



HAL
open science

Conception et réalisation d'un système de gestion de véhicules partagés : de la multimodalité vers la co-modalité

Karama Jeribi

► **To cite this version:**

Karama Jeribi. Conception et réalisation d'un système de gestion de véhicules partagés : de la multimodalité vers la co-modalité. Autre. Ecole Centrale de Lille, 2012. Français. NNT : 2012ECLI0033 . tel-00794506

HAL Id: tel-00794506

<https://theses.hal.science/tel-00794506>

Submitted on 26 Feb 2013

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

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

N° d'ordre : 211

ECOLE CENTRALE DE LILLE

THESE

Présentée en vue d'obtenir le grade de

DOCTEUR

En

Spécialité : Automatique, Génie Informatique, Traitement du Signal et Image

Par

Karama Jeribi

DOCTORAT DELIVRE PAR L'ECOLE CENTRALE DE LILLE

Titre de la thèse :

Conception et réalisation d'un système de gestion de véhicules partagés : de la multimodalité
vers la co-modalité

Soutenue le 12 Décembre 2012 devant le jury d'examen :

| | |
|---------------------------|---|
| Président | <i>Emmanuel, Duflos, Professeur, Ecole Centrale de Lille</i> |
| Rapporteur | <i>Alain, Quilliot, Professeur, Université Blaise Pascal de Clermont-Ferrand</i> |
| Rapporteur | <i>Aziz, Moukrim, Professeur, Université de Technologie de Compiègne</i> |
| Membre | <i>Serge, Boverie, HDR, Continental Engineering Services Toulouse</i> |
| Membre | <i>Jean Marc, Blosseville, Directeur du Laboratoire LEMCO, IFSTAR Versailles-Satory</i> |
| Directeur de thèse | <i>Slim, Hammadi, Professeur, Ecole Centrale de Lille</i> |
| Co-encadreur | <i>Hayfa, Zgaya, MCU, Université Lille 2</i> |

Thèse préparée dans le Laboratoire LAGIS UMR CNRS 8219 à l'école Centrale de Lille

Ecole Doctorale SPI 072 (EC Lille)

PRES Université Lille Nord-de-France

A mon papa, Hamadi, pour son amour, son affection et son soutien qui ne me quittent jamais

A ma maman, Afifa, pour tout ce qu'elle est pour moi, ma mère, ma sœur, mon amie, ma vie

A mon frère, Hassen, pour son amour, sa confiance et le grand frère qu'il a toujours été pour moi

A mon futur époux, Riadh, pour son amour, sa patience et le bonheur qu'il me fait vivre

A ma nièce, Emna et mon neveu Mohamed, pour tout le bonheur qu'ils ont apporté à ma vie

A ma très chère, Ayda, pour son amitié et sa présence avec moi

A Nouha, pour tous les bons moments qu'on a passés ensemble

A toute ma famille

A tous mes amis et amies

*Je vous dédie ce travail, sans vous ma vie ne serait pas ce qu'elle est aujourd'hui. Je vous aime tous
infiniment.*

Remerciements

Je tiens à exprimer en premier lieu ma gratitude à mon directeur de thèse Monsieur Slim HAMMADI, professeur à l'École Centrale de Lille. Je le remercie pour son encadrement scrupuleux, sa confiance, son encouragement tout au long de cette thèse et surtout de m'avoir fait découvrir réellement le monde de la recherche.

Je remercie tout aussi chaleureusement Mademoiselle Hayfa ZGAYA, maître de conférences à la faculté d'Ingénierie et Management de la Santé à l'Université de Lille 2 et Co-encadrante de cette thèse, pour sa présence, son encouragement, sa perspicacité et son énorme soutien scientifique.

J'adresse aussi mes vifs remerciements au Professeur Aziz MOUKRIM, Professeur à l'Université de Technologie de Compiègne et au Professeur Alain QUILLIOT, Professeur à l'Université Blaise Pascal de Clermont-Ferrand, qui m'ont fait le grand honneur d'accepter de rapporter cette thèse. Je les remercie infiniment pour le temps consacré à cet effet en dépit de toutes les responsabilités qu'ils ont.

Je souhaite remercier tout particulièrement Messieurs Emmanuel Duflos, Professeur à l'École Centrale de Lille, Jean Marc BLOSSEVILLE, directeur du laboratoire LEMCO à IFSTTAR Versailles Satory et Serge BOVERIE, HDR et responsable au Continental Engineering Services à Toulouse, pour avoir évalué ce travail et accepté de participer au jury de cette thèse.

Je tiens aussi à remercier l'ensemble du personnel LAGIS et tous les membres de l'équipe OSL en particulier mes collègues de bureau wided et safa, témoignant de la bonne ambiance dans laquelle j'ai évolué durant ces trois années.

Tous mes remerciements vont à ma très chère Ayda qui a été présente pour moi dès le premier jour et à qui je dois tout. Je la remercie pour son soutien, son amour et sa douceur.

Je tiens tout aussi à remercier mon amie Nouha pour tous les bons moments qu'on a passés ensemble.

Enfin, je souhaite exprimer ma profonde reconnaissance et tout mon amour pour toute ma famille et ma belle-famille. Je les remercie pour tout l'amour, la présence, la confiance dont ils ont fait preuve. Je leur dédie ce travail qui n'aurait jamais vu le jour sans leurs soutien et encouragements.

Table des matières

Contenu

| | |
|--|----|
| Table des matières | 7 |
| Index des figures | 14 |
| Index des Tableaux..... | 16 |
| Index des Tables..... | 17 |
| Glossaire..... | 19 |
| Introduction générale..... | 23 |
| I. Chapitre I Transport : de la multimodalité vers la co-modalité | 25 |
| I.1. Introduction | 25 |
| I.2. Les transports : Leur évolution et leur impact sur l'environnement..... | 25 |
| I.3. Les transports et la mobilité durable..... | 28 |
| I.4. Transport : De la multimodalité vers la co-modalité..... | 29 |
| I.4.1. Le transport monomodal | 29 |
| I.4.2. Le transport intermodal | 29 |
| I.4.3. Le transport multimodal | 29 |
| I.4.4. Le transport co-modal | 32 |
| I.5. La notion de véhicule partagé..... | 33 |
| I.5.1. Le Covoiturage | 34 |
| I.5.1.1. Définitions..... | 34 |
| I.5.1.2. Types de trajets de covoiturage..... | 36 |
| I.5.1.3. Formes du covoiturage..... | 37 |
| I.5.1.4. Covoiturage : Systèmes existants..... | 37 |
| I.5.2. L'autopartage | 40 |

| | |
|--|----|
| Table des matières | |
| I.5.2.1. Fonctionnement de l'autopartage..... | 40 |
| I.5.2.2. Autopartage : Historique et systèmes existants | 41 |
| I.6. Les systèmes d'Information du transport co-modal | 43 |
| I.6.1. Définition d'un système d'information | 43 |
| I.6.2. Systèmes d'information monomodaux..... | 43 |
| I.6.3. Systèmes d'information multimodaux | 44 |
| I.6.4. Systèmes d'information co-modaux..... | 45 |
| I.7. Limites des systèmes existants et proposition d'un système de transport co-modal..... | 48 |
| I.8. Conclusion | 50 |
| II. Chapitre II : Alliance Systèmes multi-agents et optimisation au service du transport co-modal..... | 53 |
| II.1. Introduction | 53 |
| II.2. Quelle architecture et quelle approche de résolution choisir ?..... | 53 |
| II.3. Les Systèmes Multi-Agents..... | 57 |
| II.3.1. Définition d'un agent | 58 |
| II.3.2. Propriétés d'un agent..... | 59 |
| II.3.3. Définition d'un système multi-agents | 60 |
| II.3.3.1. Environnement dans un système multi-agents..... | 60 |
| II.3.3.2. La communication et l'interaction dans un système multi-agent | 61 |
| II.3.3.3. Organisation Multi-Agents (OMA) | 62 |
| II.3.4. Les SMA dans le domaine du transport | 65 |
| II.4. Optimisation | 67 |
| II.4.1. Définition d'un problème d'optimisation..... | 67 |
| II.4.2. Complexité des problèmes d'optimisation..... | 68 |
| II.4.3. Optimisation combinatoire | 69 |
| II.4.4. Optimisation multi-objectif | 70 |
| II.4.4.1. Les méthodes scalaires..... | 73 |

| | |
|--------------------|--|
| Table des matières | |
| II.4.4.2. | Les méthodes interactives 73 |
| II.4.4.3. | Les méthodes floues..... 73 |
| II.4.4.4. | Les méthodes d'aide à la décision 73 |
| II.4.4.5. | Les méthodes exploitant une métaheuristique 74 |
| II.4.5. | Les métaheuristiques 74 |
| II.4.5.1. | L'approche de recherche locale 74 |
| II.4.5.2. | L'approche constructive..... 77 |
| 2.1.1.1. | L'approche évolutive 77 |
| II.4.6. | L'optimisation dans le domaine du transport..... 82 |
| II.5. | L'alliance entre les Systèmes multi-agents et l'optimisation 83 |
| II.6. | Conclusion 84 |
| III. | Chapitre III : Architecture à base d'agents communicants intégrant des graphes distribués pour la combinaison des services de transport 87 |
| III.1. | Introduction..... 87 |
| III.2. | Formulation du problème..... 88 |
| III.3. | Complexité du problème..... 91 |
| III.4. | Architecture multi-agents proposée 93 |
| III.4.1. | Comportement des agents dans le système..... 95 |
| III.4.1.1. | Agent Interface (IA) 95 |
| III.4.1.2. | Super Agent (<i>SupA</i>) 96 |
| III.4.1.3. | Agent Service de Transport (TSA)..... 99 |
| III.4.1.4. | Agent Information Transport (TIAgent)..... 100 |
| III.4.1.5. | Agent Route (RA)..... 101 |
| III.4.1.6. | Agent Evalueur (EA) 101 |
| III.5. | Algorithmes de recherche et de composition d'itinéraires 103 |
| III.6. | Sélection du domaine de recherche..... 103 |
| III.7. | Identifications des plus courts chemins 106 |

Table des matières

| | | |
|------------|---|-----|
| III.7.1. | Définition des graphes | 106 |
| III.7.1.1. | Généralités | 106 |
| III.7.1.2. | Graphe co-modal | 107 |
| III.7.1.3. | Graphe de transfert | 109 |
| III.7.2. | Calcul du plus court chemin dans chaque composant <i>Ci</i> | 110 |
| III.7.2.1. | Algorithmes de plus court chemin dans le domaine du transport..... | 111 |
| III.7.2.2. | Les algorithmes de correction d'étiquettes (label Correction Algorithms) ... | 112 |
| III.7.2.3. | Les algorithmes de fixation d'étiquettes (label Setting Algorithms)..... | 112 |
| III.7.3. | Algorithme du plus court chemin sur un graphe dynamique et distribué..... | 114 |
| III.7.3.1. | Modélisation du problème | 114 |
| III.7.3.2. | Réseaux dynamiques FIFO et non FIFO (First In First Out) | 114 |
| III.7.3.3. | Modélisation du graphe dynamique distribué | 116 |
| III.7.3.4. | Algorithme du plus court chemin distribué dans un composant <i>Ci</i> | 118 |
| III.7.3.5. | Le graphe d'intersection complet <i>Gcint</i> | 119 |
| III.7.3.6. | Le graphe d'intersection étendu <i>Gveint</i> | 120 |
| III.7.4. | Graphe de Transfert de plus court chemin (SPTG) | 122 |
| III.8. | Conclusion | 123 |
| IV. | Chapitre IV : Approche évolutionnaire pour la recherche et la composition des itinéraires co-modaux | 125 |
| IV.1. | Introduction..... | 125 |
| IV.2. | D'une optimisation monocritère vers une optimisation multicritère | 125 |
| IV.3. | Les algorithmes évolutionnaires | 126 |
| IV.3.1. | Choix d'un codage..... | 128 |
| IV.3.2. | Opérateur de sélection | 128 |
| IV.3.3. | Opérateur de croisement | 129 |
| IV.3.4. | Opérateur de mutation | 131 |
| IV.3.5. | Convergence des algorithmes évolutionnaires | 131 |

Table des matières

| | | |
|-----------|--|-----|
| IV.4. | L'optimisation multicritère et les algorithmes évolutionnaires | 132 |
| IV.4.1. | Approche de résolution à priori | 134 |
| IV.4.2. | La méthode somme pondérée (SP) | 134 |
| IV.4.2.1. | La méthode par but | 135 |
| IV.4.2.2. | La méthode ϵ -contrainte | 135 |
| IV.4.2.3. | La méthode lexicographique | 135 |
| IV.4.3. | Approche de résolution à postériori | 136 |
| IV.4.3.1. | La méthode Vector Evaluated Genetic Algorithm (VEGA) | 136 |
| IV.4.3.2. | La méthode Multiple Objective Genetic Algorithm (MOGA) | 137 |
| IV.4.3.3. | La méthode Non dominated Sorting Genetic Algorithm (NSGA) | 137 |
| IV.4.4. | Approche de résolution progressive | 137 |
| IV.4.4.1. | La méthode STEM | 138 |
| IV.4.4.2. | La méthode Steuer | 138 |
| IV.5. | Approche évolutionnaire proposée pour une optimisation multicritère | 138 |
| IV.5.1. | Génération des agents Route | 139 |
| IV.5.2. | Fonctions d'évaluation du chromosome VeSAR | 140 |
| IV.5.2.1. | Critère coût CRg1 pour une route Rg | 141 |
| IV.5.2.2. | Critère temps CRg2 pour une route Rg | 142 |
| IV.5.2.3. | Critère émission des gaz à effet de Serre (GES) CRg3 pour une route Rg | 143 |
| IV.5.3. | L'opérateur de croisement | 145 |
| IV.5.3.1. | Algorithme de croisement | 145 |
| IV.5.3.2. | Exemple d'application de l'opérateur de croisement | 146 |
| IV.5.3.3. | Opérateur de correction de l'opérateur de croisement | 147 |
| IV.5.3.4. | Exemple d'application pour l'opérateur de correction de croisement | 148 |
| IV.5.4. | L'opérateur de mutation | 148 |
| IV.5.4.1. | Algorithme de mutation | 148 |
| IV.5.4.2. | Exemple d'application pour l'opérateur de mutation | 149 |

| | |
|---|-----|
| Table des matières | |
| IV.5.5. Choix du meilleur chromosome | 150 |
| IV.6. Coalition des RAs pour la formation des combinaisons de routes | 150 |
| IV.6.1. Formulation de la coalition des RAs | 151 |
| IV.6.2. Coalition : protocole d'interaction..... | 152 |
| IV.7. Meilleure combinaison de routes pour chaque requête..... | 155 |
| IV.8. Conclusion | 156 |
| V. Chapitre V : Implémentation et Scénarios du système SITCoMo | 157 |
| V.1. Introduction | 157 |
| V.2. Choix de l'outil informatique | 157 |
| V.3. Choix de la plateforme Multi-agents | 158 |
| V.3.1. Zeus | 159 |
| V.3.2. MadKit | 159 |
| V.3.3. Jade..... | 159 |
| V.4. Présentation et aspect graphique..... | 161 |
| V.4.1. Interface..... | 161 |
| V.4.2. Google Maps | 162 |
| V.4.3. Cartocom | 163 |
| V.5. Implémentation et simulation des algorithmes du système SITCoMo..... | 164 |
| V.5.1. Données utilisées..... | 164 |
| V.5.2. Tests et scénarios..... | 167 |
| V.5.3. Identification du domaine de recherche : Application du DSSA | 169 |
| V.5.3.1. Calcul des premiers plus courts chemins : construction des graphes et application du DSRA. | 170 |
| V.5.3.2. Optimisation, coalition des RAs et construction des solutions finales | 175 |
| V.6. Impact de la variabilité des services et des requêtes | 182 |
| V.7. Conclusion | 184 |
| Conclusion générale | 185 |

Table des matières

Bibliographie..... 189

Index des figures

| | |
|--|----|
| Figure I-1 Evolution des volumes de transports intérieurs de voyageurs | 26 |
| Figure I-2 Emission de Gaz à effet de serre en Europe..... | 27 |
| Figure I-3 Emissions de GES globales des modes de transports aux échelles urbaines et périurbaines (Deloitte, 2008)..... | 27 |
| Figure I-4 Intermodalité | 29 |
| Figure I-5 Multimodalité..... | 30 |
| Figure I-6 Transport co-modal | 33 |
| Figure I-7 Exemple de covoiturage | 36 |
| Figure I-8 Villes disposant d'un ou plusieurs services d'autopartage (2008)..... | 42 |
| Figure I-9 Services de transport offerts par SITCoMo | 50 |
| Figure II-1 Comparaison des différentes architectures | 56 |
| Figure II-2 Cycle de vie d'un agent | 59 |
| Figure II-3 Différentes formes d'interactions | 62 |
| Figure II-4 OMA à structure hiérarchique | 63 |
| Figure II-5 OMA à structure de marché..... | 63 |
| Figure II-6 OMA à structure de communauté..... | 64 |
| Figure II-7 OMA à structure de société | 64 |
| Figure II-8 Les différents minima | 68 |
| Figure II-9 Méthodes d'optimisation monobjectif..... | 72 |
| Figure II-10 Algorithme Recherche Taboue | 75 |
| Figure II-11 Algorithme Recuit simulé..... | 77 |
| Figure II-12 Fonctionnement générique d'un algorithme évolutionnaire..... | 81 |
| Figure III-1 Fonctionnement du système SITCoMo..... | 88 |
| Figure III-2 Un tronçon d'itinéraire : une Route <i>R_g</i> | 90 |
| Figure III-3 Une combinaison de Route <i>R_{ck,p}</i> | 90 |
| Figure III-4 Architecture du système multi-agents | 94 |
| Figure III-5 Comportement du système pendant une période de formulation | 95 |
| Figure III-6 Diagramme d'activité de l'agent interface IA | 96 |
| Figure III-7 Les rôles du Super Agent (<i>SupA</i>)..... | 97 |

Index des figures

| | | |
|---|-----------------------------------|-----|
| Figure III-8 Clonage du SupA pour la résolution du problème des plus courts chemins dans chaque composant | 98 | |
| Figure III-9 Diagramme d'activité de l'agent SupA | 99 | |
| Figure III-10 Diagramme d'activité de l'agent TSA..... | 100 | |
| Figure III-11 Diagramme d'activité de l'agent TIA..... | 100 | |
| Figure III-12 Diagramme d'activité de l'agent RA..... | 101 | |
| Figure III-13 Diagramme d'activité de l'agent Evaluator (EA)..... | 102 | |
| Figure III-14 Comportement du système SITCoMo..... | 102 | |
| Figure III-15 Exemple d'un graphe d'adjacence pour un service de transport i..... | 104 | |
| Figure III-16 Graphe non orienté | Figure III-17 Graphe orienté..... | 107 |
| Figure III-18 Graphe co-modal | 108 | |
| Figure III-19 Exemple de graphe de transfert | 110 | |
| Figure III-20 Différence entre Réseau FIFO et non FIFO | 114 | |
| Figure III-21 Graphe de transfert avec trois composants..... | 117 | |
| Figure III-22 Graphe distribué d'un composant..... | 118 | |
| Figure III-23 Graphe d'intersection | 120 | |
| Figure III-24 Graphe étendu virtuel $G_{veint}(A, B, WAB)$ | 121 | |
| Figure III-25 Le plus court chemin entre A et B dans $C1$ | 122 | |
| Figure IV-1 Les différents rôles du SupA..... | 126 | |
| Figure IV-2 Etapes d'un algorithme évolutionnaire | 127 | |
| Figure IV-3 Exemple d'un codage binaire | 128 | |
| Figure IV-4 Exemple de croisement à un seul point..... | 130 | |
| Figure IV-5 Exemple de croisement multi-points..... | 130 | |
| Figure IV-6 Exemple de croisement uniforme..... | 131 | |
| Figure IV-7 Exemple de mutation..... | 131 | |
| Figure IV-8 Illustration d'un ensemble de solutions dans l'espace des objectifs | 133 | |
| Figure IV-9 Traitement de problème d'optimisation multicritère : Approches et exemples de méthodes de résolution..... | 134 | |
| Figure IV-10 Méthode lexicographique | 135 | |
| Figure IV-11 Principe de l'algorithme VEGA..... | 136 | |
| Figure IV-12 Coalition des agents | 151 | |
| Figure IV-13 Coalition des RAs..... | 151 | |
| Figure IV-14 Proposition de coalition..... | 154 | |
| Figure IV-15 Formation de coalition | 155 | |

| | |
|--|-----|
| Index des figures | |
| Figure V-1 Architecture logicielle de la paletform JADE | 160 |
| Figure V-2 Interface utilisateur | 161 |
| Figure V-3 Exemple d'itinéraire sur Google Maps..... | 162 |
| Figure V-4 Interface Cartocom | 163 |
| Figure V-5 Le réseau de transport étudié | 165 |
| Figure V-6 Stations de voitures d'autopartage..... | 166 |
| Figure V-7 Stations de Vlille | 166 |
| Figure V-8 Liste des opérateurs pour chaque requête pour le scénario 1 | 169 |
| Figure V-9 Liste des opérateurs pour chaque requête pour le scénario 3 | 170 |
| Figure V-10 Communication entre les agents <i>supA</i> , <i>TSAi</i> et <i>TIAi, j</i> | 170 |
| Figure V-11 Réseau de transport pour le scénario 1 | 171 |
| Figure V-12 Graphe de transfert pour le scénario 1 | 171 |
| Figure V-13 Classes du composant C1 | 173 |
| Figure V-14 Graphe d'intersection | 173 |
| Figure V-15 Comparaison entre le graphe de transfert CT et SPTG | 174 |
| Figure V-16 Communication entre les RA pour former des coalitions | 176 |
| Figure V-17 Solutions obtenues pour l'exemple..... | 177 |
| Figure V-18 Solutions obtenues sur Google Maps | 178 |
| Figure V-19 Résultats obtenus avec CARTOCOM pour I1, I2 et I3 | 179 |
| Figure V-20 Résultats obtenus avec CARTOCOM pour I4, I5 et I6 | 180 |
| Figure V-21 Nouvel itinéraire pour I1 | 181 |
| Figure V-22 Nouvel itinéraire pour I1 sur Google Maps | 181 |
| Figure V-23 Evolution du nombre des nœuds et du nombre d'arcs pour chaque requête en fonction du nombre des services | 182 |
| Figure V-24 Evolution du nombre des nœuds, nombre des points PTC et nombre des arcs en fonction des requêtes..... | 183 |

Index des Tableaux

| | |
|---|-----|
| Tableau III-1 Comparaison entre les problèmes du voyageur de commerce et de l'optimisation dans un système de transport co-modal..... | 92 |
| Tableau IV-1 Vitesse moyenne de différents véhicules..... | 143 |
| Tableau IV-2 Facteurs d'émission par voiture et par Km en fonction du parcours | 144 |
| Tableau IV-3 Facteurs d'émission par véhicule par km pour différents types d'autobus..... | 145 |

Index des figures

| | |
|--|-----|
| Tableau IV-4 Facteurs d'émission par voyageur pour différents modes de transport public (données RATP 2005)..... | 145 |
| Tableau IV-5 Facteurs d'émission par voyageur par km par type de train (SNCF 2009) | 145 |
| Tableau V-1 Liste des PTC | 172 |
| Tableau V-2 Résultats du graphe de transfert pour le scénario 3..... | 174 |
| Tableau V-3 Tableau 2 Résultats du graphe SPTG pour le scénario 3 | 174 |

Index des Tables

| | |
|--|-----|
| Table IV-1 Exemple de chromosome VeSar..... | 140 |
| Table IV-2 Instance de VeSAR..... | 140 |
| Table V-1 Chromosome R(Dunkerque, Gare Lille Europe, [7h30 ; 8h55])..... | 175 |

Glossaire

ADEME : Agence de l'Environnement et de la Maîtrise de l'Energie

AE : Algorithmes Evolutionnaires

ACI : Algorithme de Calcul d'Itinéraires

AG : Algorithmes génétiques

ARENE : Agence Régionale de l'Environnement et des Nouvelles Energies

BD : Base de données

CERTU : Centre d'Etudes sur les Réseaux, les Transports, l'Urbanisme et les constructions publiques.

