux autres, ou bien de leur donner des importances égales.

$$f = \alpha.\mu(C_1) + \beta.\mu(C_2) + \delta.\mu(C_3) \quad (1)$$

C_1 , C_2 et C_3 représentent respectivement les critères d'optimisation, tels que le coût, le temps de parcours et le confort.

$\mu(C_1)$, $\mu(C_2)$ et $\mu(C_3)$ sont les valeurs correspondant à C_1 , C_2 et C_3 dans l'échelle [0,1]. Ces valeurs sont calculées par la méthode de Reardon présentée dans la section III.3.

C_1 est calculé par la formule suivante :

$$C_1 = \lambda \sum_{i=0}^z C_3^i \sum_{k=0}^M \sum_{j=0}^N C_1^{jk} \quad (2) \quad M : \text{nombre de modes, } N \text{ nombre d'arcs.}$$

L'indice i (de 0 à z) représente les zones. C_3^i représente le coefficient de confort correspondant à la zone i . Cette variable est calculée par la formule (4). C_1^{jk} est le prix du titre de transport de l'arc j en utilisant le mode k . $\lambda \leq 1$ est le coefficient utilisé par les modes de transport pour le calcul du prix inter-zones.

$$C_2 \equiv T \quad T = \sum_{i=0}^p t_i^{m_i} + \sum_{j=0}^n T_{Att}^j \quad (3)$$

n : nombre de stations, p nombre d'arcs utilisés dans la solution

$t_i^{m_i}$ est le temps de parcours de l'arc i par le mode de transport m_i . T_{Att}^j est le temps d'arrêt dans la station j .

$$C_3 = \sigma.N + \mu. \sum_{\substack{j=1 \\ k \in \text{mode}}}^N C_3^{jk} \quad (4)$$

N est le nombre de changements de mode dans un déplacement. σ est une valeur qui varie selon le souhait de l'utilisateur. Cette valeur est comprise dans l'intervalle [0,1]. μ est une valeur déterminée par les exploitants de transport selon les zones et les plages horaires. C_3^{jk} est la valeur du troisième critère pour l'arc j en utilisant le mode k . Alors que les α , β et δ sont des coefficients de pénalité pour ces critères, leur somme est égale à 1. Cette fonction

d'agrégation f ramène le problème d'optimisation multicritère à un problème d'optimisation monocritère. Mais le fait d'utiliser une méthode évolutionnaire donne à notre système plus de flexibilité. Cette méthode nous permet de proposer plusieurs solutions à l'utilisateur ce qui lui permet de choisir celle qui lui convient le mieux.

III.5 Algorithme de Dijkstra

III.5.1 Algorithme de Dijkstra classique

Publié en 1959, l'algorithme de Dijkstra est une alternative à celui de Floyd et plus complexe, mais également beaucoup plus rapide. Il n'est valable que pour les graphes à évaluations positives ou nulles, qui ne contiennent donc pas de circuits négatifs. A chaque itération, un sommet x reçoit sa distance par rapport au sommet de départ définitive, nous disons qu'il est marqué.

L'algorithme de Dijkstra est un algorithme de type glouton : à chaque nouvelle étape, on traite un nouveau sommet. Reste à définir le choix du sommet à traiter, et le traitement à lui infliger.

Tout au long du calcul, on va donc maintenir deux ensembles :

- C , l'ensemble des sommets qui restent à visiter ; au départ $C=S-\{source\}$
- D , l'ensemble des sommets pour lesquels on connaît déjà leur plus petite distance à la source ; au départ, $D=\{source\}$.

L'algorithme se termine lorsque C est vide.

Pour chaque sommet s dans D , on conservera dans un tableau « *distances* » le poids du plus court chemin jusqu'à la source, et dans un tableau « *parcours* » le sommet p qui le précède dans un plus court chemin de la source à s . Ainsi, pour retrouver un chemin le plus court, il suffira de remonter de prédécesseur en prédécesseur jusqu'à la source, ce qui peut se faire grâce à un unique appel récursif.

Initialisation

Au début de l'algorithme, le chemin le plus court connu entre la source et chacun des sommets est le chemin direct, avec une arête de poids infini, s'il n'y a pas de liaison entre les deux

sommets. On initialise donc le tableau *distances* par les poids des arêtes reliant la source à chacun des sommets, et le tableau *parcours* par *source* pour tous les sommets.

i^{ème} étape

On suppose avoir déjà traité *i* sommets, « *parcours* » et « *distances* » contiennent respectivement les poids et les prédécesseurs des plus courts chemins pour chacun des sommets déjà traités. Soit $F(s,t)$ le coût de l'arc $[s,t]$.

Soit *s* le sommet de *C* réalisant le minimum de *distances*[*s*]. On supprime *s* de *C* et on l'ajoute à *D*. Reste à mettre à jour les tableaux « *distances* » et « *parcours* » pour les sommets *t* reliés directement à *s* par un arc comme suit : si $distances[s] + F(s,t) < distances[t]$, alors on remplace *distances*[*t*] par $distances[s] + F(s,t)$ et *parcours*[*t*] par *s*.

(n-2)^{ème} étape

Au départ, il y a $(n-1)$ sommets à visiter, mais comme on le verra ci-après, la dernière étape est inutile puisqu'elle n'apporte rien. Ainsi, dès la $(n-2)$ ^{ème} étape, « *distances* » et « *parcours* » contiennent toute l'information nécessaire pour trouver des plus courts chemins de la source à chacun des autres sommets :

- *distances*[*s*] est le poids du plus court chemin de la source à *s*
- *parcours*[*t*] est le prédécesseur de *s* dans un plus court chemin de la source à *s*

L'inconvénient majeur de l'algorithme de Dijkstra est qu'il est sensible à la densité du graphe [Lacomme 03]. Le nombre d'itérations du « Répéter », au plus *N*, ne peut pas être amélioré par construction de l'algorithme. En revanche, l'essentiel du travail est dû à la boucle interne trouvant le prochain sommet *s* à fixer.

Dans la seule agglomération lilloise, on peut dénombrer plus de 1000 nœuds. Sur toute la France le nombre de nœuds total dépasse 100 000 ; si *p* est le nombre d'arcs du graphe et *n* le nombre de sommets, la complexité de l'algorithme est au mieux en $O(\max(p,n) \cdot \log(n))$ [Lacomme 03]. Sauf que le problème étudié est plus compliqué que celui de plus court chemin simple, si nous ajoutons les contraintes temporelles, multigraphe et multicritère. La complexité du problème implique qu'il est impossible de résoudre notre problème par un simple algorithme de Dijkstra : le temps de calcul qui en découle serait bien trop important.


```

Initialiser D à source
Initialiser C à S - {source} : l'ensemble des sommets non fixés
distances[source]= 0
parcours[source]=source

Répéter
Etape1 : Chercher sommet s non fixé de distance minimale
    Si distances[s] < +∞
        Ajouter s dans D
        Supprimer s de C
    Fin de si
Pour tous les sommets t de C faire
    Si distances[s] + F(s,t) < distances[t] alors
        distances[t]= distances[s] + F(s,t)
        parcours[t]= s
    fin de si
Fin de pour
Si C non vide alors retour à l'étape1
Sinon fin

```

Figure III.2: Algorithme de Dijkstra classique.

Pour contourner ce problème, on se permet une approximation dont l'origine est assez naturelle : pour aller d'un point dans une ville à un autre point dans une ville différente, nous essayons de passer par des points de correspondance, comme les grandes villes par exemple, afin de construire l'itinéraire. Voici à quoi correspond cette approximation lorsqu'elle est appliquée à notre système :

- Le territoire est subdivisé en zones d'influence, une zone par agglomération dotée d'un réseau de transports en commun. A chaque station de cette zone d'influence est associé un niveau, local ou national.
- Lorsque l'utilisateur veut aller d'une station locale à une autre station locale dans la même zone d'influence, le nombre de nœuds et d'arcs est suffisamment faible pour que l'on puisse utiliser l'algorithme de Dijkstra.
- Lorsque l'utilisateur veut aller d'une station à une autre et que les zones d'influence de ces stations sont différentes, le programme commence par calculer tous les trajets possibles entre la station de départ (son niveau est « local ») et les stations de niveau national ou bien les nœuds de correspondance avec les autres zones (il y en a peu, en général moins de 5).

Pour chacune de ces stations de niveau national, il calcule un trajet jusqu'à chaque station de niveau national dans la zone d'influence de la ville d'arrivée. Enfin, pour chacun de ces trajets partiels calculés, il détermine la fin du trajet, entre les stations de niveau national de la zone d'influence d'arrivée et la station de niveau local. Les trois trajets partiels sont calculés à l'aide de l'algorithme de Dijkstra.

Propriété 1 « Théorème de Bellman » : Les plus courts chemins possèdent la propriété suivante : *Tout sous-chemin d'un plus court chemin est lui même un plus court chemin.*

Si $p=(s,\dots,t)$ est un plus court chemin entre s et t , alors, pour tout sommet x sur le chemin, p a les propriétés suivantes :

- le sous-chemin de s jusqu'à x , (s,\dots,x) , est un plus court chemin de s à x
- le sous-chemin de x jusqu'à t , $p(x,\dots,t)$, est un plus court chemin de x à t

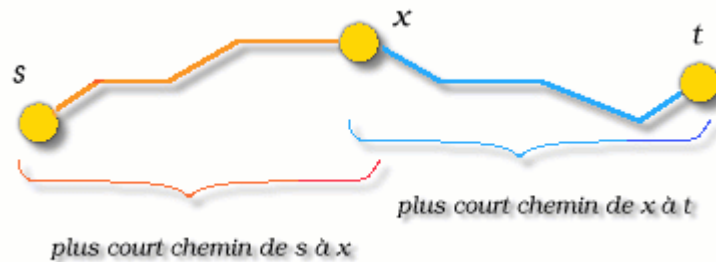


Figure III.3: Sous-chemin d'un plus court chemin.

La propriété : *Tout sous-chemin d'un plus court chemin est lui même un plus court chemin* se démontre facilement : Soit $p=(s,\dots,t)$ un plus court chemin de s à t . Soit $p_{i,j}$ un sous-chemin de p allant de s_i à x_j , il est forcément un plus court chemin, sinon p ne serait pas un plus court chemin. Ce plus court chemin serait obtenu en remplaçant dans p , p_{ij} par le chemin plus court de s_i à s_j .

Propriété 2 : Soit $p_{d,a}$ le poids du plus court chemin de d à a : on a la propriété suivante :

Pour tout s appartenant à S et (s,a) appartenant à A $p_{d,a} \leq p_{d,s} + p_{s,a}$, $p_{s,a}$ est le poids de l'arc de s à a . Il y a égalité lorsque s est sur le plus court chemin de d à a .

L'algorithme de Dijkstra dans sa version originale cherche tous les chemins optimaux entre un nœud de départ et tous les autres nœuds du graphe. Nous avons apporté deux améliorations. D'après la première propriété, nous pouvons arrêter l'algorithme de Dijkstra à

partir du moment où il trouve le nœud d'arrivée recherché, ceci constitue la première amélioration. La deuxième amélioration est l'obtention d'un ensemble de k meilleurs chemins entre deux nœuds. Ces deux améliorations nous conduisent à modifier l'algorithme de Dijkstra afin d'atteindre nos objectifs.

III.5.2 Algorithme de Dijkstra Amélioré

L'algorithme de Dijkstra est l'un des meilleurs algorithmes pour le calcul du plus court chemin dans un graphe où les poids sont positifs [Meskine 01]. Pour éviter la densité des graphes, qui présente un inconvénient de cet algorithme, on propose la division du graphe global en plusieurs sous graphes (Figure III.4) dont chacun représente une zone. Cette méthode favorise l'utilisation d'un système réparti, comme dans le cas du système allemand « DELFI ». Nous expliquerons cette répartition dans le chapitre IV. En se basant sur le théorème de Bellman, nous pouvons appliquer l'algorithme de Dijkstra sur chaque zone.

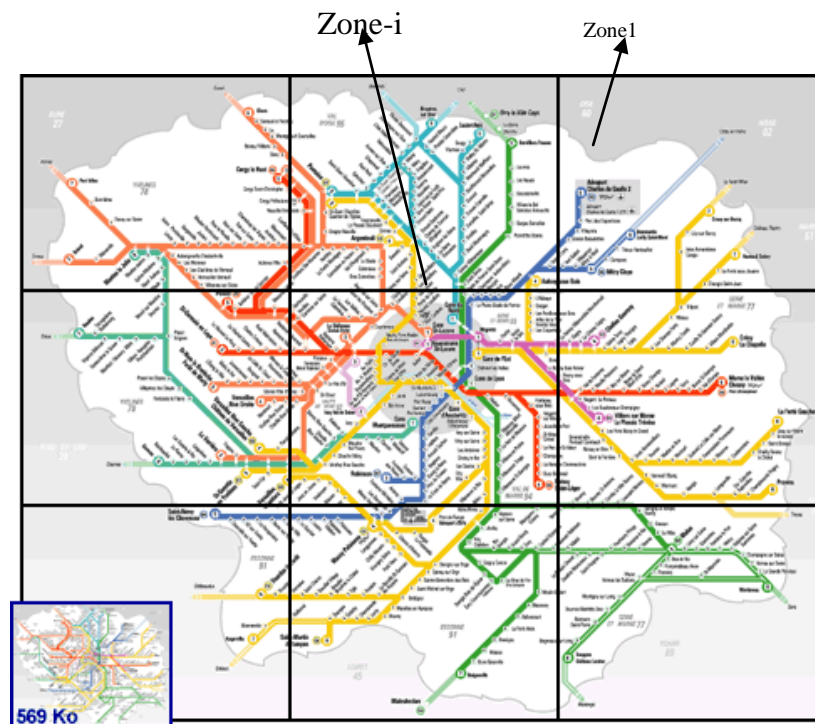


Figure III.4: Exemple de réseau pour le mode k [site RATP].

L'algorithme de Dijkstra a plusieurs variantes qui ont visé à l'améliorer, telle que la version appelée « avec tas » [Lacomme 03]. Ces versions ont focalisé leur amélioration sur la complexité et le temps de calcul. La complexité passe de $O(N^2)$ à $O(M \log N)$. Ces variantes sont l'algorithme de Sedgewick et Vitter et l'algorithme « à bucket » [Lacomme 03]. Le

premier a été conçu pour des graphes euclidiens, c'est-à-dire des graphes non orientés dont les sommets sont des points d'un espace euclidien, et les arêtes des segments entre points valués par la longueur euclidienne du segment [Sedgewick 86], alors que l'algorithme « à bucket » est intéressant quand les coûts des arcs sont des entiers [Lacomme 03]. Même si ces variantes améliorent l'algorithme de Dijkstra, elles ne restent utilisables que dans des cas particuliers. Pour cette raison, nous avons cherché d'autres modifications adaptées à notre problème. L'algorithme de Dijkstra modifié nous donne un ensemble de r meilleurs chemins entre deux nœuds. Donc la première modification concerne la condition d'arrêt. Cette modification permet que l'algorithme cherche le plus court chemin entre deux nœuds au lieu des plus courts chemins entre un nœud et tous les autres nœuds du graphe. Avec la deuxième modification, l'algorithme donne plusieurs meilleurs chemins entre les deux nœuds. Le principe de cet algorithme est la répétition de l'algorithme de Dijkstra plusieurs fois, avec un test de changement de chemin entre les itérations (Figure III.5).

- *distances*[s] est un tableau de poids du plus court chemin de la source à s
- *parcours*[t] est un tableau contenant le prédécesseur de s dans un plus court chemin de la source à s
- destination est le nœud d'arrivée souhaité par l'utilisateur

Il s'agit d'un algorithme classique. Dans sa forme originale, il calcule tous les plus courts chemins entre un nœud de départ et tous les autres nœuds d'un graphe pondéré. Dans notre cas, il s'arrête dès que le plus court chemin de l'arrivée est calculé. Nous avons rapidement pris conscience de l'importance du tri des données sur lesquelles notre programme travaille : nous avons comparé une technique matricielle de rangement des données et l'optimisation par liste d'adjacence (Figure III.6). Dans la méthode matricielle, on construit une matrice $n \times n$ (n =nombre de nœuds), dite matrice d'adjacence, et l'on met dans chaque case (m, p) les trajets entre les nœuds m et p . Mais la matrice ainsi construite contient inutilement de nombreux zéros, que l'algorithme devra pourtant parcourir en pure perte (de temps). La liste d'adjacence est la liste de tous les sommets, auxquels sont associés par un système de pointeurs tous les arcs dont le sommet en question est le nœud de départ. Cette technique améliore grandement les performances de l'algorithme, qui avait intensivement recours à la liste des arcs dont la station de départ est connue. Avec cette optimisation, nous avons pu améliorer le temps de calcul.

```

Répéter de i=0 à i=p
Initialiser D à source
Initialiser C à S-{source} : l'ensemble des sommets non fixés
distances[source]= 0
parcours[source]=source

Répéter
Etape1 : Recherche sommet s non fixé de distance minimale
Si distances[s] <+ ∞
Ajouter s dans D
Supprimer s de C
Pour tous les sommets t de C faire
    Si distances[s] + F(s,t) < distances[t] alors
        distances[t]= distances[s] + F(s,t)
        parcours[t]= s
Fin de pour
Si (C non vide & s ≠ de destination) alors retour à l'étape1
Sinon fin de Répéter
Modifier le graphe en Augmentant le poids de l'arc le moins coûteux
dans le chemin Ci
i++

```

Figure III.5: Algorithme de Dijkstra modifié.

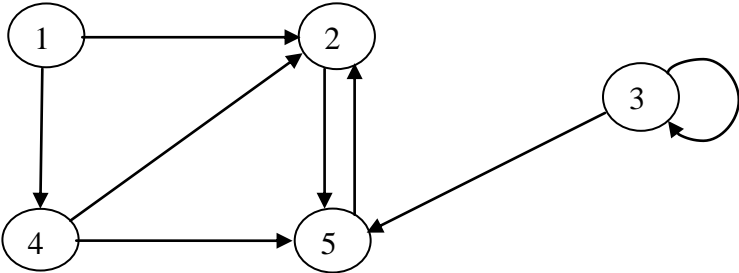


Figure III.6: Un exemple de graphe

$$Mg = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix}$$

La matrice d'adjacence...

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | → | 2 | - | → | 4 | / |
| 2 | → | 5 | / | | | |
| 3 | → | 3 | - | → | 5 | / |
| 4 | → | 2 | / | | | |
| 5 | → | 2 | - | → | 4 | / |

...et la liste d'adjacence correspondante.

Pour la programmation, nous utilisons une représentation utilisant des structures de listes. Ce sont les mieux adaptées aux structures évolutives, et elles permettent de ne parcourir que les

éléments d'un ensemble limité, et eux seuls, au lieu de visiter tous les éléments de l'ensemble de base, dont cet ensemble est un sous-ensemble.

III.6 Graphe dynamique et multimodal

Avant d'entreprendre son voyage, l'utilisateur des TC (transports en commun) va rechercher, en premier lieu, des informations sur l'état des réseaux (perturbations, mouvements sociaux...), ainsi que sur les places disponibles. Si l'analyse et la modélisation du trafic répondent à la curiosité naturelle de l'homme pour la compréhension du comportement de ses semblables, l'utilité de cette discipline apparaît véritablement lorsqu'on réalise qu'elle est indispensable à l'élaboration des stratégies de transport au niveau de la société et la nécessité de telles stratégies est aujourd'hui indiscutable [Labbé 97].

Lorsque le client prépare et effectue son trajet, ces informations vont lui permettre d'apprécier en temps réel la durée de son trajet (localisation des véhicules, horaires de passage aux arrêts, temps d'attente, temps de trajet, ...) ainsi que connaître toutes les informations relatives aux perturbations dans la zone empruntée et leurs conséquences sur le déroulement de ce trajet.

Afin de réagir en cas de perturbation, pour un système d'aide au déplacement, la gestion en temps réel est très importante. Dans ce cas, le système gère la préparation du voyage et le suivi du voyageur tout au long du déplacement. En outre, la mise à jour des informations et la prise en compte des perturbations constituent un enjeu important d'un point de vue opérationnel, car les usagers ont besoin d'informations fiables avant et pendant le voyage. Pour cette raison, un déplacement représente plusieurs cas possibles. En effet, un utilisateur sollicite l'information nécessaire à un déplacement entre un point de départ 'A' et un point d'arrivée 'G', mais ces deux points peuvent changer. Autrement dit, l'utilisateur peut changer volontairement son point de départ, ou bien sa destination, avant le début du déplacement. De même, au cours du déplacement, une perturbation éventuelle dans le réseau de transport peut entraîner des changements sur les points de départ et d'arrivée. En conclusion, on distingue cinq cas possibles :

Cas 1 : Le premier cas est statique, les deux points A et G sont fixes. Dans ce cas, le problème est la recherche d'un bon itinéraire multimodal satisfaisant les critères imposés par le client du transport.

Cas 2 : Le deuxième cas est celui où le point de départ A bouge. En effet, quand l'utilisateur change son choix de départ éventuellement à cause d'une perturbation, A n'est plus accessible par le réseau de transport (Figure III.7). De même, pendant le déplacement, une perturbation provoque un changement du prochain noeud qui peut être considéré comme un nouveau point de départ A.

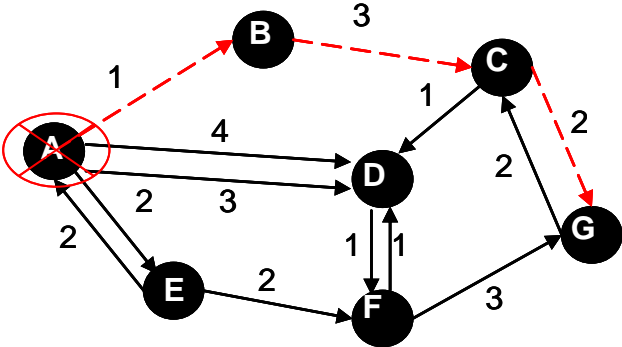


Figure III.7: Exemple de réseau nœud de départ dynamique.

Cas 3 : Le troisième cas concerne la dynamique du nœud d'arrivée G (Figure III.8). De même que dans le cas précédent, la cause peut être un nouveau choix de l'utilisateur ou une inaccessibilité due à une perturbation. Il y a donc un changement du point G.

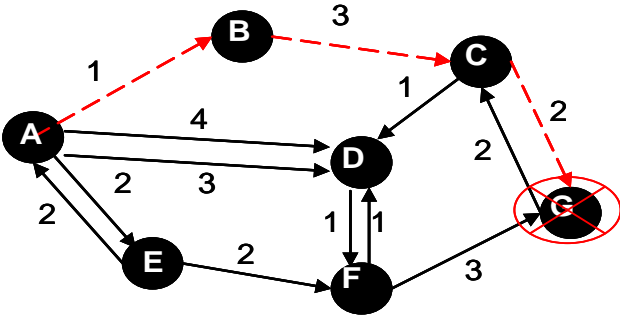


Figure III.8: Exemple de réseau nœud d'arrivée dynamique.

Cas 4 : Le quatrième cas regroupe les deux précédents. Autrement dit les deux nœuds G et A bougent simultanément (Figure III.9). La différence avec les deux cas précédents est que l'on a deux points dynamiques. Ce cas demande un traitement spécifique, parce que l'optimisation du chemin n'est plus entre deux points, mais plutôt entre deux zones.

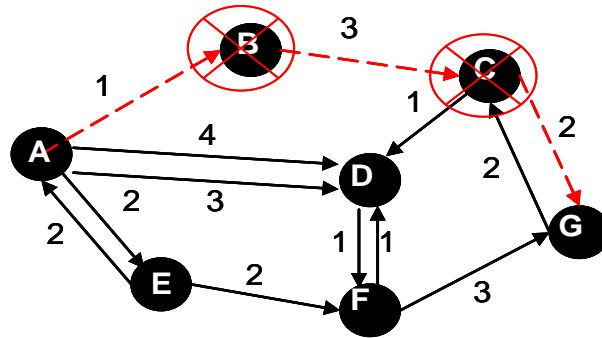


Figure III.9: Exemple de réseau avec deux nœuds dynamiques.

Cas 5 : Le dernier cas peut être le plus général. Un noeud de plus court chemin n'est plus accessible (Figure III.10). Ce cas rend notre plus court chemin impraticable. Il demande un traitement spécifique, parce qu'il faut déterminer la position de l'utilisateur, avant de lui donner une nouvelle solution. En général, cette perturbation a des influences sur toute la ligne ou bien la zone où elle se produit. Dans ce cas, quelques stations sont instantanément dynamiques.

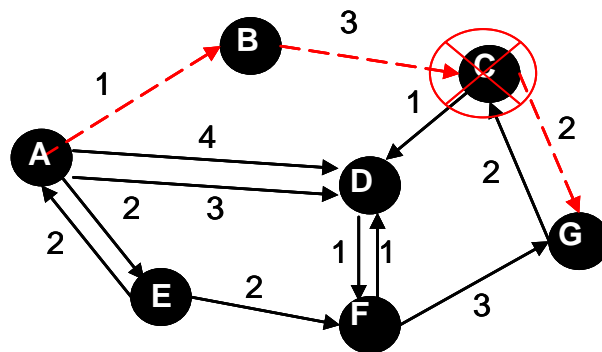


Figure III.10: Exemple de réseau avec un nœud dynamique.

Pour chaque nœud du graphe, nous avons besoin des informations concernant les modes desservant ce noeud, ainsi que les heures de passage et les arcs sortants et entrants. Soit N ($L_{modes}, T[L_{passageM}]$) la représentation du nœud N , L_{modes} est la liste des modes passant par N , $T[L_{passageM}]$ est un tableau contenant des listes de passage. Dans chaque liste sont enregistrés les horaires de passage du mode(i) par N (Figure III.11). Alors que, pour chaque arc, ou bien tronçon de chemin, nous traitons les informations des nœuds de départ et d'arrivée, les coûts (temps, prix, confort...).

| |
|--------|
| 7h 15 |
| 7h 45 |
| 8h 15 |
| 8h 30 |
| 10h 05 |
| 11h 05 |
| 12h 05 |
| 12h 35 |
| 13h 30 |

Figure III.11: Exemple d'horaires pour un nœud pendant un jour.

L'ajout de la notion d'horaire de passage par les nœuds rend l'utilisation des algorithmes de théorie des graphes existants très difficile. Les algorithmes de théorie des graphes optimisent le chemin entre deux nœuds en se basant seulement sur les informations concernant les arcs. Alors que, dans la représentation de nos graphes, nous utilisons les horaires de passage par les nœuds du graphe comme des contraintes fortes. De plus, notre objectif est d'optimiser les solutions selon plusieurs critères, d'où la nécessité d'appliquer d'autres algorithmes ou méthodes, comme les métaheuristiques.

Dans le cas d'une perturbation, notre graphe peut devenir non connexe. Avec la nouvelle hypothèse, nous sommes obligés d'ajouter le mode « marche à pied » comme mode de transport représentant une solution dans la partie du réseau affectée par la perturbation. Mais ce nouveau mode ne sera utilisé que dans les cas des destinations non desservies, ou bien mal desservies par les modes de transport en commun. Dans ces cas, le temps de parcours est calculé à partir de la distance entre les stations et une vitesse moyenne de 3 km/h

III.7 Les problèmes et les limites des algorithmes de la théorie des graphes

Un réseau de transport en commun ne peut pas être modélisé par un graphe statique, où les nœuds et les arcs sont fixés. Car ces réseaux changent selon les périodes de l'année, les jours, les horaires, etc.... Cette notion de temps, ou bien d'instant de passage d'un mode par un arrêt, ajoute une spécificité au graphe modélisant les réseaux de transport. Cette spécificité limite l'application des algorithmes de théorie des graphes existants. Les caractéristiques de ces graphes, ainsi que les objectifs attendus, compliquent les problèmes. Ces problèmes

demandent donc des modifications sur les algorithmes existants ou bien l'application de nouveaux algorithmes plus puissants. L'algorithme de Dijkstra est difficilement applicable sur des réseaux de transport à cause de leurs densités. Ainsi l'ajout de la notion de temps et les coûts des arcs des graphes exigent l'application d'autres types d'algorithmes.

Quand il s'agit de graphes qui ont peu de sommets et d'arcs, il importe peu de faire un effort de représentation qui pourrait conduire à des économies de place en mémoire. La représentation matricielle peut suffire. Par contre, dès que l'ordre du graphe est élevé, et que le graphe est peu dense, c'est-à-dire relativement pauvre en arcs, il est nécessaire de faire un effort dans la représentation, non seulement pour des économies de place mémoire, mais aussi parce qu'une telle représentation concise peut nécessiter de nombreux calculs pour retrouver certaines informations.

C'est pourquoi il faut attacher une grande importance à la recherche de la représentation la mieux adaptée au problème qu'il s'agit de résoudre. Cette représentation doit tenir compte non seulement des objets à représenter, mais également des opérations élémentaires, ou primitives, à effectuer sur eux.

III.8 L'application des algorithmes génétiques dans le SIADM

Les problèmes de cheminement sont parmi les plus anciens de la théorie des graphes. Le problème le plus typique de cette catégorie est celui du plus court chemin. En effet, il se rencontre dans de nombreuses applications. Dans la littérature, une attention particulière a été portée aux problèmes à deux critères en utilisant les méthodes exactes telles que le « branch and bound » [Visée 98], l'algorithme A* [Stewart 91], et la programmation dynamique [Carraway 90]. Ces méthodes sont efficaces pour des problèmes de petites tailles. Pour des problèmes à plus de deux critères ou de grandes tailles, il n'existe pas de procédure exacte efficace, étant donné les difficultés simultanées de la complexité NP-difficile, et le cadre multicritère des problèmes [Talbi 00] [Hao 99][Collette 02].

Les algorithmes évolutionnaires sont bien adaptés à l'optimisation multiobjectif. En effet, l'utilisation d'une population de solutions permet de pouvoir approcher un ensemble Pareto optimal en une seule exécution [Basseur 05].

Le premier travail consistant à utiliser les AGs pour résoudre des problèmes d'optimisation multiobjectif est celui de Schaffer [Schzfer 85]. L'algorithme développé VEGA (*Vector Evaluated Genetic Algorithm*) sélectionne les individus de la population courante suivant chaque objectif, indépendamment des autres (sélection parallèle). A chaque génération, la population est donc divisée en un nombre de sous-populations qui est égal au nombre de fonctions objectifs composant le vecteur objectif.

Nous présentons également les possibilités de combiner différentes méthodes pour créer des méthodes hybrides. Le mode d'hybridation qui semble le plus fécond concerne la combinaison entre les méthodes de voisinage et l'approche évolutionnaire. Les algorithmes hybrides sont sans doute parmi les méthodes les plus puissantes.

Le but de l'algorithme génétique est d'offrir une bonne approximation de la frontière Pareto exacte, mais surtout d'obtenir une population la plus diversifiée possible, en explorant d'une manière la plus intelligente possible l'espace des solutions.

Par ailleurs, il existe également des algorithmes de calcul réparti, permettant de distribuer le calcul, par exemple entre réseaux urbains et interurbains, et reconstituant l'itinéraire optimal de bout en bout ; ces solutions ont été mises en oeuvre dans le projet allemand DELFI. Ce dernier reste à notre connaissance un prototype encore en test.

Dans ce sens, l'algorithme évolutionnaire de recherche d'itinéraire (Figure III.19) commence par construire une population initiale de solutions. L'algorithme de Dijkstra modifié (DijkstraModifier) « ADM » nous donne un ensemble de r bons chemins entre deux nœuds [Zidi 04]. Le principe de cet algorithme est la répétition de l'algorithme de Dijkstra p fois avec un test de changement de chemin entre les itérations. P est la population initiale de $(2*w)$ individus résultant de cette étape de l'algorithme [Zidi_b 04]. Cette population est la population initiale pour notre algorithme génétique. Dans cet algorithme, nous proposons deux opérateurs de croisement par conjugaison du mode, OCCM [Zidi_b 04] le premier à deux-points sur les colonnes, le deuxième à un seul point (section III.8.3_b). Toutefois, le respect de faisabilité des itinéraires est toujours assuré par ces opérateurs. L'opérateur de mutation « OM » correspond à des changements aléatoires de passage par un nœud d'une solution donnée (section III.8.3_c).

III.8.1 Codage

Il peut être difficile d'adapter le codage binaire à certains problèmes. La représentation binaire traditionnelle utilisée pour les algorithmes génétiques crée des difficultés pour les problèmes d'optimisation de grandes dimensions à haute précision numérique. Par exemple, si l'on doit utiliser 100 variables appartenant au domaine $[-500, 500]$ et si l'on a besoin d'une précision de l'ordre de 10^{-6} , on doit utiliser des chromosomes d'une longueur de $l=3000$. Ces paramètres en retour, génèrent un espace de recherche de l'ordre de 10^{1000} . Pour de tels problèmes, les algorithmes génétiques basés sur des représentations binaires ont de faibles performances, comme le montre Michalewicz [Michalewicz 94].

L'utilisation du codage réel (les chromosomes sont des vecteurs de réels et non plus des chaînes de bits), pour des problèmes dans lesquels les données prennent des valeurs réelles, s'est donc répandue [Lecomte 00]. L'avantage est ainsi de conserver les variables du problème dans le codage de l'élément de population sans passer par le codage binaire intermédiaire. La structure du problème est alors plus facilement conservée dans le codage.

Le codage des individus de notre population se base sur le codage réel. Par la suite, une solution est une succession des nœuds, comme le montre la figure III.12.

| | | | | | | |
|--------|--------|--------|--------|--------|---------|--------|
| D(M,t) | 1(M,t) | 2(M,t) | 4(M,t) | 7(M,t) | 14(M,t) | A(M,t) |
|--------|--------|--------|--------|--------|---------|--------|

Figure III.12: Codage d'une solution.

On suppose ici que «D» est le nœud de départ, «A» celui d'arrivée et (1,2,4,7,14) sont les noms des nœuds traversés par cette solution. Chaque chromosome contient les informations sur le nœud qui le présente. Ces informations sont le mode (M) à utiliser pour passer au nœud suivant et les horaires de passage (t).

Il est important de noter que, puisque l'environnement étudié est dynamique, le codage qui le représente l'est aussi. En effet, une solution peut avoir un nombre de nœuds variable, en ajoutant ou éliminant le passage par des stations. Autrement dit, la taille d'un individu n'est pas fixe.

Deux solutions différentes peuvent avoir les mêmes stations à parcourir. Dans ce cas, ces solutions n'utilisent pas les mêmes modes (Figure III.13) ou bien les temps de passage par quelques stations sont différents (Figure III.14).

| Solution 1 | | | | | | |
|------------|---------|---------|---------|---------|----------|---------|
| D(M1,t) | 1(M1,t) | 2(M3,t) | 4(M2,t) | 7(M2,t) | 14(M2,t) | A(M2,t) |

| Solution 2 | | | | | | |
|------------|---------|---------|---------|---------|----------|---------|
| D(M1,t) | 1(M1,t) | 2(M3,t) | 4(M3,t) | 7(M3,t) | 14(M4,t) | A(M4,t) |

Figure III.13: Changement des modes de transport.

| Solution 1 | | | | | | |
|------------|------------|------------|------------|------------|-------------|------------|
| D(M,10:02) | 1(M,10:07) | 2(M,10:09) | 4(M,10:15) | 7(M,10:35) | 14(M,10:44) | A(M,10:54) |

| Solution 2 | | | | | | |
|------------|------------|------------|------------|------------|-------------|------------|
| D(M,10:10) | 1(M,10:15) | 2(M,10:17) | 4(M,10:23) | 7(M,10:43) | 14(M,10:52) | A(M,11:02) |

Figure III.14: Changement d'horaires de passage.

III.8.2 Initialisation de la population

L'AG démarre en général avec une population initiale composée par un tirage aléatoire de N individus (chaînes, chromosomes) dans le codage retenu. La performance de chacun de ces individus doit alors être évaluée. Le choix de la population initiale d'individus conditionne fortement la rapidité de l'algorithme. Si la position de l'optimum dans l'espace de recherche est totalement inconnue, il est naturel de générer aléatoirement des individus en faisant des tirages uniformes dans chacun des domaines associés aux composantes de l'espace d'état. Si, par contre, des informations a priori sur le problème sont disponibles, il paraît bien évidemment naturel de générer les individus dans un sous domaine particulier, afin d'accélérer la convergence. Si le problème d'optimisation à résoudre comporte des contraintes, on veille en général à ce que les individus ainsi générés respectent ces contraintes [Michalewicz 91]. En se basant sur les informations données par la modélisation graphique et l'utilisation de l'algorithme de Dijkstra modifié, nous préparons une population initiale pas complètement aléatoire. Cette partie est expliquée dans la section III.9.

Nous disposons d'une population d'individus non homogène, la diversité de la population doit être entretenue au cours des générations, afin de parcourir le plus largement possible l'espace de recherche. C'est le rôle des opérateurs de croisement et de mutation. En effet, sans ces opérateurs, l'effet de la sélection serait d'uniformiser la population.

III.8.3 Les opérateurs

Trois opérateurs jouent un rôle prépondérant dans la réussite et la convergence d'un AG : l'opérateur de sélection, l'opérateur de croisement et l'opérateur de mutation. Si le principe de chacun de ces opérateurs est facilement compréhensible, il est toutefois difficile d'expliquer l'importance isolée de chacun de ces opérateurs dans la réussite de l'AG. En effet, chacun de ces opérateurs agit selon divers critères qui lui sont propres (valeur sélective des individus, probabilité d'activation de l'opérateur, etc.) [Vallée 03].

Deux des mécanismes principaux qui déterminent la capacité d'exploration de l'espace de recherche d'un algorithme génétique sont les opérateurs de croisement et de mutation, ainsi que la méthode de diversification [Basseur 05].

a) Sélection

La sélection permet d'identifier statistiquement les meilleurs individus d'une population, de leur assurer une descendance, c'est-à-dire de favoriser leur présence dans la génération suivante, et d'éliminer les mauvais. Plusieurs techniques de sélection sont utilisées par les chercheurs ; elles peuvent être déterministes ou stochastiques [Bounsaythip 98].

Cet opérateur détermine la capacité de chaque individu à persister dans la population et à se reproduire. En règle générale, la probabilité de survie d'un individu sera directement liée à sa performance relative au sein de la population. Plus la performance d'un individu est élevée par rapport à celle des autres individus de la même population, plus il a une chance d'être reproduit dans la population. Les individus ayant une grande performance relative ont donc plus de chance d'être sélectionnés. On parle alors de sélection proportionnelle.

Supposons qu'une chaîne (chromosome pour nous) de la population possède comparativement une performance élevée, sans être optimale ou très proche de l'optimum. Disons que la performance de cette chaîne est dix fois plus grande que la performance moyenne. Il pourrait facilement arriver, après quelques générations, que la population ne soit entièrement constituée que de cette chaîne. Dans un tel cas, l'algorithme génétique n'évoluera

plus et l'optimum ne sera pas trouvé. Ce phénomène est appelé « convergence prématurée » et c'est l'un des problèmes les plus fréquents lors de l'utilisation des algorithmes génétiques.

Il existe plusieurs méthodes pour représenter la reproduction. La méthode la plus connue et la plus utilisée est la sélection par roulette pondérée de Goldberg [Goldberg 89]. Nous appliquons cette méthode dans notre algorithme. L'inconvénient majeur de cette méthode repose sur le fait qu'un individu qui n'est pas le meilleur peut tout de même dominer la sélection. Cette méthode peut aussi engendrer une perte de diversité par la dominance d'un individu. Un troisième inconvénient est sa faible performance vers la fin du processus, quand les individus se ressemblent (car la population est alors dominée par des individus qui ont des performances relatives similaires). [Dawid 99] résume très bien tous ces inconvénients. Nous proposerons une amélioration reposant sur le contrôle de l'évolution des individus entre les générations dans la partie évaluation et contrôle de l'évolution.

b) Opérateur de croisement

Nous proposons deux types de croisements : le premier à deux-points sur les colonnes et le deuxième à un seul point. Toutefois, le respect de la faisabilité des itinéraires est toujours assuré par ces opérateurs.

Deux points de croisement sont choisis au hasard, l'échange concerne les colonnes qui se trouvent entre ces deux points. De même, le choix d'un point de croisement est aléatoire, l'échange concerne les colonnes après ce point.

Les figures ci-dessous illustrent le fonctionnement de ce croisement. Par exemple, pour le parent 1, l'itinéraire (d, 1, 2, 4, 5, 8, 7, 9, f) est effectué par le premier mode. De l'autre côté, le parent 2 suit l'itinéraire (d, 10, 2, 11, 6, 7, 14, f). Les points de croisement entre ces deux individus sont les stations 2 et 7 (Figure III.15). Dans le deuxième exemple (Figure III.16), le croisement est à un seul point qui est le nœud 7.

La zone qui subit l'échange est composée par le chemin réalisé par les arrêts 2, 4, 5, 8, 7. Les zones externes du parent 1 (respectivement 2) sont transmises à l'enfant 1 (respectivement 2) et la zone interne du parent 2 (respectivement 1) est copiée dans celle de l'enfant 1 (respectivement 2).

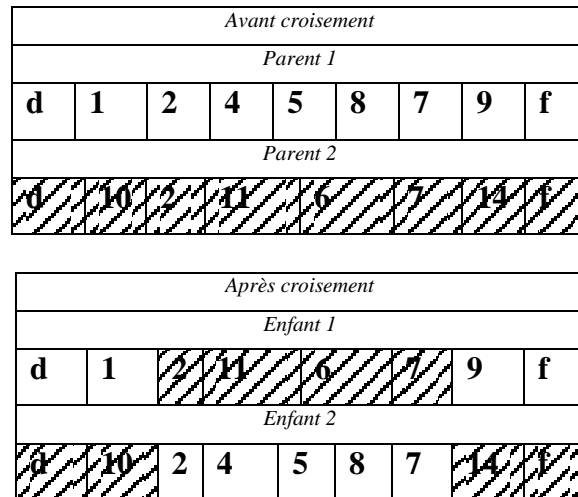


Figure III.15: Croisement à deux points.

Pour le deuxième cas de croisement, les descendants gardent chacun la première partie de chemin avant le point de croisement de l'un des parents. Le reste du chemin est échangé avec la deuxième partie de l'autre parent. Dans l'exemple suivant, l'enfant 1 hérite le bout de chemin (d, 1, 2, 4) du parent 1. Puis, il hérite du reste de chemin (7,14, f) du parent 2.

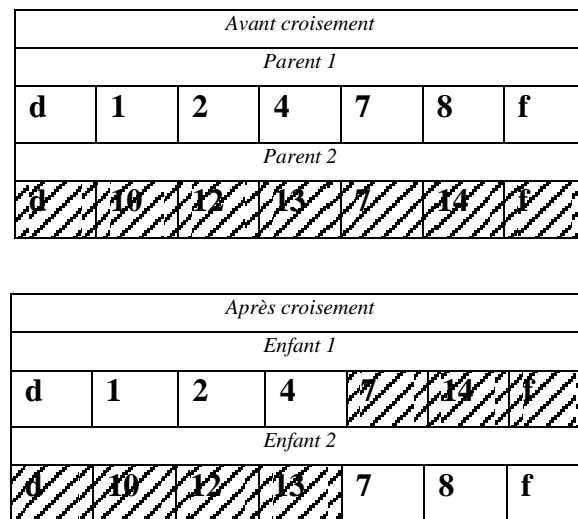


Figure III.16: Croisement à un point

Les croisements sont à la base des algorithmes génétiques. Quoiqu'il en soit, il se peut que l'action conjointe de la reproduction et du croisement soit insuffisante pour assurer la réussite de l'AG.

c) Opérateur de mutation

La mutation des individus correspond à des changements aléatoires de passage par un nœud d'une solution. Cependant, ces changements doivent respecter la réalisabilité du chemin. La

figure suivante montre un exemple de mutation appliquée sur l'enfant 1 du dernier croisement (figure III.17). Dans cet exemple, l'individu 1 perd le gène 14 (le nœud 14). Cet opérateur montre que la longueur des chromosomes n'est pas fixe, puisque nous utilisons un codage réel.

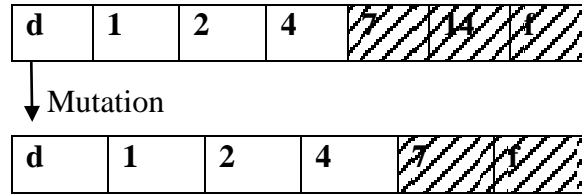


Figure III.17: Exemple de mutation.

La mutation est traditionnellement considérée comme un opérateur intervenant à la marge, dans la mesure où sa probabilité est en général fixée assez faible (de l'ordre de 1%). Mais elle confère aux algorithmes génétiques une propriété très importante : l'ergodicité (i.e. tous les points de l'espace de recherche peuvent être atteints). Cet opérateur est donc d'une grande importance et il est en fait loin d'être marginal. Il a de fait un double rôle : celui de permettre une recherche locale et/ou de sortir d'un piège (optima locaux) [Vallée 03].

c) Fonction d'évaluation et contrôle de l'évolution

Chaque chromosome apporte une solution potentielle au problème à résoudre. Néanmoins, ces solutions n'ont pas toutes le même degré de pertinence. C'est à la fonction de performance de mesurer cette efficacité, pour permettre à l'AG de faire évoluer la population dans un sens bénéfique pour la recherche de la meilleure solution [Vallée 03]. Pour évaluer un individu dans une population, nous utilisons le coût final C_f . Ce dernier est calculé par la formule (1) (section III.4). L'évolution d'une population est contrôlée donc par la minimisation de C_f .