CODAC : Covoiturage Optimisé Dynamique basé sur les Agents Communicants

DSRA : Distributed Shortest Route Algorithm

DSSA : Domain Search Selection Algorithm

EA : Agent Evaluator

FIFO : First In First OUT

GES : Gaz à effet de serre

FIPA : Foundation for Intelligent Physical Agents

IA : Agent Interface

INSEE : Institut National de la Statistique et des Etudes Economiques

PDA : Personal Digital Assistant

PVC : Problème Voyageur de Commerce

RA : Agent Route

SIAD : Système d'Information d'Aide au Déplacement

SITCoMo : Système d'Information de Transport Co-modal

SMA : Système Multi-Agents

SNDD : Stratégie Nationale de Développement Durable

SP : Somme Pondérée

SPTG : Graphe de Transfert des plus courts chemins

SupA : Super Agent

TIA_{ij} : Agent Information de Transport j du service i

TSA_i : Agent Service de Transport i

VeSAR : Vehicle Sharing Assignment Representation

Formulation du problème :

N : Requêtes formulées pendant un intervalle de temps Δ_ε

Δ_ε : Période d'acquisition

Δ_∞ : Période d'inactivité

I_t : Ensemble de requêtes à l'instant t

I_k : Une demande d'itinéraire à l'instant t

d_k : Point de départ de la requête I_k

a_k : Point d'arrivée de la requête I_k

$W_k = [td_k, ta_k]$: Intervalle de temps souhaité de la requête I_k

td_k : le temps de départ au plus tôt de d_k

ta_k : le temps d'arrivée au plus tard à a_k

RC_k : Ensemble de toutes les combinaisons de route possibles identifiées pour répondre à une demande d'itinéraire I_k

$RC_{k,p}$: Une combinaison de route possible formant une solution à I_k

$R_g(d_g, a_g, W_g)$: une route allant de d_g à a_g dans l'intervalle de temps W_g

$V_h^{R_g}$: un véhicule qui assure V_h qui assure à l'instant t une seule route $R_g(d_g, a_g, W_g)$

H : le nombre total de véhicules disponibles pour servir la route R_g

$C_{V_h^{R_g}}^i$: Valeur du critère i pour le véhicule $V_h^{R_g}$

CR : le nombre total de critères à considérer

S_t : Société d'agents à l'instant t

Algorithme DSSA :

- $Ga_i(Na_i, Ea_i)$: Graphe d'adjacence du TSA_i
- Na_i : Ensemble des nœuds du graphe Ga_i représentant l'ensemble des agents $TIA_{i,j}$
- Ea_i : Ensemble des arcs du graphe Ga_i représentant les possibilités de passage du réseau d'un opérateur à un autre.

Définitions des graphes :

- $G = (N, E)$: un graphe avec N ensemble de nœuds et E ensemble d'arcs
- $N^-(x)$: La liste des nœuds prédécesseurs d'un nœud $x \in N$
- $N^+(x)$: la liste des nœuds successeurs d'un nœud $x \in N$

Glossaire

- $G = (N, E, M)$: Graphe co-modal
- $N = \{n_1, \dots, n_j\}$: Ensemble des nœuds
- $E = \{e_1, \dots, e_L\}$: Ensemble des arcs
- $M = \{m_1, \dots, m_K\}$: Ensemble des services de transport
- $e_l = (n_p, n_q)_{m_r}$: un arc co-modal exprimant qu'il est possible d'aller du nœud n_p vers le nœud n_q en utilisant le service de transport m_r
- $\omega_{e_l} = \omega(n_p, n_q)_{m_r}$: Poids ou le coût de l'arc e_l
- $RC_{n_1, n_l} = (n_1 \rightarrow n_l)$: Une combinaison de route dans le graphe co-modal pour aller de n_1 à n_l .
- $G_T = (C, TR)$: Graphe de transfert
- $C = \{C_1, C_2, \dots, C_k\}$: l'ensemble des composants du graphe de transfert représentant l'ensemble des réseaux de transport
- TR : Ensemble des arcs de transfert qui relient les différents composants du graphe de transfert. Ils symbolisent la possibilité de transférer d'un service de transport à un autre.
- $C_i = (N_i, E_i, M_i, PTC_i)$: Graphe relié à chaque composant C_i du graphe de transfert
- PTC_i : Points de transfert co-modaux d'un composant C_i
- RC_{d_k, a_k}^{*i} : Le plus court chemin qui part du nœud d'origine d_k et arrive au nœud d'arrivée a_k dans la classe C_i .
- RC_{d_k, PTC_i}^{*i} : Le plus court chemin qui part du nœud d'origine d_k et arrive à un nœud de transfert PTC_i dans la classe C_i .
- RC_{PTC_i, PTC_j}^{*i} le plus court chemin qui part d'un nœud de transfert PTC_i et arrive à un autre nœud de transfert PTC_j dans la classe C_i .
- RC_{PTC_i, a_k}^{*i} : Le plus court chemin qui part d'un nœud de transfert PTC_i et arrive au nœud d'arrivée a_k dans la classe C_i .

Calcul du plus court chemins dans les composants : Algorithme DSRA :

- $C_{i,j}$ une classe représentant un opérateur j d'un service C_i
- $G_{i,j}(N_{i,j}, E_{i,j})$: Graphe relatif à la classe $C_{i,j}$ avec $N_{i,j}$ et $E_{i,j}$ sont respectivement l'ensemble de nœuds et l'ensemble des arcs relatifs à la classe $C_{i,j}$
- $RC_{x,y}^{*i,j}$ le plus court chemin local dans une classe $C_{i,j}$ pour aller de x à y
- $RC_{x,y}^{*i}$ le plus court chemin dans le composant C_i pour aller de x à y .

Glossaire

- $G_{cint}(N_{cint}, E_{cint})$: Graphe complet d'intersection
- $rec(x, y)$: Etiquette pour connaître à quelle classe appartient l'arc (x, y)
- $G_{veint}(d_k, a_k, W_k)$: Graphe virtuel d'intersection pour la requête I_k
- $\varphi_{d_{ki}}$: coût du chemin qui va de d_k vers i
- $\omega_{i,j}(t)$: poids de l'arc (i,j) à l'instant t
- $pred(i)$: nœud précédent de i
- $G_F = (N_f, E_f)$: graphe SPTG

Algorithmes évolutionnaires :

- $X_{V_h}^{R_g}$: transport de la personne en utilisant le véhicule V_h
- $C_{R_g}^i$: coût du critère i pour une route $R_g(d_g, a_g, W_g)$
- C_{km} : le coût de parcours d'un km.
- $D(d_g, a_g)$ est la distance entre le point de départ et le point d'arrivée de la route en Km.
- $T(d_g, a_g)$: le temps de parcours du point de départ d_g au point d'arrivée a_g d'une route avec un véhicule $V_h^{R_g}$.
- $V_m(V_h^{R_g})$: Vitesse moyenne du véhicule $V_h^{R_g}$ parcourant la route R_g .
- $Fe(V_h^{R_g})$: Facteur d'émission du véhicule par Km
- p_c : probabilité de croisement
- p_m : probabilité de mutation
- C_{R_g} : Coût d'une route R_g
- $\alpha_{k,i}$: pondération pour chaque critère i
- $C_{R_{gmax}}^i$: valeur maximale obtenue pour le critère i
-

Coalition des agents

- $A = \{RA_1, RA_2, \dots, RA_n\}$: un ensemble de n RAs
- $RC_k = \langle a, I_k \rangle$: une coalition RC_k
- Y_{R_g} : l'itinéraire incluant R_g pour répondre à la requête $I_k(d_k, a_k, W_k)$.
- $CR_{k,p}$: Coût de chaque combinaison de route

Introduction générale

De nos jours, la dispersion spatiale de l'habitat et les activités contribuent à une croissance considérable du trafic et de l'utilisation des voitures. La flexibilité, l'efficacité, la rapidité et le confort de la voiture particulière ont fait que ce moyen de transport devient le moyen le plus populaire et préféré des utilisateurs. Elle représente la liberté et la technologie. Selon le rapport de la CERTU en 2008, les statistiques ont montré que la voiture particulière demeure prédominante en représentant 60% des déplacements urbains. Les autres moyens de transport comme la marche à pieds, le transport en commun, le vélo et les deux roues motorisés représentent respectivement 27%, 9%, 2% et 2% des déplacements. Cependant, ce moyen de transport possède plusieurs inconvénients qui peuvent détériorer notre qualité de vie. En effet, essentiellement l'émission de CO₂ ainsi que d'autres gaz à effet de serre cause la pollution atmosphérique. De nos jours, l'intérêt porté à la préservation de l'environnement à travers la réduction des émissions de gaz à effet de serre prend de plus en plus d'ampleur. Dans un premier lieu, une politique multimodale d'encouragement de l'utilisation de différents types de transports en commun a été mise en place. Depuis 2006, cette politique a évolué vers une politique co-modale qui n'oppose plus la voiture au transport public mais encourage une combinaison de tous les modes de transport sans favoritisme dans le but d'une optimisation du service. En passant de la concurrence vers la complémentarité, le transport en général et le transport des personnes en particulier évoluent vers une vision efficace que ce soit dans le contexte environnemental (la combinaison la moins polluante), le contexte économique (le moindre coût) ou tout simplement la combinaison la plus pertinente à la situation.

Placés dans ce cadre, le but de cette thèse est de concevoir un système de transport co-modal capable de satisfaire les demandes des utilisateurs en leur fournissant des itinéraires co-modaux optimisés en terme de temps, coût et émission des gaz à effet de serre tout en respectant leurs préférences et priorités.

Ces travaux de recherche visent alors à mettre en œuvre un système de gestion de véhicules partagés qui recouvre tous les services de transports existants tel que le transport public, le covoiturage, l'autopartage, les vélos en libre service...

Toutefois un ensemble de problème se pose. Afin de générer un itinéraire co-modal, il faut consulter les données et les systèmes d'information de tous les opérateurs de tous les services de transport existants. L'aspect dynamique et distribué dans l'espace et dans le temps des

données à traiter rend le problème assez complexe et difficile à résoudre. En plus, dans les réseaux du transport, les perturbations (accidents, grèves, problèmes techniques, ...) sont assez fréquentes et gênantes pour les utilisateurs. Dans ce cas, il faut prendre en compte tous les changements d'horaires de tous les services de transport inclus dans l'itinéraire. Enfin, il ne faut pas oublier qu'il faut garantir l'optimalité des solutions en fonction des préférences et des priorités des usagers.

Pour mettre en place un tel système, une stratégie de résolution efficace mettant à profit une mixture de concepts ; à savoir les systèmes multi-agents et l'optimisation a été mise en place. Nous avons tiré profit de l'efficacité de ces deux technologies et méthodes pour les allier et générer de suite un système intelligent, distribué et varié grâce à la complémentarité entre les véhicules partagés et les autres moyens de transport existants.

La présente thèse est organisée en 5 chapitres comme suit :

- Dans le premier chapitre nous élaborons une étude générale sur le transport des personnes partant du transport monomodal jusqu'au transport co-modal. Nous décrivons ainsi un ensemble de systèmes d'information de transport existants en mettant en relief leurs apports ainsi que leurs limites.
- Le deuxième chapitre est consacré à un état de l'art général sur les systèmes multi-agents et les méthodes d'optimisation ainsi que l'apport de leur combinaison et alliance dans le domaine du transport.
- Dans le troisième chapitre, nous présentons le premier niveau de notre approche d'optimisation distribuée pour la mise en œuvre d'un système d'Information de Transport Co-modal (SITCoMo) basé sur une architecture multi-agents et une première approche d'optimisation appliquée sur un graphe co-modal.
- Le quatrième chapitre est dédié à la présentation de notre deuxième niveau d'optimisation avec les approches évolutionnaires proposées pour la recherche et la composition des itinéraires co-modaux optimisés en termes de trois critères.
- Dans le dernier chapitre, nous présentons des scénarios d'utilisation du système adopté pour démontrer l'efficacité des solutions proposées. Les résultats obtenus justifient l'importance de la co-modalité et la nécessité de mettre à profit la complémentarité entre les véhicules partagés et les autres moyens de transport à travers un système intelligent, unique et global.

I. Chapitre I Transport : de la multimodalité vers la co-modalité

I.1. Introduction

Aujourd'hui le monde entier est touché par des crises économiques, sociales et financières sans oublier la persistance des enjeux environnementaux. Dans ce contexte, le comité interministériel pour le développement durable a proposé « La stratégie Nationale de Développement Durable (SNDD) 2010-2013 ». « La SNDD pose les bases d'un mode de développement durable organisé autour d'une économie verte et équitable ». En effet, nos modes de vie doivent changer en de nouveaux modes d'organisation, de production et de consommation qui préservent les ressources naturelles et luttent contre la pollution ce qui nous mènera à une diminution des gaz à effet de serre (GES).

La SNDD s'articule autour de neuf défis : la société de la connaissance, la gouvernance, le changement climatique et l'énergie, les transports et la mobilité durable, la conservation et la gestion durable de la biodiversité et des ressources naturelles, la santé publique et la gestion des risques, la démographie, l'immigration et l'inclusion sociale et enfin les défis internationaux en matière de développement durable et de lutte contre la pauvreté dans le monde.

Les transports et la mobilité durable sont considérés comme un défi et un objectif majeur à atteindre. Ceci a donné lieu à l'élaboration de programmes collectifs rassemblant plusieurs états et faisant ainsi la preuve du développement d'une culture mondiale de respect de l'environnement. Parmi ces programmes, nous trouvons la sensibilisation des personnes et des voyageurs vers la nouvelle notion de co-modalité et spécialement le recours aux véhicules privés dans leur sens public. En effet, la voiture particulière a tourné vers une autre forme : la voiture partagée.

I.2. Les transports : Leur évolution et leur impact sur l'environnement

Au cours des vingt dernières années, l'évolution des comportements de mobilité a été marquée par une stabilisation du nombre de déplacements locaux par personne et de leur durée totale par jour ; en revanche, la distance parcourue par déplacement a fortement augmenté tout comme leur vitesse moyenne. Les déplacements sont de plus en plus rapides, distants, répartis dans l'espace et dans le temps.

Cette répartition de mobilité coïncide avec l'évolution des modes de vie urbains, de l'organisation du travail et de la structure urbaine elle-même. La ville est donc aujourd'hui multipolaire et son fonctionnement repose non plus sur la proximité mais sur la mobilité qui devient première dans la détermination des modes de vie.

L'évolution des conditions de la mobilité ont conduit forcément à l'évolution du nombre de kilomètres parcourus par les voyageurs.

Selon l'Institut National de la Statistique et des Etudes Economiques (INSEE) en 2010, le transport intérieur de voyageurs accélère tout en restant à un rythme modéré (+ 0,8 % en **voyageurs-km**). La voiture particulière, qui représente près de 82 % des transports de personnes, augmente au même rythme que l'ensemble malgré la hausse du prix des carburants en seconde moitié de l'année. La circulation routière repart à la hausse et cette augmentation concerne l'ensemble des véhicules. Le transport a connu un recul en 2009 à cause de la crise mais il retrouve en 2010 le niveau atteint en 2008 avec une croissance de +0.6 voyageurs.km. Cette croissance est remarquée pour les transports urbains tandis que le transport régional (ferroviaire, bus et cars non urbains) augmente modérément, et, globalement, les liaisons interurbaines longue distance (ferroviaire et aérien) reculent. (Commisariat général du développement durable, 2011).

Évolution des volumes de transports intérieurs de voyageurs

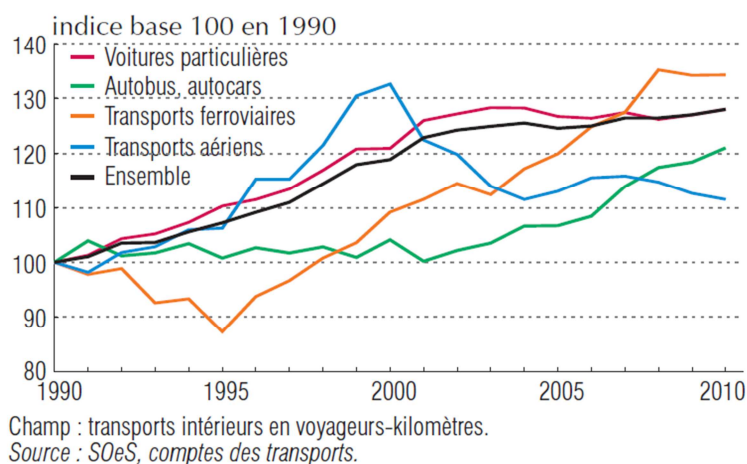
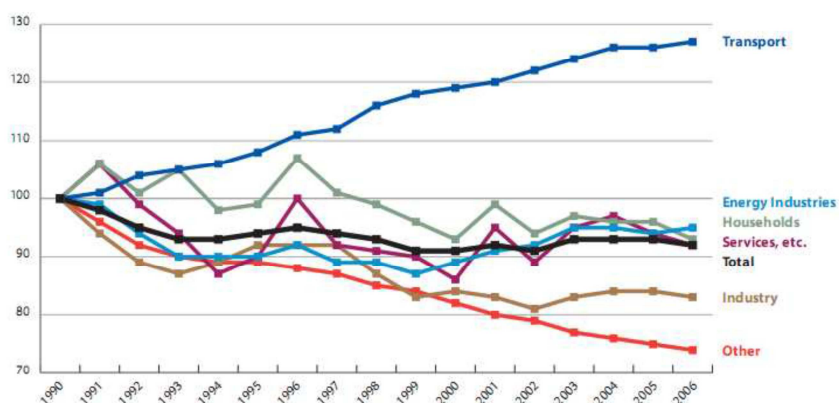


Figure I-1 Evolution des volumes de transports intérieurs de voyageurs

La figure I-2 représente l'évolution à la hausse des émissions des gaz à effet de serre entre 1990 et 2006.

GHG emissions in the EU-27, by sector (1990 = 100)



Source: EU energy and transport in figures — Statistical pocketbook 2009.

Figure I-2 Emission de Gaz à effet de serre en Europe

Selon les données de l'Agence pour l'Environnement de l'Union européenne, les transports représentent presque le quart (23,8%) des émissions totales de GES et un peu plus d'un quart (27,9%) du total des émissions de CO₂ dans l'UE-27 en 2006. Le plus inquiétant est que ces valeurs ont tendance à croître. En outre, bien que le total des émissions soit en légère baisse, les émissions des transports continuent à augmenter. Il devient de plus en plus urgent d'essayer de remédier à ce problème vu la forte croissance de ces taux.

La Figure I-3 montre les émissions globales de GES de différents modes de transport dans les zones urbaines et périurbaines (Paris).

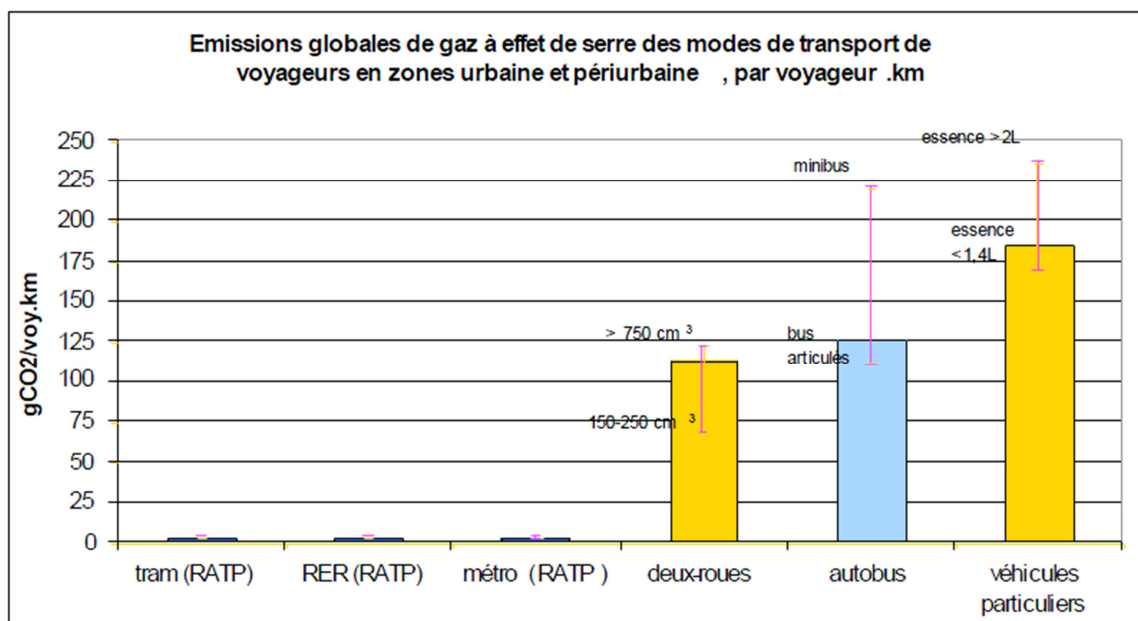


Figure I-3 Emissions de GES globales des modes de transports aux échelles urbaines et périurbaines (Deloitte, 2008)

Nous pouvons voir que les émissions de GES des transports en commun ferrés sont très faibles par rapport à celles des deux roues, autobus et surtout la voiture particulière.

I.3. Les transports et la mobilité durable

En 1987, la Commission mondiale sur l'environnement et le développement des Nations Unies a formulé la définition du développement durable comme étant « un développement qui répond aux besoins des générations du présent sans compromettre la capacité des générations futures à répondre aux leurs ». Toutes nos techniques et nos organisations doivent évoluer de telle façon elles protègent et partagent les ressources naturelles limitées de notre planète.

En prenant conscience des enjeux du développement durable, toute la communauté internationale a décidé en 2002 d'élaborer des politiques globales cohérentes. Cependant, l'intégration du développement durable s'est faite lentement vu que certains pays ne semblaient pas trop convaincus par l'intérêt d'intégrer ce genre de stratégie dans leurs politiques publiques. Mais depuis 2008, la crise économique qui a submergé le monde a poussé les états à s'inquiéter et à se poser des questions sur les causes des déséquilibres qui ont fait trembler les systèmes économiques et financiers.

En plus de la crise économique vient la crise écologique qui ne cesse d'évoluer et de prendre une ampleur assez importante. En effet, le Groupe d'experts Intergouvernemental sur l'Evolution du Climat (GIEC) estime que les températures mondiales devraient augmenter de 2 à 3 °C durant les cinquante prochaines années. En plus, le monde ne cesse de voir la hausse importante des émissions des gaz à effet de serre devant la capacité décroissante de la planète à capturer et séquestrer le carbone. En France, le développement durable a été intégré pour la première fois dans la stratégie nationale de développement durable 2003-2008 (SNDD). Actuellement, cela continue en essayant de réussir la SNDD 2010-2013 en préservant l'équilibre entre les dimensions environnementale, sociale et économique du développement durable et en conciliant les droits des générations présentes et futures. Grâce à cette stratégie, la France pourrait être dans le futur l'un des acteurs majeurs de l'économie verte, une économie sobre en ressources naturelles. Parmi les défis que la SNDD veut atteindre, nous citons celui des transports et mobilité durable. Ce dernier vise la réduction des déplacements contraints et le développement des systèmes innovants qui répondent aux besoins de performances économiques, écologiques et de cohésion sociale.

I.4. Transport : De la multimodalité vers la co-modalité

Un mode de transport ou un moyen ou un système de transport est une forme particulière de transport qui se distingue principalement par le véhicule utilisé, et par conséquent par l'infrastructure qu'il met en œuvre. Il s'agit d'un accessoire pour l'être humain afin de se déplacer du point A au point B.

La combinaison de plusieurs modes de transport pour des déplacements donne plus de flexibilité aux voyageurs et rend le réseau de transport plus rentable. Cette combinaison de modes de transport a donné naissance à plusieurs nouvelles notions dans le domaine du transport tel que l'intermodalité, la multimodalité pour arriver à la co-modalité. Nous commençons par définir ces différents modes de transport.

I.4.1. Le transport monomodal

Le transport monomodal concerne les déplacements réalisés avec un seul mode tout au long du trajet à effectuer ; désignant ainsi le fait d'utiliser un mode particulier pour réaliser les opérations de transport et considère ainsi un type spécifique de véhicule particulier ou public (e.g. Bus, Métro, voiture, vélo, etc.).

I.4.2. Le transport intermodal

A l'opposé de la monomodalité, l'intermodalité consiste à utiliser au moins deux modes de transport pour réaliser un déplacement d'un point d'origine à un point de destination. D'après l'ARENE (ARENE, 2002), l'intermodalité vise à réduire l'usage de la voiture particulière au profit de transport moins polluant (transports collectifs, vélos ...).



Figure I-4 Intermodalité

I.4.3. Le transport multimodal

La multimodalité offre plusieurs moyens de transport pour un déplacement entre une origine et une destination. La multimodalité a été le sujet de plusieurs travaux récents vu que l'offre de transport est un axe important d'amélioration de la qualité de service du transport, en adéquation avec les enjeux environnementaux, économiques et par la suite politiques.

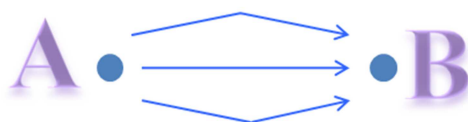


Figure I-5 Multimodalité

La multimodalité est basée sur la notion de choix où le client doit choisir le moyen de transport qu'il va prendre selon le jour, l'heure ou le motif de son déplacement. En général, le client cherche à optimiser l'usage de la gamme de transport disponible en tirant profit des avantages de performances intrinsèques à chaque mode.

L'information multimodale est l'information qui permet de renseigner les utilisateurs sur toute information liée au déplacement qu'il souhaite effectuer ou qu'il est en train d'effectuer. Cette information permet de le mettre au courant des différentes modalités (intermodalité et multimodalité) ainsi que les conditions de son déplacement et elle peut également l'encadrer et l'orienter pendant son parcours. Les services de l'information multimodale doivent être facilement accessibles en étant diffusés par différents canaux et supports afin d'accompagner les voyageurs dans leurs déplacements urbains au quotidien (guichet, Web, téléphone, PDA, bornes, etc.). Ces services peuvent correspondre aussi bien à des services de transport (calcul d'itinéraire, consultation d'horaire, informations sur les perturbations et retards, tarifs de déplacement, disponibilités des places de parking, etc.) qu'à des services connexes au transport (événements culturels, météo, informations touristiques, etc.).