Allenson [Allenson 92] a proposé la méthode avec genre pour traiter un problème bi-objectif. Cette méthode utilise la notion de genre (masculin ou féminin), et l'attracteur. En affectant un objectif à chaque genre, l'auteur espère minimiser les deux objectifs simultanément, car un genre sera toujours jugé d'après l'objectif qui lui a été associé [Basseur 05].

Dans l'étude d'Allenson, où l'on se trouve dans un cas bi-objectif, les deux genres sont mâle et femelle. La reproduction n'est alors permise qu'entre un individu mâle et un individu femelle, le genre étant affecté aléatoirement à la création de l'individu. Il utilise ce croisement dans un algorithme génétique classique.

En général, une perturbation concerne un ensemble de stations dans la zone ou bien un ensemble de modes de transport. Pour gérer les cas de perturbation, nous avons besoin d'une population finale dont les individus sont dispersés sur toute la zone et utilisant le plus de modes possibles. Autrement dit, si nous disposons de solutions robustes représentatives des différentes régions et des différents modes de transport, nous augmentons aussi la probabilité de trouver une solution de secours en temps réel lors des perturbations qui surviennent dans le réseau de transport. Ceci est possible grâce à la diversification des solutions dans notre population finale et le contrôle de l'évolution [Zidi_b 05].

Après l'utilisation de chaque opérateur génétique, nous devons choisir les individus à garder pour la génération suivante. En se basant sur une idée proposée par Allenson, nous utilisons un mécanisme pour contrôler ces opérateurs (Figure III.18). Dans ce mécanisme, nous utilisons deux variables : la première A_{ij} , qui est le nombre d'arcs communs entre deux solutions i et j , minimiser cette variable permet d'augmenter l'exploration des solutions sur la zone. L'avantage de cette méthode est d'augmenter la probabilité de trouver une solution en cas de perturbation. Cette augmentation représente l'un de nos objectifs. Pour choisir entre les individus parents d'une génération G , et leurs enfants issus de l'opération de croisement, nous utilisons la sélection par tournois (section II.8.3.2). L'originalité dans cette utilisation, par rapport à la méthode classique de sélection par tournois, est que la compétition n'est plus entre deux individus, mais entre deux couples d'individus. Aussi le critère utilisé dans cette compétition est la variable A_{ij} . Le couple qui a la variable minimale sera gardé pour la génération $G+1$.

La deuxième variable M_i^j est le rang de la solution i selon le nombre utilisation du mode j dans un ensemble de solutions. Cette variable est utilisée avec l'opérateur de mutation. Nous classons les parents d'une génération G et leurs enfants issus de l'opérateur de mutation selon la variable M_i^j . Pour chaque solution, qui utilise plusieurs modes, elle prend la variable minimale de M_i^j selon j . Par la suite, nous utilisons la sélection par tournois, avec le rang des individus comme valeur de performance. Entre deux individus, celui qui a le meilleur rang, sera dans la génération $G+1$. Avec la maximisation de nombre de modes utilisés, nous améliorons la population, en garantissant la multimodalité des solutions, ainsi que la diversification. Cette diversification est intéressante pour la population finale pour donner plus de choix aux usagers. De plus, elle augmente aussi la probabilité de trouver une solution, en cas de perturbation du réseau de transport.

Soit P la probabilité de trouver une solution dans la population finale en cas de perturbation.

Nb_{mj} : Nombre de solutions utilisant le mode j dans la population finale.

Nb_{pi} : Nombre de solutions passant par la partie i de la zone.

Supposons que nous ayons m modes de transport et que la zone soit coupée en k parties géographique. Une perturbation concerne en général un mode de transport ou bien une partie géographique de la zone.

La probabilité de solutions non valables dans la population finale si un mode j est perturbé

$$\text{est : } P_m = \frac{1}{m} \cdot Nb_{mj} \quad (5)$$

La probabilité de solutions non valables dans la population finale si une partie i est perturbée

$$\text{est : } P_p = \frac{1}{k} \cdot Nb_{pi} \quad (6)$$

$$P = 1 - (P_m + P_p) \quad (7)$$

Si Nb_{pi} décroît, alors P_p diminue aussi, ce qui permet à P d'augmenter. De même aussi si

Nb_{mj} décroît, alors P_m diminue aussi, et P augmente.

III.9 Hybridation entre l'algorithme de Dijkstra et un AG

Notre objectif est d'améliorer la qualité des services pour les clients de transport en commun. Ce client a besoin de planifier ses déplacements selon plusieurs critères. Donc notre problème est d'optimiser un chemin entre deux stations, ou bien adresses à adresses, suivant les objectifs du client. Bien que le calcul de plus court chemin dans un graphe valué soit un problème polynomial, la gestion de la multimodalité, ainsi que les notions temporelles dans le réseau de transport, rendent ce problème NP-Difficile, comme nous l'avons montré dans le chapitre précédent. L'avantage de l'utilisation d'un algorithme de Dijkstra est qu'il nous donne des solutions partiellement optimales.

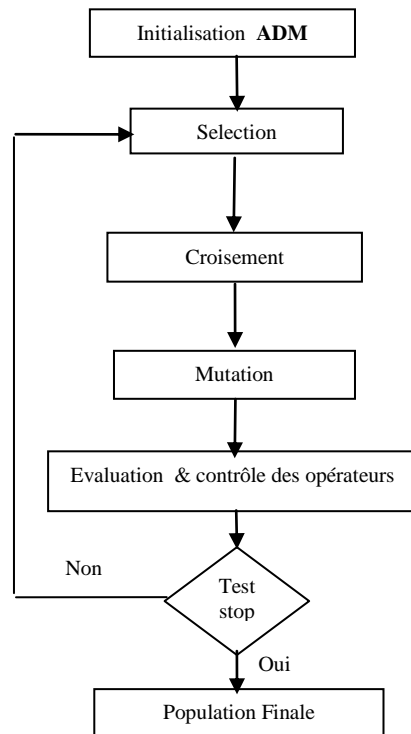


Figure III.18: Structure globale de l’algorithme génétique.

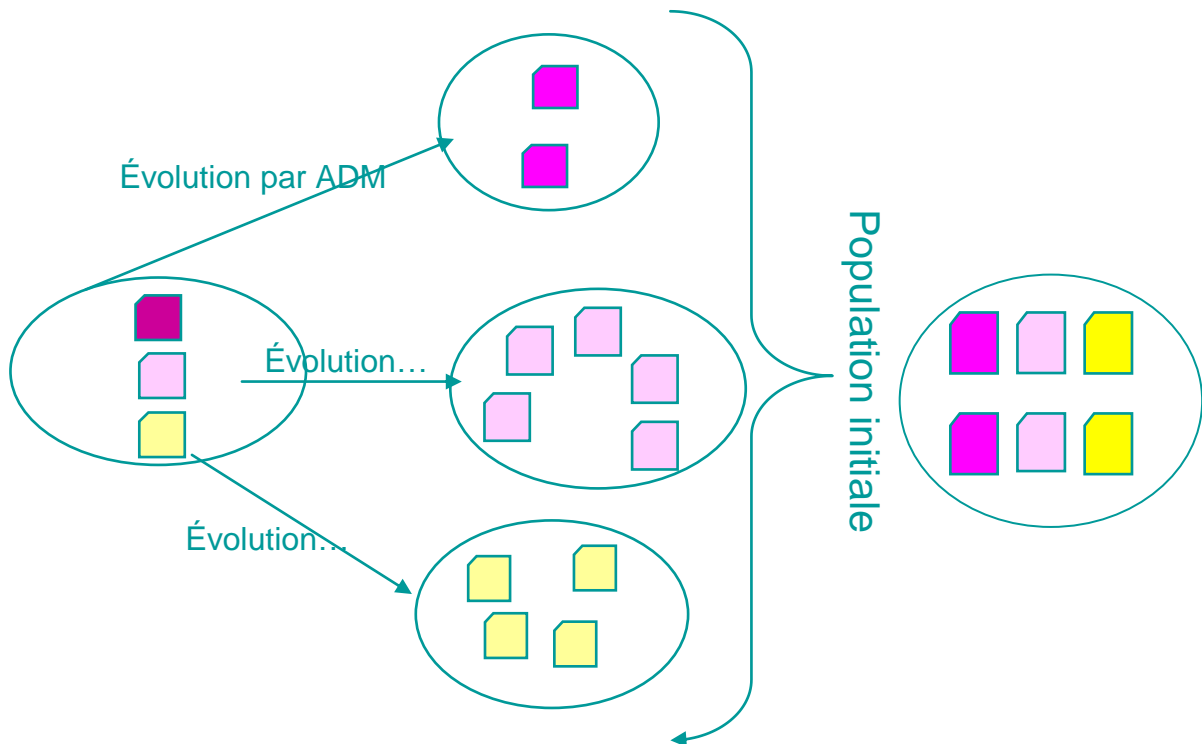


Figure III.19: Hybridation entre algorithme de Dijkstra et un AG

L'application de l'algorithme de Dijkstra modifié sur une zone, ou bien sur un mode du réseau de transport en commun, nous permet de trouver des solutions optimales. Ces solutions seront utilisées comme population initiale pour notre algorithme génétique (Figure III.19). Les graphes d'une zone ou bien un mode ne sont généralement pas denses. Cette propriété nous permet d'exploiter l'efficacité de l'algorithme de Dijkstra modifié. La concaténation de ces solutions partielles forme des solutions complètes. Ces dernières constituent la population initiale de notre algorithme génétique.

Malgré la simplicité des AG, ils assurent bien l'exploration stochastique orientée d'un espace de recherche. Si, avec cette approche, nous avons proposé une solution aux problèmes d'optimisation des déplacements, il reste d'autres problèmes d'ordre organisationnel et technique, que nous proposons de résoudre via les systèmes multi-agents.

III.10 Architecture du système d'aide au déplacement

Du fait de la popularisation de l'Internet et de la technologie du Web, la réalisation des applications coopératives dans cet environnement distribué est de plus en plus nécessaire. La tendance actuelle est de donner la capacité de coopérer entre des applications dispersées sur les réseaux pour fournir un service aux utilisateurs. Dans ce contexte, nous nous sommes intéressés à la réalisation d'applications coopératives, en appliquant la technologie d'agents et des systèmes multi-agents (SMA). De nos jours, cette technologie a trouvé sa place dans les systèmes manufacturiers, les systèmes financiers, les loisirs, les télécommunications, les systèmes embarqués et bien d'autres applications [Chaib-draa 96].

Dans le but d'adapter la gestion de ces réseaux par un SMA, nous avons besoin de bien modéliser leurs caractéristiques et leurs complexités. Dans la première partie de ce chapitre, nous présenterons la modélisation du réseau de transport par une architecture multi-zone. Cette architecture nous montre l'aspect distribué du système, les interactions et les relations qui peuvent avoir lieu entre les différentes zones.

Afin de cerner au mieux les particularités du SMAAD proposé, nous présenterons, dans la suite de ce chapitre, les différents agents qui le composent, ainsi que leur architecture interne. Nous décrirons aussi la nature de l'organisation du système considéré, afin de situer les interactions qui le caractérisent. Nous illustrerons les comportements des agents SYSTEME,

RESPONSABLE ZONE, CALCUL CHEMIN, PERTURBATION, SUPERVISEUR et GESTIONNAIRE D'INFORMATION, via le langage de modélisation UML, dans le but de mettre en valeur les actions qu'ils réalisent dans le processus de gestion temps réel du réseau.

III.10.1 Modélisation globale du système

La méthode traditionnelle pour la réalisation d'un système d'aide au déplacement consiste généralement à centraliser dans une base unique l'ensemble des informations nécessaires aux déplacements. Ce schéma s'applique très bien dans le cas d'un transporteur unique, mais devient contraignant quand plusieurs fournisseurs de services interviennent. Dans notre objectif, la multimodalité est un critère très important. La gestion de cette multimodalité pose un certain nombre de problèmes :

- Problème d'agrément, il faut que l'ensemble des responsables et fournisseurs d'information acceptent de mettre à disposition leurs données ;
- Problème de responsabilité : qui devient responsable de l'exactitude de la mise à jour et de la maintenance des données ?
- Problème de rationalité : est-il nécessaire de centraliser dans une même base des informations de régions très éloignées et aux inter-relations improbables ?
- Problème de fiabilité de la base, dont les mises à jour dépendent d'un grand nombre d'intervenants.

C'est ainsi qu'est née l'idée d'un système à l'architecture répartie ou distribuée. En proposant non pas de centraliser les données, mais de mettre en réseau les différentes sources d'information, le projet préserve la multitude des systèmes et fournisseurs d'information existants. D'un point de vue organisationnel, chaque fournisseur conserve son autonomie et la compétition est maintenue à la fois entre fournisseurs de systèmes d'information et fournisseurs de services.

III.10.2 Architecture multi-zones

Afin de résoudre les problèmes de mise à jour des données, ainsi que de responsabilité, notre but est de réaliser un système distribué. Dans cette approche, le système est divisé en plusieurs niveaux (Figure III.20). Le premier niveau est celui des systèmes des exploitants du réseau de transport. Dans ce niveau, chaque exploitant est responsable de ses bases de données, ainsi que de leur mise à jour. Le deuxième niveau représente les sous-systèmes

centraux. Autrement dit, chaque sous système s'occupe d'une zone ou bien d'une région. Le rôle de ce système est d'assurer l'interopérabilité entre les systèmes des exploitants de cette zone. Ce système n'a besoin de garder que les informations nécessaires pour les interactions entre les systèmes des exploitants et les informations d'interconnexion avec les autres zones voisines. Ce dernier niveau peut lui-même être divisé en plusieurs niveaux. En effet, chaque zone peut regrouper plusieurs zones, par exemple la France peut regrouper tous les départements et agglomérations en même temps, elle représente une zone à l'échelle européenne. Enfin le dernier niveau est le système central qui n'a besoin que des informations concernant les correspondances entre les sous-systèmes. Cette approche nous permet de faciliter la gestion des données, ainsi que de réduire la zone de travail de chaque sous-système. Pour faire face à cette approche, on propose la division du graphe global en plusieurs sous-graphes (Figure III.20) dont chacun représente une zone. Cette méthode favorise l'utilisation d'un système réparti, comme dans le cas du système allemand « DELFI ».

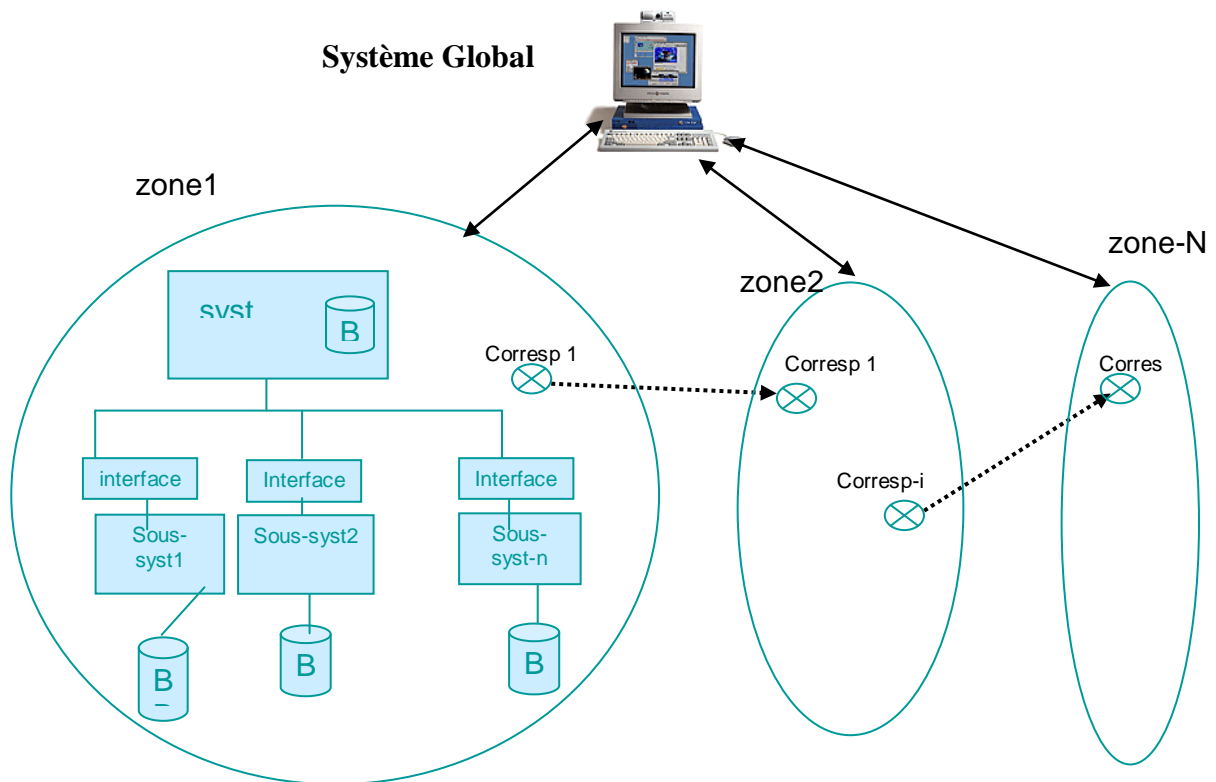


Figure III.20: Architecture multi-zones

III.10.3 Le Système Distribuer

Les données dynamiques constituent le cœur de l'information en temps réel. Actuellement, seul l'exploitant est en mesure de produire ce type de données. La constitution d'un service d'information multimodal ne peut donc se faire qu'en utilisant les données produites par les exploitants.

Il faut toutefois noter que l'élaboration d'une donnée pour l'exploitation n'est pas identique à une élaboration dans un objectif d'information des utilisateurs (surtout dans un contexte multimodal, donc multi-exploitants). Un processus de transformation de la donnée brute en donnée utilisable pour l'information aux utilisateurs est donc très important. La difficulté liée à l'homogénéité des données entre les exploitants évoquée plus haut se retrouve également au niveau des données dynamiques.

A partir d'un certain niveau de complexité et de taille, il ne devient plus possible techniquement de réaliser le service d'information par la constitution d'une base centralisant toutes les données utiles, indépendantes de l'exploitant. Cette "impossibilité" s'explique par des contraintes fortes sur les volumes et fréquences de mise à jour. On peut évoquer aussi le cas de la réservation de place dans un transport en commun ou dans un parking qui exige également une gestion de la concurrence d'accès à la ressource. La difficulté technique serait encore accrue s'il s'agissait d'un service intégrant la réservation, la commande et le paiement de billets (e-commerce) dans un contexte multimodal. Il faut alors imaginer un dialogue entre services répercutant le dialogue entre l'utilisateur final et le fournisseur de service d'information. Le "fournisseur de données" est alors en fait aussi un "fournisseur de services d'information".

Les systèmes multi-agents (SMA) se présentent aujourd'hui comme une solution naturelle pour concevoir des applications distribuées dans des environnements hétérogènes, ouverts et dynamiques, tels que Internet [Nierstrasz 00]. Dans ces environnements, les protocoles de communication sont amenés à évoluer et les serveurs d'agents à changer. Ces modifications seront de plus en plus fréquentes, et pour la plupart d'entre elles, imprévisibles lors du développement des systèmes [Viet 04].

III.11 Système Multi-Agent d'Aide au Déplacement (SMAAD)

Afin de planifier des déplacements et gérer les cas des perturbations, nous proposons un Système Multi-Agent d'Aide au déplacement (SMAAD), qui a pour rôle d'assister

l'utilisateur de transport en commun avant et pendant ses déplacements. Ce système est composé des agents SYSTEME, RESPONSABLE ZONE, CALCUL CHEMIN, PERTURBATION, SUPERVISEUR et GESTIONNAIRE D'INFORMATION. Ces agents sont regroupés dans des groupes qui modélisent les niveaux de l'architecture multi-zones. Dans la partie suivante, nous décrivons les comportements des agents, ainsi que les interactions entre eux [Zidi 05] [Ben Khaled 05].

III.11.1 Architecture

L'architecture de notre système SMAAD est basée sur le regroupement des agents sur plusieurs niveaux [Zidi 05]. Ces niveaux sont attachés à la division en zones. Chaque zone élémentaire regroupe l'ensemble des agents suivants : RESPONSABLE ZONE, CALCUL CHEMIN et GESTIONNAIRE D'INFORMATION. Comme chaque zone peut être elle-même divisée en plusieurs zones élémentaires, alors dans ce cas elle regroupe un agent RESPONSABLE ZONE avec les agents RESPONSABLE ZONE élémentaires. Par exemple : nous pouvons diviser la France en plusieurs zones où ces zones représentent les régions. Dans une échelle plus grande où nous modélisons le système d'aide aux déplacements européens, on divise l'Europe en zones, où chacune des zones représente un pays. Dans ce dernier cas, chaque zone représente un ensemble de zones plus petites. Cette architecture nous permet d'ajouter ou bien de supprimer autant de zones que nécessaire, ce qui caractérise l'ouverture de l'environnement de notre système. A partir de cette caractéristique, de nouveaux agents « responsable zone » peuvent être introduits ou retirés. On ne peut pas savoir à l'avance où s'arrête ce système, ni le nombre d'agents qui travaillent dans ce système. Cette architecture est présentée sur la figure III.21.

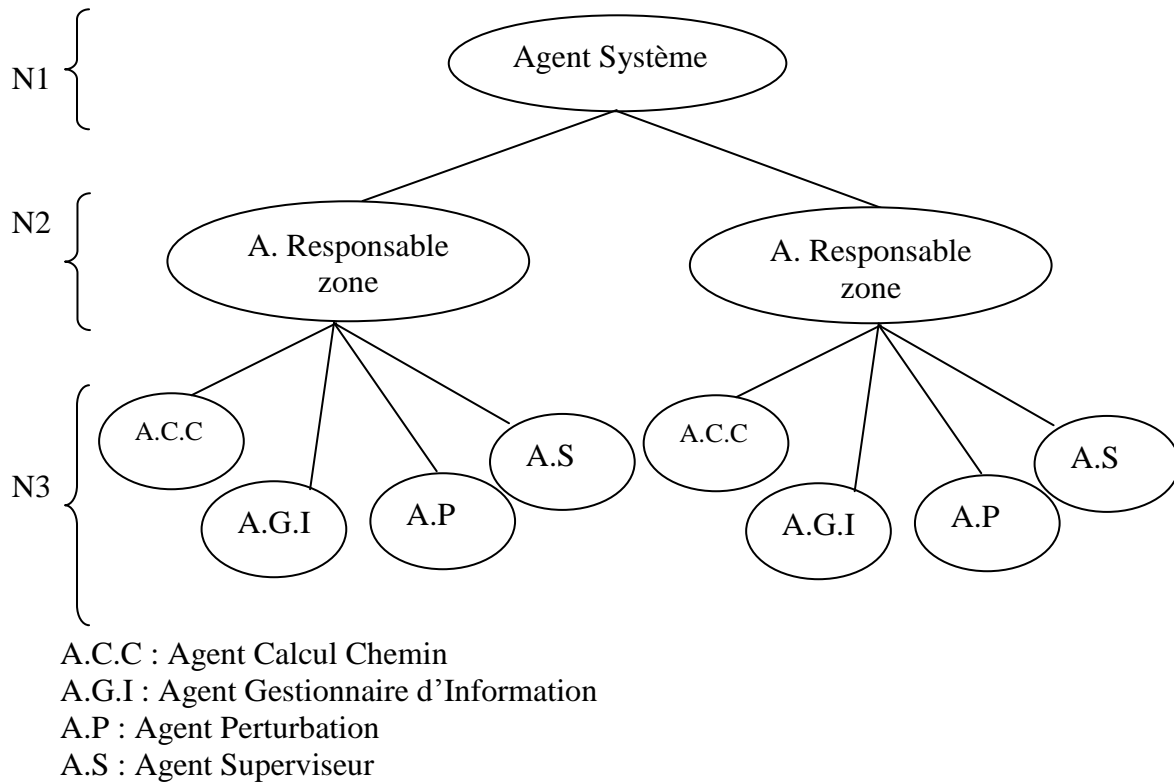


Figure III.21: Architecture du SMA.

III.11.2 Les agents de SMAAD

III.11.2.1 Agent « SYSTEME »

Toutes les régions sont placées sous la responsabilité de l'agent SYSTEME, responsable du fonctionnement général du SMAAD. Cet agent doit aussi pouvoir gérer ses messages et communiquer avec un acteur humain, le client. Il est défini comme suit :

| | |
|----------------------|--|
| Identité | Communauté : Transport Groupe : Système Rôle : Responsable de groupe |
| Etat | Libre, occupé |
| Connaissances | les modes de transport les nœuds de chaque zone les nœuds d'intersection entre zones |

Si nous considérons une requête de calcul du chemin multicritère entre une origine A et une destination B, les actions que réalise l'agent SYSTEME sont :

- Déterminer les zones correspondant aux points A et B : limitation des zones concernées par la requête.
- Rechercher les points d'intersections (correspondances) : si l'origine et la destination n'appartiennent pas à la même zone.
- Envoyer des requêtes de calcul de chemin : si une seule zone est concernée, alors l'agent système envoie une seule requête pour le calcul du chemin optimal direct entre A et B, sinon, il envoie diverses requêtes à toutes les zones concernées pour le calcul du chemin entre les points A, B et les correspondances. Dans ce cas, l'agent SYSTEME utilise un algorithme de Dijkstra simple pour déterminer les zones concernées. L'ensemble des zones est représenté par un graphe dont chaque nœud modélise une zone et les arcs représentent les liaisons de voisinage entre les zones.
- Enregistrer les portions des chemins trouvées : chaque zone, après le calcul du chemin dans sa région, envoie le résultat à l'agent SYSTEME.
- Concaténer les portions de chemins : Réalisation du chemin final suite aux réponses des agents RESPONSABLES ZONE (RZ), en concaténant les parties de chemin proposées par chacun d'entre eux.

Le comportement de l'agent SYSTEME est représenté (Figure III.22) à travers le séquençement des tâches qu'il exerce.

III.11.2.2 Agent « RESPONSABLE ZONE »

L'agent RESPONSABLE ZONE est chargé d'assurer l'interopérabilité entre les différents exécutants de sa zone. Il garantit également la communication avec l'agent SYSTEME.

Cet agent est identifié comme suit :

| | |
|----------------------|---|
| Identité | Communauté : Transport Groupe : Zone i Rôle : Responsable de groupe |
| Etat | Libre, occupé ou interruptible |
| Connaissances | les nœuds de sa zone |

Les principales actions que nous intégrons dans un tel agent sont (Figure III.23):

- Déclencher une recherche par l'agent GESTIONNAIRE D'INFORMATION : vérification si le chemin a déjà été calculé, afin d'éviter le temps de recalcul d'une requête répétitive ;

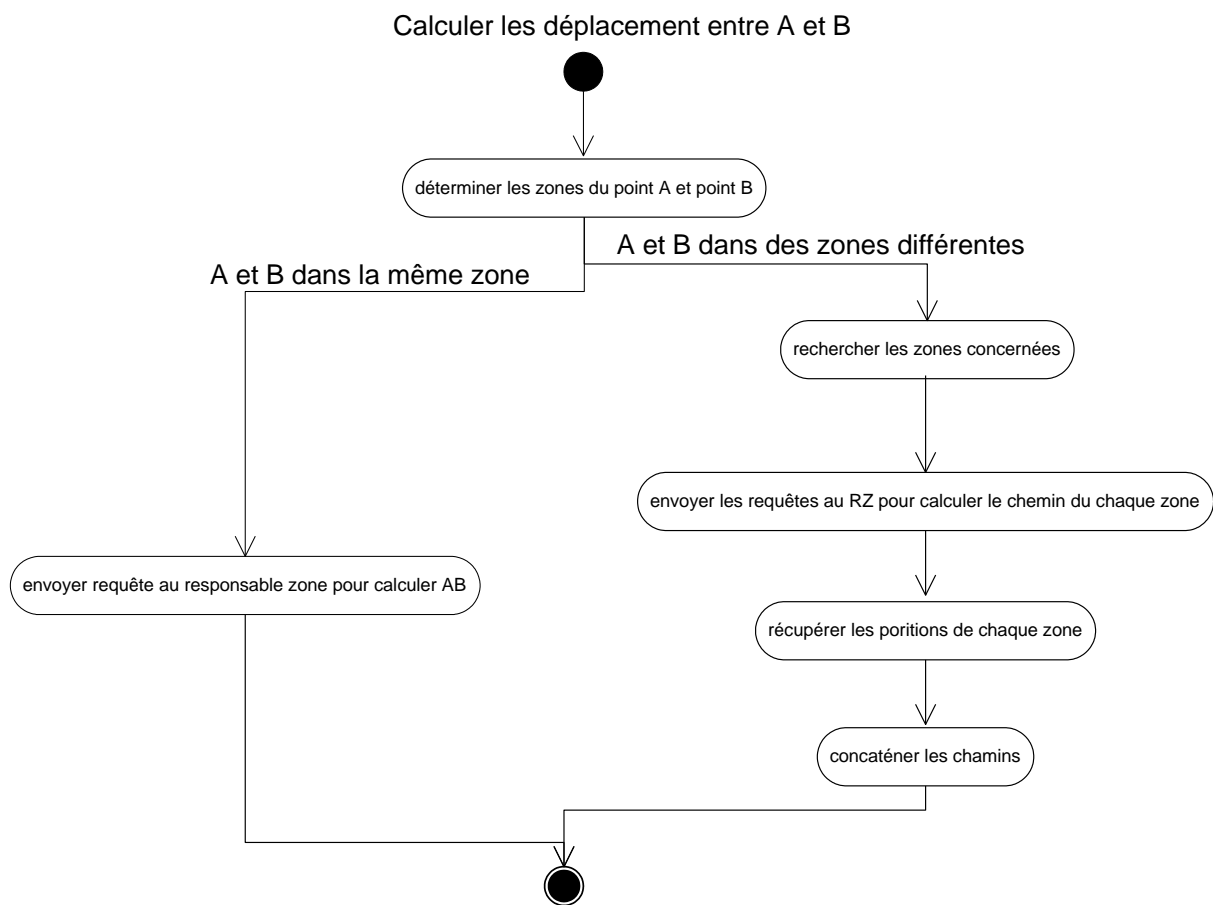


Figure III.22: Comportement de l'agent SYSTEME.

- Lancer l'agent CALCUL CHEMIN : en cas de nouvelle requête, l'agent RESPONSABLE ZONE demande à l'agent CALCUL CHEMIN de déterminer le chemin optimal, puis il l'enregistre dans la base de données, grâce à l'agent GESTIONNAIRE D'INFORMATION ;
- Envoyer le résultat à l'agent SYSTEME.

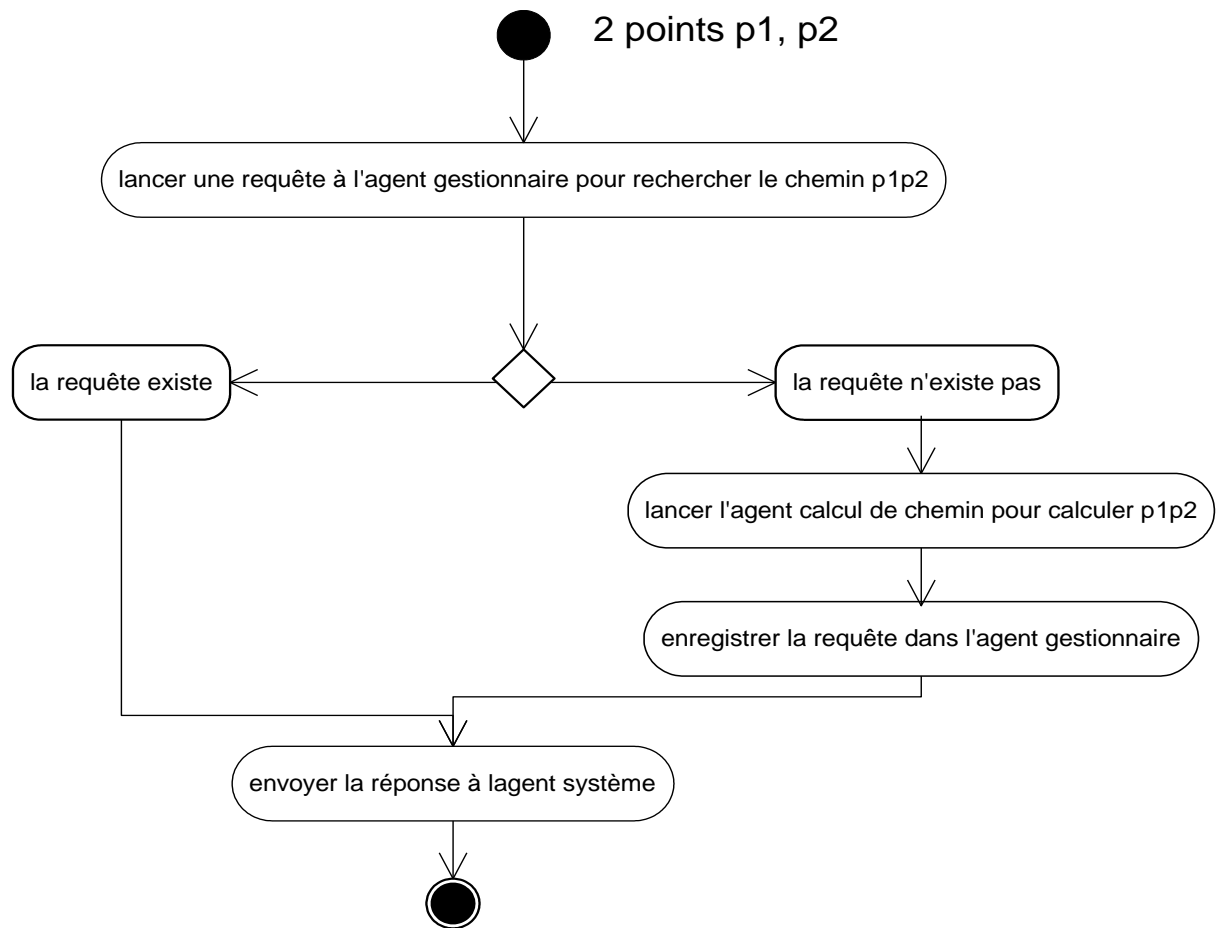


Figure III.23: Comportement de l'agent RESPONSABLE ZONE.

III.11.2.3 Agent « CALCUL CHEMIN »

Cet agent a pour rôle de calculer le chemin optimal entre deux nœuds dans sa zone en utilisant une méthode hybride (couplage entre l'algorithme de Dijkstra modifié et un algorithme génétique), que nous avons détaillée dans le chapitre précédent.

| | |
|----------------------|--|
| Identité | Communauté : Transport Groupe : Zone i Rôle : Membre |
| Etat | Libre ou occupé |
| Connaissances | Graphe de nœuds de sa zone |

Dans l'algorithme implémenté, nous avons prévu le calcul d'une population finale de 30 ou 50 solutions si possible. Les trois meilleurs chemins possibles entre la station de départ et la station d'arrivée seront proposés à l'utilisateur, alors que le reste de la population sera stocké chez l'agent GESTIONNAIRE D'INFORMATION, afin de l'utiliser en cas de perturbation.

Cet agent a ainsi une durée de vie liée à la fin de requête qui lui est associée. Il ne s'autodétruit qu'après le calcul et l'enregistrement des solutions de la requête.

III.11.2.4 Agent « GESTIONNAIRE D'INFORMATION »

L'agent GESTIONNAIRE D'INFORMATION gère tout le flux d'informations dans sa zone et s'occupe de sa mise à jour. Il établit une liaison avec la base de données de sa région qui compte cinq champs : nœud de départ, nœud d'arrivée et les trois meilleurs chemins possibles entre l'origine et la destination trouvés par l'agent CALCUL CHEMIN.

Les actions réalisées par cet agent sont alors :

- Fournir les informations nécessaires pour l'agent CALCUL CHEMIN ;
- Sauvegarder les solutions trouvés par l'agent CALCUL CHEMIN selon chaque requête : utilisation de la structure de données Liste de taille n en gardant les requêtes de plus grand nombre d'occurrences.
- Mettre à jour les informations : lors d'une perturbation, certains chemins qui étaient possibles peuvent se bloquer et d'autres se créent.

III.11.2.5 Agent « SUPERVISEUR »

L'agent SUPERVISEUR s'occupe de contrôler l'état des réseaux pour signaler les perturbations, puis de créer des agents PERTURBATION. Chaque groupe d'agents zone a un agent SUPERVISEUR. Cet agent permet de surveiller les bases de données des exploitants ou bien leurs Systèmes d'Aide à l'Exploitation (SAE), afin de signaler les changements. Dernièrement, plusieurs exploitants utilisent des systèmes multi-agents pour aider les régulateurs à résoudre les conflits créés par les perturbations [Chaib-draa 96] [Laïchour 01] [Balbo 00] [Fayech 02]. Ces systèmes offrent à notre agent de bons supports, via les moyens de coopération et communication entre les plate-formes des SMA. Cette possibilité est à développer ultérieurement.

III.11.2.6 Agent « PERTURBATION »

Ces agents sont créés par l'agent SUPERVISEUR, dès l'apparition d'une perturbation. Ils sont responsables de la gestion des incidents. Un agent PERTURBATION doit contrôler toutes les tâches liées à la perturbation, comme l'échange des informations avec les autres agents, la proposition de solutions ou le suivi. Il a ainsi une durée de vie liée à celle de la perturbation qui lui est associée. Il ne s'autodétruit qu'à la disparition complète de la perturbation ou après l'application réussie des nouvelles solutions.

III.11.3 Structures organisationnelles

III.11.3.1 Coordination et coopération

Les besoins de l'utilisateur nécessitent que l'agent «SYSTEME» coopère avec d'autres agents pour fournir un service satisfaisant l'utilisateur. Cette coopération est simple, comme une simple relation de « client/serveur ». Par exemple, on peut trouver la demande des informations sur les services de voyage (horaires, prix, etc.) et la réponse donnant des informations concernées. D'autres exemples plus complexes réalisés par plusieurs messages, comme trouver un voyage par avion de prix minimal, les agents vont coopérer suivant le protocole d'appel d'offres.

En ce qui concerne l'allocation des tâches, SMAAD utilise deux stratégies :

- L'affectation des responsabilités et des ressources nécessaires à la résolution de problèmes à un agent est parfois statique et parfois dynamique. Celle-ci est déterminée à l'avance pour les agents statiques (RESPONSABLE ZONE, SUPERVISEUR et GESTIONNAIRE D'INFORMATION) de chaque zone et l'agent « SYSTEME ».
- L'allocation dynamique et flexible des tâches, pour les agents PERTURBATION et CALCUL CHEMIN.

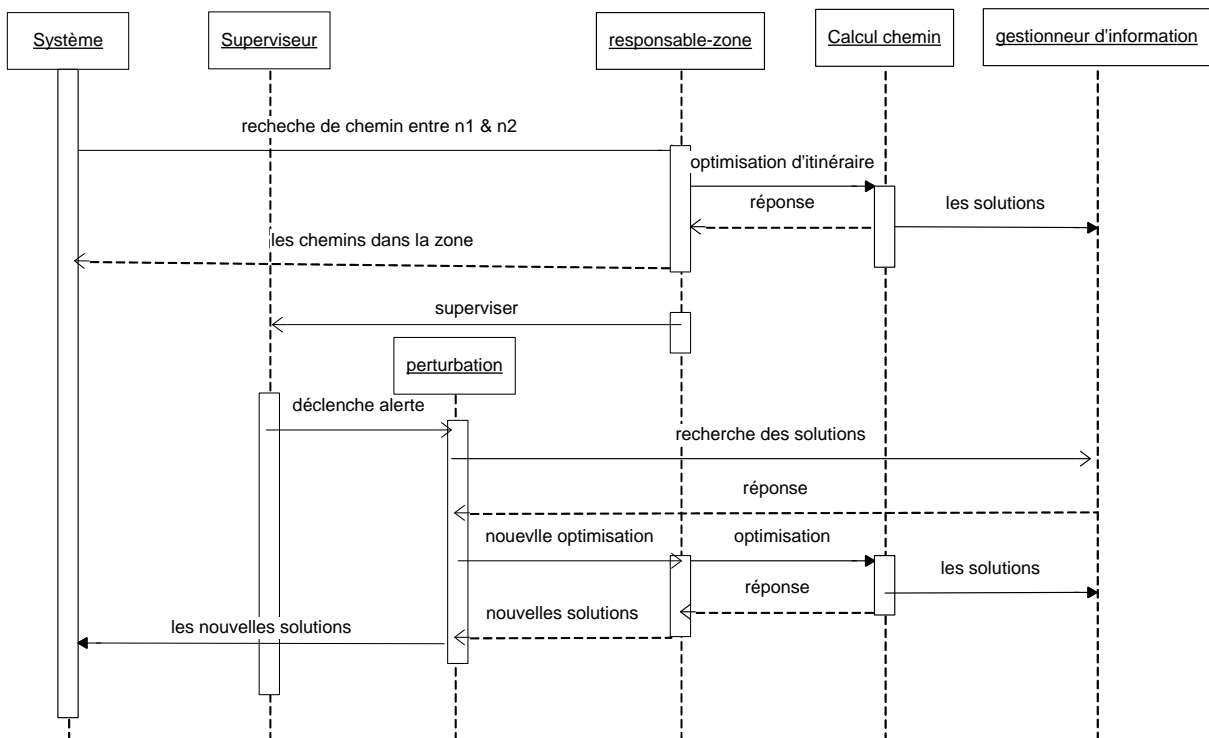


Figure III.24: Diagramme de séquences d'interaction entre agents.

III.11.3.2 Communication

Le Transport des Messages (Agent Message Transport) traite de la représentation et de la livraison des messages à travers plusieurs types de réseaux. Un message consiste en une enveloppe et un corps. L'enveloppe contient des informations, permettant au Service de Transport des Messages (Message Transport Service ou MTS) de traiter et d'acheminer les messages. Le corps du message est exprimé en FIPA-ACL. La communication entre Agents (Agent Communication) définit l'ACL et la séquence des échanges entre les agents, permettant ainsi de structurer les interactions.

De nombreuses plates-formes existent pour le développement d'applications à base d'agents logiciels. Le choix d'une plate-forme est généralement motivé par sa facilité de prise en main et par sa facilité de mise en oeuvre des différents modèles d'agent. En effet, certaines caractéristiques relatives aux agents sont plus ou moins faciles à mettre en oeuvre suivant la plate-forme de développement utilisée. La plate-forme JADE, que nous utilisons, est fondée sur la spécification de FIPA et comprend tous les composants qui permettent la gestion de la dite plate-forme selon FIPA. Ces composants sont : le ACC (*Agent Communication Channel*), le AMS (*Agent Management System*) et le DF (*Directory Facilitator*). Les agents communiquent à travers les échanges de messages. Ces messages sont représentés dans le langage de communication FIPA-ACL. Cette plate-forme nous permet d'ajouter des nouvelles ontologies. Une ontologie est une spécification d'objets, de concepts et de relations dans un domaine d'intérêt. L'intérêt d'une ontologie quand elle est partagée par des agents pour représenter leur connaissance est qu'ils ont les moyens de comprendre les « mots » utilisés dans une communication.

Un agent a la possibilité d'envoyer un message de trois façons différentes:

- 1) L'agent A envoie à son ACC local le message. L'ACC prend alors toutes les précautions pour envoyer le message à l'ACC distant correct en utilisant le MTP approprié. L'envoi du message à l'agent B sera alors géré par l'ACC distant.
- 2) L'agent A envoie directement le message à l'ACC de l'AP distant où l'agent B réside. L'ACC distant délivre ensuite le message à l'agent B. Pour utiliser cette méthode, l'agent A doit supporter l'accès à une interface MTP de l'ACC distant.

3) L'agent A envoie directement son message à l'agent B en utilisant un mécanisme de transfert direct. Le transfert du message, l'adressage et les messages d'erreurs doivent alors être gérés par les agents. Ce mode de communication n'est pas couvert par la FIPA.

Le schéma de la figure III.25 résume ces différents modes de communication:

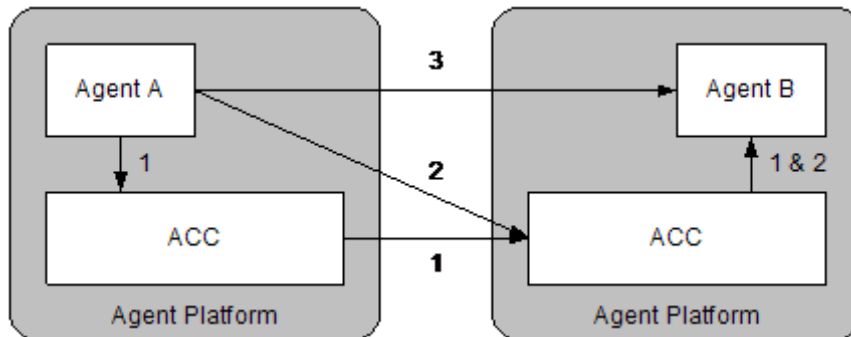


Figure III.25: Echange des messages entre les agents.

III.11.3.3 Protocole de communication

La technologie des agents est susceptible de jouer un rôle primordial dans cette révolution technologique, en permettant d'automatiser les processus d'acquisition et de traitement de l'information assurant, ainsi une prestation de services plus intelligents.

Afin d'atteindre le but souhaité, il est essentiel de fournir les standards, qui permettront aux agents de communiquer et de coopérer.

Les agents échangent des messages entre eux selon un protocole. Ce protocole respecte bien l'ACL définie par FIPA. Chaque message est composé de deux parties. La première consiste en l'en-tête du message, la deuxième représente le corps du message. Comme dans la majorité des messages utilisés dans la communication, l'en-tête donne juste une idée sur le contenu du message lui-même, ainsi que la source et la destination (Figure III.26). Alors que le corps contient le contenu du message et les paramètres à utiliser (Figure III.27). Pour appliquer ce protocole, nous avons ajouté une nouvelle ontologie Transport. Pour le moment, cette ontologie ne s'occupe que des itinéraires. Dans l'avenir, nous travaillerons pour l'étendre sur tout ce qui concerne le domaine de transport.