Un réseau de transport multimodal étendu sur une vaste zone géographique ne peut être que la réunion de plusieurs sous-réseaux. Le réseau de transport public en France réunit un réseau national SNCF avec plusieurs réseaux de transport régionaux liés par les pôles d'échanges *de la SNCF*, représentant les stations communes. Un réseau régional peut également réunir plusieurs réseaux départementaux et un réseau départemental, à son tour, peut aussi réunir plusieurs réseaux de communes. Au niveau des départements et des communes, les liaisons des réseaux se fait par un ensemble de lignes de bus et de TER régionaux. Dans tous les cas, chaque exploitant, privé ou public, dispose de l'information multimodale du réseau de transport qu'il exploite. La génération d'une information multimodale sur une vaste zone géographique implique donc les différents opérateurs dont les réseaux de transport couvrent au moins une partie de la zone concernée.

Une multitude de travaux traite le sujet du transport multimodal que ce soit du côté régulateur ou côté client. En effet, le transport côté régulateur représente un axe de recherche pour la régulation du trafic par les systèmes d'aide à la décision, cet axe est également relié aux

problèmes d'aménagement, de planification et d'exploitation. Traitant ce problème, nous pouvons citer les travaux de recherche effectués dans notre équipe par OuldSidi (OuldSidi, 2006) et Fayeche (Fayeche, 2003). (OuldSidi, 2006) a développé un système d'aide à la décision, à partir des informations fournies par le SAE (Système d'Aide à l'Exploitation) dans le but d'élaborer, évaluer et proposer des solutions efficaces aux régulateurs, en tenant compte de leurs préférences. (OuldSidi, 2006) se base sur une méthode d'optimisation utilisant la théorie de la logique floue et les algorithmes évolutionnaires pour optimiser les solutions de régulation. Cette approche utilise également une méthode d'évaluation des décisions basée sur l'application de l'intégrale de Choquet, comme opérateur d'agrégation, pour l'évaluation des décisions. (Fayeche, 2003) a présenté un Système d'Aide à la Décision pour la régulation dans le but de traiter le problème de l'adaptation de l'offre aux conditions réelles d'exploitation et donc assister les régulateurs dans les différentes tâches qui leur sont attribuées. Le système proposé effectue la surveillance du trafic, la détection des incidents, leur diagnostic et la régulation et 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. La méthode proposée par (Fayeche, 2003) se base sur 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.

Concernant le transport du côté client, les travaux effectués dans ce domaine ont pour but de fournir au client toutes les informations nécessaires à son voyage, sans qu'il soit obligé de se connecter à plusieurs systèmes d'information, d'optimiser et de composer lui-même son itinéraire. L'avantage d'un système de transport multimodal est d'épargner au client la gestion d'un grand volume de données sur des réseaux d'information de plus en plus étendus en l'encadrant durant son trajet et en lui proposant différents services qui concernent non seulement le transport et les itinéraires mais aussi des services connexes qui peuvent l'orienter pendant son voyage et lui servir même de divertissement. Dans notre équipe, un certain nombre de chercheurs se sont également intéressés à ce type de problème. Kamoun (Kamoun, 2007) propose un système d'information orienté agents qui permet d'interroger les systèmes d'information existants et fournit par la suite aux clients une information multimodale et multi-opérateurs. Feki (Feki, 2010) a mis en place un Système d'Aide au Déplacement (SIAD) des voyageurs proposant des itinéraires les plus courts selon l'offre existante dans l'intervalle de temps demandé par l'utilisateur.

Les travaux de recherche de Zgaya (Zgaya, 2007) ont permis la conception, la mise en œuvre et l'optimisation d'un système d'Information de Transport Multimodal ouvert, dynamique et capable de fournir aux clients des réseaux de Transport une information pertinente en temps réel avant et pendant leurs déplacements.

Avec l'intégration de la notion de multimodalité afin de trouver des solutions aux problèmes environnementaux et sociaux rencontrés, l'offre multimodale concerne en général le transport public (bus, métro, tramway...). Cependant, certains travaux ont inclus les véhicules privés ou particuliers pour être un autre moyen de transport à offrir aux clients. Malgré que les auteurs de ces travaux continuent à considérer la combinaison entre le transport public et privé comme étant transport multimodal, une nouvelle notion ou un nouveau mode de transport a été introduit : le transport co-modal.

I.4.4. Le transport co-modal

La notion de co-modalité a été introduite en 2006 par la commission européenne dans le domaine de la politique des transports. Elle est définie comme « la combinaison optimale des différents modes sur la chaîne de transport » (Commission Européenne, 2007) représentant ainsi « le recours efficace à différents modes de transport isolément ou en combinaison ». Il s'agit d'un mode particulier adoptant le transport sous toutes ses formes afin de fournir un service souple offrant un maximum de flexibilité à l'utilisateur. Cette politique encourage les combinaisons rassemblant les moyens de transports en commun (tramway, train, métro, etc.) et privés (voiture privée, etc.) à l'opposé de l'intermodalité qui incite à aller plutôt vers les transports en commun suscitant de ce fait une sorte de rivalité entre les services de transport se basant sur l'un et l'autre des pratiques collectives ou privées (Commission Européenne, 2001). La co-modalité vient par ailleurs contrer cette concurrence pour imposer et instaurer un certain équilibre mettant en exergue la complémentarité entre les modes de transports collectifs et privés (Gille, Alain, 2006). D'après Giannopoulos (Giannopoulos, 2008), l'approche co-modale, de la même manière que l'approche multimodale, consiste à développer des infrastructures et prendre des mesures et actions qui assureront une combinaison *optimale* des différents modes de transport. Le but principal de cette politique est alors d'assurer la qualité et l'optimalité du service offert au niveau économique, environnemental, commercial, financier...

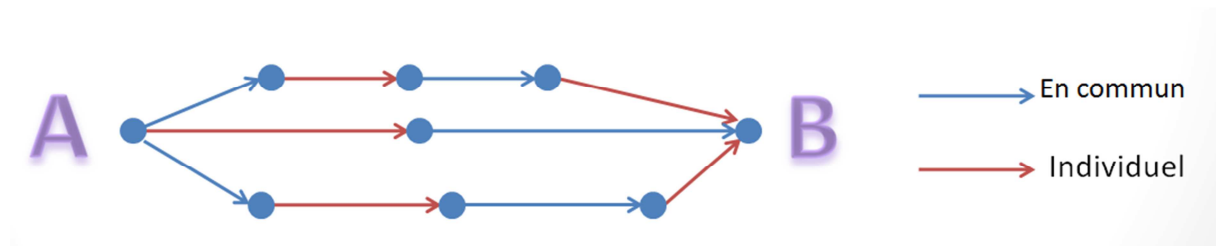


Figure I-6 Transport co-modal

Ce besoin de complémentarité est né de l'insatisfaction des voyageurs d'un seul moyen ou service de transport dans son contexte. Parmi les facteurs importants causant cette insatisfaction, nous pouvons citer à titre d'exemple la variabilité des besoins de mobilité motivant la nécessité d'adaptation des offres de transport. Un utilisateur peut avoir à se déplacer la plupart du temps en ville ou en agglomération, dans ce cas les moyens de transport les plus préconisés sont ceux relatifs aux modes de transport en commun, moins encombrants et surtout moins polluants. Par ailleurs, l'étalement périurbain vient imposer la voiture comme moyen de transport essentiel. En effet, la nécessité de déplacement hors des zones d'urbanisation, là où les modes de transport en commun sont limités, a créé un besoin récurrent de recourir à la voiture personnelle. Ainsi, grâce à sa flexibilité spatiale et temporelle, son confort, la voiture particulière devient indispensable à la majorité des personnes. Cependant, bénéficier de ces multiples avantages a effectivement un prix et se fait au détriment des budgets, de l'environnement dans lequel évolue la collectivité, des comportements, de la qualité de vie, etc. Pour cela, les nouvelles alternatives poussant à considérer à parts égales et dans une interopérabilité équitable de tous les modes de transport possibles s'impose. La co-modalité œuvre de ce fait dans le sens de l'équité entre les moyens de transports publics (collectifs ou individuels) et privés dans un contexte de combinaison favorisant plutôt la complémentarité à la concurrence (Gille, Alain, 2006).

Visant à réduire d'avantage l'impact négatif du domaine du transport particulièrement sur l'environnement, tous les regards ont été dirigés dans le sens de la voiture particulière. Etant donné qu'elle a été démontrée pour y être en grande partie responsable, accaparant la proportion d'émissions de CO₂ la plus importante, il devient de plus en plus urgent de trouver un bon remède qui saurait faire régresser les comportements des individus vers un usage plus raisonnable et donc des impacts de moindre incidence.

I.5. La notion de véhicule partagé

De nouvelles alternatives ont été adoptées dans le but de réduire le nombre de voitures en circulation, réduisant par la même occasion son impact péjoratif sur la fluidité du trafic et

l'environnement tout en maintenant accessible la flexibilité et facilité qu'elles offrent. Une solution clé se trouve dans le concept de véhicule partagé et spécialement la voiture partagée qui soigne l'image de l'automobile particulière la délogeant du contexte réquisitoire dont elle a toujours fait figure tout en préservant les avantages qu'elle présente.

Les systèmes basés sur la voiture partagée ont vu le jour suite au développement de plusieurs travaux visant l'amélioration de la qualité de vie sur le plan individuel et collectif. Parmi ces systèmes, nous pouvons essentiellement citer :

- Le covoiturage : partage d'un véhicule pendant un même intervalle de temps par plusieurs personnes pour parcourir une partie ou la totalité d'un trajet commun.
- L'autopartage : partage d'un même véhicule par période de temps par différentes personnes à chaque fois.

I.5.1. Le Covoiturage

« Le covoiturage consiste en l'utilisation commune d'un véhicule par un conducteur non professionnel et un (ou plusieurs) passager(s) dans le but d'effectuer tout ou une partie d'un trajet commun (Ballet, et al., 2007)». Cette nouvelle forme de transport a contribué à l'amélioration de l'image environnementale, sociale et économique de la voiture en réduisant les émissions de CO₂, les dépendances à la voiture, les frais de transport engagés dans le cadre d'une mobilité individuelle, etc. Le covoiturage permet aussi de restaurer une certaine communication qui a disparu dans les transports en commun en fortifiant les liens sociaux en regroupant des personnes dans la même voiture.

I.5.1.1. Définitions

Nous commençons tout d'abord par présenter certaines notions de base relatives au concept de covoiturage pour pouvoir en aborder les principaux éléments par la suite.

Covoiturage : Le covoiturage (dit *carpooling*, *ride-sharing* ou encore *lift-sharing* en anglais) se réfère à l'utilisation partagée d'une voiture par un conducteur non professionnel et un ou plusieurs passagers généralement pour effectuer des déplacements ensemble. Selon le concept énoncé du covoiturage, deux façons subsistent pour utiliser le ou les véhicules dans ce contexte :

- Un véhicule différent de l'un des covoiturés est utilisé à chaque fois, les trajets sont ainsi effectués à tour de rôle en utilisant l'un des véhicules propriétaires. Chaque individu intégrant un groupe de covoiturés est ainsi alternativement une fois conducteur, les autres passager.

- Un seul véhicule, propriété de l'un des covoiturés, est utilisé. Dans ce second cas, les autres passagers doivent contribuer aux frais de déplacement, à savoir le carburant et éventuellement le péage pour effectuer le trajet en question. Une participation forfaitaire est ainsi calculée selon le montant total des charges et le nombre de passagers au total sur le même trajet. Les usagers de ce service s'en voient ainsi réduire les frais de transport grâce à cette tarification avantageuse. Toutefois, il n'est pas envisageable que des voyages organisés dans ce cadre deviennent une source de profit pour l'utilisateur mettant son véhicule personnel à disposition des autres. En effet, le conducteur ne doit en aucun cas faire de bénéfices.

Covoitureurs : il s'agit des conducteurs d'un véhicule dans le contexte de covoiturage avec un ou plusieurs autres passagers. Ces usagers du service utilisent dans le cas général leur propre véhicule ou parfois des voitures de location. Dans ce dernier cas, les frais de location incombent aussi à tous les passagers du véhicule en question. Par ailleurs, l'initiateur du voyage, à savoir le conducteur, a ses propres besoins de déplacement qu'il définit à travers la soumission d'une offre de covoiturage.

Offre de covoiturage : elle désigne les paramètres de déplacement du conducteur (généralement propriétaire de la voiture utilisée pour le covoiturage). Ces paramètres définissent les spécificités du trajet à parcourir : origine, destination, date, heure de départ, nombre de places disponibles, etc.

Covoiturés : ce terme désigne les passagers d'une voiture de covoiturage. Ce sont des piétons qui sont en quête d'une éventuelle offre de covoiturage pouvant les ramener à un endroit précis. Elles sont ainsi définies comme intervenants du service initiant des demandes pour se faire conduire entre deux points donnés dans le cadre d'un déplacement souhaité.

Demande de covoiturage : un déplacement en covoiturage ne peut se faire que s'il existe une demande. Il s'agit d'une requête d'utilisateur émettant le souhait de se déplacer en voiture d'un endroit à un autre. Une demande concerne un besoin spécifique de déplacement en fonction duquel l'utilisateur détermine la date et l'heure du voyage, l'endroit où il veut aller, etc.

Contraintes de correspondance : Pour pouvoir effectuer un covoiturage, il faut que certaines conditions soient satisfaites. Dans les systèmes de covoiturage classiques, seules des contraintes basiques sont vérifiées. Comme le montre la Figure I-7, ces contraintes se réfèrent essentiellement à la correspondance entre offre et demande permettant ainsi de vérifier les :

- Dates et heures des voyages offert et requis : les dates et heures des covoitureurs et covoiturés doivent se correspondre ou se confondre.
- Coïncidences entre les trajets : le déplacement concernant la requête du covoituré doit de ce fait appartenir à l'itinéraire à parcourir par le covoitureur pour que sa demande puisse être considérée.
- Nombres de places disponibles dans une voiture de covoiturage qui doivent être suffisantes pour pouvoir accueillir les usagers demandeurs du même trajet
- Etc.

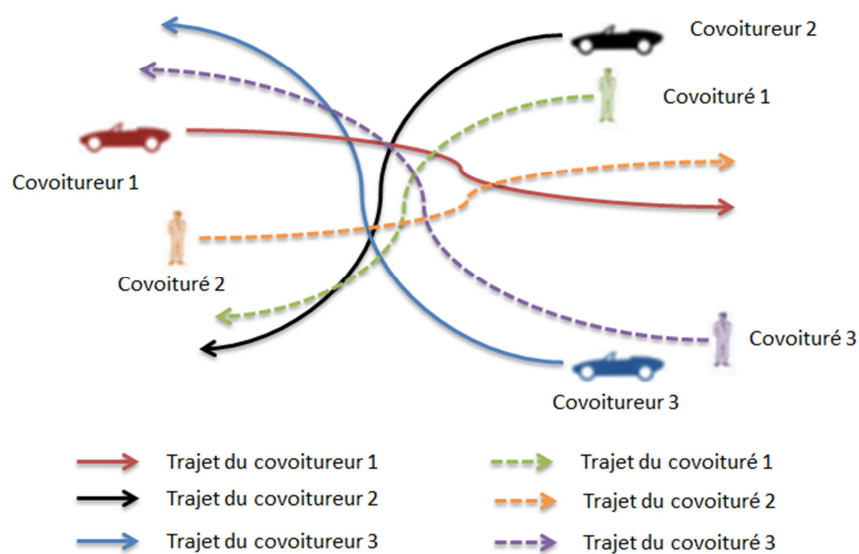


Figure I-7 Exemple de covoiturage

I.5.1.2. Types de trajets de covoiturage

On distingue trois types de trajets de covoiturage (Ballet, et al., 2007) :

- les trajets réguliers, c'est-à dire les trajets qui se font au moins une fois par semaine ou bien aussi les trajets quotidiens entre le lieu de résidence et le lieu de travail ou également les trajets scolaires.
- Les trajets occasionnels sont les trajets qui ne se répètent pas souvent dans le temps. Ce sont des trajets principalement longue distance très utilisés pour les départs en vacances par exemple
- Les trajets événementiels qui se font lors des évènements (concerts, festivals...) ou bien dans le cas aussi de panne ou de grève de transport en commun.

I.5.1.3. Formes du covoiturage

Le covoiturage peut prendre deux formes : spontané ou organisé.

- Le covoiturage spontané : Il fait référence à toutes les personnes qui covoiturent sans passer par une structure de mise en relation. Il peut bien être l'autostop ou bien le covoiturage informel. Il s'agit de covoitureurs indépendants de tout système, s'étant entendu mutuellement sur les modalités de déplacement (horaire, date, frais de transport...). Ce covoiturage informel correspond à des regroupements volontaires de personnes, qu'ils soient amis, collègues ou voisins, qui décident de s'organiser pour réaliser un trajet.
- Le covoiturage organisé : Dans le cas de covoiturage organisé, il faut passer par une entité intermédiaire de mise en relation pour trouver un covoitureur potentiel. Cette entité peut se décliner sous différentes formes : Centrale de mobilité (structure qui centralise toutes les informations concernant l'offre en transport), associations (un cadre privilégié pour encourager le covoiturage) et collectivités, entreprises/ administrations, Particuliers (sites Internet de covoiturage).

I.5.1.4. Covoiturage : Systèmes existants

Pour la concrétisation du processus de développement durable et la mise en place d'une mobilité avancée, le concept de covoiturage a connu un succès de plus en plus important. Ce concept a émergé dans deux formes statique et dynamique.

Covoiturage Statique

Le covoiturage statique est assuré par un système non automatisé accessible via le web, intégrant les fonctionnalités de gestion des offres et demandes de covoiturage. La majorité des systèmes existants font partie de cette catégorie. L'utilisation de tels systèmes nécessite une inscription préalable afin de pouvoir avoir accès aux différentes fonctionnalités offertes par le biais de ces supports. Malheureusement, ces fonctionnalités se résument dans le cas général à un simple dépôt ou consultation d'une offre ou demande de covoiturage selon que l'internaute soit covoitureur ou covoituré. En effet, ces sites web, dont la plupart sont dynamiques disposant d'une base de données, ont pour fonctionnalité principale de sauvegarder les offres et demandes en vue d'une éventuelle demande de consultation. Ils doivent donc être dotés de capacités de stockages nécessaires en adéquation avec les besoins de sauvegarde.

Mis à part ces sites qu'on appelle sites à grand public, il existe aussi des sites à accès restreints où un code est nécessaire pour l'utilisation du service. Selon les cas, un code

entreprise / administration, un code salarié, le numéro d'étudiant ou encore le numéro de sécurité sociale sont nécessaires pour l'utilisation du service.

Un opérateur de covoiturage est une structure qui propose ses compétences et son savoir-faire à d'autres structures pour la réalisation et la gestion d'un site de covoiturage spécifique à un territoire. Il s'agit, dans la plupart des cas, d'une société mais il existe des opérateurs sous forme associative. Leurs attributions sont diverses : certains opérateurs sont spécialisés dans le covoiturage comme Green Cove, Ecolutis, LaRoueVerte ou encore covoiturage.fr. D'autres ont plusieurs compétences et le covoiturage en fait partie ; c'est le cas de la société LB Mobilité (service en management de la mobilité) et de Hippocampe (agence de conseil en communication). Ce sont des sociétés qui utilisent leurs savoir-faire pour multiplier leurs références et diversifier leur activité.

Les sites de covoiturage sont considérés comme des systèmes statiques étant donné qu'il faut consulter et réserver à l'avance le trajet qu'on va faire. Il s'agit en quelque sorte de sites de rendez-vous là où le conducteur et les passagers se mettent d'accord sur le lieu et le temps de rendez-vous ainsi que le coût du trajet. En France, jusqu'à 2007, il existe plus de 78 sites de covoiturage à accès public (Ballet, et al., 2007). Ce nombre de sites ne cesse d'augmenter. Cependant, ils restent des sites de rencontre et manque de flexibilité.

Récemment, en parallèle, un nouveau concept est entrain de se développer : le covoiturage dynamique. Il présente un fort potentiel de développement du fait de la souplesse de service qu'il vise à apporter grâce à ses grands principes : le temps réel, l'optimisation des trajets et la garantie d'un service fiable (Legrand, et al., 2009).

Covoiturage dynamique

Le covoiturage dynamique se caractérise par une grande souplesse d'utilisation et moins d'interdépendance que le covoiturage « classique ». Le fonctionnement est le même que le covoiturage classique sauf qu'on peut trouver une course dans un délai rapide (moins d'une demi-heure par exemple). L'accès au service et son mode de fonctionnement doivent être faciles et souples. Sous réserve d'une inscription préalable, le trajet peut être proposé quasiment instantanément et traité en quelques minutes seulement : l'organisation de ce service en temps réel et en réseau est rendue possible grâce aux nouvelles technologies. En effet, avec les smartphones, les PDAs, les tablettes PC, etc. munis de technologies BlueTooth, GPS, GPRS... il est devenu possible de bénéficier d'applications mobiles intégrées grâce à des systèmes embarqués sur son propre support. Ces technologies révolutionnaires sont très

pertinentes pour la mise en œuvre de mesures de performances et d'efficacité dans un contexte de mobilité continue. Ceci étant, des fonctionnalités telles que la géolocalisation instantanée et continue, l'Internet mobile, la communication en temps réel, etc. jadis impensables, sont devenues aujourd'hui banalisées avec la réussite explosive de ces technologies.

Depuis quelques années, des expérimentations de covoiturage dynamique voient le jour, notamment, aux États-Unis et en Europe occidentale.

Aux Etats Unis, deux systèmes de covoiturage dits dynamiques ont vu le jour : Goloco et Ride Now. GoLoco¹ est un site de covoiturage associé au réseau social Facebook et Ride Now² était une expérimentation de covoiturage dynamique qui s'est déroulée de Novembre 2005 à Mai 2006 au niveau de la station Dublin/Pleasanton du métro de San Francisco. D'autres systèmes de covoiturage dynamique ont été testés comme EasyRider³ en Hollande, T.ecovoiturage⁴ en France ou Carlos⁵ en Suisse.

Par ailleurs, malgré l'avancement technologique et les efforts déployés, ces expérimentations sont restées au stade embryonnaire à cause de failles de sécurité, d'automatisation ou à cause d'aspects d'optimisation leur faisant défaut. D'autant plus qu'ils sont pratiquement similaires à ceux précédemment décrits, à la seule différence qu'ils présentent l'avantage de consulter en temps réel la liste des offres des véhicules en circulation. L'intégration de services de géolocalisation est à la base de la possibilité de mise en place de tels services. Les fonctionnalités supplémentaires des services de covoiturage dynamique ainsi définis se résument globalement à la récupération de données GPS actualisées et de les fournir à leurs abonnés.

Dans le sens du covoiturage dynamique, les travaux effectués par (Sghaier, 2011) ont mené à la proposition d'un système de covoiturage dynamique optimisé *CODAC* (Covoiturage Optimisé Dynamique basé sur les Agents Communicants). D'autres recherches ont aussi été menées par d'autres chercheurs partout dans le monde et ont abouti à des résultats satisfaisants (Calvo, et al., 2004) (Yan, et al., 2011) (Stach, 2011) (Bai, et al., 2007).

¹ <http://www.goloco.org>

² http://www.ridenow.org/4113_ACCMADynamicRidesharing.pdf

³ <http://www.ecolane.com>

⁴ <http://www.covoiturage-campus.com>

⁵ <http://www.carlos.ch/>

I.5.2. L'autopartage

L'autopartage est un service de mobilité qui consiste à disposer d'une voiture « à la carte », moyennant une adhésion préalable au service et réservation avant chacune des prises d'un véhicule (Clavel, et al., 2008). Les membres d'un service d'autopartage se partagent donc l'usage d'un parc de véhicules successivement pour des déplacements de courte durée et de faible distance. L'autopartage est un service économique et pratique pour les utilisateurs, et écologique pour la collectivité⁶.

L'Autopartage permet ainsi la réduction du nombre de véhicules en circulation (effets positifs sur la qualité de l'air, le confort sonore et l'espace public). Il s'agit d'un service complémentaire aux autres modes de transports favorisant l'intermodalité et la co-modalité avec une simplicité et souplesse d'utilisation. En plus de son avantage environnemental en réduisant les émissions des gazs à effet de serre en général et le CO2 en particulier, il possède un avantage économique que ce soit pour l'individu ou les collectivités.

I.5.2.1. Fonctionnement de l'autopartage

Grâce à cette pratique, tout individu ayant un permis de conduire et qui s'est procuré ce droit peut disposer d'une voiture à volonté et sans restriction délaissant ainsi le souhait ou le besoin d'en acquérir une pour ses déplacements personnels. L'autopartage est donc défini comme un service de mobilité prodiguant à ses usagers le droit de disposer d'une voiture « à la carte » moyennant une adhésion préalable lui donnant la possibilité d'accéder au service. Pour profiter des fonctionnalités offertes par ce dernier, tout usager doit effectuer une réservation (par téléphone, via Internet...) à l'avance pour pouvoir prendre un véhicule. Il s'agit plus précisément d'un partage de parcs de véhicules par les membres du service. L'ensemble des voitures de la flotte est ainsi disponible à une prise autonome moyennant une carte à puce utilisée pour débloquer un boîtier contenant les clés ou un ordinateur embarqué sur la voiture (Clavel, et al., 2008).

Outre le coût d'adhésion, la tarification se fait selon l'usage du véhicule. Une formule basée sur la durée d'utilisation (temps) et la distance parcourue (kilomètre) a ainsi été mise en place pour la tarification du service selon l'utilisation des abonnés. Cette formule s'avère dans la plupart des cas beaucoup plus avantageuse et moins coûteuse pour les individus que les frais engendrés par l'acquisition d'une voiture personnelle (prix du véhicule, assurance, carburant, etc.). Par ailleurs, des mesures spécifiques ont été prises pour encourager les individus à aller vers ce mode de transport, telles que la gratuité des frais de parking.

⁶ <http://www.franceautopartage.com/>

Un service d'autopartage compte en moyenne 8 véhicules/employé. La durée moyenne d'un trajet est de 5 heures, l'utilisation principale se faisant durant le week-end. La distance moyenne par trajet, pour un service donné, est généralement comprise entre 50 et 60 km. Elle peut atteindre 90 km ou ne pas dépasser 5 km dans certains cas qui sont rares. Enfin, nous avons un ratio moyen de 15 adhérents/véhicule d'autopartage, sur l'ensemble des services existants pour l'année 2008 (Clavel, et al., 2008).

I.5.2.2. Autopartage : Historique et systèmes existants

L'autopartage a vu le jour en 1948 à Zurich avec le premier club de conducteurs qui cotisaient pour acheter une voiture. Le concept d'auto-partage traverse l'Atlantique en 1983. La première organisation américaine fut Mobility Entreprise dans le cadre d'un programme de recherche universitaire en 1983 dans l'Indiana. C'est à partir de 1999 que la France voit s'implanter l'auto-partage durablement. Ainsi, ce mode de déplacement est très développé depuis plusieurs années dans de nombreux pays, en Europe et ailleurs : Belgique, Allemagne, Suisse, Italie, Espagne, Angleterre, Canada et Etas Unis.

Les travaux de recherche réalisés jusqu'aujourd'hui peuvent être subdivisés en deux catégories suivant les critères de gestion des réservations adoptés au sein du système. La première catégorie se base sur une gestion statique des réservations alors que la deuxième traite l'aspect dynamique (i.e. en temps réel).

En 2008, le nombre de structures qui proposent un service d'autopartage en France s'élève à 19, principalement dans des villes de plus de 100 000 habitants (Figure I-8).



Figure I-8 Villes disposant d'un ou plusieurs services d'autopartage (2008)

En France, des services d'autopartage ont été mis en place grâce à certains travaux et projets. C'est ainsi que des systèmes tels que City Roul' à Rennes, Lilas à Lille, Mobilib à Toulouse, Autolib' à Lyon, etc. ont intégré ce réseau pour engendrer une expansion très remarquable de ce concept en France allant jusqu'à s'installer dans les petites provinces. En parallèle, l'autopartage a aussi été très développé dans le monde entier : l'Europe , les états unis, la Grande Bretagne, etc. Parmi les systèmes qui relèvent de la gestion statique des réservations, on peut citer : mobility qui est le leader du marché de l'autopartage en Suisse, Communauto au Canada, CityCarClub en Angleterre, CambioStadt dans la ville de Bremen en Allemagne, Autopia et Cambio qui sont les deux alternatives d'autopartage en Belgique, ZipCar et I-GO aux états unis, etc.

Par ailleurs, en matière de réservations dynamiques, quelques systèmes existent mais la plupart d'entre eux sont restés à un stade embryonnaire n'intégrant que très partiellement l'aspect temps réel dans leurs services. Ayant développé un concept innovant de nouvelle mobilité urbaine de type *VIP* pour Véhicules Individuels Publics sans obligation de stationnement dans des sites dédiés à cet effet ; le projet le plus en vue aujourd'hui en France est celui baptisé Cité Vu implémenté à Antibes et réalisé par la société Vu Log. Dans le

même contexte de gestion dynamique des réservations, nous pouvons aussi citer le projet Autolib implanté à Paris.

I.6. Les systèmes d'Information du transport co-modal

Cette partie est dédiée à l'état de l'art des systèmes d'information du transport co-modal. Il s'agit de décrire les systèmes existants et de définir leurs limites pour enfin aboutir à notre proposition : notre système d'information pour le transport co-modal.

I.6.1. Définition d'un système d'information

Un Système d'information (SI), est défini par l'ensemble des éléments pour la gestion, le stockage, le traitement, le transport et la diffusion de l'information dans une institution. Il offre les fonctionnalités suivantes :

- Une aide à la régulation des services même en cas de perturbations.
- Une aide à la billetterie
- Une aide à la maintenance
- Une aide à la sécurité des employés et des clients
- Une aide à l'information voyageur.

Dans le domaine du transport, un Système d'Information d'Aide à l'Exploitation (*SIAE*) regroupe les fonctionnalités citées ci-dessus et intègre spécialement l'aide à la régulation des services en cas de perturbation, l'aide à la billetterie et l'aide à l'information des voyageurs. Nous nous intéressons à cette dernière fonctionnalité qui correspond à un module particulier d'un SIAE, appelé, Système d'Information d'Aide au Déplacement (*SIAD*).

Ce module vise à assister les clients de transport à consulter l'information dont ils ont besoin pour les aider à se déplacer.

Actuellement, les systèmes d'information destinés aux clients peuvent être monomodaux, multimodaux ou co-modaux.

I.6.2. Systèmes d'information monomodaux

Ce sont des systèmes à un seul exploitant qui ne concernent qu'un seul mode de transport. Nous pouvons citer pour le cas du transport en commun, le système InfoBus⁷ de la ville de Metz. Ce système est ajusté à un Système d'Aide à l'Exploitation et à l'Information (SAEI) qui indique en temps réel la position exacte des autobus à un poste central de régulation. Nous

⁷ www.mairie-metz.fr

pouvons également parler des systèmes des deux opérateurs Tadao⁸ et Fibleu⁹ qui sont des systèmes pour les réseaux des bus correspondants aux deux villes Lens et Tours.

A côté du transport public, nous pouvons citer les systèmes existants tels que les sites web « covoiturage.fr » ou « 123envoiture » qui proposent uniquement un seul service de transport qui est le covoiturage. De même pour l'autopartage, les systèmes d'informations de Lilas à Lille ou autolib à Paris sont des systèmes monomodaux puisqu'ils ne proposent que des flottes de voitures pour les partager.

I.6.3. Systèmes d'information multimodaux

En général, nous insinuons par systèmes d'information multimodaux les systèmes de transport en commun. En effet, ce sont des systèmes qui offrent différents modes tels que le métro, train ou tramway... Ces systèmes en général ne concernent qu'un seul opérateur et donc sont mono-opérateur. Ces systèmes se présentent sous forme de site web. Ils offrent en plus des informations usuelles (horaires, disposition des stations dans une carte) et un calculateur d'itinéraire interne. Il s'agit d'un moteur de recherche d'itinéraire accédant aux données locales d'un seul exploitant. Parmi les exploitants proposant de tels sites nous pouvons citer Transpole¹⁰ pour la métropole Lilloise et RATP¹¹ pour l'île de France.

Tous ces systèmes demeurent mono-opérateur. De plus en plus de projets sont mis en place dans le but d'intégrer les données issues de plusieurs opérateurs. Les deux projets les plus connus sont le projet allemand « DELFI » et le projet du Royaume Uni « Journeyweb ».

Ces deux projets sont basés sur une approche répartie qui vise l'intégration des systèmes d'information d'aide au déplacement existants.

- *DELFI*¹² : Initié en 1996 en France, le projet 'DELFI' « Durchgängige Elektronische FahrplanInformation » vise à intégrer les différents calculateurs d'itinéraires locaux des différents opérateurs relatifs aux divers états ('landers') de la République Fédérale allemande. Pour déterminer un itinéraire, DELFI commence par déterminer les deux calculateurs locaux relatifs à l'origine et à la destination. Les pôles d'échanges entre les opérateurs sont calculés et fournis par le serveur de l'opérateur National. Un calculateur principal va concaténer les offres recueillies pour composer les différentes alternatives de la requête globale.

⁸ www.tadao.fr

⁹ www.fibleu.fr

¹⁰ www.transpole.fr

¹¹ www.ratp.fr

¹² www.delfi.de

*JourneyWeb*¹³ : Ce projet a été initié en 2000 par le ministère de transport public Anglais pour la réalisation d'un système national d'information multimodale. JourneyWeb utilise un protocole de communication spécialement mis en place pour le projet. Il vise à faciliter l'échange des données entre les systèmes d'information des différents opérateurs. Ainsi, chaque opérateur serait capable de répondre à des requêtes globales relatives à plusieurs opérateurs.

JourneyWeb exploite deux bases de données :

- NPTG (National Public Transport Gazetteer) : La NPTG est une base de données nationale regroupant l'ensemble des villes et des villages pour les différentes régions. Par ailleurs, elle inclut les pôles d'échange nationaux qui permettent de basculer d'un réseau régional vers le réseau national.
- NaPTAN (National Public Transport Access Node) : il s'agit aussi une base de données nationale, répertoriant l'ensemble des points d'accès (arrêts et stations) au réseau de transport d'un opérateur local, et ce, pour les différentes régions.

Afin de répondre à une requête d'itinéraire globale, chaque calculateur est capable de rechercher un itinéraire d'un point de départ appartenant à la zone qu'il dessert vers un pôle d'échange national permettant l'accès au réseau incluant la destination. Ceci suppose que chaque calculateur local dispose d'une carte des opérateurs nationaux, information fournie par le NPTG et le NaPTAN. Cette information est ensuite concaténée à l'itinéraire de l'opérateur d'arrivée constituant ainsi une réponse à la requête globale.

D'autres projets ont traité le problème du transport multimodal mais avec une approche différente qui vise à intégrer les données de chaque opérateur dans le système d'information.

Nous pouvons citer le projet suisse « *GOFAS* » qui est composé de deux sous-systèmes :

- Infopool qui permet l'importation des données horaires et géographiques à partir des bases de données des différents opérateurs et de les stocker dans une même base de données centralisée couvrant tout le territoire national.
- Internet-Gis qui permet de calculer les itinéraires à partir de la base de données centralisée.

I.6.4. Systèmes d'information co-modaux

Tel que son nom l'indique, un système d'information co-modal doit offrir aux voyageurs des itinéraires co-modaux avec différents modes de transport public ou privé. Plusieurs travaux

¹³ www.journeyweb.org.uk

ont été établis dans ce sens. Cependant, dans la littérature nous trouvons plusieurs chercheurs qui proposent des systèmes de transport appelés multimodaux mais qui combinent plusieurs modes de transport (voiture, vélo, métro, train...). En se basant sur la définition d'un système de transport co-modal, nous considérons qu'effectivement ces travaux sont inclus dans le cadre des systèmes d'information co-modaux.

Mettant le focus sur la complémentarité des modes de transport, plusieurs travaux ont été effectués dans le but d'organiser les voyages des clients et de leur fournir des itinéraires qui peuvent être monomodaux ou co-modaux optimisés selon différents critères.

(Zhang, et al., 2011) proposent un système qui fournit aux voyageurs avant leur départ un itinéraire qui peut être composé de quatre modes de transport possibles : le bus, le vélo, la voiture ou la marche à pied. Il a utilisé l'algorithme de Dijkstra pour la résolution de son hypergraphe en optimisant le temps de parcours.

De même pour (Ayed, et al., 2010), qui ont traité le problème de transport co-modal en considérant différents modes de transport en proposant une approche basée sur un algorithme parallèle appliqué sur un graphe de transfert.

Un assistant personnel de voyage a été élaboré par (Rehrl, et al., 2007) pour suivre et assister en temps réel chaque voyageur avant, pendant et après son voyage grâce à la technologie de géolocalisation et la technologie mobile. Le système est très intéressant spécialement pour les personnes à mobilité réduite puisqu'il lui indique la direction qu'il faut suivre pour prendre l'ascenseur par exemple.

D'autres travaux ont été effectués dans ce sens aussi mais visant une optimisation multiobjective par (Modesti, et al., 1998).

En Allemagne, une application RHURPILOT¹⁴ a été appliquée à Ruhr dans le but de proposer des itinéraires utilisant le transport en commun et la voiture. Un autre système a vu le jour en Suède à Stockholm en 2009 (Green and ITS, 2009). Ce système permet la combinaison de tous les modes de transport y compris la voiture particulière. La marche à pied est également considérée comme un mode de transport.

Tous les systèmes d'information co-modaux se trouvent devant la situation de traiter d'énormes données distribuées appartenant à différents services et différents opérateurs de transport. Pour cela, il est possible de procéder par :

- Une architecture distribuée qui vise à intégrer les services existants pour la production de l'information co-modale.

¹⁴ <http://www.ruhrpilot.de/?jsessionid=C2A1385A8DB822AAAC6C1093A6B59CE8>

– Une architecture centralisée qui vise à intégrer les données des différents opérateurs.

Dans les deux cas, chaque opérateur dispose d'un Système d'Information d'Aide au Déplacement. Généralement, ces systèmes et spécialement ceux du transport public disposent d'une base de données (BD) sur laquelle ils exécutent un Algorithme de Calcul d'Itinéraires (ACI) permettant de déterminer le meilleur chemin.

La base de données décrit, via un ensemble d'informations statiques, la globalité du réseau en termes de stations, d'horaires de passages, de tarifs, de lignes et de ressources disponibles. L'ACI exploite cette base de données locale pour trouver le ou les itinéraires qui conviennent au mieux à la requête de l'utilisateur.

Pour produire l'information co-modale, il est nécessaire de procéder à une intégration de l'information locale à chaque opérateur de chaque service. Deux stratégies d'intégration sont alors à considérer. La première consiste à intégrer les différentes bases de données dans une même base globale et centralisée et la deuxième consiste à intégrer les différents SIAD existants d'une manière automatique.

Approche centralisée :

La première approche tend à regrouper et uniformiser toutes les données suivant un seul et unique modèle relationnel. Une telle approche peut être avantageuse si l'on n'exécute qu'un seul ACI global sur une seule base pour la composition d'itinéraires. Cette stratégie implique une uniformisation des données hétérogènes selon un standard donné et entraîne des contraintes relatives à la synchronisation de la base. En effet, il faudra prendre en compte la différence des données générées par différents opérateurs et différents services de transport sans oublier les changements effectués par les différents opérateurs sur leurs bases respectives et mettre à jour la base de données centrale. Cela a de fortes chances de dégrader les performances du système spécialement si le nombre d'opérateurs à inclure dans le système est assez important. Par ailleurs, pour la mise en place et l'exploitation de cette base, il faut prévoir assez d'espace pour l'ensemble des données et s'attendre à un temps de réponse considérable.

Approche distribuée :

L'idée de cette approche est d'aller chercher l'information ciblée sur le SIAD correspondant. Une telle stratégie entraînerait des communications supplémentaires et donc, à première vue, une dégradation des performances. Cependant, les ACI s'exécutent sur des bases de données moins importantes que celles dans la stratégie centralisée et par conséquent, le temps de réponse de la base est amélioré. Cette approche est mieux adaptée à la distribution des

systèmes étant donné que les requêtes s'exécutent sur des bases de données différentes. On pourra ainsi gagner en performance tout en évitant la mobilisation de toute la base. En effet, si l'on considère plusieurs requêtes d'itinéraires, il est possible d'exécuter ces requêtes de façon séquentielle, car on mobilise la base pour chaque requête. Dans l'approche distribuée, les différentes requêtes peuvent concerner des SIAD différents et par conséquent on pourra traiter plusieurs requêtes en parallèle. De même, l'approche distribuée ne nécessite pas de synchronisation des données ce qui fait qu'elle est moins coûteuse.

I.7. Limites des systèmes existants et proposition d'un système de transport co-modal

Bien que l'utilisation de la voiture soit en régression, elle demeure le mode de transport dominant en Europe. L'impact sur l'environnement est tangible, car elle provoque, en amont, la surexploitation des ressources pétrolières et augmente, en aval, le taux des émissions des GES. Une première politique européenne allait dans le sens d'une mise en concurrence des transports en commun vis-à-vis de la voiture particulière en essayant ainsi de diminuer l'utilisation de cette dernière. Mais les limites de cette politique furent vite atteintes. En effet, l'utilisation de la voiture occupe toujours la plus grande part du transport urbain et ceci malgré les différents avantages que proposent les transports en commun. Il est donc plus logique d'aller dans le sens actuel du mouvement qui tend, au lieu de concurrencer les véhicules particuliers, à profiter de leur maniabilité, leur confort tout en préservant l'environnement. Un objectif réalisable si l'on exploite efficacement la voiture particulière, en encourageant, par exemple, une utilisation collective de cette dernière. La notion de multimodalité est dès lors remplacée par celle de co-modalité, mise en avant par la nouvelle politique du transport urbain en Europe. En effet, il n'est plus question de mettre en concurrence les transports en commun et la voiture particulière, il s'agit plutôt de trouver une complémentarité entre les différents modes. Cette complémentarité peut se traduire par une composition d'itinéraire utilisant des tronçons en voiture et des tronçons en transport en commun. De plus, afin de minimiser la pollution émise par la voiture, il est nécessaire d'augmenter le nombre de personnes transportées durant le voyage et par la suite réduire le nombre de voitures en circulation.

Une étude des systèmes d'information de transport existants a montré que nous pouvons distinguer trois types de systèmes : monomodaux, multimodaux et co-modaux. Les systèmes monomodaux traitent chaque mode de transport dans son contexte, c'est le cas des systèmes proposant des services de covoiturage ou d'autopartage par exemple.

D'autres systèmes qui traitent les informations de transport en commun de différents opérateurs à la fois sont considérés comme des systèmes multimodaux. Quant aux systèmes co-modaux, nous avons vu que plusieurs chercheurs partout dans le monde s'intéressent de plus en plus à ce type de transport. La majorité des travaux ont proposé des systèmes qui offrent des itinéraires qui combinent à la fois le transport public et le transport privé. Par privé, ils entendent particulièrement l'usage de la voiture privée.

Cependant, nous avons constaté qu'il n'existe pas de systèmes co-modaux qui intègrent les services basés sur la notion de la voiture partagée en complément avec les autres modes de transport.

Concernant les approches proposées par les systèmes précédents, une panoplie de méthodes d'optimisation ont été adoptées. La recherche et composition d'itinéraires est généralement résolue grâce à l'algorithme de Dijkstra (Zhang, et al., 2011) ou bien une version modifiée de ce dernier. D'autres travaux, ont adopté des approches hybrides en combinant les algorithmes de recherche des plus courts chemins et une métaheuristique. Les métaheuresistiques généralement appliquées sont les colonies de fourmis (Ayed, et al., 2010) ou les algorithmes génétiques.

Proposition d'un système d'information de transport co-modal

En pratique comme en théorie, le véhicule partagé est un concept riche en apport. De lourde incidence touchant à plusieurs aspects, le covoiturage et l'autopartage ont des portées positives sur l'environnement, les comportements des individus, leurs relations sociales, l'économie, les budgets, etc. et méritent donc d'être considéré avec grand intérêt. Leur combinaison et complémentarité avec les systèmes de transport déjà existant font le succès du transport co-modal offrant la satisfaction des usagers. Le concept de la co-modalité accapare aujourd'hui une attention bien particulière doublée d'un grand intérêt pour l'évolution dans ce domaine. Nos travaux de recherche s'inscrivent dans ce cadre et tentent une poussée technologique pour essayer de gagner du terrain par rapport aux travaux recensés dans la littérature. Nous proposons de ce fait un **Système d'Information de Transport Co-modal (SITCoMo)** qui englobe tous les services de transport. Nous préférons parler de services de transport plutôt que de modes de transport puisque un service peut offrir plusieurs modes de transport. En effet, nous considérons tout service de transport existant actuellement, comme le transport en commun qui est un service qui offre plusieurs modes tel que le bus, le métro...

D'autres services de transport proposés sont le service de covoiturage et celui des véhicules en libre service que ce soit des vélos, ou des voitures grâce à l'autopartage.

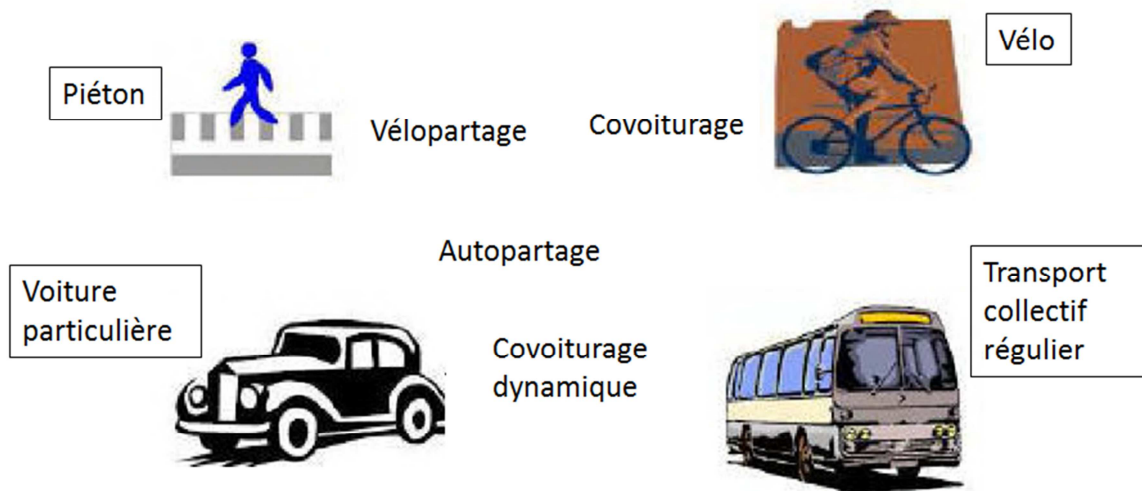


Figure I-9 Services de transport offerts par SITCoMo

Ainsi, en plus de la combinaison du transport public et privé, nous offrons à notre système une flexibilité et une efficacité en intégrant aussi des services basés sur la notion du véhicule partagé. Nous ne pouvons pas passer à côté de tels services qui n'arrêtent pas de s'imposer dans notre vie actuelle et à long terme.

En vue de répondre au mieux aux attentes des usagers du service et de respecter la vraie définition du transport co-modal, l'optimalité de ce dernier réside aussi bien dans leur satisfaction par rapport aux réponses fournies. Dans ce cadre, la prise en compte de différents critères d'optimisation tels que le temps total du trajet, le coût et les émissions des gaz à effet de serre s'impose.

Joint au concept de l'optimisation, l'aspect dynamique et distribué du système nous pousse à profiter de l'apport des systèmes multi-agents dans ce domaine. Une alliance optimisation et systèmes multi-agent a été adoptée dans ce travail et qui va être expliquée dans les prochains chapitres.

I.8. Conclusion

L'intégration du transport co-modal dans la vie quotidienne est devenue une nécessité à cause des problèmes de congestion, problèmes environnementaux et sociaux. En plus des transports en commun, les systèmes basés sur le partage de véhicules aident à l'amélioration des conditions environnementales dans lesquelles évoluent les individus. Les concepts de covoiturage et d'autopartage présentent de multiples avantages, dont la réduction du nombre

de voitures en circulation au kilomètre, la minimisation du taux d'émissions de CO2 et GES aussi bien la réduction des budgets alloués au transport.

Dans ce sens, nos travaux s'articulent autour d'un Système d'Information de Transport Co-modal (SITCoMo) qui respecte la notion de co-modalité et offre aux utilisateurs une combinaison optimale de plusieurs services et modes de transport en respectant leurs préférences et leurs contraintes.

En se basant sur les travaux et projets existants, nous avons remarqué que jusqu'à aujourd'hui certes la co-modalité intéresse les chercheurs de plus en plus, mais ces systèmes restent toujours limités. A travers notre système, nous essayons de faire partie de ces nombreux travaux et d'apporter un plus au domaine de la recherche et de la composition d'itinéraires dans un système distribué. Il intègre le concept d'optimisation multiobjective dans une alliance avec les Systèmes Multi-Agent (SMA). Cette alliance vise la mise en place de traitements efficaces dans un système performant fournissant une qualité de service optimale.

Avant de proposer nos approches de résolution basées sur des agents communicants, toutes les notions de systèmes multi-agents et d'optimisation ainsi que leur alliance sont détaillées dans le chapitre II.

II. Chapitre II : Alliance Systèmes multi-agents et optimisation au service du transport co-modal

II.1. Introduction

Dans le chapitre précédent, nous avons évoqué un état de l'art sur les systèmes d'information de transport co-modal existants. Néanmoins, un système d'information pour le transport co-modal et les différentes techniques de composition et d'intégration de l'information, se basent sur une méthodologie de modélisation et une approche de résolution.

Nous présentons donc dans ce chapitre, la méthodologie multi-agents de modélisation ainsi que les techniques d'optimisation existantes dans le domaine du transport.

Dans le reste de chapitre, nous allons commencer par argumenter le choix de l'architecture que nous allons adopter à savoir l'architecture à base d'agents et nous introduisons le lien entre les systèmes multi-agents et les méthodes d'optimisation, en évoquant ces deux domaines complètement différents dans le même chapitre. Nous procédons par la suite par un état de l'art sur les systèmes multi-agents : définitions, caractéristiques, typologies, classifications en démontrant les avantages de ce paradigme pour les systèmes d'information et en présentant quelques travaux dans le domaine du transport se basant sur le concept SMA. Ensuite, nous nous intéressons aux techniques d'optimisation existantes de même à quelques cas d'emploi des méthodes d'optimisation dans le domaine du transport. Avant de conclure ce chapitre, nous allons démontrer l'intérêt d'associer et d'allier les concepts de SMA et optimisation.