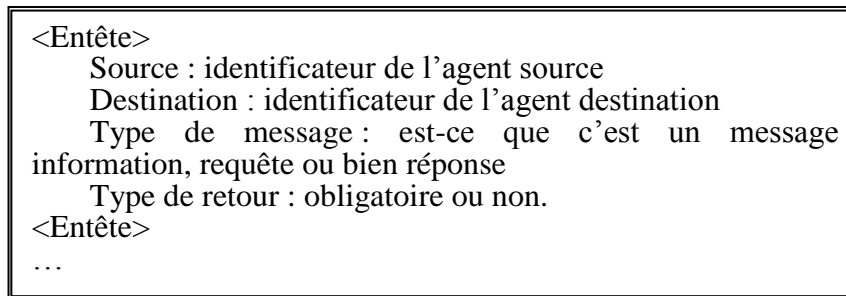


Figure III.26: Format de l'en-tête du message échangé entre les agents.

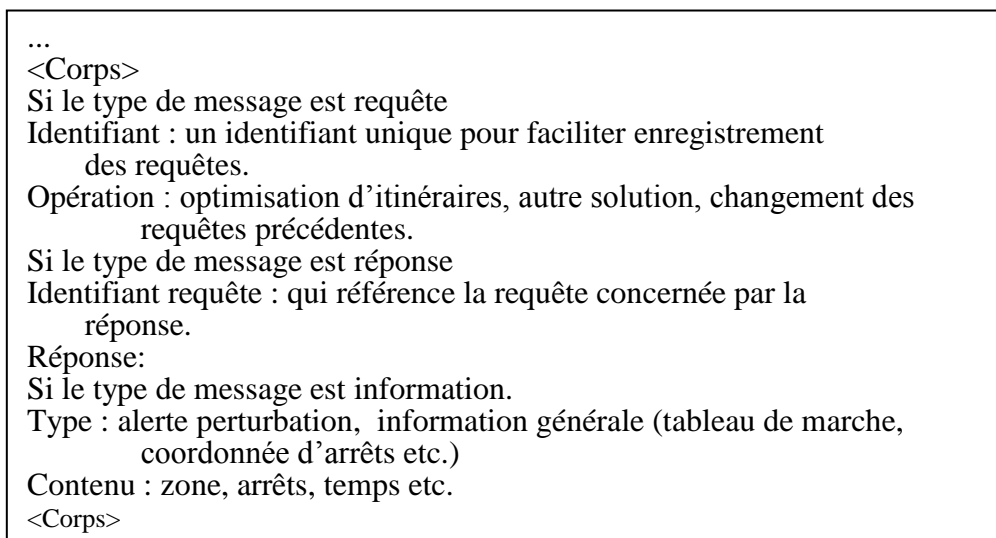


Figure III.27: Format de corps du message échangé entre les agents.

III.12 Conclusion

Bien que nous ayions pu modéliser le réseau de transport en commun par un multigraphe, la modélisation graphique de la multimodalité, du multiparamétrage et des contraintes a montré la complexité du problème d'aide au déplacement. Avec les contraintes temporelles sur les graphes, la modélisation du réseau de transport doit être dynamique. Ce phénomène est dû au changement périodique des données et aux perturbations. Avec cette complexité, les algorithmes de la théorie des graphes seuls sont incapables de résoudre ce problème. Nous avons proposé une approche d'optimisation adaptée au problème d'aide au déplacement. Cette approche utilise une méthode multicritère de recherche d'itinéraire, qui s'appuie sur une hybridation entre un algorithme de Dijkstra modifié et un algorithme génétique, pour générer une population de chemins minimums.

Afin de simplifier la complexité de l'aide au déplacement, nous avons procédé à la division du réseau de transport en plusieurs zones. Cette méthode a permis de modéliser ce réseau par

une architecture multi-zones. Cette architecture nous a montré l'aspect distribué du système, les interactions et les relations qui peuvent avoir lieu entre les différentes zones. Par la suite, nous avons présenté notre Système Multi-Agents d'Aide au Déplacement, SMAAD, pour un réseau de transport multimodal.

Pour bien expliquer les particularités du SMAAD proposé, nous avons présenté les différents agents qui le composent, ainsi que leurs architectures internes. Nous avons décrit aussi la nature de l'organisation du système considéré, afin de situer les interactions, communications et coordinations entre ces agents.

Chapitre IV

Applications et résultats

IV.1 Introduction

Dans ce dernier chapitre, nous présentons les différents résultats des simulations que nous avons réalisées. Nous commençons par présenter des applications de notre approche évolutionnaire pour l'aide au déplacement, afin de valider nos méthodes de résolution. Dans ce but, nous nous intéressons à plusieurs scénarii de perturbations qui affectent le trafic au sein d'un réseau multimodal. Ces scénarii sont appliqués sur des données réelles de l'agglomération lilloise. Nous testons la capacité de notre approche à trouver des solutions alternatives, sans recalculer les chemins.

Dans la deuxième partie de ce chapitre, nous exposons le fonctionnement global du SMAAD que nous avons proposé dans le chapitre IV. Pour valider le système que nous avons conçu, nous avons effectivement simulé l'exploitation d'une partie d'un réseau de transport, afin d'illustrer le comportement des différents agents du SMAAD que nous avons développé, en utilisant des plate-formes existantes.

IV.2 Outil informatique utilisé

Pour développer les algorithmes utilisés dans notre approche d'optimisation, nous avons utilisé le langage de programmation orienté objet JAVA. Ce langage reprend en grande partie la syntaxe du langage C++, très utilisé par les informaticiens. Néanmoins, Java a été épuré des concepts les plus subtils et à la fois les plus déroutants du C++, tels que l'héritage multiple. Les concepteurs ont privilégié l'approche orientée objet, de sorte qu'en Java, tout est objet à l'exception des types primitifs (nombres entiers, nombres à virgule flottante, etc.).

Notre choix est basé sur les caractéristiques de ce langage. Les applications Java peuvent être exécutées sur tous les systèmes d'exploitation pour lesquels a été développée une plate-forme Java, dont le nom technique est JRE (*Java Runtime Environment* - Environnement d'exécution

Java). Cette dernière est constituée d'une Machine Virtuelle Java JVM (*Java Virtual Machine*), le programme qui interprète le code Java et le convertit en code natif. Mais le JRE est surtout constitué d'une bibliothèque standard, à partir de laquelle doivent être développés tous les programmes en Java. C'est la garantie de portabilité qui a fait la réussite de Java dans les architectures client-serveur, en facilitant la migration entre serveurs, très difficile pour les gros systèmes.

La seconde caractéristique du langage, l'indépendance vis-à-vis de la plate-forme, signifie que les programmes écrits en Java peuvent fonctionner sur divers types de matériel informatique sans réécriture, selon le slogan « *write once, run anywhere* » (écrire le programme une seule fois, et le faire fonctionner sur n'importe quel ordinateur).

La portabilité du langage Java est obtenu, en compilant le code source d'un programme dans un pseudo-code intermédiaire, (appelé "*bytecode Java*", des instructions machine standardisées et destinées à un processeur virtuel). Le code est par la suite exécuté par une implémentation de cette machine virtuelle (JVM), un programme écrit en code natif sur le matériel cible qui traduit le pseudo-code Java en code utilisable pour le matériel concerné. De plus, des bibliothèques sont offertes pour fournir l'accès à des fonctionnalités propres à la machine cible, comme le graphisme ou le réseau, d'une manière unifiée. Le langage Java offre aussi le support des programmes avec plusieurs processus légers (*threads*).

IV.3 Validation des algorithmes d'optimisation

Suite à l'étude des méthodes existantes, nous avons choisi l'utilisation d'une approche hybride entre un algorithme de Dijkstra modifié et un algorithme évolutionnaire. Ce dernier repose sur un algorithme génétique. Nous avons effectué des tests afin de bien choisir P_c et P_m . Ces deux paramètres ont une grande influence sur la convergence de l'algorithme. Pour éviter la convergence prématurée, nous devons choisir des bonnes valeurs pour ces paramètres. Le tableau suivant représente les résultats trouvés pour des valeurs moyennes calculées sur 40 tests. En se basant sur les valeurs utilisées en général, nous avons effectué les tests sur des valeurs allant de 0,5 à 0,8 pour la probabilité de croisement et des valeurs comprises entre 0,01 et 0,03 pour la probabilité de mutation.

| P_c | 0,5 | | | 0,6 | | | 0,7 | | | 0,8 | | |
|---------------|------|------|------|------|------|------|------|------|------|------|------|------|
| P_m | 0,01 | 0,02 | 0,03 | 0,01 | 0,02 | 0,03 | 0,01 | 0,02 | 0,03 | 0,01 | 0,02 | 0,03 |
| F_{op} | 17 | 15,1 | 15,3 | 14,8 | 14,3 | 15,3 | 14,7 | 15,4 | 17,2 | 16,1 | 16,2 | 15 |
| T_{emp} (s) | 3,5 | 3,7 | 3,8 | 3,5 | 3,3 | 3,5 | 3,2 | 3,7 | 4 | 4,2 | 3,8 | 5,1 |
| σ | 2 | 1,3 | 1,5 | 0,8 | 0,1 | 1,3 | 0,4 | 1,6 | 3,7 | 2,8 | 2,5 | 2,6 |

Tableau IV.1 : Recherche de P_m et P_c .

F_{op} est la valeur moyenne de la fonction objectif.

T_{emp} est le temps de calcul.

$\sigma = F_{op} - 14,3 + T_{emp} - 3,2$ est la somme des distances (écarts) entre chaque solution et les meilleurs valeurs de la fonction objectif (chiffre grisé de la ligne F_{op}) et le meilleur temps de calcul (chiffre grisé de la ligne T_{emp}).

On compare les valeurs optimums des solutions obtenues pour les différentes valeurs de P_c et de P_m . Le deuxième critère de comparaison est le temps de calcul. Le troisième critère est l'écart entre chaque solution et la meilleure solution.

D'après le tableau précédent, on voit que la meilleure solution obtenue est en utilisant $P_c=0.6$ et $P_m=0.02$, alors que le minimum de temps de calcul est atteint pour les valeurs $P_c=0.7$ et $P_m=0.01$ (chiffres grisés).

Pour cette optimisation, nous avons utilisé une sélection par roulette de Goldberg, un renouvellement d'une partie de la population proportionnel à P_c , sur laquelle on applique des croisements en un ou deux points. En outre, nous utilisons l'opérateur de mutation avec la probabilité P_m . Afin de déterminer quels individus (parents) P et leurs enfants issus des opérateurs génétiques, nous utilisons la sélection par tournois.

Nous avons testé l'algorithme de calcul du plus court chemin et le système distribué sur des zones de la région lilloise. Dans notre exemple, nous simulons un réseau avec 4 modes de transport en commun. Ces modes sont le métro, le bus, le tramway, et le train régional. Les trois premiers modes sont assurés par Transpole et le dernier par la SNCF. Nous avons divisé le réseau en trois zones. Chaque zone est représentée par un graphe de n stations et m arcs. Dans nos simulations, nous utilisons ($n=78$, $m=80$) pour la première zone, ($n=92$, $m=116$) pour la deuxième et ($n=108$, $m=134$) pour la troisième.

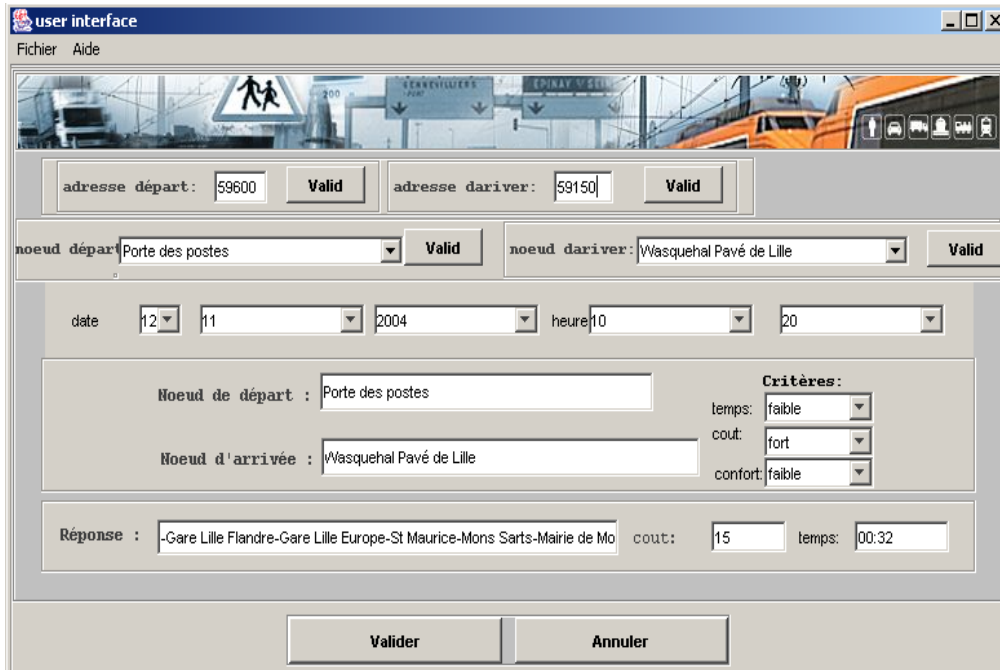


Figure IV.1: Exemple de l'interface utilisateur.

Dans le tableau IV.2, nous montrons les résultats de notre méthode appliquée sur une de nos bases de données. Ces solutions représentent une partie des individus de notre population finale. Il reste à signaler que l'affichage de ces résultats pour les utilisateurs n'est pas encore terminé. Pour ce point, l'amélioration de l'interface utilisateur (en lui proposant plusieurs solutions) est envisageable prochainement. On notera aussi que les coûts et les temps donnés sont calculés en utilisant des données de test que nous avons introduites dans nos bases des données.

Remarques :

- Les indices M1, M2 indiquent les lignes de métro 1 et 2, alors que T1 et T2 indiquent les tramways 1 et 2.
- (T2 10:32) traduit le changement de mode vers le tramway 2 qui part à 10h 32 minutes, LXX représente la ligne de bus numéro XX.
- Le coût n'indique pas que le prix, il représente aussi le résultat rendu en utilisant la fonction objectif du chapitre III appliquée sur nos données de test avec les trois critères à optimiser. L'objectif ici est de montrer les coûts optimisés par notre méthode, alors que les résultats affichés à l'utilisateur lui indiquent le prix et le temps de parcours.

| Solutions | Coût C_f | Temps de parcours(mn) |
|--|------------|-----------------------|
| (M1) Porte des postes,Wazemmes,Gambetta, Republique, Rihour, (M2) Gare Lille Flandres, Gare Lille Europe,St Maurice,Mons Sarts,Mairie de Mons,Fort de Mons,les Prés,Jean Jaurès,Wasquehal Pavé de Lille | 14 | 00:32 |
| (M2) Porte des postes,port d'Arras,Port de Douai,Port de Valenciennes, Lille Grand palais,Mairie de Lille,Gare Lille Flandres, Gare Lille Europe,St Maurice,Mons Sarts,Mairie de Mons,Fort de Mons,les Prés,Jean Jaurès,Wasquehal Pavé de Lille | 15 | 00:37 |
| (M1) Porte des postes,Wazemmes, Gambetta, Republique, Rihour, (T2 10:32) Gare Lille Flandres, Gare Lille Europe, Romarin,Botanique,St Maur, Buisson,Brossolette, Clemenceau, (T1 10: 56) Croisé Laroche,Acacias,Pt de Wasquehal,La Terrasse,Wasquehal Pavé de Lille | 29 | 00:45 |
| (M1) Porte des postes,Wazemmes,gambetta,Republique, Rihour, Gare Lille Flandres, Coulier, Fives, Marbrerie, Hellemmes, Lezennes, (L41 10:38) Pont de bois,Stadium, Château,Chanterelle,Parc Urbain, Comices, courtilles, Cuisinerie,I.U.T.Crest, Peupliers,La Fontaine,Collège Molière, Cité Babylone, Luis Constant,(M2)Jean Jaurès, Wasquehal Pavé de Lille | 45 | 00:57 |
| (M2 10:22) Porte des postes,Montebello,Cormontaigne (L12 10:25) Cormontaigne, Leclerc,Colbert,Faculte catholique,Sacre coeur,Foch, De Gaulle,Théâtre,(M2) Gare Lille Flandres, Gare Lille Europe,St Maurice, Mons Sarts,Mairie de Mons,Fort de Mons,les Prés,Jean Jaurès,Wasquehal Pavé de Lille | 45 | 00:45 |
| (M2 10:22) Porte des postes,Montebello,Cormontaigne (L12 10:25) Cormontaigne, Leclerc, Colbert,Faculte catholique,Sacre coeur,Foch, De Gaulle,Théâtre, (M1)Gare Lille Flandre, Coulier, Fives, Marbrerie, Hellemmes, Lezennes, (L41 10:57) Pont de bois,Stadium, Château, Chanterelle,Parc Urbain,Comices, courtilles, Cuisinerie, I.U.T.Crest, Peupliers, La Fontaine,Collège Molière, Cité Babylone, Luis Constant, (M2)Jean Jaurès, Wasquehal Pavé de Lille | 76 | 01:15 |
| (M2 10:22) Porte des postes,Montebello,Cormontaigne (L12 10:25) Cormontaigne, Leclerc,Colbert,Faculte catholique,Sacre coeur,Foch, De Gaulle,Théâtre, (T1 10: 35) Gare Lille Europe,Romarin, Botanique,St Maur,Buisson,Brossolette, Clemenceau, Croisé Laroche, Acacias,Pt de Wasquehal,La Terrasse, Wasquehal Pavé de Lille | 59 | 01 :07 |

Tableau IV.2 : Exemple de résultats pour un scénario.

On distingue deux cas de perturbations : une perturbation se produit avant le début de voyage et une autre pendant le voyage. Si la première demande juste de rechercher une solution alternative dans la population, pour la deuxième on doit déterminer la position du client. Pour

simplifier, nous utilisons la localisation selon le temps de passage et l'instant de perturbation. Par exemple, si la perturbation est signalée à 10h25 l'utilisateur n'a pas encore dépassé la station « Gare Lille Flandres » dans la troisième solution.

Supposons que l'on a une perturbation avant le début de déplacement. Nous testerons les trois scénarios suivants :

Scénario1 : sur le métro 1 : dans ce cas, nous remarquons que la première solution du tableau ci-dessus n'est plus valide alors que la deuxième l'est.

Scénario2 : sur le métro 2 : Dans ce deuxième cas, ni la première ni la deuxième solution ne sont valables alors que la troisième l'est.

Scénario3 : En générant d'autres types de perturbations qui concernent un ensemble de stations, comme par exemple la partie proche de la station « gare Lille Europe », nous remarquons que la population finale contient des solutions qui ne passent pas par les mêmes stations, comme l'avant-dernière solution par rapport aux autres. Donc si une perturbation couvre une partie de la zone, la probabilité de trouver la solution dans la population finale est grande, grâce à la diversification des individus dans la population par rapport aux zones. Dans ce cas, il faut éviter l'utilisation du métro (M2), le tramway (T2) à côté de la zone perturbée donc, la quatrième solution est la plus probable.

Quant aux perturbations pendant le voyage, nous remarquons que les points de changement de mode sont nombreux dans les solutions générées. Le changement du trajet dans ces points explique les cas de graphe dynamique que nous avons détaillé auparavant. L'application de l'algorithme de la gestion de la population finale a donné de très bonnes solutions de secours dans ces types de perturbations.

Supposons que la perturbation soit survenue avant que l'utilisateur arrive à la station « Gare Lille Flandres »

Scénario1 : sur le métro 1 : dans le tableau, on n'a pas de solution, donc le système propose à l'utilisateur de continuer à pied jusqu'à la station ou de revenir à la station de départ.

Scénario2 : sur le métro 2 : la solution trois est toujours valable donc l'utilisateur peut changer à la station gare Lille Flandres en utilisant le T2, au lieu de M2.

Scénario3 : au tour de la station « Gare Lille Europe » : l'utilisateur peut continuer avec la quatrième solution.

Afin de tester l'efficacité des valeurs obtenues par nos approches, nous avons effectué des comparaisons avec un algorithme de colonie de fourmis (ACF), développé par un chercheur

de Lille 1 S. ZIDI [Zidi 06] dans le cadre de ses travaux dans le domaine de la régulation de trafic. L'objectif est de trouver des solutions dans des cas de perturbation. La comparaison est faite sur les valeurs optimales obtenues par les deux algorithmes. Le graphe utilisé dans cette application est la première zone du réseau de transport de Lille. Ce graphe comprend 58 arcs et 51 stations. Le réseau de transport utilise deux modes : deux lignes de métro (ligne1 et ligne2) et 3 lignes de bus (ligne12, ligne14 et ligne17).

Pour effectuer une comparaison entre les deux approches (AG et ACF), on doit les appliquer au même problème et avec des conditions équivalentes. Donc, on a commencé par le premier problème d'aide au déplacement des usagers de ce réseau de transport de Lille. Pour adapter l'algorithme de colonie de fourmis au premier problème de déplacement, on a changé sa fonction de coût par notre fonction *fitness* évoquée dans le chapitre III. Ce qui nous a permis une première comparaison des deux approches.