II.2. Quelle architecture et quelle approche de résolution choisir ?

Nous nous intéressons à la conception et la mise en place d'un système d'information de transport co-modal. La co-modalité étant définie comme « la combinaison optimale des différents modes sur la chaîne de transport », elle met l'accent sur l'importance et la nécessité d'inclure l'optimisation dans un tel système. En effet, l'apport d'un système d'information de transport co-modal est non seulement d'offrir tous les différents modes de transport existants privé ou public mais aussi de fournir des itinéraires optimaux en termes de temps, coût et d'émission des gaz à effet de serre. Mais la question qui se pose : quelle approche d'optimisation faut-il adopter pour garantir une efficacité et une meilleure qualité de service ?

Le système doit pouvoir accéder en temps réel aux systèmes d'information de tous les services de transport existants et s'adapter automatiquement à leur offre de service. Ensuite il faut pouvoir appliquer des algorithmes de recherche d'itinéraires et d'optimisation complexes tout en gérant l'intégration à chaud de nouveaux SI.

Pour cela, il faut penser à une architecture qui facilite la communication du système avec son environnement et même la communication entre les différentes entités du système lui-même. En plus, le système ne doit pas être affecté et doit être fiable en cas de perturbations. En effet, il doit permettre une inscription et une désinscription automatique des SIAD sans avoir à modifier le code source. Sans oublier évidemment, le calcul complexe qu'il doit effectuer pour la recherche d'itinéraires co-modaux et le calcul des meilleures solutions en termes de trois critères : le temps, le coût et les émissions des GES.

Pour mettre en place un tel système, une autre question se pose alors : quelle technologie ou méthodologie choisir ?

Dans le domaine du génie logiciel, de nouveaux concepts structurels et architecturaux sont apparus vu la demande croissante en systèmes d'information et en logiciels de plus en plus complexes. Parmi les architectures les plus répandues, nous trouvons les architectures à base de composants, de services et d'agents.

- Architecture à base de composants

La technologie à base de composants est une architecture extraite des approches orientées objets (Chardigny, et al., 2007) (Booch, 1992). En se basant sur le concept du circuit électrique, elle consiste à considérer le logiciel informatique comme un ensemble de plusieurs composants. Selon (Chardigny, et al., 2007), « un composant est un élément logiciel qui est composable sans modification, peut être distribué de manière autonome, encapsule une fonctionnalité, et qui adhère à un modèle de composant ». Il comporte des interfaces d'entrée et de sortie qui lui permettent d'interagir avec d'autres composants.

La programmation à base de composants trouve son efficacité grâce à la réutilisation. La notion de composant vient combler les limites de l'approche objet en termes de granularité de réutilisation en assurant ainsi une meilleure sécurité de la réutilisation des composants (Szypersky, 1998) (Meijler, et al., 1998). En effet, contrairement à un objet qui risque de faire des réutilisations non sécurisées (suite à des appels à des services externes sans spécification explicite), un composant ne peut utiliser que des services bien spécifiés compte tenu de sa conception et des précisions sur ses interfaces.

- Architecture orientée services

L'architecture orientée services (ou Service Oriented Architecture, SOA) est une nouvelle organisation applicative permettant une interaction entre des composants distants de l'application à travers des services. Il s'agit d' « une manière d'organiser les applications logicielles isolées à travers une mise en place d'une infrastructure accueillant un ensemble de services interconnectés. Chaque service étant accessible au travers de standards et de protocoles d'échange de message. » (Rouillard, et al., 2007)

La notion du service peut être représentée ici par un objet produit par un fournisseur et consommé par un client. La SOA propose un nouveau concept facilitant l'échange de messages, réutilisable et avec une bonne sécurité (utilisation de protocoles standardisés). Le service n'étant pas exclusivement un service Web, mais peut-être tout type de service respectant un/des protocole(s) et une description précise mise à la disposition du client. Par exemple les WSDL (pour Web Services Description Language).

- Architecture à base d'agents

Le concept agent est une évolution de l'intelligence artificielle distribuée née de la fusion entre les systèmes distribués et l'intelligence artificielle (Moulin, et al., 1996).

Un agent est « une entité autonome capable de communiquer, disposant de connaissances et d'un comportement privé ainsi que d'une capacité d'exécution propre. Un agent agit pour le compte d'un tiers (un autre agent, un utilisateur) qu'il représente sans être obligatoirement connecté à lui » (Leriche, 2006). L'association des connaissances de plusieurs agents et leur coopération permet de résoudre les problèmes complexes et de donner plus d'élan dans les capacités de résolution.

Afin de choisir l'architecture la mieux adéquate à notre problématique nous allons nous baser sur la synthèse de comparaison effectuée par (Feki, 2010) entre les différentes architectures selon la figure II-1.

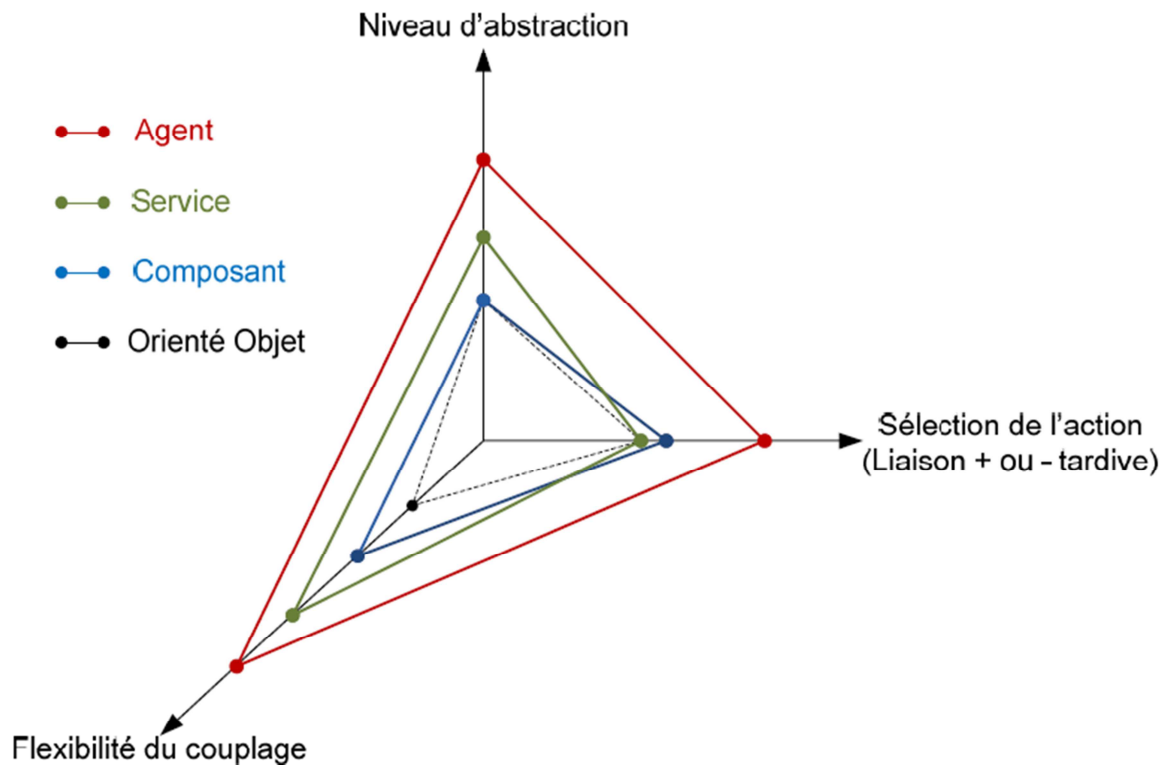


Figure II-1 Comparaison des différentes architectures

Dans sa comparaison, (Feki, 2010) se base sur trois axes : le niveau d'abstraction qui est défini par le niveau d'expression des concepts offerts aux concepteurs et aux programmeurs. Le deuxième axe concerne la sélection de l'action qui identifie quand et comment l'entité logicielle active l'action du code correspondant. Finalement, le troisième axe tourne autour de la flexibilité du couplage définie par la capacité à mettre en relation plusieurs entités logicielles.

D'après (Feki, 2010), l'architecture Composant et celle orientée objet se rapprochent avec un léger devancement de l'architecture composant au niveau de la « sélection de l'action » et « Flexibilité du couplage ». Cependant, son « niveau d'abstraction » demeure limité. L'architecture Orientée Service présente un niveau plus élevé d'abstraction et une meilleure flexibilité de couplage, mais pas d'innovation dans la « Sélection de l'action ». Finalement, l'architecture multi-agent est la plus évoluée sur les trois axes. Sa faiblesse existe peut-être ailleurs, au niveau de la mise en place ou des performances.

En se basant sur le système de transport co-modal décrit auparavant et sur la comparaison des différentes architectures, nous pouvons déduire que la meilleure architecture à adopter est celle à base d'agents. En effet, vu que nous sommes confrontés à des transferts de message intra-système et inter-systèmes, l'architecture agent est la plus développée en matière de

transfert de message. Contrairement à l'architecture composant et service, on n'a pas besoin de connaître l'interface de son interlocuteur.

Notre problématique nous impose également l'exécution d'un algorithme dans un environnement distribué. Dans ce genre de contexte, différentes entités doivent exécuter des instructions indépendamment des autres et communiquer entre elles par la suite pour regrouper les différents résultats dans une réponse globale. L'architecture est la plus adéquate pour ce genre de système complexe et distribué.

Ainsi, nous choisissons d'utiliser une architecture multi-agents alliée avec des algorithmes de recherche d'itinéraires co-modaux et des algorithmes d'optimisation multi-critères afin de garantir en temps réel aux clients des itinéraires co-modaux optimaux selon leurs préférences et leurs contraintes.

Avant de décrire notre approche pour la résolution du problème, nous allons par la suite introduire les différentes propriétés de l'architecture multi-agents avec un état de l'art des travaux à base d'agents dans le domaine du transport. L'optimisation avec ses différentes définitions et méthodes de résolution sera aussi détaillée dans ce chapitre.

II.3. Les Systèmes Multi-Agents

Vu les besoins des nouveaux systèmes d'information distribués, hybrides et hétérogènes, à caractère dynamique et à sémantique riche, variée et évolutive, le domaine de l'Intelligence Artificielle (IA) prend de plus en plus sa place. Il vise à simuler les capacités de l'être humain tels que le raisonnement, le langage naturel pour la communication et l'apprentissage. L'évolution de ce domaine devient alors d'un très grand intérêt à résoudre des problèmes plus complexes, plus réalistes et de plus grande échelle qui dépassent les capacités d'un agent artificiel unique, dont les capacités sont limitées par ses propres connaissances, ressources de calcul et perspectives. D'où la création d'organisations artificielles de résolution de problèmes, qui sont capables d'interagir, d'apprendre et d'évoluer conjointement. Le génie logiciel se dirige donc vers une conception en termes d'unités autonomes distribuées en interactions pour résoudre des problèmes indépendants : il s'agit du domaine de l'Intelligence Artificielle Distribuée (IAD). Faisant partie de ce domaine, les SMA sont de plus en plus utilisés dans plusieurs disciplines de recherche : communication, santé, production, etc. et plus particulièrement les systèmes logistiques

II.3.1. Définition d'un agent

Dans la littérature, plusieurs définitions ont été attribuées à la notion d'agent. La définition proposée par (Flores-Mendez, 1999) reste la définition la plus explicite et basée sur d'autres définitions.

« Un agent est une entité interactive qui existe en tant que partie d'un environnement partagé par d'autres agents. Il s'agit d'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. »

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

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

Ceci permet de ressortir le cycle de vie d'un agent : Perception de l'environnement en utilisant les connaissances et les comportements, prise de décision de l'action suivante en exploitant les connaissances et l'intelligence et enfin l'exécution de l'action choisie (Figure II-2).

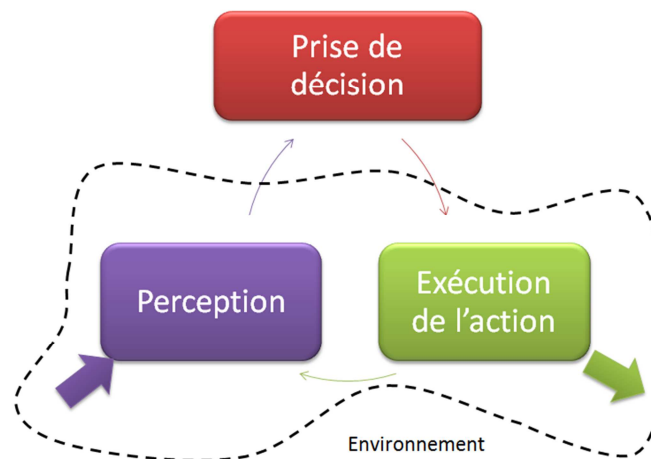


Figure II-2 Cycle de vie d'un agent

II.3.2. Propriétés d'un agent

Les principales propriétés d'un agent sont son autonomie, sa réactivité, sa proactivité et sa sociabilité :

- l'*autonomie* : Un agent est autonome s'il est capable de prendre des initiatives et agir sans l'intervention d'autres agents et notamment de l'être humain.
- la *réactivité* : Un agent est réactif s'il est capable de percevoir son environnement et de réagir automatiquement en fonction des changements discernés.
- la *proactivité* : Un agent est proactif si, en plus de répondre aux événements, il possède des processus internes qui lui permettent de prendre des initiatives pour atteindre ses propres buts et objectifs dont il a la capacité de se fixer.
- la *sociabilité* : Un agent est sociable s'il est capable d'interagir avec d'autres agents pour atteindre leurs buts respectifs.

Un agent peut être aussi *cognitif* quand il raisonne avant d'agir. Il est souvent associé au trio bouclant : perception-raisonnement-action. Il possède, en plus de ses buts, des notions psychologiques qui peuvent être exprimés par le biais des attitudes mentales comme les croyances, les intentions et les désirs.

Nous trouvons également des agents stationnaires ou mobiles selon leur capacité à parcourir les réseaux informatiques et de migrer d'une station vers une autre pour exécuter un code qu'il transporte. Un agent *stationnaire* est dépourvu de mobilité. Cet agent agit localement, pendant tout son cycle de vie, dans la machine là où il a été implanté initialement. Cependant,

un agent *mobile* est capable de se déplacer à travers un réseau, d'un nœud à un autre pour agir à son propre compte ou à la demande d'un autre agent. C'est un paradigme de plus en plus utilisé dans le contexte des réseaux largement distribués.

II.3.3. Définition d'un système multi-agents

Un système multi-agent est « un système composé d'un ensemble d'entités autonomes et intelligentes qui coordonnent leurs connaissances pour atteindre un objectif ou résoudre un problème (Weiss, 1999) ».

Selon (Ferber, 1995), un SMA est un système qui se compose d'un environnement, d'un ensemble d'objets passifs et d'un ensemble d'agents actifs. Les objets sont reliés entre eux par un ensemble de relations définies et sont manipulés par les agents qui les perçoivent (création, modification et destruction).

Les connaissances d'un seul agent ne sont pas assez suffisantes pour définir un système complet. Dans un SMA, l'aspect d'organisation, communication, interaction et environnement est très important. Les différentes entités d'un SMA cohabitent dans un environnement en commun. Cette cohabitation implique l'utilisation d'une organisation particulière qui facilite et canalise la communication entre les entités. Dans cette organisation, un rôle est attribué à chaque agent définissant son degré de collaboration avec les autres agents. Ainsi les tâches globales sont décomposées en tâches élémentaires exécutées par l'ensemble des agents.

La puissance d'un SMA réside dans l'intelligence globale du système. Dans un contexte particulier, le changement de l'organisation pourrait faciliter ou accélérer la résolution d'un problème. On dit qu'un SMA est auto-organisé lorsqu'il s'adapte à son environnement en changeant d'organisation. Ce changement d'organisation obéit à des règles particulières (Serugendo, et al., 2004) et constitue un aspect de la prise de décision du système entier et non pas d'un agent en particulier.

II.3.3.1. Environnement dans un système multi-agents

L'environnement symbolise le monde dans lequel les agents évoluent. Il n'est pas juste un environnement de déploiement avec des serveurs, une plateforme de déploiement, un réseau de communication et tout le matériel utilisé pour exploiter les agents, mais bien plus que ça. L'environnement d'un SMA représente une dimension de conception à part entière qui peut intégrer une grande partie de la complexité du système principalement en termes de service.

De ce fait, l'environnement du SMA contient aussi tout type de système, objet, service en interaction avec le SMA lui-même (Viroli, et al., 2006).

Un environnement peut être statique ou dynamique, déterministe ou non déterministe, discret ou continu, accessible ou non accessible (Hanon, 2006).

Lorsque l'état de l'environnement ne dépend que ses états antérieurs et des actions réalisées par le système, il est dit environnement *statique* (*dynamique* par opposition).

L'environnement d'un système est *déterministe* si une action du système sur cet environnement a un effet unique et certain. L'environnement d'un acteur virtuel peut donc être non déterministe, par exemple, deux acteurs virtuels ne peuvent pas prédire la réaction d'un objet sur lequel ils agissent simultanément.

Si l'ensemble des perceptions et l'ensemble des actions possibles sur cet environnement sont finis, l'environnement est *discret*. Dans le cas contraire, il est *continu*.

L'environnement d'un système est *accessible* si ce système peut être perçu, à chaque instant, d'une manière complète et précise.

II.3.3.2. La communication et l'interaction dans un système multi-agent

Les interactions, telles que définies par (Doniec, 2006) sont une relation dynamique instaurée entre plusieurs agents du fait de leurs actions combinées et réciproques. Ces interactions sont essentielles et très importantes dans les organisations. Elles constituent un concept inhérent et indissociable de celui des SMA. En effet, étant donné que la sociabilité est une priorité principale d'un agent, il faut modéliser ses comportements sociaux et déterminer ses interactions avec les autres membres qui constituent un SMA.

Une communication est entretenue entre une société d'agents pour assurer ces actions. Il ne s'agit pas seulement d'échanges de messages mais aussi d'échanges de données et d'objets tout en obéissant à des règles, des protocoles et des ontologies.

L'interaction peut avoir plusieurs formes : la coopération, la coordination, la négociation, la collaboration, etc (Weiss, 1999).

Pendant leur communication, les agents se trouvent devant deux possibilités : soit ils s'entraident et donc coopèrent pour essayer d'atteindre l'objectif global en évitant les situations de conflits, soit se retrouvent dans une situation de compétition qui implique une phase de négociation.

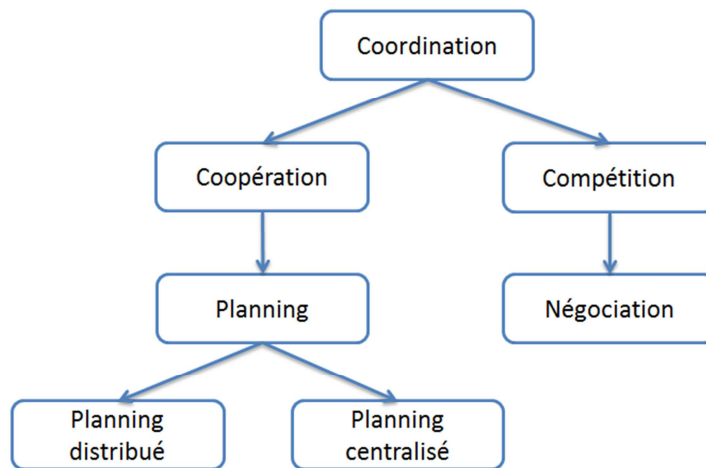


Figure II-3 Différentes formes d'interactions

Comme le montre la figure II-3, les différentes formes d'interactions sont :

- *La Coopération* : les agents coopèrent lorsqu'ils sont motivés collectivement pour atteindre un but global. Dans ce cas, n'étant pas dans une situation de concurrence, les agents essaient donc de s'accommoder sans se déranger. Pour coopérer, les agents collaborent pour partager les tâches et les ressources.
- *La Négociation* : chaque agent est motivé individuellement pour atteindre son propre objectif, en dépit de ceux des autres agents, mais en essayant de conserver certaines propriétés au niveau du groupe. La négociation est définie comme étant un processus grâce auquel plusieurs parties aboutissent successivement à une décision commune, après avoir diffusé initialement, leurs demandes respectives (Verrons, 2004). Ces demandes initiales convergent donc petit à petit, vers un accord final, par une suite de concessions ou par la recherche de nouvelles alternatives (Pruitt, 1981).

II.3.3.3. Organisation Multi-Agents (OMA)

L'organisation d'un système multi-agent définit l'architecture globale du système. Elle définit les différentes entités inter-reliées qui composent le système, leurs rôles, les relations qui les rassemblent, les règles d'interaction entre elles, leurs activités ainsi que les buts et les critères d'évaluation de ces activités (R.Mandiau, et al., 1999).

Les différentes typologies des organisations dans les systèmes multi-agents peuvent être classées en deux catégories : typologie relationnelle et typologie structurelle (Adam, 2000) (Mandiau, et al., 2002).

La typologie relationnelle dépend de la typologie des interactions. Cette classification distingue les OMA à structure hiérarchique de celles à structure hétérarchique :

- Les *OMA* à *structure hiérarchique* : un agent de niveau supérieur, « chef hiérarchique» possède le contrôle et est le seul à donner des ordres à des exécutants de plus bas niveau, ce sont donc des structures rigides qui poursuivent un seul objectif opérationnel par niveau hiérarchique.

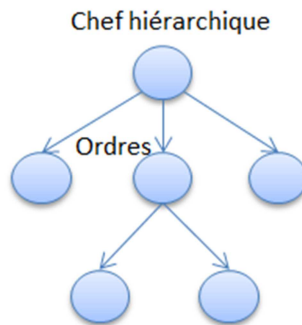


Figure II-4 OMA à structure hiérarchique

- Les *OMA* à *structure hétéroarchique* : ce type de structure se décompose, à son tour, en trois catégories de types *marché*, *communauté* et *société* :
 - Les organisations de type *marché* : dans ce type d'organisation le contrôle est partagé sur plusieurs agents coordinateurs hiérarchiques qui visent des objectifs opérationnels différents.

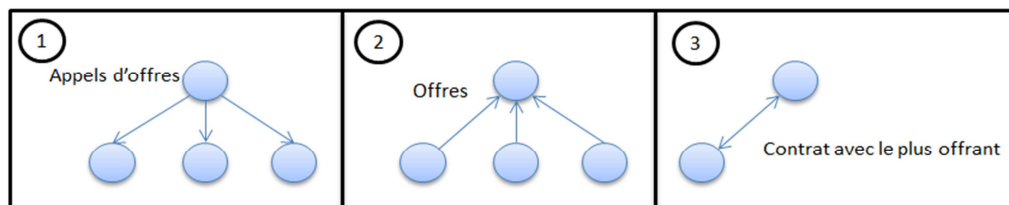


Figure II-5 OMA à structure de marché

- Les organisations de type *communauté* : c'est une structure fortement décentralisée qui ne présente aucune forme de hiérarchie car tous les agents sont de même type, ont les mêmes capacités et partagent le même niveau de contrôle. Dans ce type d'organisation, il n'y a pas de règles d'interaction, il s'agit donc d'échanges de flux de données.

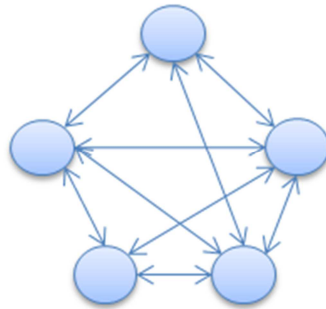


Figure II-6 OMA à structure de communauté

- Les organisations de type *société* : il s'agit d'un croisement entre les organisations hiérarchiques et les organisations de communauté, en définissant précisément les règles d'interaction. Le contrôle est partagé par plusieurs unités qui n'ont pas nécessairement les mêmes capacités. *Adam* (Adam, 2000) parle d'ajustement mutuel entre les différentes unités par des principes de négociation.

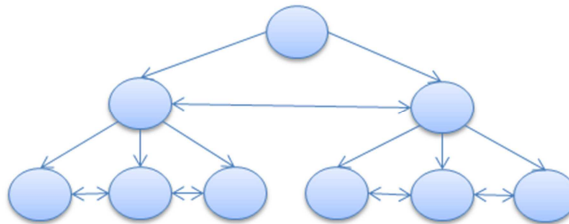


Figure II-7 OMA à structure de société

Quant à la typologie *structurelle*, elle fait la différence entre une organisation prédéfinie (tout est connu à l'avance) et une organisation découverte à posteriori (structure émergente) résultant des interactions entre agents. Cette classification rejoint celle de *Mandiau* (Mandiau, et al., 2002) qui classe les organisations selon leurs processus de formation et distingue alors deux modèles d'organisation :

- Organisation *émergente* : cette organisation est définie progressivement avec l'évolution du système à travers les interactions inter-agents ;
- Organisation *support d'activités* : cette organisation est définie dès la conception du système. Elle impose le fonctionnement du contrôle global du système. Cette structure organisationnelle précise également pour chaque agent, sa fonction dans l'organisation, ses compétences ainsi que les possibilités d'interactions le concernant.