Le tableau IV.3 montre les résultats des deux algorithmes pour 4 scénarii distincts avec différentes pondérations des 3 critères.

scénario1 : Une perturbation affectant les services allant de la station Faculté Catholique à la station CHR B Calmette.

scénario2 : Une perturbation affectant les services allant de la station Wazemmes à la station Isly.

scénario3 : Une perturbation affectant les services allant de la station Cormontaigne à la station Jaurès.

scénario4 : Une perturbation affectant les services allant de la station Place de la solidarité à la station H Renault.

| pondération | | | scénario1 | | scénario2 | | scénario3 | | scénario4 | |
|-------------|---------|----------|-----------|-----|-----------|-----|-----------|------|-----------|-----|
| α | β | δ | AG | ACF | AG | ACF | AG | ACF | AG | ACF |
| 0 | 0 | 1 | 7 | 7 | 6 | 6,3 | 12 | 11 | 13 | 13 |
| 0 | 0,25 | 0,75 | 7,5 | 7,5 | 6,5 | 6,5 | 10,4 | 10,2 | 14 | 15 |
| 0 | 0,5 | 0,5 | 7,3 | 7,8 | 6,2 | 6,1 | 9 | 9 | 15 | 15 |
| 0 | 0,75 | 0,25 | 7,5 | 7,5 | 5,5 | 5,5 | 9,3 | 9,3 | 18 | 18 |
| 0 | 1 | 0 | 7 | 7 | 7 | 6,7 | 8,6 | 8,8 | 21 | 21 |
| 0,25 | 0 | 0,75 | 8,1 | 8 | 7 | 7 | 10,5 | 10,5 | 15 | 16 |

| | | | | | | | | | | |
|------|------|------|------|------|-----|-----|-----|-----|------|------|
| 0,25 | 0,25 | 0,5 | 8,25 | 8,25 | 7,2 | 7,2 | 9,2 | 9 | 16 | 16 |
| 0,25 | 0,5 | 0,25 | 8,5 | 8,5 | 6,9 | 7 | 8,5 | 8,4 | 20 | 20 |
| 0,25 | 0,75 | 0 | 7,2 | 7,4 | 6,5 | 6,5 | 7,5 | 7,5 | 14,7 | 14 |
| 0,5 | 0 | 0,5 | 8,7 | 8,9 | 9,3 | 9,3 | 10 | 10 | 18 | 19 |
| 0,5 | 0,25 | 0,25 | 8,3 | 8,3 | 7,9 | 8,1 | 8,4 | 8,6 | 17,3 | 17,3 |
| 0,5 | 0,5 | 0 | 9 | 9 | 8 | 8 | 8 | 8 | 16 | 18 |
| 0,75 | 0 | 0,25 | 9 | 8,25 | 9,3 | 9,6 | 8,2 | 8,2 | 17 | 17 |
| 0,75 | 0,25 | 0 | 8,6 | 8,6 | 9,5 | 9,5 | 9,1 | 9 | 19,1 | 18,7 |
| 1 | 0 | 0 | 11 | 8 | 9 | 9 | 10 | 10 | 23 | 23 |

Tableau IV.3 : Les coûts minimaux trouvés par les deux algorithmes pour différentes pondérations.

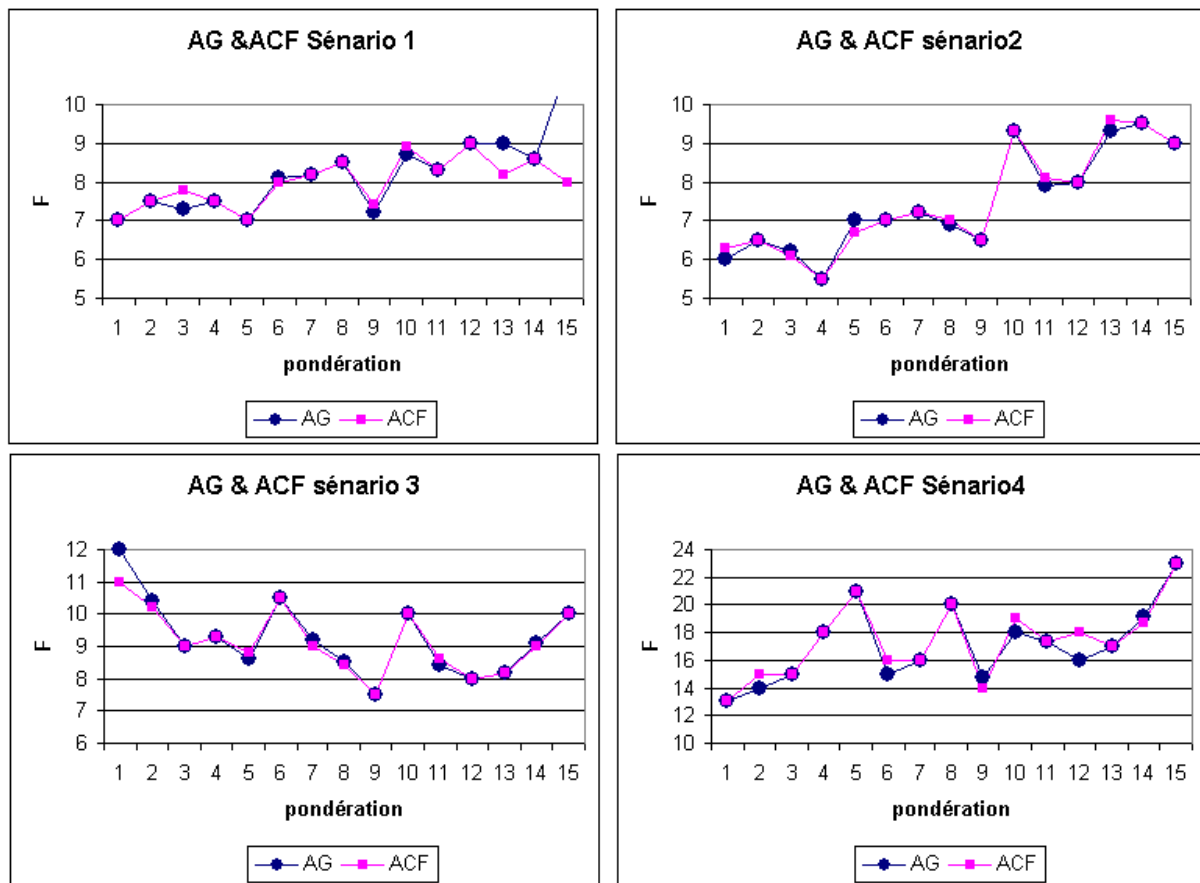


Figure IV.2: Comparaison entre AG et ACF pour 4 scénarii de perturbation.

Nous proposons d'utiliser l'algorithme génétique en temps anticipé pour générer des solutions et ainsi choisir, parmi ces solutions, celle qui convient le mieux pour résorber une perturbation.

En comparant les temps de calcul de notre algorithme sur les trois zones, nous avons remarqué qu'en général il n'y a pas une grande différence, alors que les nombres des nœuds et des arcs sont très différents entre les zones. Ces résultats font l'objectif attendu de l'utilisation des métaheuristiques dans notre approche, alors que si nous utilisons un algorithme de la théorie des graphes seul le temps de calcul va augmenter exponentiellement avec l'augmentation du nombre des nœuds et des arcs.

Avec une population de taille 30 individus par génération, nous avons obtenu les résultats du tableau IV.4 pour 100 itérations, en notant P la probabilité de trouver une solution alternative dans la population finale en cas de perturbation. Les valeurs de cette probabilité sont calculées statistiquement. Après plusieurs scénarii de perturbation de test, nous avons trouvé les résultats reportés dans le tableau. Nous avons effectué plusieurs tests d'optimisation en changeant le nombre de critères à optimiser. Par la suite, nous générons des perturbations aléatoires pour calculer les chances de trouver une solution alternative dans la population finale de l'algorithme d'optimisation.

| perturbation en mode | | C ₁ | C ₂ | C ₃ | C ₂ ,C ₁ | C ₁ ,C ₃ | C ₂ ,C ₃ | C ₁ ,C ₂ ,C ₃ |
|----------------------|----------|----------------|----------------|----------------|--------------------------------|--------------------------------|--------------------------------|--|
| sans contrôle | P | 74% | 73% | 74% | 79% | 81% | 80% | 83% |
| avec contrôle | P | 91% | 92% | 89% | 96% | 95% | 92% | 97% |

Tableau IV.4 : Des résultats (perturbation en mode) des deux approches (sans & avec) contrôle des opérateurs.

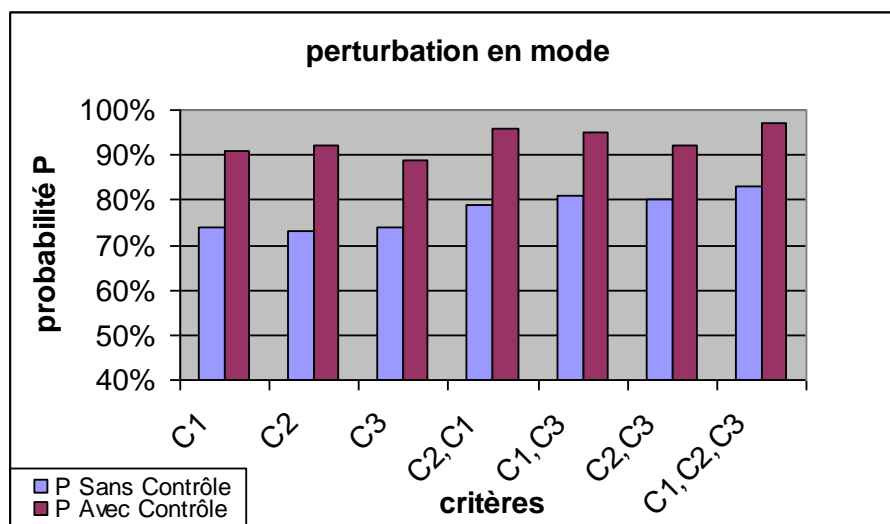


Figure IV.3: Probabilité de trouver une solution en cas de perturbation sur les modes.

D'après la figure IV.3, nous remarquons que les chances de trouver une solution alternative dans la population finale sont plus élevées si on applique le contrôle des opérateurs. Dans le

cas d'une perturbation sur les modes, les résultats donnés par l'utilisation du contrôle des opérateurs sont toujours meilleurs, même si le nombre de critères à optimiser varie.

Pour les résultats du tableau IV.3, les perturbations générées aléatoirement ne concernent que les modes de transport. Alors nous avons effectué des scénarii de test avec des perturbations en zone. Après plusieurs tests en changeant les critères à optimiser, nous avons obtenu des statistiques montrant l'évolution de la probabilité entre le cas de contrôle des opérateurs et l'application de ces opérateurs sans contrôle dans l'algorithme évolutionnaire étudié (tableau IV.5).

| perturbation en zone | | C ₁ | C ₂ | C ₃ | C ₂ ,C ₁ | C ₁ ,C ₃ | C ₂ ,C ₃ | C ₁ ,C ₂ ,C ₃ |
|----------------------|----------|----------------|----------------|----------------|--------------------------------|--------------------------------|--------------------------------|--|
| sans contrôle | P | 78% | 78% | 73% | 85% | 78% | 80% | 83% |
| avec contrôle | P | 92% | 95% | 91% | 97% | 93% | 94% | 98% |

Tableau IV.5 : Des résultats (perturbation en zone) des deux approches (sans & avec) contrôle des opérateurs.

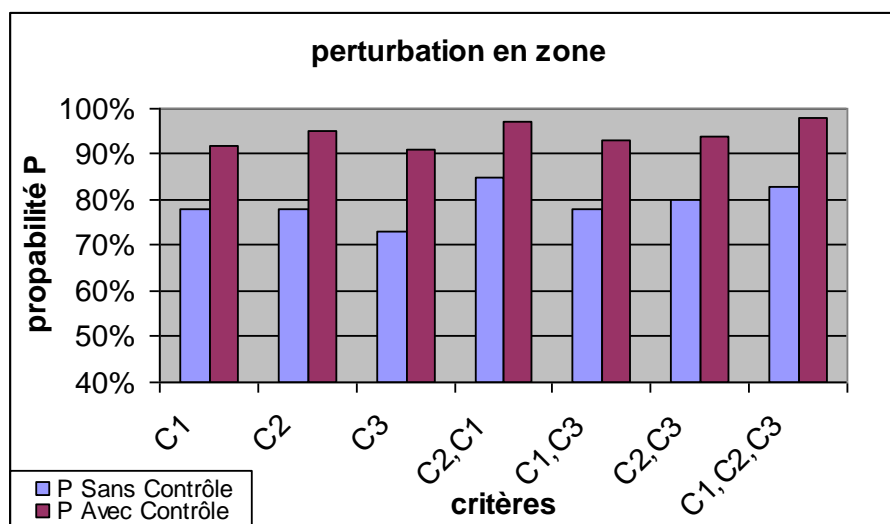


Figure IV.4: Probabilité de trouver une solution en cas de perturbation en zone.

Les histogrammes précédents (Figure IV.4) montrent que les chances de trouver une solution dans la population finale en cas de perturbation en zone sont améliorées en utilisant les critères de contrôle des opérateurs génétiques. Cette amélioration est garantie, pour l'optimisation d'un, deux ou trois critères, selon le choix de l'utilisateur.

Suite à ces résultats, nous avons vérifié le test d'arrêt de l'algorithme évolutionnaire utilisé en terme de nombre d'itérations. Ainsi, suite à des tests sur plusieurs scénarii nous avons remarqué que l'algorithme converge souvent avant les 100 premières itérations.

| Convergence (itérations) | C ₁ | C ₂ | C ₃ | C ₂ ,C ₁ | C ₁ ,C ₃ | C ₂ ,C ₃ | C ₁ ,C ₂ ,C ₃ |
|--------------------------|----------------|----------------|----------------|--------------------------------|--------------------------------|--------------------------------|--|
| sans contrôle | 63 | 67 | 60 | 70 | 76 | 69 | 91 |
| avec contrôle | 57 | 60 | 56 | 71 | 67 | 60 | 82 |

Tableau IV.6 : Convergence de l'algorithme dans les deux approches (sans & avec) contrôle des opérateurs.

Suivant le tableau IV.6, nous avons remarqué que nous pouvons changer le test d'arrêt de notre algorithme. En vérifiant la convergence avant de terminer toutes les itérations, nous améliorons les performances de notre algorithme.

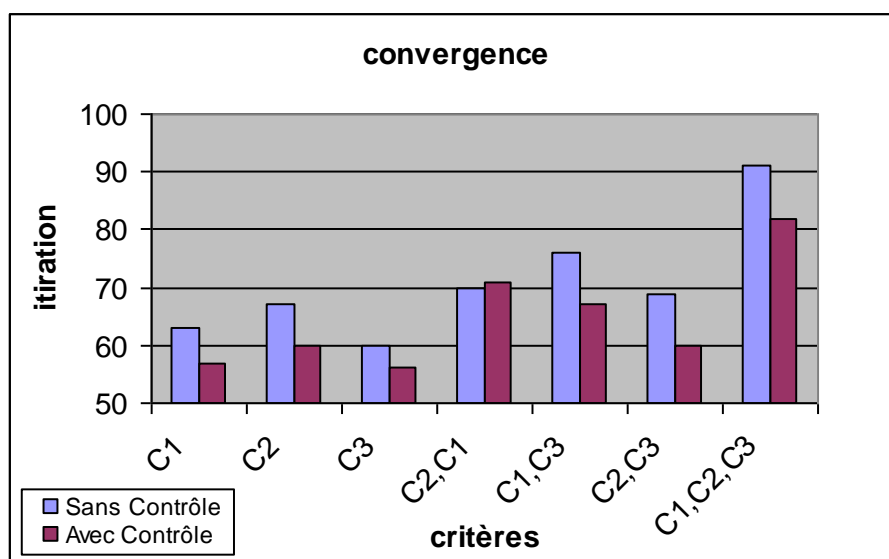


Figure IV.5: Comparaison de la convergence entre les deux méthodes d'optimisation.

D'après la Figure IV.5, nous remarquons que l'utilisation des critères de contrôle des opérateurs génétiques accélère la convergence de notre algorithme. Bien que cette accélération améliore le temps d'exécution de l'algorithme, elle peut aussi engendrer un phénomène de convergence prématuré. Ce phénomène peut influencer dans le mauvais sens sur la qualité des solutions, même si nos études statistiques n'ont pas mis ces cas en évidence.

IV.4 Système multi-agent

Pour développer des SMA, il est pratique de choisir une plate-forme. A l'heure actuelle, plusieurs outils ont été développés pour la réalisation d'agents. Si ces plates-formes sont très nombreuses, elles ne répondent pas toutes aux critères que nous nous sommes fixés. Ce

paragraphe relate notre démarche dans le choix d'une plate-forme de développement d'agents.

Etant donné que la validation de notre système n'est pas réalisée sur un site réel, il a fallu simuler le comportement du réseau pour que le SMAAD reste dynamique, utiliser des zones peu denses en terme de nœuds, et créer des fichiers de description de graphe avec des capacités que nous avons fixées arbitrairement pour l'instant. Après une étude détaillée des plate-formes existantes, nous avons procédé par élimination pour n'en garder que trois, en l'occurrence Magique, MadKit et Jade. Après des tests d'implémentation, nous avons éliminé la première, parce qu'elle demande plus de développement pour définir les agents. Entre les deux autres, le choix était plus difficile, parce qu'elles proposent presque les mêmes caractéristiques et les deux proposent les fonctionnalités attendues pour notre système. En premier lieu, nous avons implémenté notre système en utilisant la plate-forme Madkit. Dans la partie suivante, nous montrons les résultats de cette implémentation. Dans notre équipe, nous travaillons afin de développer une plate-forme générique qui regroupe tous les services autour du transport. Dans ce contexte, et après plusieurs réunions de travail avec mes collègues, on a opté pour l'utilisation de la plate-forme Jade, qui assure la migration des agents entre les sites. Vu que les deux plate-formes séparent la structure et les méthodes ou compétences des agents, le changement de madKit vers JADE n'était pas difficile, parce que ces compétences sont implémentées séparément. De plus, les deux plate-formes utilisent Java comme langage de programmation.

IV.4.1 Plate-forme multi-agent : MadKit

Dans cette partie, nous présenterons la plate-forme MadKit que l'on a utilisée, en premier lieu, pour développer le système multi-agent proposé. Nous allons commencer par voir brièvement ce qu'est MadKit et ce que cette plate-forme permet de faire. Ensuite, nous détaillerons son architecture, ainsi que son fonctionnement. Enfin, nous terminerons par voir comment développer sous Madkit.

IV.4.1.1 Qu'est-ce que MadKit ?

MadKit est une plate-forme de conception et d'exécution de systèmes multi-agents écrite en Java. Elle a l'originalité d'être basée sur un modèle organisationnel, plutôt qu'une architecture d'agents ou un modèle d'interaction spécifique.

MadKit est aussi une plate-forme distribuée, toutes les considérations à propos des composants basiques de distributions, comme les « sockets » ou les « ports » sont totalement transparentes. Cette plate-forme a été développée par Olivier Gutknecht, Jacques Ferber ainsi que Fabien Michel du LIRMM (Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier).

IV.4.1.2 Modèle organisationnel : Agents/Groupes/Rôles (AGR)

Le modèle AGR permet d'exprimer et d'analyser divers systèmes multi-agents, en utilisant avant tout des concepts organisationnels. Les principes de ce modèle sont les suivants :

- Chaque agent est membre d'un ou de plusieurs groupes et possède un ou plusieurs rôles.
- Chaque groupe est composé d'un ou de plusieurs agents qui ont un rôle au sein de ce groupe.
- Chaque rôle est défini au sein d'un groupe et appartient à un ou plusieurs agents.

Le SMAAD proposé a été spécifié de manière répartie. Chaque zone présente un groupe à part et comporte un module d'assistance, composé des agents : RESPONSABLE ZONE, CALCUL CHEMIN, PERTURBATION, SUPERVISEUR et GESTIONNAIRE D'INFORMATION et les agents exécutants, un agent CALCUL CHEMIN, agent PERTURBATION, agent SUPERVISEUR et agent GESTIONNAIRE D'INFORMATION. Un « agent système », du groupe *Système*, supervise toutes les régions du réseau de transport. Tous les groupes du SMAAD font partie de la communauté Transport.

Etant donné que toutes les zones ont une architecture identique de trois agents qui jouent le même rôle dans chaque zone, nous nous sommes limités à la création de trois zones sur MadKit pour des raisons de simplicité, puisque trois zones avec l'agent SYSTEME suffisent pour mettre en évidence les fonctionnalités de notre SMAAD (Figure IV.6).

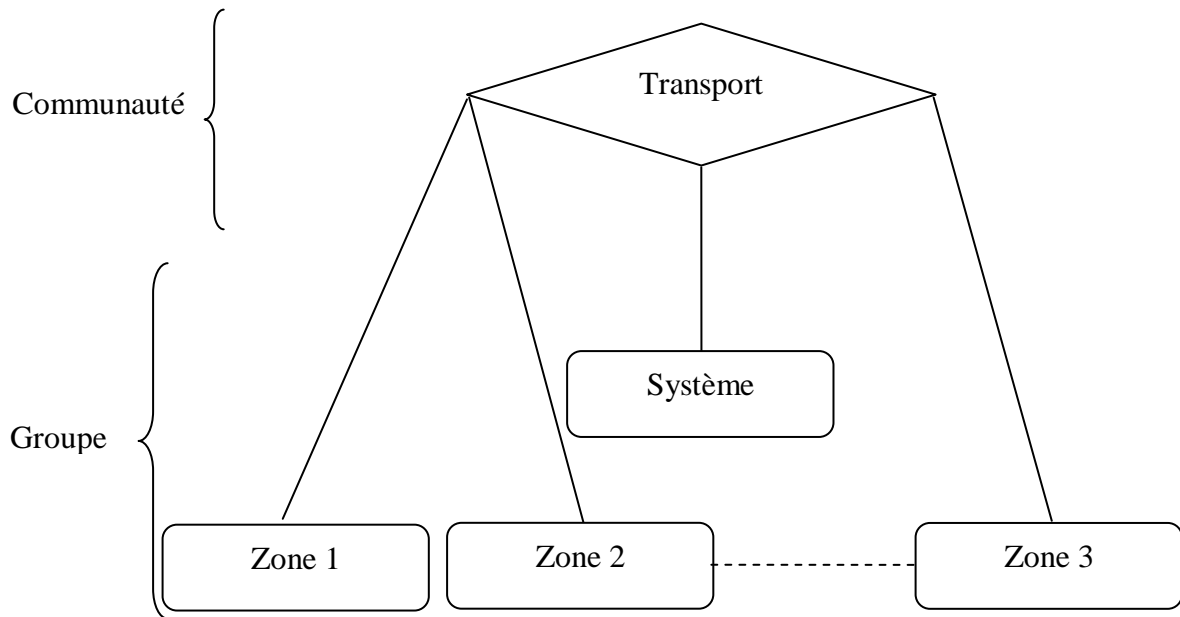


Figure IV.6: Hiérarchie par niveau du SMAAD réalisé.

Nous avons d'abord implémenté notre application sur une structure monoposte, ce qui limite les ressources et réduit la rapidité d'exécution. Par conséquent, nous avons opté pour une architecture répartie, afin d'assurer une réelle distributivité et une exécution asynchrone des agents. La plate-forme multi-agent devient alors un ensemble de sous-systèmes spécialisés qui communiquent via Intranet. Nous retrouvons un système qui gère la communication et les flux de données (SYSTEME) entre les zones, un système qui simule le fonctionnement de la zone (agent RESPONSABLE ZONE, CALCUL CHEMIN ...) et un système lié aux données statiques et dynamiques de la zone sur une base de données, l'agent GESTIONNAIRE D'INFORMATION.

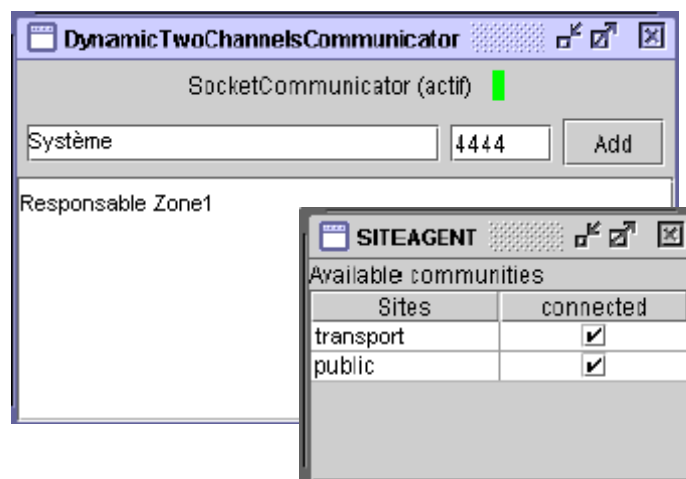


Figure IV.7: Image de « Communicator » et « SiteAgent ».