II.3.4. Les SMA dans le domaine du transport

Le recours des chercheurs et industriels aux plateformes multi-agents pour la mise en place de systèmes performants et efficaces montrent le grand succès des SMA dans plusieurs domaines. En effet, les SMA se sont imposés par :

- Leur capacité d'adaptation à n'importe quel type de problème
- L'intégration et la gestion aisée d'une quantité importante d'informations distribuées et hétérogènes faisant évoluer la littérature vers le génie logiciel distribué et l'intelligence ambiante qui règne aussi efficacement sur un ensemble d'applications réparties qu'au sein d'une application unique.
- La prise de décision collective à travers des mécanismes de coordination et collaboration assurées par la communication, échanges et interactions entre les agents.
- Etc.

Différentes études dans plusieurs domaines ont ainsi émergé intégrant ce concept pour la résolution de problèmes variant de complexes à difficiles de point de vue architectural ou autre. Nous nous intéressons au domaine du transport qui a été influencé également par l'efficacité des SMA. Parmi les problèmes inhérents à ce domaine, nous pouvons citer la recherche d'itinéraires, les systèmes d'information et systèmes d'aide au déplacement des voyageurs, les services liés au transport, la gestion du trafic, l'ordonnancement et logistique pour la gestion des places de parking ou de circulation dans un carrefour, etc. les problématiques sont multiples et le nombre de travaux ayant émergé est très important. Traitant d'optimisation pour améliorer la qualité des services offerts ou tout autre concept, différents problèmes de transport ont été traités avec le paradigme agent que ce soit du transport aérien (Ljungberg, et al., 1992) ou terrestre (Chaib-draa, 1995) (Fayech, 2003) (OuldSidi, 2006) (Durfee, et al., 1991) (Chaib-draa, 1996), transport de marchandises (Fischer, et al., 1999) ou transport des personnes (Feki, 2010) (Kamoun, 2007) (Zidi, 2006).

Concernant le transport des personnes, en plus des travaux établis dans notre équipe par (Feki, 2010) (Kamoun, 2007) (Zidi, 2006) (Zgaya, 2007) pour l'optimisation, la recherche et la composition des services liés au transport ainsi que les itinéraires multimodaux, (Fayech, 2003) et (OuldSidi, 2006) pour les problèmes de régulation des réseaux de transport multimodal et (Sghaier, 2011) pour la résolution du problème de covoiturage dynamique, d'autres travaux et d'autres systèmes ont vu le jour dans ce sens.

PerSyst (Anli, et al., 2006) est un système de personnalisation de l'information multimodale. Il s'agit d'une plate-forme à base d'agents logiciels, pouvant servir de support pour la

conception de Système de Personnalisation. PerSyst facilite la gestion et la transmission des résultats ou des recommandations dans un système distribué et évolutif.

D'autres chercheurs (Belgueliel, et al., 2012) se sont intéressés aussi à l'amélioration de la qualité de service offert aux usagers du transport en commun. Il s'agit essentiellement d'améliorer la qualité de l'information de transport en termes de facilité d'accès, de disponibilité et d'efficacité, pour aider les clients à se déplacer moyennant un système d'information voyageur. Ils proposent ainsi un travail s'appuyant sur la théorie multi-agents qui vise à concevoir un système d'information voyageur d'aide au déplacement multimodal. Ce travail a pour but d'utiliser une approche de composition des services web pour calculer l'itinéraire le plus court en termes de coût, de temps de parcours et de confort en vue d'offrir une assistance pour les voyageurs.

Un projet plus poussé, le projet Terra Dynamica¹⁵ 2010-2012, vise à développer les technologies permettant d'animer une ville virtuelle réaliste en la peuplant d'habitants et de véhicules réalisant leurs activités habituelles mais aussi capables de réagir aux événements et de s'adapter aux évolutions de l'environnement de manière crédible. Les défis technologiques sont nombreux, citons entre autres le haut niveau d'interactivité visé et le nombre élevé d'agents à gérer, ce qui oriente vers des architectures d'agents autonomes modélisés avec des granularités variables (individu, groupe, foule).

La problématique de covoiturage n'a pas été épargnée non plus, donnant lieu à l'incorporation du concept des SMA dans des systèmes de covoiturage. Les SMA ont pour effet notoire de faire croître l'autonomie des composantes et acteurs intervenant dans de tels systèmes, facilitant par la même occasion l'intégration de plusieurs plateformes dédiées (Giorgini, et al., 2006) (Burmeister, et al., 1997) (Bellemans, et al., 2012). Afin de mieux décrire ces systèmes, dans le domaine du covoiturage en particulier, nous citons à titre d'exemple les travaux qui ont tourné autour de ce concept pour la mise en place d'une architecture multi-agent au service des fonctionnalités intégrées dans les systèmes de type support classique basé sur le web (Kothari, 2004). Les fonctionnalités principales à noter dans ce type de structures concernent essentiellement la gestion des données de covoiturage dont dispose le système mais aussi la recherche, à la place des usagers, d'offres correspondantes à leurs demandes.

Aussi, parmi les systèmes proposés dans la littérature, nous pouvons citer la plateforme MobiAgent (Carabelea, et al., 2005) où le paradigme agent a été introduit pour bénéficier d'une facilité d'accès à certains services (e.g. recherche dans le web, applications de contrôle

¹⁵ <http://www-poleia.lip6.fr/~corruble/TerraDynamica/accueil.html>

à distance, etc.). Ces services seront désormais accessibles grâce au concept agent pour tout utilisateur disposant d'un smartphone ou *PDA*.

Nous remarquons que parmi les systèmes cités ci-dessus, nombreux sont ceux qui considèrent les problèmes d'optimisation que ce soit au niveau de l'information multimodale ou modale ou bien au niveau de la recherche des itinéraires par exemple. En effet, le concept optimisation est un concept très important et très large dans le domaine du transport.

II.4. Optimisation

Dans divers secteurs, les ingénieurs sont contraints de faire face à des problèmes technologiques de complexité grandissante tel est le cas par exemple de l'électronique, le traitement des images ou dans la recherche opérationnelle. Le problème à résoudre peut fréquemment être exprimé sous la forme générale d'un problème d'optimisation. Un problème d'optimisation se définit comme la recherche de l'optimum (minimum ou maximum) d'une fonction f donnée, appelée fonction objectif ou fonction de coût ou encore critère d'optimisation. Ce besoin d'optimiser découle de la nécessité de fournir à l'utilisateur un système qui puisse répondre au mieux à ses exigences.

II.4.1. Définition d'un problème d'optimisation

Un problème d'optimisation consiste à trouver, parmi un ensemble donné appelé ensemble des variables de décision, un élément minimisant ou maximisant une fonction donnée appelée fonction objectif. On parle de problème d'optimisation contraint lorsque les variables de la fonction à optimiser sont contraintes d'évoluer dans un espace de recherche restreint (Colette, et al., 2002).

Un problème d'optimisation revient à minimiser une fonction objectif $f(\vec{x})$ en prenant en compte m contraintes d'inégalité et p contraintes d'égalité. L'ensemble des contraintes diminue et délimite l'espace de recherche de la solution optimale.

$$\left| \begin{array}{l} \text{minimiser } f(\vec{x}) \text{ (fonction à optimiser)} \\ \text{avec } \vec{g}(\vec{x}) \leq 0 \text{ (} m \text{ contraintes d'inégalités)} \\ \text{et } \vec{h}(\vec{x}) = 0 \text{ (} p \text{ contraintes d'égalité)} \end{array} \right.$$

Avec $\vec{x} \in \mathfrak{R}^n$, $\vec{g}(\vec{x}) \in \mathfrak{R}^m$ et $\vec{h}(\vec{x}) \in \mathfrak{R}^p$.

Ainsi, les termes « *Minimum global* », « *Minimum local fort* » et « *Minimum local faible* » peuvent être définis de la manière suivante :

Définition 1 : Le point \vec{x}^* s'appelle le *minimum global* de la fonction f tel que

$$f(\vec{x}^*) < f(\vec{x}) \quad \forall \vec{x}^* \neq \vec{x}$$

Définition 2 : Le point \vec{x}^* s'appelle le *minimum local fort* de la fonction f quand il existe un voisinage $V(\vec{x}^*)$ tel que $\vec{x} \in V(\vec{x}^*)$ tel que

$$f(\vec{x}^*) < f(\vec{x}) \quad \forall \vec{x}^* \neq \vec{x}$$

Définition 3 : Le point \vec{x}^* s'appelle le *minimum local faible* de la fonction f quand il existe un voisinage $V(\vec{x}^*)$ tel que $\vec{x} \in V(\vec{x}^*)$ tel que

$$f(\vec{x}^*) \leq f(\vec{x}) \quad \forall \vec{x}^* \neq \vec{x}$$

Nous prenons l'exemple de la Figure II.8. Dans cette figure, P_3 est un minimum global, P_2 et P_4 sont deux minimum locaux forts et P_1 est un minimum local faible.

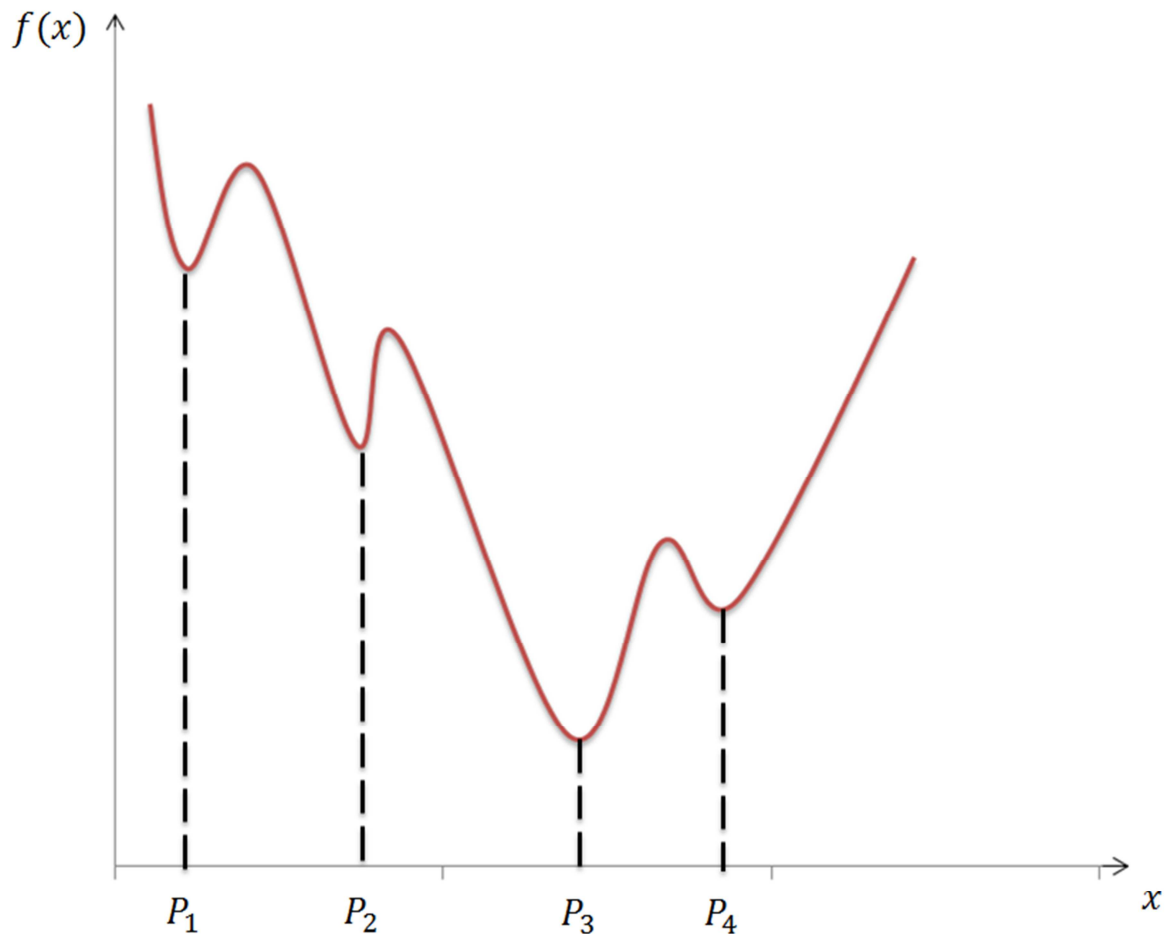


Figure II-8 Les différents minima

II.4.2. Complexité des problèmes d'optimisation

Les méthodes de résolution des problèmes d'optimisation dépendent de la complexité du problème. Un problème est dit *polynomial* s'il existe un algorithme efficace permettant de trouver une solution optimale pour toutes ses instances en un temps polynomial par rapport à

la taille de l'instance. Cependant pour la majorité des problèmes d'optimisation combinatoire aucun algorithme polynomial n'est connu pour le moment. En effet, la plupart des problèmes étudiés en optimisation appartiennent à la classe des problèmes NP-difficiles (Basseur, 2005). 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" (Colette, et al., 2002). 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, 2005). Nous pouvons citer l'exemple du problème de voyageur de commerce (*Traveling Salesman Problem*) qui est le plus connu dans la classe du problème NP-complets où il s'agit de minimiser la longueur d'un circuit hamiltonien dans un graphe pondéré (Dréo, et al., 2003). Il existe une quantité innombrable de problèmes de ce type dans la littérature.

II.4.3. 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).

Dans un problème d'optimisation combinatoire (f), il est très difficile d'envisager une méthode qui énumère les solutions du problème car le nombre des solutions est très grand, bien qu'il soit fini. 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 sont généralement *NP*-difficiles. Plusieurs méthodes ont été introduites durant les cinq dernières décennies pour résoudre les problèmes d'optimisation combinatoire.

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.4.4. Optimisation multi-objectif

Dans un problème d'optimisation multi-objectif, on ne cherche pas à optimiser une fonction objectif unique mais un vecteur de fonctions. Cette extension est d'autant plus nécessaire que de plus en plus de problèmes nécessitent la considération de manière simultanée d'une multitude d'objectifs. L'optimisation multicritère consiste donc à choisir parmi un ensemble infini d'alternatives une seule (ou plusieurs) lorsqu'une multitude de critères doivent être considérés (Othmani, 1998). L'ensemble de ces alternatives varie généralement dans un domaine continu. Depuis plus de quatre décennies déjà, la problématique multicritère n'a pas arrêté d'évoluer témoignant de la naissance et du développement d'une grande panoplie de méthodes. Deux groupes distincts ont émergé pour réaliser la classification de celles-ci ; le premier est constitué des méthodes qui s'appuient sur un critère unique de synthèse, lesquelles sont dites *classiques* ; alors que le deuxième inclue des méthodes réactives au sens où elles intègrent un processus interactif de décision (Roy, et al., 1993).

Les méthodes *classiques* sont des méthodes de programmation linéaire qui sont efficaces lorsque la fonction objectif et les contraintes s'expriment linéairement en fonction des variables de décision.

Formellement, la définition d'un problème d'optimisation multiobjectif reprend celle d'un problème d'optimisation classique tout en considérant un ensemble de fonctions objectifs regroupées dans $\vec{f}(\vec{x})$ telle que $\vec{g}(\vec{x}) \leq 0$ et $\vec{h}(\vec{x}) = 0$ avec

$$\left| \begin{array}{l} \vec{x} \in \mathfrak{R}^n : n \text{ variables de décision} \\ \vec{f}(\vec{x}) \in \mathfrak{R}^k : k \text{ fonctions objectif} \\ \vec{g}(\vec{x}) \in \mathfrak{R}^m : m \text{ contraintes d'inégalité} \\ \vec{h}(\vec{x}) \in \mathfrak{R}^p : p \text{ contraintes d'égalité} \end{array} \right.$$

Un problème d'optimisation multicritère présente l'avantage de tolérer les degrés de liberté qui faisaient défaut à l'optimisation monoobjectif. Par ailleurs, cette flexibilité n'est pas sans

conséquence sur l'espace de solutions qui sont passées d'une seule à plusieurs et qui dépendent fortement de la démarche d'optimisation suivie. En effet, les objectifs étant souvent contradictoires, optimiser un objectif peut influencer négativement sur un ou plusieurs autres. De ce fait, une solution optimale n'existe pas dans l'absolu, nous parlons plutôt, dans ce contexte, de solutions « *optimisées* ». En effet, il est quasi-impossible, dans la majorité des cas, d'optimiser tous les critères en même temps et il est indispensable, par conséquent, de privilégier un ou plusieurs par rapport à d'autres. C'est dans ce cadre que le concept de compromis a été introduit, dans le sens où certains critères s'en verront privilégiés (optimisés) au détriment d'autres dont la qualité de performance s'en serait réduite dans l'ensemble des solutions finalement extraites. Par ailleurs, ces solutions ne sont jugées utiles et adéquates qu'au sens de l'utilisateur sur lequel se basera le décideur pour la modélisation de la fonction objectif. En effet, ces solutions feront l'objet d'une étape de sélection constituant l'arbitrage final de l'utilisateur.

L'ensemble des solutions ainsi extraites sont dites *solutions de Pareto* et constituent la *surface de compromis*. Une solution est jugée intéressante lorsqu'elle présente une relation de dominance par rapport aux autres solutions, on dit que \vec{x}_1 domine \vec{x}_2 si :

\vec{x}_1 est strictement meilleur que \vec{x}_2 dans au moins un objectif

\vec{x}_1 est au moins aussi bon que \vec{x}_2 dans tous les objectifs

Grâce à cette définition, la relation de dominance sert à filtrer les mauvais éléments pour faire émerger seules les solutions ne pouvant être comparées entre elles. Plusieurs critères distinguent les solutions ainsi retenues leur faisant valoir le qualificatif de *solutions optimales au sens de Pareto* (ou encore *solutions non dominées*) si elles ne se dominant pas entre elles mais dominant les autres ; de

On parle de *solutions optimales localement au sens de Pareto* si elles sont optimales au sens de Pareto sur une restriction de l'ensemble \mathfrak{R}^n .

Il existe dans la littérature un large panel de méthode d'optimisation multi-objectif (Colette, et al., 2002). Le choix de la méthode à appliquer doit être adéquat selon le problème d'optimisation auquel l'on se trouve confronté. La sélection de la méthode d'optimisation peut se faire parmi trois grandes familles (Veldhuizen, 1999) distinctes. Nous retrouvons ainsi :

- Les méthodes d'optimisation a priori : le compromis est fixé avant l'application de la méthode d'optimisation.

- Les méthodes d'optimisation progressive : le compromis est fixé en interagissant avec le décideur au fur et à mesure de l'exécution de la méthode d'optimisation. Ceci afin de pouvoir orienter la recherche vers les zones auxquelles le décideur est le plus sensible et qui satisfont le compromis qu'il souhaite opérer entre les fonctions objectifs.
- Les méthodes d'optimisation a posteriori : où après la recherche d'une sélection d'une large panoplie de solutions, différentes et bien réparties, qui se fera pour être exposée au décideur ; ce dernier pourra ainsi choisir la solution qui lui convient le plus par rapport au reste. Toutes ces méthodes seront détaillées dans le chapitre IV.

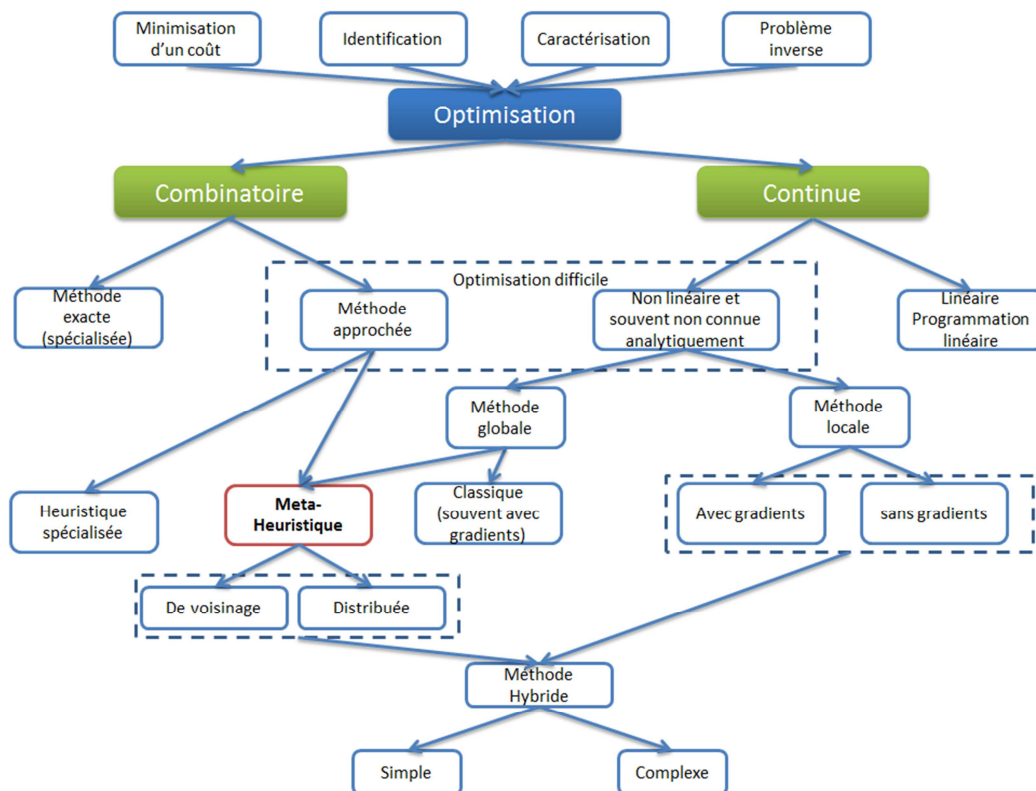


Figure II-9 Méthodes d'optimisation monobjectif

Parallèlement à l'optimisation continue, les efforts déployés dans le domaine de l'optimisation combinatoire ont abouti au développement d'un grand nombre d'*heuristiques*. Celles-ci ont pour mission principale de produire des solutions proches de l'optimum et ont été conçues pour répondre de manière spécifique à un problème bien déterminé.

Afin d'avoir plus de flexibilité, chercheurs et praticiens ont été amenés à considérer le développement de techniques génériques capables de s'adapter à n'importe quel domaine. Dans le but de réaliser cette généricité des méthodes, les efforts déployés dans ce sens ont mené à la conception d'une nouvelle classe de méthodes d'optimisation, nommées « *métaheuristiques* ». Ces dernières marquent une réconciliation entre les deux domaines

d'optimisation continue et combinatoire puisqu'elles peuvent s'appliquer à toute sorte de problème à variables discrètes aussi bien que continues.

Plusieurs méthodes de résolution existent pour les problèmes d'optimisation multiobjectif. Ces méthodes sont classées en cinq groupes (Leriche, 2006) :

- Les méthodes scalaires ;
- Les méthodes interactives ;
- Les méthodes floues ;
- Les méthodes d'aide à la décision ;
- Les méthodes exploitant une métaheuristique ;

II.4.4.1. Les méthodes scalaires

Ces méthodes ont pour but de convertir le problème multi-objectif à résoudre en un problème d'optimisation monobjectif, dont il existe de nombreuses méthodes de résolution. Il suffit de déterminer une fonction objectif qui somme toutes les fonctions objectif, en amplifiant chaque fonction par un coefficient de pondération.

II.4.4.2. Les méthodes interactives

Ce sont des méthodes progressives qui ne permettent de chercher qu'une seule solution. Elles permettent à l'utilisateur de déterminer ses préférences vis-à-vis d'un compromis entre objectifs. Ces préférences peuvent être choisies avant ou après le lancement de l'optimisation.

II.4.4.3. Les méthodes floues

La logique floue peut être intégrée pour la résolution des problèmes d'optimisation multi-objectif. Grâce à la logique floue il est possible de traiter l'incertitude et l'imprécision des connaissances humaines ainsi que les transitions progressives entre états, ce qui n'est pas le cas pour la logique classique. Avec cette dernière, tout est décrit en termes de VRAI ou FAUX.

II.4.4.4. Les méthodes d'aide à la décision

Ce sont les seules méthodes de résolution de problèmes d'optimisation multi-objectif qui ne travaillent que sur des ensembles discrets de points et qui ne soient pas basées sur des relations de dominance. Cette méthode génère un ensemble de solutions grâce à l'établissement d'une relation d'ordre entre les différents éléments.

II.4.4.5. Les méthodes exploitant une métaheuristique

Ces méthodes génèrent un ensemble de solutions grâce à l'établissement d'une relation d'ordre entre les différents éléments.

Les métaheurstiques ont initialement été proposées dans la recherche dédiée aux problèmes d'optimisation difficile. Le principal avantage de ces méthodes, constituant aussi leur principale source d'efficacité, se trouve être leur capacité à éviter les pièges des minima locaux, et ce contrairement aux méthodes d'optimisation classiques.

II.4.5. Les métaheurstiques

Dans (Blum, et al., 2003), les auteurs résument les propriétés attachées à la notion de métaheuristique.

Les métaheurstiques :

- permettent d'explorer efficacement l'espace de recherche afin de trouver une solution proche de l'optimalité.
- possèdent des techniques qui vont de la simple recherche locale à des procédures complexes d'apprentissage.
- sont des algorithmes de résolution incertains et souvent non déterministes.
- peuvent intégrer des mécanismes évitant d'être piégé dans une zone de l'espace de recherche.
- sont décrites suivant un niveau d'abstraction indépendant du problème spécifique à traiter.
- peuvent encapsuler les informations spécifiques au problème sous la forme de sous-heurstiques contrôlées à un niveau supérieur.
- introduisent des mécanismes pour adapter et guider la recherche dynamiquement. Il s'agit d'approches adaptatives et autoadaptatives.