La méthode pour interconnecter plusieurs postes est la suivante.

Pour commencer, il faut démarrer l'agent "Communicator", qui est fourni avec la plate-forme, ce dernier démarre un autre agent, le "SiteAgent" qui affiche la liste de toutes les communautés disponibles. Pour se connecter à un autre noyau, il suffit d'entrer dans le champ texte de l'agent "Communicator" le nom de la machine où se trouve le micro-noyau auquel on veut se connecter, puis entrer le numéro de port, comme indiqué sur la Figure IV.7.

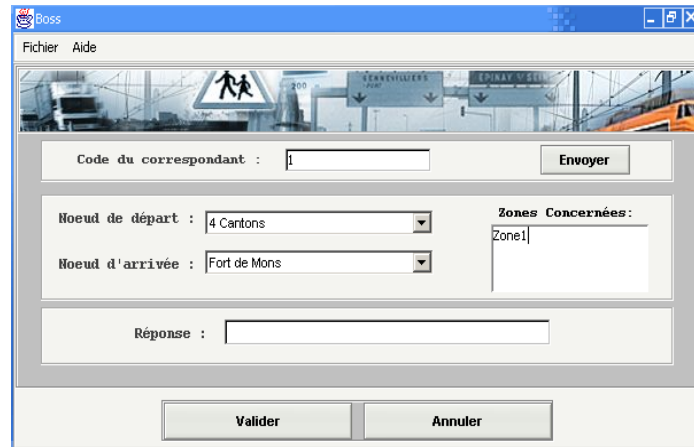


Figure IV.8: Interface de l'agent « SYSTEME ».

Pour que cela fonctionne, il faut que l'autre plate-forme à laquelle on veut se connecter ait aussi lancé l'agent "Communicator". Après avoir réalisé cela, les 2 micros-noyaux devraient être connectés et les agents de ces 2 plates-formes différentes peuvent maintenant communiquer de manière tout à fait transparente (pour le programmeur, c'est comme si les agents se trouvaient sur la même machine).

Une fois les agents implémentés sur différentes plates-formes MadKit communicantes, nous pouvons lancer maintenant nos agents. Le client de transport fait entrer dans les cases correspondantes de l'interface de l'agent SYSTEME son origine et sa destination (Figure IV.8). Par la suite, l'agent SYSTEME recherche, dans ses connaissances des nœuds des régions, les zones concernées par cette requête, qu'il visualise sur son interface. Ensuite, ce même agent vérifie s'il s'agit d'une seule zone concernée ou de plusieurs. Dans ce second cas, il cherche les points de correspondances entre les zones trouvées.

Une fois les zones et les intersections trouvées, l'agent SYSTEME lance des requêtes parallèles pour les agents RESPONSABLE ZONE des zones qu'il vient de noter. Chaque agent RESPONSABLE ZONE va effectuer son travail, afin d'avoir la portion de chemin

optimal dans sa région. Ceci est possible en lançant un calcul de chemin multicritère par l'agent CALCUL CHEMIN.

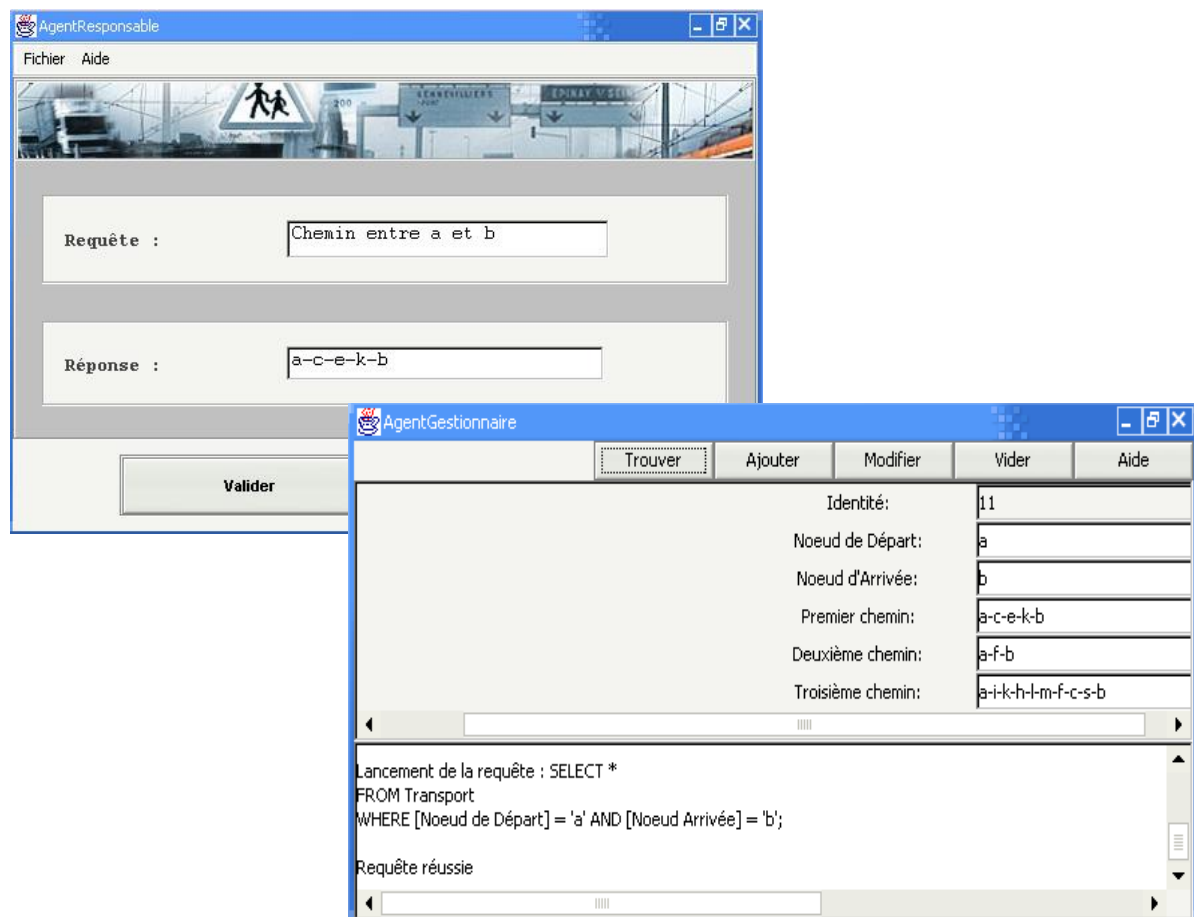


Figure IV.9: Exemple de communication entre deux agents.

Finalement, chaque RESPONSABLE ZONE communique le résultat trouvé à l'agent SYSTEME, qui va concaténer toutes les portions de chemin, pour aboutir au chemin global optimal qu'il affiche dans la case réponse (Figure IV.9).

Ainsi, le client a pu avoir un chemin multicritère, multimodal et optimal pour partir de l'origine qu'il a précisée au préalable et atteindre la destination souhaitée.

IV.4.2 Plate-forme multi-agent : JADE

La plate-forme JADE (*Java Agent DEvelopment Framework*) est un environnement de développement d'agents implanté totalement dans le langage JAVA. Il facilite la mise en place d'un système multi-agent répondant aux spécifications de FIPA (*Foundation for Intelligent Physical Agent*) à travers un ensemble d'outils.

La plate-forme JADE est fondée sur la spécification de FIPA et comprend tous les composants qui permettent la gestion de ladite plate-forme selon FIPA. Ces composants sont : le ACC (*Agent Communication Channel*), le AMS (*Agent Management System*) et le DF (*Directory Facilitator*). Les agents communiquent à travers les échanges de messages. Ces messages sont représentés dans le langage de communication FIPA-ACL.

La plate-forme JADE peut être répartie sur plusieurs hôtes. Une seule application JAVA et par conséquent une seule machine virtuelle Java est exécutée sur chaque hôte. Chaque machine virtuelle Java est un conteneur des agents fournissant un environnement complet d'exécution de ces agents et permettant à plusieurs agents concurrents de s'exécuter sur le même hôte (Figure IV.10).

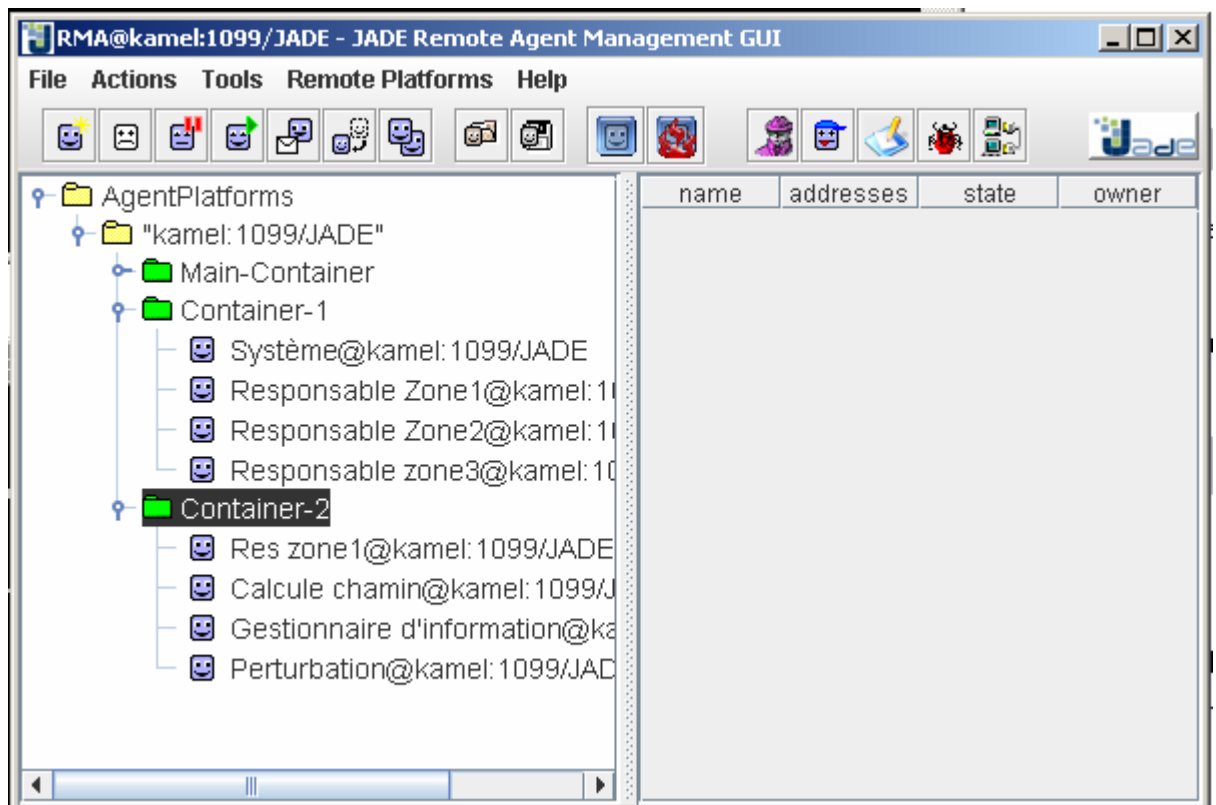


Figure IV.10: Exemple des conteneurs sur la même machine sous JADE.

En définitif, JADE est la plate-forme qui se rapproche le plus de nos critères. C'est une plate-forme de création d'agent qui prend en compte les spécifications de la FIPA pour l'interopérabilité des systèmes multi-agents. Les agents communiquent à travers des messages représentés en FIPA-ACL. Le concept d'agent est vu par JADE comme un processus

autonome et indépendant qui a une identité, qui requiert la communication (collaboration, compétition...) avec les autres agents, dans le but de remplir ses missions.

Nous avons testé avec succès la création de plusieurs « containers » sur une même machine et la communication entre différents agents se trouvant sur différents *containers*.

Ensuite, nous avons démarré des *containers* sur des hôtes distants, pour tester la communication entre agents se trouvant sur ces différents hôtes. Cette opération s'est déroulée avec succès.

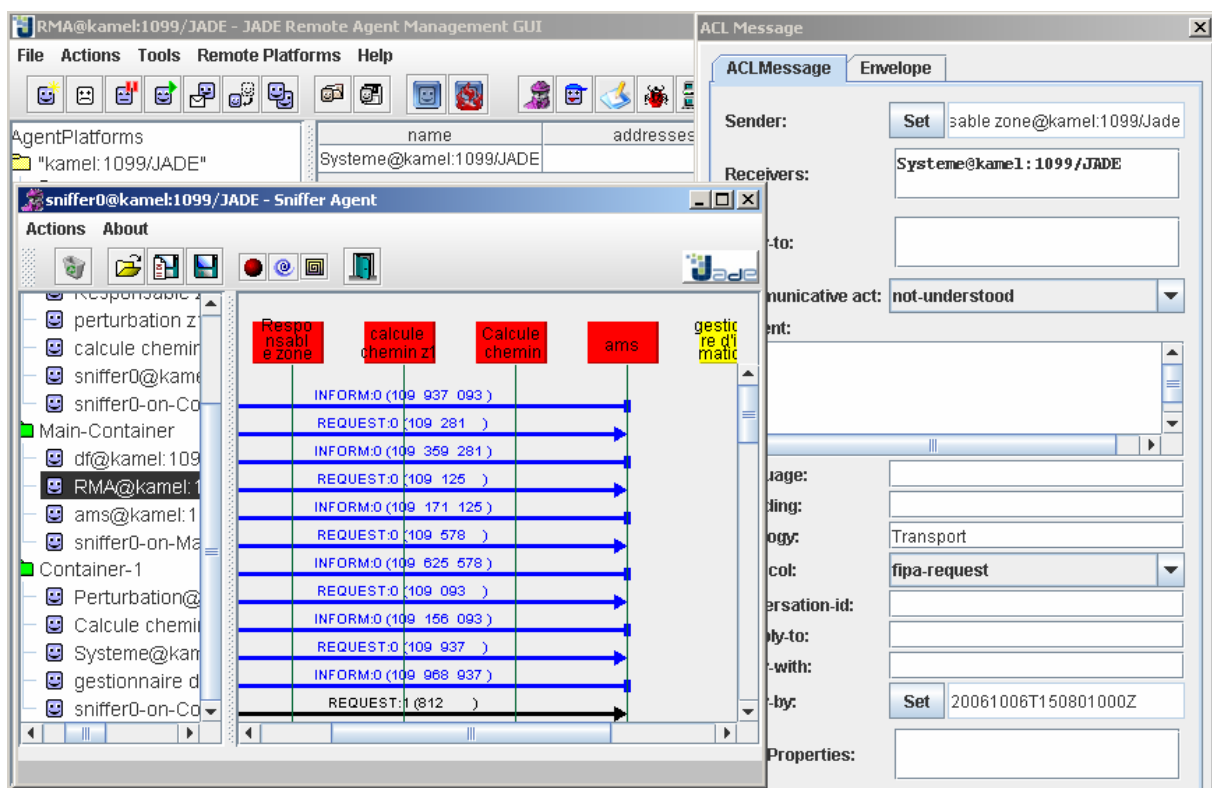


Figure IV.11: Exemple des communications et échanges de messages sous JADE.

On a vu qu'une propriété importante d'un agent est son autonomie : un agent ne doit pas se limiter à réagir aux événements externes, mais il doit être aussi capable de prendre l'initiative de nouveaux actes communicatifs d'une façon autonome.

Donc, le développeur d'agents doit étendre la classe Agent et implémenter les tâches spécifiques de l'agent par une ou plusieurs classes Comportement, les instancier et les ajouter à l'agent. La classe Agent représente une super-classe commune pour tous les agents définis par l'utilisateur. Du point de vue du programmeur, la conséquence est qu'un agent JADE est

simplement une classe Java qui étend la classe de base Agent. Cette relation permet à l'agent d'hériter un comportement fondamental caché (qui traite toutes les tâches liées à la plateforme, telles que l'enregistrement, la configuration, la gestion à distance, etc.), et un ensemble de méthodes qui peuvent être appelées pour implémenter les tâches spécifiques à l'agent, par exemple envoi des messages, utilisation des protocoles d'interaction standard, enregistrement sur plusieurs domaines, etc.

IV.5 Conclusion

Dans ce chapitre, nous avons présenté quelques résultats de simulation. Vu les difficultés d'accéder à des bases de données réelles, nous nous sommes restreints à des réseaux de petite et moyenne tailles. Ces réseaux font partie du réseau urbain de l'agglomération lilloise. Ces simulations ont montré l'intérêt des approches proposées dans l'optimisation des déplacements dans un réseau multimodal normal et de l'accompagnement dans les cas des perturbations. Pour le système multi-agent, nous avons testé les fonctionnements et les coordinations des agents en utilisant deux plates-formes différentes, MadKit et JADE. Il reste à vérifier le comportement du système global dans des cas de grandeur réelle, ainsi que l'intégration de ce travail avec d'autres applications en cours de réalisation dans notre équipe.

Conclusion et perspectives

Dans ce travail de recherche, nous avons traité le problème d'aide au déplacement dans des réseaux de transport multimodal. Ce problème préoccupe de plus en plus les compagnies de transport, car il a un impact direct sur la qualité du service offert aux usagers. En effet, en cas de perturbation, ces derniers souhaitent avoir de l'information fiable et des solutions alternatives, afin de terminer leurs voyages.

Dans l'objectif de favoriser la multimodalité et l'utilisation de transport en commun, notamment dans les grandes villes, plusieurs projets lancés par les acteurs de transport ou les pouvoirs publics visent à améliorer les services annexes au service de transport en commun. Les caractéristiques des systèmes d'information existants, en général, sont monomodaux ou n'assurent pas l'accompagnement pendant les déplacements. Après l'étude de ces systèmes et de leurs principales fonctionnalités, nous avons étudié la complexité du problème d'optimisation d'aide au déplacement, ainsi que les algorithmes nécessaires pour le résoudre.

Nous avons étudié les méthodes existantes afin de présenter notre approche. La modélisation du réseau de transport en commun par un multigraphe a montré la complexité du traitement de la multimodalité et l'aspect temporel du transport. Ce phénomène est dû au changement périodique des données et aux perturbations. Ces problèmes ont mis en évidence l'incapacité des algorithmes classiques de la théorie des graphes pour résoudre le problème d'aide au déplacement. Dans notre approche, on a proposé une hybridation entre l'algorithme de Dijkstra et les algorithmes évolutionnaires pour optimiser des déplacements selon plusieurs critères.

Cette approche utilise une méthode multicritère de recherche d'itinéraire, qui s'appuie sur une hybridation entre un algorithme de Dijkstra modifié et un algorithme génétique pour trouver une population de chemins optimaux. Cette approche nous a permis de trouver des solutions optimisées à présenter aux utilisateurs et d'avoir un ensemble de solutions à utiliser en cas des perturbations. En plus de l'optimisation multicritère des déplacements, ces méthodes nous permettent d'anticiper le traitement des perturbations.

Afin de garantir la multimodalité, nous étions confrontés à d'autres problèmes d'ordre organisationnel et technique, que nous avons rencontrés pendant la réalisation de notre système. Les problèmes organisationnels sont provoqués par l'implication de plusieurs exploitants ou organismes de transport. Les problèmes techniques sont dus à l'impossibilité

de modifier les systèmes existants et l'hétérogénéité des bases de données utilisées par les exploitants. Afin de simplifier la complexité de ces problèmes, nous avons procédé à la division du réseau de transport en plusieurs zones. Cette méthode a permis de modéliser ce réseau par une architecture multi-zones. Cette architecture nous a montré l'aspect distribué du système, les interactions et les relations qui peuvent avoir lieu entre les différentes zones. A partir de cette architecture, on a pu préserver l'indépendance et l'autonomie des exploitants, en même temps on a utilisé un système multi-agent qui traite les interactions et l'interopérabilité des systèmes existants. Nous avons présenté notre Système Multi-Agent d'Aide au Déplacement, SMAAD, pour un réseau de transport multimodal. Ce système est organisé sur plusieurs niveaux. Chaque niveau regroupe un ensemble d'agents qui sont autonomes pour l'aide au déplacement dans leur zone. Ils coopèrent et communiquent avec les autres groupes afin de compléter des tâches plus complexes. Notre architecture est « holonique » : chaque zone peut être aussi divisée en plusieurs zones et par la suite le groupe d'agents responsables d'une zone devient groupe central de ces zones.

Nous avons expliqué les particularités du SMAAD proposé en présentant les différents agents qui le composent ainsi que leurs architectures internes. Nous avons décrit aussi la nature de l'organisation du système considéré, afin de situer les interactions, communications et coordinations entre ces agents.

Des simulations présentées dans le dernier chapitre ont montré les intérêts de ces approches proposées pour améliorer le service d'aide au déplacement dans les réseaux multimodaux de transport. Il serait ainsi intéressant de pouvoir approfondir ce travail en développant d'autres aspects qui ont été envisagés. Différentes perspectives peuvent alors surgir dans ce sens. En effet, dans le cadre de l'approche évolutionnaire d'optimisation d'itinéraires, nous pouvons intégrer une recherche multicritère basée sur la méthode Pareto, afin de saisir les dominances des solutions par rapport aux trois critères considérés.

D'autre part, dans le calcul du troisième critère qui est le confort, nous avons simplifié les méthodes, en choisissant des valeurs discrètes bien déterminées. Nous pouvons améliorer cette partie en utilisant la logique floue par exemple.

En ce qui concerne les agents de notre système, nous avons d'autres perspectives. Le premier est l'intégration de notre système à d'autres systèmes multi-agents d'information multimodaux en cours de réalisation dans notre équipe. Avant et pendant l'implémentation,

nous avons pris en considération l'existence de tels systèmes, mais on n'a pas terminé la phase d'intégration entre les deux.

Dans nos perspectives globales, nous envisageons d'intégrer ce travail avec d'autres travaux, dans une plate-forme complète de transport. Cette plate-forme vise à regrouper trois systèmes dans le domaine de transport, en l'occurrence, l'aide au déplacement, l'information multimodale et la régulation des trafics.

Dans le même sens, au cours de la thèse, nous avons travaillé avec des collègues du LAMIH (Laboratoire d'Automatique, de Mécanique, et d'Informatique Industrielles et Humaines) de Valenciennes, afin de préparer un démonstrateur régional de transport, dans le cadre du projet VIATIC Mobilité du pôle de compétitivité I-Trans. Ces travaux ont influé sur le choix technique de développement, mais il reste à mettre en place les différentes parties qui composent ce démonstrateur.