Nous décrivons dans ce qui suit les différents types de métaheurstiques.

II.4.5.1. L'approche de recherche locale

Cette approche représente une évolution des méthodes classiques d'amélioration itérative en acceptant des solutions voisines moins bonnes que la solution courante pour échapper aux optima (minima ou maxima) locaux. Les méthodes de l'approche de recherche locale commencent à partir d'une solution réalisable x_0 , choisie arbitrairement de l'ensemble des solutions réalisables X . Ces méthodes fonctionnent avec une seule configuration courante à la fois, actualisée au cours des itérations successives. Les différentes versions de cette approche

varient selon le choix de la solution voisine et le critère d'arrêt. Le passage d'une solution réalisable à une autre se fait selon un ensemble de modifications élémentaires qui dépend de la résolution adoptée. Le voisinage d'une solution x , noté par $N(x)$ est défini comme l'ensemble des solutions réalisables, atteignables depuis x en effectuant des modifications élémentaires. La solution x' voisine à x est donc choisie telle que $f(x') = \min_{x'' \in N(x)} f(x'')$. Les méthodes les plus connues sont : la méthode Tabou, le recuit simulé et les méthodes d'acceptation à seuil.

- *La méthode taboue ou la recherche taboue*

Le principe de la recherche taboue est de garder une trace des solutions récemment visitées dans une liste taboue. Cette liste est utilisée lors des déplacements dans le voisinage de manière à ne pas revenir sur des solutions déjà visitées. Pour cela, lors de la recherche locale, le voisinage de la solution courante est réduit aux solutions ne faisant pas partie de la liste taboue. Les déplacements réalisés consistent alors à sélectionner la solution de meilleur coût dans le voisinage. Il faut remarquer que celle-ci est choisie quand bien même elle possède un coût supérieur à la solution courante, de manière à pouvoir s'échapper des optima locaux. La liste taboue ne conserve que les solutions visitées les plus récentes. Elle est nommée en conséquence "mémoire à court terme".

| Algorithme Recherche taboue |
|--|
| Initialiser la liste T (T vide) x_0 point de départ N : nombre d'itération maximal $i=0$ $x_c=x_0$ (solution courante) Pour $i < N$ répéter $E = N(x_c) \setminus T$ (les voisins de x_c sans les éléments de T) soit $y \in E$ et y meilleur solution dans E $T = T \cup \{y\}$ $x_c = y$ $i = i + 1$ Fin de répéter |

Figure II-10 Algorithme Recherche Taboue

- *Le recuit simulé*

Le recuit simulé a fait son apparition en optimisation combinatoire en 1983 dans (Kirkpatrick, et al., 1983). Il s'agit d'une approche métaheuristique parmi les plus anciennes. Elle est fondée sur une analogie avec un phénomène de physique portant sur le refroidissement de matériaux. L'idée fondamentale est d'accepter des déplacements dans le voisinage d'une

solution en dépit du fait qu'ils dégradent le coût de la solution et cela suivant une probabilité calculée à partir d'une température T du système à un moment donné. L'algorithme débute avec l'initialisation d'une solution courante s à une température T . À chaque itération une nouvelle solution s_0 est prise aléatoirement dans le voisinage de la solution courante. Cette nouvelle solution s_0 est acceptée comme solution courante si son coût est inférieur à celui de la solution courante ($f(s_0) < f(s)$ dans le cas d'un problème de minimisation) sinon l'acceptation de la solution dépend d'une probabilité calculée à partir de la température T et de la différence de coût $f(s_0) - f(s)$. Durant la recherche, la température T décroît, simulant un phénomène de refroidissement. Ainsi, au début la probabilité d'accepter une solution de moins bonne qualité est élevée, puis progressivement le processus devient une recherche locale classique acceptant uniquement des mouvements améliorant la solution.

Algorithme Recuit simulé

X : Ensemble des solutions réalisables et $N(x)$: Le voisinage d'une solution réalisable $x \in X$

Paramètres :

n : Nombre d'itérations d'un cycle
 a : Coefficient de refroidissement
 t_0 : Température initiale du système

Variables :

x, x' et x^* solution courante, solution voisine et meilleure solution
 k compteur d'itérations global
nouveau_cycle booléen qui indique s'il faut effectuer un nouveau cycle d'itérations
nbCycle compteur d'itérations, interne à un cycle
 t température courante du système
 q probabilité générée uniformément dans l'intervalle $[0,1[$

1. Initialisation :

Choisir une solution initiale réalisable $x \in X$
 $x^* = x, k = 0, \text{nouveau_cycle} = \text{vrai}, t = t_0$

2. Processus itératif :

Tant que *nouveau_cycle* est vrai faire :

nbCycle = 0, *nouveau_cycle* = faux

Tant que *nbCycle* < n faire :

Incrémenter k et *nbCycle* de 1

Générer aléatoirement une solution $x' \in N(x)$

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

Si $\Delta f < 0$ alors faire

$x = x', \text{nouveau_cycle} = \text{vrai}$

Sinon

$\text{prob}(\Delta f, t) = \exp(-\Delta f / t)$

Générer q uniformément dans l'intervalle $[0,1[$

Si $q < \text{prob}(\Delta f, t)$ alors faire

$x = x', \text{nouveau_cycle} = \text{vrai}$

| |
|--|
| $\text{Si } f(x) < f(x^*) \text{ alors faire}$ $x^* = x$ $t = a.t$ |
|--|

Figure II-11 Algorithme Recuit simulé

- Les méthodes d'acceptation à seuil

Elles représentent une dérivation directe de l'algorithme du Recuit Simulé. Ces méthodes décident de l'acceptation d'une moins bonne solution sans avoir recours aux principes du recuit thermodynamique. Plusieurs versions existent et varient selon deux caractéristiques : une fonction auxiliaire et un seuil. Ce dernier peut dépendre de l'évaluation de la meilleure solution trouvée par la fonction objectif. Le Recuit Simulé a donné de très bons résultats pour plusieurs problèmes, souvent de grandes tailles mais ses limites se présentent dans les temps de calcul qui peuvent devenir très importants, ce qui a conduit à des parallélisations de la méthode. Les inconvénients résident également dans les « réglages » pour gérer la décroissance de la température. Ces réglages demandent une expérience et un savoir faire de la part des utilisateurs.

II.4.5.2. L'approche constructive

L'idée de base d'une approche constructive est de réduire la taille du problème à chaque étape pour limiter progressivement l'ensemble des solutions réalisables. A chaque étape k , l'ensemble des solutions réalisables X est donc réduit en un ensemble $X^k \subseteq X$.

L'algorithme NEH (Nawaz, et al., 1983) représente un exemple d'approche constructive, développé dans le cadre de la recherche d'une séquence de durée minimale sur un atelier à cheminement unique (« flow shop ») (Lopez, et al., 2000). Cet algorithme se base sur l'hypothèse d'une tâche qui est prioritaire par rapport à une autre si son temps total d'exécution est plus élevé.

L'approche constructive est caractérisée par sa rapidité et sa simplicité mais son principal défaut réside dans la mauvaise qualité des solutions obtenues.

2.1.1.1. L'approche évolutive

La principale différence d'une approche évolutive par rapport aux autres approches citées précédemment, c'est qu'elle manipule un groupe de solutions réalisables, appelé *population*, à chaque étape du processus de recherche. Cet aspect est inspiré des sciences de la vie et des processus naturels là où les populations d'êtres vivants évoluent selon leurs propriétés collectives et leur environnement. De ce fait, et par analogie à la science de la vie, l'approche évolutive génère une population initiale, créée arbitrairement, qu'elle fait évoluer le plus

naturellement possible dans le but de trouver des bonnes solutions dans l'espace des solutions réalisables. En général, la taille d'une population reste constante tout au long d'un processus de recherche, qui évolue cycliquement en deux phases se succédant à tour de rôle : une phase de coopération collective entre tous les individus d'une même population et une phase d'adaptation individuelle. Une population courante *converge* lorsqu'elle contient un pourcentage élevé de solutions identiques. Ce phénomène doit être évité de se produire prématurément pour laisser assez de temps pour une population d'évoluer, afin de se rapprocher au mieux de la solution optimale. Plusieurs techniques existent pour éviter cet inconvénient, la solution la plus courante étant d'évaluer le degré de diversification d'une population afin de prévoir la diversité et empêcher la convergence prématurée. Nous décrivons dans ce qui suit les trois méthodes évolutives les plus connues, à savoir les algorithmes de colonies de fourmis et les algorithmes évolutionnaires.

- *Les colonies de fourmis*

Les études réalisées par les éthologistes ont montré que certains comportements collectifs des insectes sociaux étaient auto-organisés (Deneubourg, et al., 1989). 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, et al., 2003). Ce mécanisme permet d'améliorer progressivement les solutions au cours des itérations.

- *Les algorithmes évolutionnaires*

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, 1992). Les algorithmes génétiques (AG) ont été ensuite développés par d'autres chercheurs comme Goldberg, Davis et Michalewicz (Goldberg, 1989) (Michalewicz, 1994). Les AG constituent certainement l'exemple le plus connu (ou populaire) des algorithmes évolutionnaires. Depuis, d'autres variantes de ces algorithmes ont été aussi développées, telles que les algorithmes à stratégie d'évolution, la programmation génétique et la programmation évolutionnaire.

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.

Les individus d'une population dans un algorithme évolutionnaire évoluent durant une succession d'itérations appelées *générations*. Le critère d'arrêt de ces itérations est arbitraire, il peut dépendre par exemple de la qualité des solutions obtenues et/ou d'un nombre fixé d'avance de générations. A chaque itération, qui correspond à une nouvelle génération, des opérateurs de sélection et de variation sont appliqués sur les individus d'une population pour faire évoluer la population courante vers une nouvelle population qui devient la population courante de la prochaine génération. Ces opérateurs permettant aux individus de se reproduire, de survivre, de se modifier génétiquement au fil des générations ou de disparaître complètement.

- *Opérateurs de sélection* : Deux types d'opérateurs de sélection existent :

- ✓ La *sélection* : il s'agit de sélectionner une ou plusieurs fois dans la même génération, les individus qui vont subir un ou plusieurs opérateurs de variation ;

- ✓ Le **remplacement** : il s'agit de sélectionner la future population de la prochaine génération à partir de la population courante qui a évolué grâce aux opérateurs de variation.

L'évolution et l'adaptation des espèces selon le credo darwiniste ne laisse évoluer que les meilleurs individus ; cependant, il est parfois préférable d'agir sur les opérateurs de sélection pour donner leur chance aux individus faibles de survivre et de se reproduire.

- **Opérateurs de variation** : Appelés également opérateurs de recherche, les opérateurs de variation permettent de faire évoluer les individus d'une génération. Ces opérateurs sont classés en deux catégories :

- ✓ **Les opérateurs de mutation** (« mutation ») : un opérateur de mutation est un opérateur unaire qui agit sur un individu pour le modifier ;
- ✓ **Les opérateurs de croisement** (« crossover ») : un opérateur de croisement est un opérateur N-aire qui génère un ou plusieurs individus enfants à partir de plusieurs individus parents (deux en général).

Plusieurs techniques de sélection existent, les plus connues sont [53] la sélection proportionnelle, la sélection par tournois, la sélection déterministe et la sélection par remplacement.

Pour résumer le fonctionnement d'un algorithme évolutionnaire générique, ce dernier procède par l'initialisation d'une première population, puis il se déroule en quatre étapes à chaque « tour de boucle » (Figure II-10) :

1. Evaluation des individus de la nouvelle population ;
2. Sélection d'un ensemble d'individus pour le croisement ;
3. Sélection d'un ensemble d'individus pour la mutation ;
4. Remplacement de l'ancienne population donc déduction de la nouvelle population.

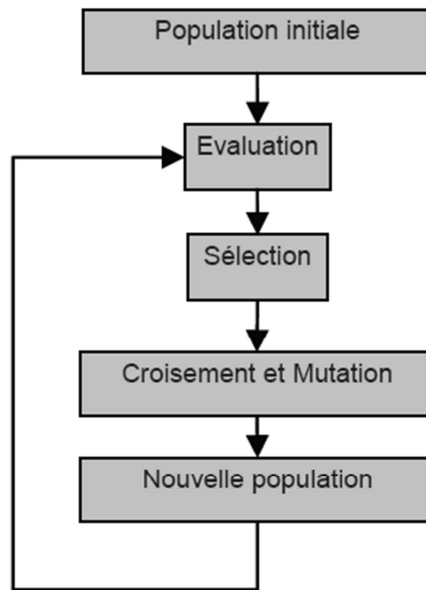


Figure II-12 Fonctionnement générique d'un algorithme évolutionnaire

Pour les algorithmes évolutionnaires, la façon de coder les solutions admissibles ainsi que les structures des opérateurs de variation et de sélection, dépendent du type et des contraintes du problème et des techniques adoptées pour le résoudre. Les résultats de ces algorithmes dépendent considérablement du choix de la structure de chacun de ces concepts, ce qui nécessite un travail minutieux pour leur élaboration. Cependant, grâce à leur grande adaptabilité, les algorithmes évolutionnaires peuvent s'accorder à n'importe quel type de problème. L'exploration parallèle de l'espace de recherche, basée sur des paramètres aléatoires et la disponibilité des meilleures solutions à un moment donné, représentent les principaux atouts des algorithmes évolutionnaires par rapport aux autres techniques d'optimisation. Ainsi, ces algorithmes ont connu un réel succès dans la résolution des problèmes d'optimisation combinatoire.

Pour la modélisation d'un système de transport, il faut prendre en compte toutes les données liées au problème d'optimisation dans un réseau de transport en question, les arrêts, les stations, les lignes du réseau, les véhicules et les distances inter-arrêts, etc. Les réseaux de transport sont souvent modélisés par des graphes illustrant les différents déplacements. Les représentations par des graphes diffèrent cependant d'un modèle à un autre.

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...

II.4.6. L'optimisation dans le domaine du transport

L'optimisation est assez large pour inclure différentes problématiques touchant à plusieurs domaines. Des problèmes d'ordonnancement, d'affectation, de logistiques ou autres, en Informatique, automatique, robotique, etc.

Parmi les problèmes de transport qui ont fait l'objet de travaux approfondis considérant les aspects d'optimisation, nous pouvons citer ceux relatifs à la recherche d'itinéraires que ce soit dans un contexte monomodal, multimodal ou comodal, mono-opérateur ou multi-opérateur ; les problèmes de tournées de véhicules, de ramassage et dépose de marchandises, d'objets ou de personnes, etc.

Pour le problème de recherche d'itinéraires dans un contexte de transport multimodal-multiopérateur, nous pouvons citer les travaux de (Feki, 2010) (Kamoun, 2007) et (Zidi, 2006). (Feki, 2010) et (Kamoun, 2007) se sont basés sur l'application d'une forme modifiée de l'algorithme classique de Dijkstra tandis que (Zidi, 2006) s'est penché vers une hybridation entre l'algorithme de Dijkstra et les algorithmes évolutionnaires pour optimiser des déplacements selon plusieurs critères.

D'autres travaux dans un contexte co-modal intégrant le mode de la voiture particulière, du vélo et de la marche à pied à côté des modes du transport en commun, ont opté pour l'algorithme de Dijkstra afin de résoudre le graphe de transport co-modal (Zhang, et al., 2011). Par contre, (Ayed, et al., 2010) se sont intéressés plutôt à une hybridation de l'algorithme de Dijkstra et celui des colonies de Fourmis.

Un autre problème de transport est concerné par les problèmes d'optimisation : le transport à la demande (TAD). Là aussi les algorithmes d'optimisation ont fait leur entrée pour pallier aux problèmes de routage de véhicules complexes essentiellement dans le cas de prise en charge instantanée de demandes dynamiques. Des heuristiques développées dans les travaux de (Horn, 2002) ont ainsi été proposées pour les algorithmes de routage en faveur de l'optimisation de l'insertion des demandes individuelles. Une réoptimisation régulière de tous les itinéraires est aussi proposée après l'insertion de chaque demande (origine et destination nouvelles).

Dans le même volet, les travaux de (Dessouky, et al., 2004) s'inscrivent dans le cadre de propositions pour l'optimisation des problèmes de transport à la demande de personnes handicapées ou personnes âgées (DARP : Dial-A-Ride Problem) en considérant les fenêtres

de temps. Les auteurs proposent dans ce contexte une heuristique visant comme objectif principal la minimisation d'une fonction généralisée de regret considérant à la fois la distance parcourue par les véhicules, le temps supplémentaire de trajet subi par le client par rapport au trajet direct et le temps inoccupé des véhicules. Dans l'algorithme ainsi proposé, les auteurs considèrent en premier lieu l'insertion des demandes ayant les heures de départ les plus proches et les points d'origine les plus éloignés du dépôt. Par la suite, c'est le reste des demandes qui est inséré en second lieu suivant un processus spécifique minimisant l'augmentation du coût dans chaque itinéraire.

II.5. L'alliance entre les Systèmes multi-agents et l'optimisation

Un système d'information de transport est un système qui doit intégrer un large nombre d'utilisateurs et doit être ainsi capable de traiter instantanément une quantité d'informations combinatoires. L'aspect dynamique et instable de l'environnement rend ces informations susceptibles de changer d'un instant à l'autre. Le concept des SMA s'accorde parfaitement à ce contexte et est capable d'apporter l'efficacité requise dans le cadre d'une dynamique contraignante. Compte tenu de ce constat, les agents sont donc bien appropriés vu leurs aptitudes d'intégration et d'adaptation efficaces sous ces contraintes. Ils sont en effet capables de réaliser des traitements sur des données hétérogènes et dynamiques grâce aux différentes coopérations qui peuvent subsister mais aussi aux perceptions continuellement mises à jour qu'ils ont de l'environnement dans lequel ils évoluent.

En plus du besoin continu de trouver des solutions optimales aux utilisateurs et dans des brefs délais, les agents intervenant dans cet environnement peuvent intégrer des méthodes d'optimisation adaptées à leurs compétences et connaissances.

Une coopération entre optimisation et agents permet ainsi de réaliser un service optimisé tout en étant efficace et performant. L'alliance de ces deux concepts s'impose ainsi d'elle-même et a pour effet d'incorporer le sens de l'optimisation dans les agents (i.e. optimisation locale intégrée dans les actions et comportements des agents), eux-mêmes régis globalement par des méthodes d'optimisation distribuées.

L'intégration du concept d'optimisation aux SMA dans ce cadre a servi à la mise en place de services optimaux en termes de critères de satisfaction par rapport aux attentes des utilisateurs (coût, temps de production, rentabilité, etc.).

Dans notre équipe, cette alliance ne cesse d'évoluer d'un projet à l'autre. En effet, tous les travaux de (Kamoun, 2007) (Feki, 2010) (Zidi, 2006) (Zgaya, 2007) (Sghaier, 2011) ont intégré le paradigme agent dans des systèmes d'optimisation et ont prouvé leur efficacité et

leurs hautes performances. D'autres travaux dans le domaine du transport proposent un modèle dynamique de transport basé sur les activités (Ma, 2007). Ce modèle permet de traiter les choix des destinations, des chaînes des modes de transport, des itinéraires et des temps de départ dans un réseau de transport co-modal avec la voiture particulière et le transport en commun comme modes de transport. Ils proposent pour la résolution du problème un algorithme basé sur la méthode d'Entropie Relative ainsi qu'un algorithme basé sur les colonies de fourmis.

Une autre alliance optimisation et multi-agents a été proposée pour Modélisation et Optimisation du Problème de Transport à la Demande Multicritère et Dynamique (Zidi, 2012). Une approche basée sur l'algorithme de recuit simulé a été développée pour la résolution de ce problème. La première contribution a été la résolution du PTD statique en utilisant l'algorithme de Recuit Simulé (RS) avec une méthode agrégative. Vu que cette méthode présente plusieurs limites telles que (la favorisation d'un objectif par rapport à un autre), l'algorithme de Recuit Simulé Multi-Objectif (RSMO) pour la résolution du PTD a été appliqué pour remédier à cette lacune. La performance de l'approche a été améliorée pour s'adapter aux critères de dynamique et de l'utilisation des véhicules hétérogènes. Cette adaptation est faite via la distribution du traitement sur un Système Multi-Agents (SMA).

Nous allons suivre ces démarches décrites ci-dessus en les adaptant à notre problématique. En effet, vu la haute performance et efficacité des SMAs et des différentes méthodes d'optimisation, nous ne pouvons que tirer profit de cette alliance.

II.6. Conclusion

Notre but est de concevoir et d'optimiser un système d'information de transport co-modal qui fournit aux utilisateurs des itinéraires co-modaux optimisés en terme de trois critères : le temps, le coût et les émissions des gaz à effet de serre, tout en respectant leurs contraintes et préférences. Il s'agit d'un problème multi-objectif à résoudre au sein d'un système largement distribué et dynamique, qui interconnecte des fournisseurs d'information de différents services de transport. L'étude bibliographique que nous avons établie dans ce chapitre nous incite à concevoir un Système d'information multi-agent de transport co-modal, en se basant sur une approche hybride intégrant l'algorithme de Dijkstra et un algorithme évolutionnaire pour la résolution de problème difficile multiobjectif. En effet, un algorithme modifié de Dijkstra sera appliqué sur le graphe co-modal pour une première résolution et ensuite vient intervenir l'algorithme évolutionnaire. Grâce à leur grande adaptabilité, les algorithmes évolutionnaires peuvent s'accorder à n'importe quel type de problème par une exploration en parallèle de l'espace de recherche. D'un autre côté, l'approche multi-agent s'adapte aux

besoins des nouveaux systèmes d'information à caractère dynamique et à sémantique riche, variée et évolutive, en prenant en considération l'aspect distribué et hétérogène de l'information.

III. Chapitre III : Architecture à base d'agents communicants intégrant des graphes distribués pour la combinaison des services de transport

III.1. Introduction

Le problème de l'optimisation distribuée de recherche et de composition d'itinéraires co-modaux pour les usagers découle de l'inconstance et l'accroissement continu des sources d'information distantes et distribuées sur les réseaux de transport co-modal. En plus, la diversité des modes et des services de transport existants de nos jours, ne fait qu'alimenter le besoin de concevoir des systèmes performants qui trouvent une solution optimale et offrir aux utilisateurs en temps réel des itinéraires co-modaux optimisés selon différents systèmes de transport.

Pour la résolution de ce problème, nous proposons un Système d'Information de Transport Co-modal (SITCoMo). La conception préliminaire de ce système nous a permis de déceler deux besoins d'optimisations. Une première optimisation qui permet par rapport à une requête utilisateur de limiter le domaine de recherche. Une deuxième optimisation consiste à rechercher et composer les meilleurs itinéraires co-modaux en termes de différents critères comme réponses aux requêtes utilisateurs.

Nous proposons alors une solution à trois niveaux d'optimisation dissimulée dans un système d'information multi-agent ouvert et dynamique. La première optimisation permet de limiter le nombre d'opérateurs de transport, quel que soit leur service, indispensables pour fournir toutes les données et toutes les routes possibles qui peuvent construire les solutions globales. La deuxième optimisation concerne la recherche et la composition des itinéraires co-modaux optimisés en termes de différents critères. Mais vu la complexité du problème nous avons opté pour une hybridation entre un algorithme de Dijkstra distribué pour la résolution du graphe co-modal et une approche évolutionnaire pour l'optimisation multicritère. Le deuxième niveau d'optimisation permet ainsi de trouver les plus courts chemins en termes de temps dans un graphe co-modal en se basant sur un graphe spécial et l'application de l'algorithme distribué de Dijkstra. La troisième optimisation prend comme paramètres les solutions du

deuxième niveau et continue à chercher la combinaison de route optimale pour chaque utilisateur.

Dans ce chapitre, nous présentons le système SITCoMo avec l'architecture multi-agents proposée ainsi que les deux premiers niveaux d'optimisation. En effet, l'identification du domaine de recherche et l'approche co-modale adoptée pour la recherche des premiers courts chemins sont détaillés dans ce chapitre.

III.2. Formulation du problème

Un Système d'Information de Transport Co-modal (SITCoMo) vise à satisfaire les requêtes des utilisateurs en leur fournissant des solutions adéquates tout en respectant les délais des réponses et en optimisant trois essentiels critères : le temps, le coût du trajet et l'émission des gazs à effet de serre pendant le trajet. Ce problème devient difficile à résoudre si les requêtes utilisateurs sont nombreuses, simultanées et formulées par un ensemble de clients du réseau de transport via différents dispositifs (ordinateur portable, ordinateur de bureau, GSM, PDA, etc.). Nous proposons donc de gérer les requêtes reçues par « bloc » simultané et procéder ensuite par la recherche et la composition des itinéraires. Cette recherche décompose les requêtes sous forme de sous requêtes qui sont considérés comme des tronçons d'itinéraires que nous appelons *routes*. Un itinéraire peut être composé par une ou plusieurs *routes*. Il est possible qu'il soit proposé par plusieurs services de transport à la fois avec différents coûts. En effet, l'itinéraire peut être monomodal assuré par un seul service ou mode de transport comme il peut être également co-modale assuré par plusieurs services de transport. L'itinéraire est donc décomposé en *routes* qui peuvent être proposées également par plusieurs services de transport.