Références bibliographique

- [Alaya 05] I. Alaya, C. Solnon et K. Ghédira "Algorithme fournis avec différentes stratégies phéromonales pour le sac à dos multidimensionnel" MHOSI'2005 24-26 Avril 2005, Hammamet, Tunisie.
- [Allenson 92] R. Allenson. Genetic algorithms with gender for multifunction optimisation. Technical raport EPCC-SS92-01, Edinburg Parallel computing center, Edinburg, Scotland, 1992.
- [Aloulou 02] M. A. Aloulou, "Structure flexible d'ordonnancements à performances contrôlées pour le pilotage d'atelier en présence de perturbations", thèse en Informatique à l'Institut Nationale Polytechnique de Lorraine, Soutenue le 16 décembre 2002.
- [Babin 97] G. Babin, Z. Maamar, and B. Chaib-draa. Metadatabase meets distributed AI. In *First Int. Workshop on Cooperative Information Agents (CIA-97)*, 1997.
- [Basseur 05] M. Basseur, conception d'algorithmes coopératifs pour l'optimisation Multiobjectif : application aux problèmes d'ordonnement de type flow-shop. Mémoire de thèse. USTL 21/06/2005
- [Ben Khaled 05] I. Ben Khaled, M. A. Kamoun, K. Zidi, S. Hammadi. Vers un système d'information voyageur multimodal (SIM) à base de système multi-agent (SMA). REE N° 1 Janvier 2005 pp: 41-47.
- [Berger 02] T. Berger, Y. Sallez, D. Trentesaux, C. Tahon. Two Heterarchical Multiagent Approaches for FMS Dynamic Scheduling. Taylor & Francis Volume 42, Number 5 / 2002 pages : 757 - 768
- [Berro 04] A. Berro, O. Heguy. Agents évolutionnaires pour l'optimisation multiobjectif (*Evolutionary agents for multiobjective optimization*). RFIA 28-30 janvier 2004 Toulouse.
- [Bodin 90] L. D. Bodin. Twenty years of routing and scheduling. Operations Research, 38:571-579,1990
- [Bounsaythip 98] C. K. Bounsaythip, "Algorithmes heuristiques et évolutionnaires : application à la résolution du problème de placement de formes irrégulières", Thèse de doctorat en Productique, université des sciences et technologies de Lille, 9 octobre 1998.

- [Bridges 91] C. L. Bridges, D. E. Goldberg – An analysis of multipoint crossover, In : Proceedings of the Foundation Of Genetic Algorithms. FOGA. – 1991.
- [Bulletin 05] Bulletin statistique du SES Transport. Ministère des transports, de l'équipement, du tourisme et de la mer (direction des affaires économiques et internationales). Juin 2005.
- [Cammarata 83] S. Cammarata, D. McArthur, and R. Steeb. Strategies of cooperation in distributed problem solving. In Proceedings of the Eighth International Joint Conference on Artificial Intelligence (IJCAI-83), Karlsruhe, Germany, 1983.
- [Carraway 90] R. L. Carraway, T. L. Morin and H. Moskowitz. Generalized dynamic programming for multicriteria optimization. European Journal of Operational Research, 44:95-104, 1990.
- [Caux 94] C. Caux, H. Pierreval, M. C. Portmann, “Les algorithmes génétiques et leurs applications aux problèmes d'ordonnancement”, Actes des journées d'étude : Ordonnancement et entreprise, applications concrètes et outils pour le futur, 16 et 17 juin 1994, Toulouse, France.
- [Chaib-draa 94] B. Chaib-draa. Distributed Artificial Intelligence : An overview. In A. Ken, J. G. Williams, C. M. Hall, and R. Kent, editors, Encyclopedia Of Computer Science And Technology, volume 31, pages 215-243. Marcel Dekker, Inc, 1994.
- [Chaib-draa 96] B. Chaib-draa. Interaction between agents in routine, familiar and unfamiliar situations. International Journal of Intelligent and Cooperative Information Systems, 1(5):7-20, 1996.
- [Ciesielski 98] V. Ciesielski and P. Scerri, “Real time genetic scheduling of aircraft landing times”, IEEE International Conference on Evolutionary Computation, ICEC98, in D. Fogel (Editor), pp.360-364, Anchorage, May 1998.
- [Collette 02] Y. Collette et P. Siarry. Optimisation multiobjectif. Eyrolles, 2002.
- [Corkill 83] D. D. Corkill and V. R. Lesser. The use of meta-level control for coordination in a distributed problem solving network. In Proceedings of the Eighth International Joint Conference on Artificial Intelligence, pages 748-756, Karlsruhe, Federal Republic of Germany, August 1983.
- [Danflous 00] D. Danflous. Déploiement national des systèmes d'information multimodale. Centre de documentation du CERTU. CE06 5749 Aout 2000
- [Danflous 01] D. Danflous, “Déploiement national des systèmes d'information multimodale”, GOFAS : l'exemple suisse”, Rapport d'étude, CERTU/CETE Méditerranée (Centre des études techniques et de l'équipement), Octobre 2001.

- [Dawid 99] H. Dawid Adaptive learning by Genetic Algorithm. Analytical results and applications to economic models, Springer, Berlin. P39 1999
- [Deb 00] K. Deb A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multiobjective Optimization : NSGA II. Parallel problem Solving form Nature – PPSN VI, Springer Lecture Notes in Computer Science (2000) 849-858.
- [Deb 98] K. Deb and P. Chakroorty, “Time scheduling of transit systems with transfer considerations using genetic algorithms”, *Evolutionary Computation* 6(1), pp.1-24, 1998.
- [Delahaye 96] D. Delahaye, N. Durand, J. M.Alliot and M. Schoenauer, “Genetic algorithms for air traffic control systems”, ENAC, INFAUTOM'96, Toulouse, France, 14-15 mars 1996.
- [Deneubourg 89] J.L.Deneubourg, S.Goss, "Collective patterns and decision-making," *Ethology, Ecology & Evolution*, Vol.1, 295–311, 1989.
- [Dréo 03] J. Dréo, A. Pérowski, P. Siarry, E. Taillard. Métaheuristiques pour l’optimisation difficile. Edition Eyrolles pages 69-112, 2003.
- [Durand 96] D. Nicolas, – Optimisation de trajectoires pour la résolution de conflits en route. – Thèse de Doctorat, INPT, 1996.
- [Durfee 91] E. H. Durfee and V. R. Lesser, “Partial global planning: A coordination framework for distributed hypothesis formation”, *IEEE Transactions on Systems, Man, and Cybernetics*, 21(5), pp.1167-1183, September-October 1991.
- [El Hmam 05] M. Said El Hmam, D. Jolly, H. Abouaissa, A. Benasser. Modélisation Hybride du Flux de Trafic. Workshop avec école intégrée Méthodologies et Heuristiques pour l’Optimisation des Systèmes Industriels 24-26 Avril 2005, Hammamet, Tunisie. Pages193-198
- [Ennigrou 04] M. Ennigrou, K. Ghédira, Approche Multi-Agents basée sur la Recherche Tabou pour le Job Shop flexible, 14ème Congrès Francophone AFRIF-AFIA de Reconnaissance des Formes et Intelligence Artificielle RFIA, Toulouse, France, 28-30 Janvier 2004.
- [Fagin 95] R. Fagin, I. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning about Knowledge*. The MIT Press : Cambridge, MA, 1995.
- [Fayech 02] B. Fayech, S. Maouche, S. Hammadi and P. Borne, “Multi-Agent Decision-Support System for an Urban Transportation Network”, *World Automation Congress, CDROM Proceedings ISIAC021*, 9-13 June, Florida, USA, 2002.

- [Fayech 01] B. Fayech Chaar. Urban Bus Traffic Regulation By Evolutionary Algorithms. Proceedings of the 2001 IEEE Systems, Man, and Cybernetics conference.
- [Ferber 95] J. Ferber. Les systèmes multi-agents, vers une intelligence collective. InterEditions, 1995.
- [FIPA 97] Foundation for Intelligent Physical Agents, Specification Version 2, Part1-Part5, 1997, <http://drogo.cselt.stet.it/fipa/spec>.
- [FIPA 99] FIPA, FIPA 99 Specifications part 2 : Agent Communication Language, rev. 0.1, avril 1999. URL: <http://www.fipa.org/spec/FIPA99.html>
- [Fischer 95] K. Fischer, J. P. Müller, M. Pischel and D. Schier, “A Model for Cooperative Transportation Scheduling”, In Proceedings of the First International Conference on MAS, AAAI Press/MIT Press, pp.109-116, Menlo Park, California,1995.
- [Fonseca 93] C. M. Fonseca, P. J. Fleming: Genetic Algorithm for Multiobjective Optimization: Formulation, Discussion and Generalization. In Proceedings of the Fifth International Conference on Genetic Algorithms, San Mateo, California, (1993) 416-423.
- [Garrett 99] C. A. Garrett, J. Huang, M. N. Goltz, and G. B. Lamont. Parallel real-valued genetic algorithms for bioremediation of TCE-Contaminated groundwater. In Congress on Evolutionary Computation (CEC'99), pages 2183-2189. Washington, D.C. USA, 1999. IEEE service center
- [Goldberg 89] D. E. Goldberg, “Genetic algorithms in search, optimisation and machine learning”, Addison-Wesley, 1989.
- [Goldberg 91] D.E. Goldberg – Real-coded genetic algorithms, virtual alphabets and blocking. Complex Systems, vol. 5, 1991, pp. 139–167.
- [Golden 86] B. L. Golden and A. A. Assad. Perspectives on vehicle routing: Exciting new developments Operations Research, 34:803-810,1986.
- [Green 97] S. Green, L. Hurst, B. Nangle, P. Cunningham, F. Somers, R. Evans, “Software agents: A review”, TCS-CS-1997-06, Trinity College Dublin, Broadcom Eireann Research Ltd., 27 may 1997.
- [Gruer 01] P. Gruer, V. Hilaire and A. Koukam, Multi-Agent Approach to Modelling and Simulation of Urban Transportation Systems, Proceedings of the 2001 IEEE SMC Conference, 6-10 October 2001, Tucson, Arizona, USA, pp.2499-2504.
- [Hao 99] J.-K. Hao, P. Galinier, M. Habib. Méthaheuristiques pour l’optimisation combinatoire et l’affectation sous contraintes. Revue d’intelligence artificielle Vol : No.1999 <http://www.info.univ-angers.fr/pub/hao/papers/RIA.pdf>

- [Holland 92] J. Holland, “Adaptation in natural and artificial systems”, Second Edition, MIT Press, 1992.
- [Horn 93] J. Horn, N. Nafpliotis: Multiobjective Optimisation using the Niche Pareto Genetic Algorithm. Illigal TR. n° 93005 (1993)
- [Huissman 01] D. Huisman, R. Freling, A. P. M. Wagelmans, “A Dynamic Approach to Vehicle Scheduling”, Econometric Institute, Erasmus University Rotterdam, Netherlands, Report EI2001-17 (accepted for Transportation Science (focussed issue on Real-time Fleet Management) on August 22, 2002 with a new title "A Robust Solution Approach to the Dynamic Vehicle Scheduling Problem").
- [Jennings 00] N. R. Jennings. Agent methodology for software engineering. *Communication of ACM*, (to appear), 2000.
- [Jouvin 00] D. Jouvin. Modèle de délégation et de composition d’agents. Conférence OCM Nantes 18 mai 2000.
- [Kabachi 96] N. Kabachi, L. Vincent, Y. Ouzrout, “Une société d’agents pour la prise de décision dans les organisations productives”, Actes des journées Francophones d’IA Distribuée et SMA, pp.35-46, 1996.
- [Kaplan 98] F. Kaplan. A new approach to class formation in multi-agent simulation of language evolution. In *Proceedings of the Second International Conference on Multi-Agent Systems (ICMAS-2)*, pages 158-165, Paris, Fr, 1998.
- [Kobsa 89] A. Kobsa. *User Models in Dialog Systems*. Springer-Verlag : Heidelberg, Germany, 1989.
- [Küchenhoff 93] V. Küchenhoff, Novel Search and Constraints – an Integration, Technical Report ECRC–CORE–93–9, European Computer-Industry Research Centre, 1993.
- [Labbé 97] M. Labbé, Ph. Toint. Recherche opérationnelle et transports. Revu “Nouvelles de la Science et des Technologies” 20 novembre 1997.
- [Labrou 97] Y. Labrou and T. Finin, A Proposal for a new KQML Specification, rapport technique TR CS-97-03, C.S. and E.E. dep., uni. of Maryland Baltimore County, février 1997.
- [Lacomme 03] P. Lacomme, C. Prins, M. Sevaux. Algorithmes de graphes. Eyrolles 2^e édition 2003. ISBN 2-212-11385-4.
- [Lacomme 05] P. Lacomme, méthodes exactes et approchées pour l’optimisation des systèmes à moyens de transport. Mémoire pour l’obtention de l’Habilitation à

Diriger des Recherches de l'Université Blaise Pascal (Clermont –Ferrand II).06/07/2005 : NO :194 ;

- [Lacomme_b 03] P. Lacomme, C. Prins, A. Tanguy (2003). "Optimisation par colonies de fourmis pour les tournées sur arcs". *4e Conférence Francophone de MOdélisation et SIMulation MOSIM03*, Toulouse (France).
- [Laïchour 01] H. Laïchour, S. Maouche and R. Mandiau, Traffic Control Assistance in Connection nodes, Proceedings of the 2001 IEEE SMC Conference, 6-10 October 2001, Tucson, Arizona, USA.
- [Lakemeyer 99] G. Lakemeyer and H. Levesque. Query evaluation and progression in AOL knowledge bases. In Proceedings of the Sixteenth International Joint Conference on *Artificial Intelligence (IJCAI-99)*, pages 124-131, Stockholm, Sweden, 1999.
- [Laporte 95] G. Laporte, I. H. Osman. Routing problems: A bibliography. *Annals of Operations Research*, 61:227-262,1995.
- [Lecomte 00] N. Lecomte, R. Patesson. Le panel des voyageurs : Une étude des activités et des besoins d'information des utilisateurs des transports publics. In Actes de la conférence ERGO-IHM, D.L.Scapin et E.Vergison(eds) . pp. 129-135, Biarritz, France, 3-6 oct 2000.
- [Lévy 94] G. Lévy, Algorithmique combinatoire méthodes constructives. DUNOD, Paris 1994, ISBN 2100021494
- [Lind 98] J. Lind and K. Fischer, "Transportation scheduling and Simulation in a railroad scenario: a multi-agent approach", German Research Center for AI (DFKI), Research Report, DFKI-TM-98-05, December 1998.
- [Linden 96] G. Linden, S. Hanks, N. Lesh, "Interactive Assessment of User Preference Models: The Automated Travel Assistant", 1996.
- [Ljungberg 92] M. Ljungberg and A. Lucas, "The oasis air-traffic management system", In Proceedings of the Second Pacific Rim International Conference on Artificial Intelligence, PRICAI '92, Seoul, Korea, 1992.
- [Marco 99] N. Marco, J.-A. Désidéri , S. Lanteri. Multi-Objective Optimization in CFD by Genetic Algorithms. Rapport de recherche n° 3686 INRIA (SOPHIA ANTIPOLIS) Avril 1999.
- [Mesghouni 99] K. Mesghouni, "Application des algorithmes évolutionnistes dans les problèmes d'optimisation en ordonnancement de la production", thèse de doctorat en Automatique et Informatique Industrielle, université des sciences et technologies de Lille, 5 janvier 1999.

- [Meskine 01] A. Meskine, P. Gendre. Algorithmes et calculs d'optimisation d'itinéraires pour l'information multimodale ; implémentation d'un prototype pour les transports collectifs avec horaires. Rapport d'étude, CERTU/EMSE (Ecole des Mines de Saint-Étienne et Centre d'études sur les réseaux, les transports, l'urbanisme et les constructions urbaines). Novembre 2001.
<http://www.predim.org>.
- [Michalewicz 91] Z. Michalewicz, C. Z. Janikov. Handling constraints in genetic algorithms, In: Proceedings of the Fourth International Conference on Genetic Algorithm. ICGA. – 1991.
- [Michalewicz 94] Z. Michalewicz. (1994), “Evolutionary computation techniques for nonlinear programming problems”, International Transactions in Operational Research 1(2), 223–240.
- [Monmarché 00] N. Monmarché, 2000. Algorithme de fourmis artificielles : application à la classification et à l'optimisation, Thèse à l'université de François Rabelais Tour.
- [Ngamchai 00] S. Ngamchai and D. J. Lovell, “Optimal time transfer in bus transit route network design using genetic algorithms”, 8th international Conference CASPT, Computer Aided Scheduling of Public Transportation, Berlin, Germany, 21-23 June 2000.
- [Nierstrasz 00] O. Nierstrasz, J.-G. Schneider, F. Achemann, “Agents Everywhere, All the Time”, ECOOP 2000 Workshop on Component-Oriented Programming, 2000.
- [Nwana 99] H. S. Nwana and D. T. Ndumu. Agents of change in future communication systems. *Applied AI Journal*, 13(1), 1999.
- [Ould Sidi_a 05] M. M. Ould Sidi, S. Hammadi, S. Hayet, P. Borne. Urban transport disrupted networks regulation strategies making and evaluation: A new approach. IESM 2005, Marrakech (2005).
- [Ould Sidi_b 05] M. M. Ould Sidi, S. Hammadi, S. Hayet. AEMC pour l'aide à la régulation du trafic d'un réseau de transport collectif. MHOSI 2005, Hammamet (2005).
- [Pierreval 03] H. Pierreval, C. Caux, J. L. Paris and F. Viguier, “Evolutionary approaches to the design and organization of manufacturing systems”, Computers and Industrial Engineering, Vol.44, pp.339-364, 2003.
- [Potvin 96] J. Y. Potvin, “Genetic algorithms for the travelling salesman problem”, Annals of Operation Research, Netherlands, 63, pp.339-370, 1996.

- [Rousseau 85] J. M. Rousseau and J. Y. Blais, “HASTUS: An interactive systems for buses and crew scheduling”, *Computer Scheduling of Public Transport-2*, J.M.Rousseau ed., pp.45-60, Amsterdam, North Holland, 1985.
- [Salim 97] V. Salim and X. Cai, “A genetic algorithm for railway scheduling with environmental considerations”, *Environmental Modelling and Software*, Vol.12, No.4, pp.301-309, 1997.
- [Saussol 00] B. Saussol, S. Maouche, S. Hayat, A. Dekokere, A. Dumont, “Elaboration et mise au point d’un système d’aide à la décision pour la gestion du réseau de transport collectif de Montbéliard”, *Rapport d’étape INRETS-I3D-LAIL, Appui à la modélisation du système multi-agents*, juin 2000.
- [Schwefel 97] H. P. Schwefel and T. Bäck, “Artificial Intelligence: How and why?”, *Genetic Algorithms and evolution strategies in engineering and computer science*, Edited by D. Quagliarella & al, J.Wiley Editions, 1997.
- [Schzfer 85] D. Schzfer. Multiple objective optimization with vector evaluated genetic algorithm. *Proceedings of First international conference on genetic algorithms*, page 93-100. 1985
- [Sedgewick 86] R. Sedgewick, J.S Vitter, Shortest paths in euclidean graphs, *Algorithmic* 1, 31-48, 1986
- [Siarry 02] P. Siarry, Y. Collette. *Optimisation multiobjectif*. Eyrolles ISBN: 2-212-11168-1. 2002.
- [Solnon 05] C. Solnon. Mémoire pour l’obtention de l’Habilitation à Diriger des Recherches de l’Université Claude Bernard Lyon 1. Contributions à la résolution pratique de problèmes combinatoires des fourmis et des graphes. 6/12/2005.
- [Srivinas 93] N. Srivinas, K. Deb, : Multiobjective Optimization using Nondominated Sorting in Genetic Algorithms. Technical Report, Departement of Mechanical Engineering, Institute of Technology, India, (1993).
- [Stevens 95] G. Stevens, “An approach to scheduling aircraft landing times using genetic algorithms”, Honours thesis, RMIT, Department of Computer Science, November 1995.
- [Stewart 91] B. S. Stewart et C. C. White. Multiobjective A*. *Journal of the ACM*, 38(4) : 775-814, 1991.
- [Talbi 00] E-G. Talbi, “Contribution à la résolution parallèle de problèmes d’optimisation combinatoire”, *Rapport de H DR en Informatique à l’université de Lille 1*, 2000.

- [Uster 01] G. Uster. Information Multimodale voyageurs aspects Institutionnels et Juridiques. [Centre de documentation de l'INRETS, Lille - Villeneuve d'Ascq](#) code : LI-UTG0078. 2001.
- [Vallée 03] T. Vallée, M. Yildizoglu. Présentation des algorithmes génétiques et de leurs applications en économie. Décembre 2003, v. 4.2
- [Viet 04] T. P. Viet, L. Magnin, H. Sahraoui, Adaptation dynamique des systèmes multi-agents basée sur le concept de méta-CATN Intl. Conf. RIVF'04 February 2-5, Hanoi, Vietnam
- [Visée 98] M. Visée, J. Teghem, M. Pirlot, and E. L. Ulungu. Two-phases method and branch and bound procedures to solve Knapsack problem. *Journal of Global Optimization*, 12: 139-155, 1998.
- [Zhu 95] K. Q. Zhu, "A New Genetic Algorithm for VRPTW", *Journal of Combinatorial Optimization*, April 2000.
- [Zidi 04] K. Zidi, S. Hammadi, CGOMFP Control Genetic Operators with Management of the Final Population to optimize a multimodal transport moving. IEEE SMC 2004 the Hague- de 10 au 13 October 2004.
- [Zidi 05] K. Zidi, S. Hammadi. DMAS: Distributed Multi-Agents System for assist users in the multimodal travels. International Conference on Industrial Engineering and Systems Management. IESM 2005, May 16 – 19, Marrakech (Morocco)
- [Zidi 06] S. Zidi, K. Zidi, S. Maouche. Approches métaheuristiques pour des problèmes de transport multimodal. Workshop avec école intégrée Logistique et Transport (LT2006) 30 Avril au 02 Mai 2006, Hammamet, Tunisie
- [Zidi_b 04] K. Zidi, S. Hammadi, P. Borne. Méthode évolutionniste pour l'aide au déplacement dans le transport multimodal perturbé. 5e Conférence Francophone de MOdélisation et SIMulation "Modélisation et simulation pour l'analyse et l'optimisation des systèmes industriels et logistiques" MOSIM'04 – du 1er au 3 septembre 2004 - Nantes (France).
- [Zidi_b 05] K. Zidi, S. Hammadi. Algorithme génétique avec contrôle des opérateurs pour l'optimisation multicritère d'un déplacement dans un réseau de transport multimodal. Workshop avec école intégrée Méthodologies et Heuristiques pour l'Optimisation des Systèmes Industriels 24-26 Avril 2005, Hammamet, Tunisie Sélectionné et publié dans la Revu électronique e-STA volume 2 quatrième trimestre 2005.