Figure III-1 Fonctionnement du système SITCoMo

Le système SITCoMo est en relation avec plusieurs services de transport qui peuvent proposer plusieurs modes de transport.

Le fonctionnement de ce système commence alors par la réception des requêtes avec toutes les données requises (points de départ et d'arrivée, les horaires de départ et d'arrivée souhaités, les préférences et les contraintes par rapport aux modes de transport, la priorité des critères...). Ensuite, afin d'identifier toutes les *routes* possibles qui peuvent répondre aux requêtes, le système doit tout d'abord localiser tous les points de départ et d'arrivée et communiquer par la suite avec les opérateurs de tous les services de transport pour identifier toutes ces *routes*. Une *route* est considérée comme un tronçon d'itinéraire comme elle peut être aussi un itinéraire complet si ses points de départ et d'arrivée correspondent à ceux d'une requête. Elle peut être assurée par différents véhicules dans la fenêtre de temps qui correspond à la fenêtre de temps demandée par l'utilisateur. Par la suite, à partir de ces routes identifiées, le système doit organiser toutes les *combinaisons de routes* possibles qui composent les réponses aux requêtes. Les *combinaisons de routes* sont la jonction des différentes *Routes*. Ainsi, le problème posé pour chaque *route* est « comment choisir un véhicule pour cette *route* pour un utilisateur en prenant en compte ses contraintes et ses préférences concernant les critères à optimiser. »

A un instant t , le problème décrit ci-dessus est défini par :

- N requêtes formulées pendant un intervalle de temps Δ_ε . L'ensemble de ces requêtes est noté par I_t .
- $I_k(d_k, a_k, W_k) \in I_t$ est une demande d'itinéraire formulée par un usager k à l'instant t à partir d'un point de départ d_k vers un point d'arrivée a_k pendant l'intervalle de temps $W_k = [td_k, ta_k]$ avec td_k et ta_k correspondent respectivement au temps de départ au plus tôt (le temps de départ minimal de d_k) et le temps d'arrivée au plus tard (le temps maximal d'arrivée à a_k) avec $t \leq td_k \leq ta_k$.
- $R_g(d_g, a_g, W_g)$ est une *Route* identifiée pour répondre à une ou plusieurs requêtes appartenant à I_t . Un véhicule assure cette *route* à partir d'un point de départ d_g vers un point d'arrivée a_g pendant un intervalle de temps $W_g = [tg_d, tg_a]$ avec tg_d et tg_a correspondent respectivement au temps le plus tôt possible pour quitter d_g et au temps le plus tard possible pour arriver à a_g .

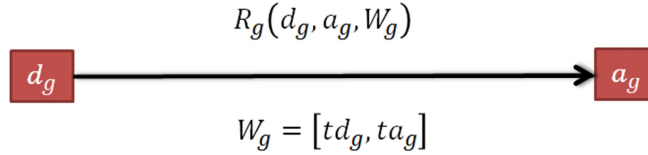


Figure III-2 Un tronçon d'itinéraire : une Route R_g

- Une jonction ou une succession de plusieurs routes $R_g(d_g, a_g, W_g)$ forment un itinéraire comme réponse à une requête $I_k(d_k, a_k, W_k) \in I_t$. Une combinaison de routes possible formant une solution est notée $RC_{k,p}$.

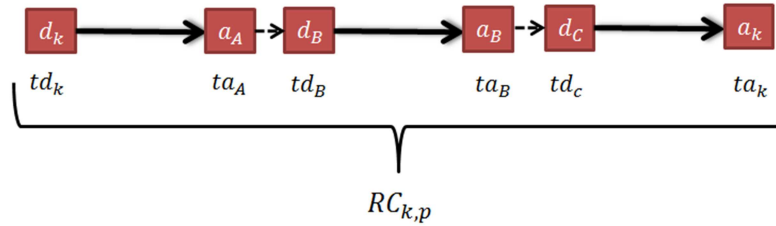


Figure III-3 Une combinaison de Route $RC_{k,p}$

- On note $RC_k = \{RC_{k,p}, p \in [1..P]\}$ l'ensemble de toutes les combinaisons de routes possibles identifiées pour répondre à une demande d'itinéraire I_k .
- Une route $R_g(d_g, a_g, W_g)$ peut être servie par plusieurs véhicules. On note $V_h^{R_g}$ le véhicule V_h qui assure à l'instant t une seule route $R_g(d_g, a_g, W_g)$ avec $1 \leq h \leq H$, H est le nombre total de véhicules disponibles pour servir la route R_g . Chaque véhicule $V_h^{R_g}$ est caractérisé par une valeur pour chaque critère obtenue par $C_{V_h^{R_g}}^i$ avec $1 \leq i \leq CR$ et CR est le nombre total des critères à optimiser.

Donc, à un instant t , un véhicule possède un seul temps de départ et une seule valeur par critère. Un véhicule $V_h^{R_g}$ ($1 \leq h \leq H$) est aussi caractérisé par un type. Nous distinguons trois types de véhicules :

- Type A : Ce type concerne les voitures particulières assurant le service de covoiturage. Pour chaque route R_g à l'instant t , un véhicule $V_h^{R_g}$ ($1 \leq h \leq H$) est caractérisé par trois propriétés dynamiques : le nombre de places disponibles dans le véhicule ($V_h^{R_g}.nb$), le nombre maximal de places autorisé dans le véhicule ($V_h^{R_g}.nbMax$). Ainsi, on peut déduire le nombre de personnes dans le véhicule avec $V_h^{R_g}.nbMax - V_h^{R_g}.nb$. Chaque véhicule possède un

temps de départ $V_h^{Rg}.dep$ pour assurer la route R_g . Ainsi, un véhicule de type A assurant une certaine route, va être noté comme suit $V_h(nb, nbMax)$;

- Type B : Ce sont les véhicules en libre service (exemple, Autolib, Vlib...). Ce type est caractérisé seulement par le nombre de véhicules disponibles $V_h^{Rg}.nb$ au point de départ de la route R_g et dans la fenêtre de temps correspondante ;
- Type C : concerne tous les moyens de transport en commun (bus, métro, tramway...). Le type C est caractérisé seulement par le temps de départ $V_h^{Rg}.dep$.

III.3. Complexité du problème

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. Dans un problème d'optimisation combinatoire, il s'agit de trouver la meilleure solution à partir d'un ensemble dit « *ensemble de solutions réalisables* ». Cet ensemble est décrit de manière implicite à travers l'ensemble des contraintes que doivent satisfaire les solutions qui y appartiennent, appelées « *solutions réalisables* ». Le nombre de celles-ci est généralement fini mais exponentiel (très grand), d'où la difficulté de résolution et ses mauvaises répercussions sur les performances du système considéré. Compte tenu de cet énoncé, un problème d'optimisation combinatoire peut avoir plusieurs solutions possibles (Khalifa, 2011).

D'après la formulation du problème ci-dessus, après l'expression des requêtes, le système va recevoir une multitude de réponses de la part des différents opérateurs des différents services de transport. Toutes ces réponses forment un graphe co-modal représentant le plan des itinéraires des différents services de transport. Un graphe co-modal ne peut être qu'un graphe très dense et contient énormément de nœuds que ce soit des nœuds de correspondances pour le transport en commun, des nœuds de rendez-vous pour le covoiturage ou des nœuds représentant des stations de véhicules en libre service. Ces propriétés engendrent des échanges intermodaux et co-modaux 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 de transport correspond à un élément, cet élément est représenté par son nœud de départ, son nœud d'arrivée et un tableau « *Correspondance* ». Ce tableau représente les modes qui partagent des nœuds de correspondance avec cet élément. Un nœud de correspondance peut être un point de transfert entre différents modes du même service de transport (bus, métro par

exemple) comme il peut aussi être un transfert co-modal vers un autre mode différent comme la voiture par exemple. Ainsi, un déplacement co-modal est équivalent à un cheminement ou un parcours entre ces éléments. Dans chaque élément, il faut chercher tous les points de correspondance pour construire un itinéraire allant d'un point de départ vers un point d'arrivée. Ce déplacement devient alors un cheminement entre plusieurs points, en passant par ces nœuds une et une seule fois.

Dans la littérature, ce problème est similaire à celui du voyageur de commerce. Nous avons fait une étude comparative du problème d'optimisation d'un système de transport co-modal et celui du problème très connu le voyageur de Commerce (PVC).

Tableau III-1 Comparaison entre les problèmes du voyageur de commerce et de l'optimisation dans un système de transport co-modal

| | | |
|-------------------|--|--|
| Problème | Problème Voyageur de Commerce (PVC) | Problème d'Optimisation dans le Système de transport Co-modal (à un instant t donné pour une requête donnée) |
| Paramètres | Villes de passage du Voyageur de Commerce | Nœuds de <i>Correspondance</i> (décrits ci-dessus) |
| Contrainte | Chaque ville doit être visitée une et une seule fois | Itinéraire unique n'engendrant pas de boucle, passage une seule fois par un nœud de <i>Correspondance</i> , disponibilité et positions des différents modes des différents services de transport (voiture de co-voiturage, bus, train, stations des véhicules en libre service...), itinéraires des voitures, nombre de places |
| Fonction Objectif | Trouver une tournée légale et optimisée : le plus court chemin reliant toutes les villes | Satisfaction de l'utilisateur : fournir un itinéraire co-modal optimisé selon les préférences et les critères de l'utilisateur |

Suite à cette analogie entre le problème de voyageur de commerce et une instance de notre problème de transport co-modal, nous pouvons voir la similitude entre les deux problèmes par rapport aux paramètres à prendre en compte (nœuds origines/destinations Vs villes...), par rapport aux contraintes (chaque nœud (ville) doit être visité une et une seule fois...), et aussi

par rapport à la fonction objectif où il s'agit de minimiser un coût (distance, durée du trajet, émission de CO₂, etc.).

Par analogie, nous pouvons dire que le problème d'optimisation d'un transport co-modal est un problème combinatoire de complexité équivalente à celle d'un voyageur de commerce.

Ainsi dans un graphe G , pour n nœuds de *Correspondance*, il y a $\frac{(n-1)!}{2}$ tours possibles, si nous considérons que l'évaluation d'un tour nécessite un temps de $O(n)$, nous obtenons alors une complexité totale en temps de $O(n!)$. Pour 6 nœuds, nous avons 720 possibilités, pour 10 nœuds, nous avons 3628800 possibilités...

Cependant, cette étude s'est basée sur la considération d'une seule requête à l'instant t . Or, nous considérons que nous recevons à l'instant t plus d'une demande d'itinéraire à traiter simultanément. Ceci rend notre problème plus complexe que celui d'un voyageur de commerce, c'est comme-ci nous avons multiples voyageurs de commerce à traiter au même instant. Sans oublier, d'autres contraintes doivent être prises en considération comme la gestion de différents services de transport, la gestion des données distribuées, la gestion en temps réel des requêtes et des perturbations...

La complexité du problème traité ainsi que le caractère distribué et dynamique du système concerné nous ont incités à choisir une modélisation multi-agent que nous présentons dans le paragraphe qui suit.

III.4. Architecture multi-agents proposée

Depuis leur apparition, les systèmes multi-agents ont connu un essor très important en faisant intervenir plusieurs entités hétérogènes pour la résolution d'un même et unique problème. Développés dans le cadre d'applications sophistiquées, les *SMA* favorisent la mise en place de processus distribués sur différentes entités autonomes (ou semi-autonomes) pour en garantir l'optimalité. Présentant l'atout principal de mise à bas des problèmes de complexité des systèmes informatiques, nous considérons alors le paradigme agent pour la résolution de notre problématique.

Nous proposons pour cela un système dynamique et ouvert (figure III-4) basé sur l'interaction de six types d'agents logiciel : les agents Interface (*IA*), les Super Agents (*SupA*), les agents Service de Transport (*TSA*), les agents Information de Transport (*TIA*), les agents Routes (*RA*) et les agents Evaluateurs (*EA*) (Jeribi, et al., 2011).

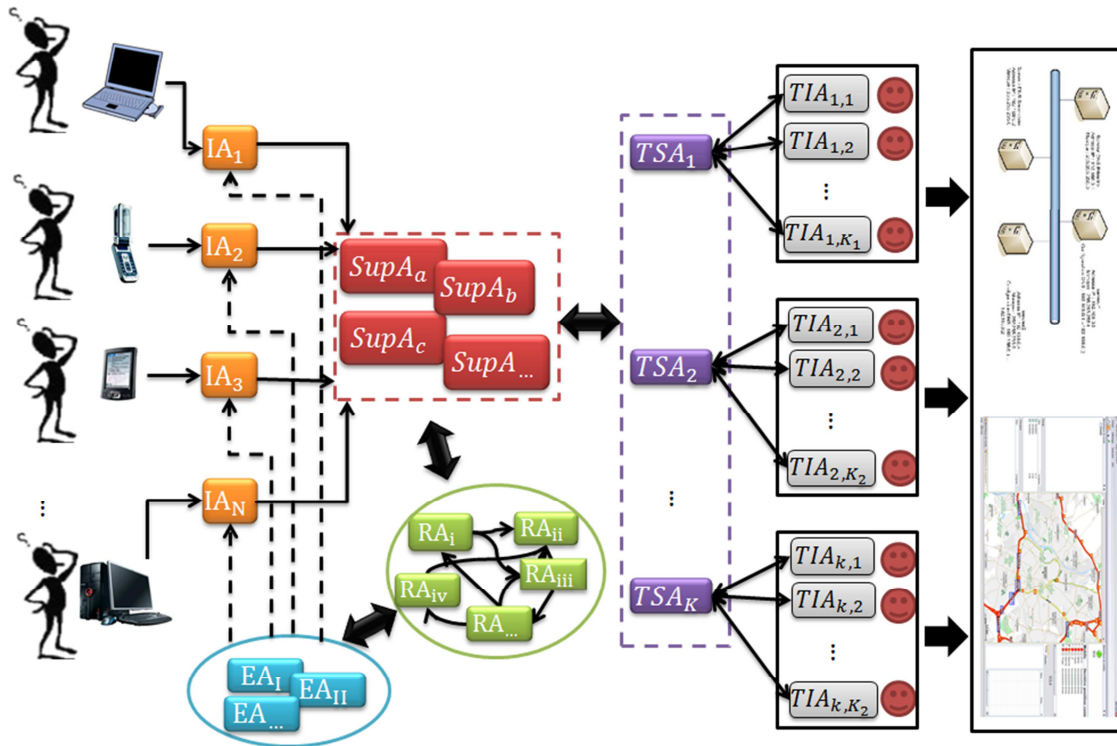


Figure III-4 Architecture du système multi-agents

L'agent *IA* joue le rôle d'interface entre les utilisateurs et le système. En recevant la requête, il la transmet au *SupA*. L'agent *SupA* s'occupe de la détermination des différentes *routes* qui peuvent répondre aux différentes requêtes en coopération avec les *TSA*s et les *TIA*s. Nous associons à chaque service de transport un agent *TSA* et à chaque opérateur de transport un agent *TIA*. Chaque *TSA* est responsable d'un ensemble d'agents *TIA*s lorsque les opérateurs correspondants appartiennent au même service de transport. La communication entre ces agents et le *SupA* permet d'identifier toutes les *routes* possibles pour répondre aux différentes requêtes. Le *SupA* applique une première approche d'optimisation (§ III.7.3) sur un graphe composé de ces *routes*. Les résultats obtenus à la fin de l'exécution de l'algorithme constituent un graphe appelé graphe de transport des plus courts chemins (SPTG). Le *SupA* crée ensuite un agent Route *RA* pour chaque *route* de ce graphe. Chaque *RA* applique par la suite une approche évolutionnaire permettant de déterminer la meilleure *route* et une coalition des *RAs* va être élaborée pour construire les différentes *combinaisons de routes* possibles pour répondre aux requêtes. A la suite de ces coalitions, l'*EA* reçoit les différentes *combinaisons de routes* pour calculer la meilleure solution et la transmettre à l'*IA*. L'*IA* va s'occuper de la transférer à l'utilisateur.

L'existence de chacun de ces agents dépend de celle des requêtes utilisateurs. En d'autres termes, s'il n'y a eu aucune requête pendant une assez longue période appelée *période d'inactivité* et notée Δ_∞ , alors aucun de ces agents n'existe pas dans le système.

Dès la formulation d'une requête utilisateur à un instant t , une société d'agents IA , $SupA$, TSA , TIA et EA est créée. Pendant une courte période de temps notée Δ_ε , chaque nouvelle requête formulée par un nouvel agent IA est gérée par la même société d'agents dont la création a été déclenchée par la première requête formulée à l'instant t . Cette société d'agents est notée S_t . Après la période Δ_ε , la formulation des nouvelles requêtes déclenche la création d'une nouvelle société si les sociétés d'agents préalablement créées sont indisponibles et ainsi de suite. La figure III-5 explique le fonctionnement dynamique du système pendant une période de formulation.

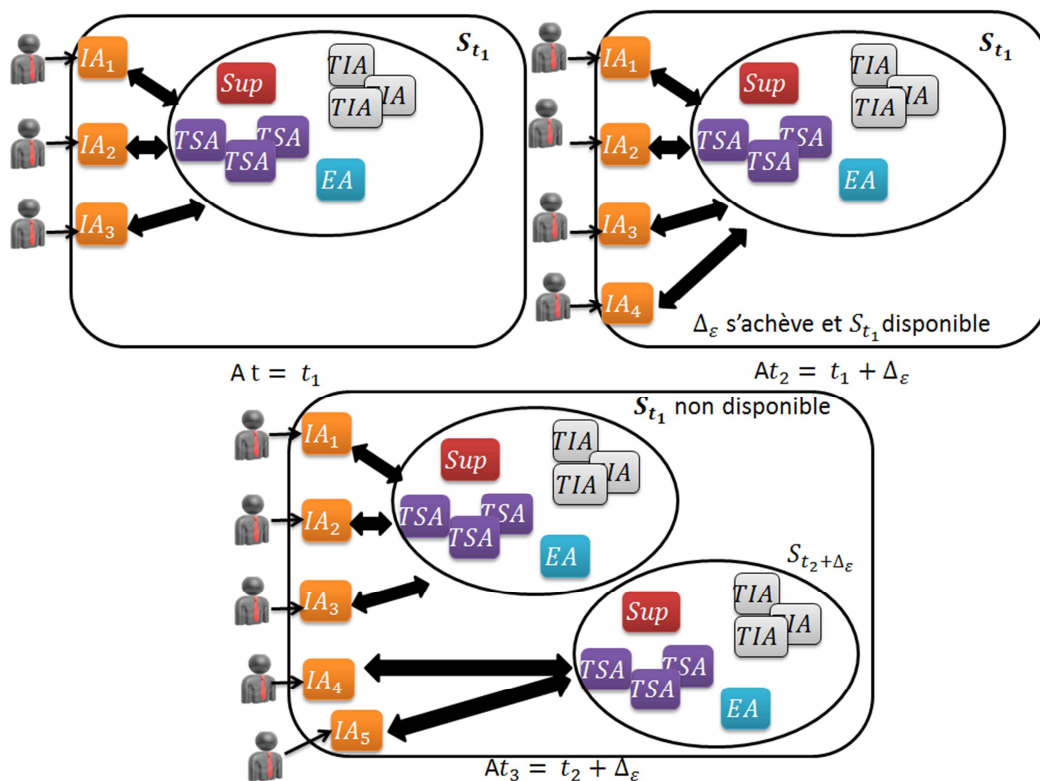


Figure III-5 Comportement du système pendant une période de formulation

III.4.1. Comportement des agents dans le système

Les comportements des différents agents du système sont décrits dans ce qui suit.

III.4.1.1. Agent Interface (IA)

Cet agent interagit avec l'utilisateur du système en lui permettant d'une part d'exprimer et d'émettre sa requête ainsi que ses préférences. Dès qu'un utilisateur accède au système, un agent IA est créé en temps réel pour l'aider et l'assister à formuler sa requête. Une requête

utilisateur formulée via un agent *IA*, est envoyée à un Super agent disponible. S'il n'existe pas encore de super agent (*SupA*) disponible dans le système, un nouvel agent *SupA* est automatiquement créé, au sein d'une nouvelle société d'agents. Dès qu'un agent *IA* reçoit une réponse donnée, il identifie le support client de diffusion approprié puis effectue une diffusion personnalisée du résultat obtenu. Le comportement de l'agent *IA* est illustré par le diagramme d'activité de la Figure III-6.

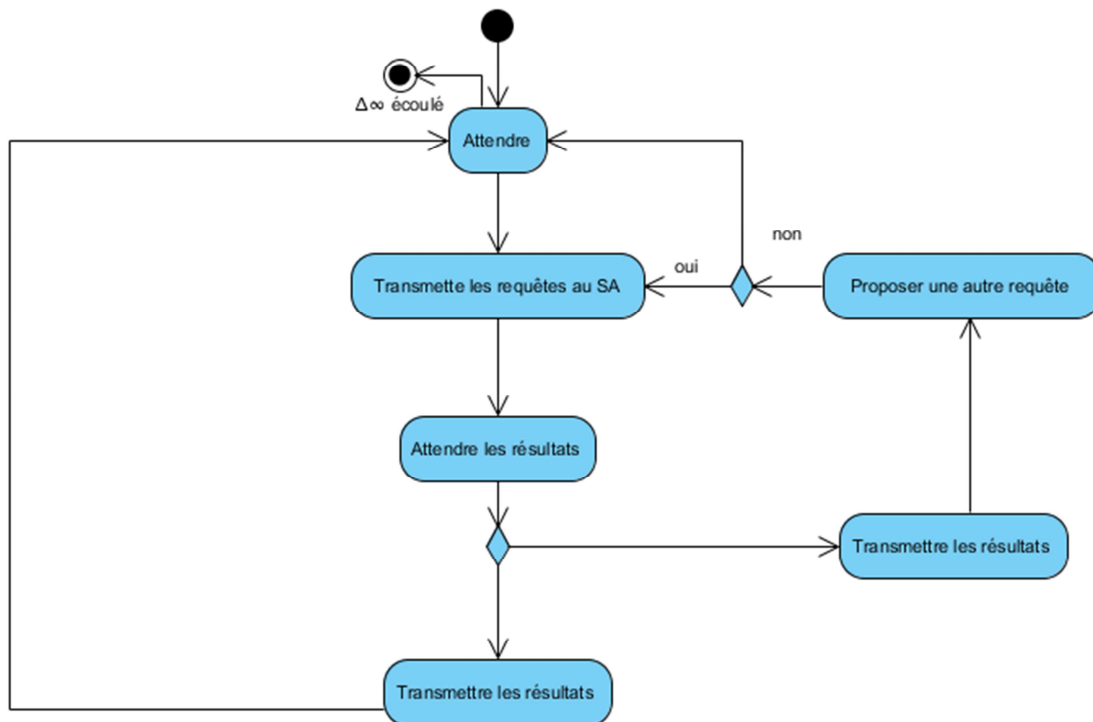


Figure III-6 Diagramme d'activité de l'agent interface IA

III.4.1.2. Super Agent (*SupA*)

Un agent *SupA* disponible peut recevoir un ensemble de requêtes simultanées pendant Δ_e . On l'appelle Super Agent car il s'agit d'un agent qui possède différents rôles (Figure III-7) et passe d'un rôle à un autre selon la nécessité.

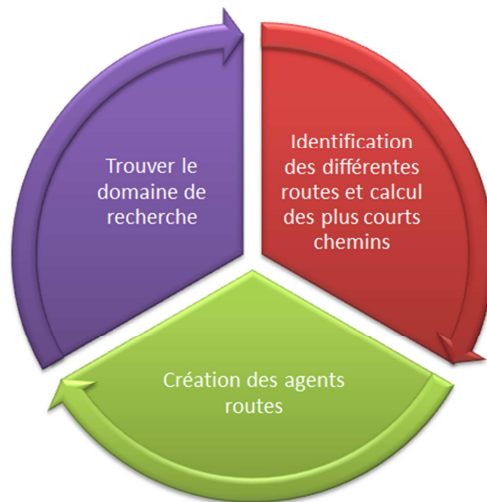


Figure III-7 Les rôles du Super Agent (*SupA*)

- Premier Rôle : Identification du domaine de recherche

En recevant les requêtes, le *SupA* joue le rôle d'identificateur et doit identifier le domaine de recherche. Pour cela, il doit tout d'abord localiser tous les points de départ et d'arrivée des différentes requêtes, ensuite il envoie une requête aux agents de service de transport (*TSA*) pour déterminer le domaine de recherche. Chaque *TSA* doit répondre avec la liste des opérateurs qui sont susceptibles de répondre aux requêtes.

- Deuxième Rôle : Détermination des différentes *routes* et calcul du plus court chemin en termes de temps.

A sa demande, les *TSAs* envoient au *SupA* la liste des *TIAs* qu'il pourra interroger. En effet, le *SupA* transmet ses requêtes aux différents *TIAs* afin d'identifier toutes les *routes* possibles qui peuvent répondre et satisfaire tous les utilisateurs. A sa réception des réponses, une approche co-modale va être appliquée. Le *SupA* construit alors le graphe co-modal avec toutes les réponses envoyées par les différents opérateurs des différents services de transport. Chaque arc de ce graphe possède un point de départ et un point d'arrivée ainsi que les horaires correspondants et le véhicule qui va le servir. Le système se doit de résoudre ce graphe pour fournir des itinéraires co-modaux optimisés et répondre aux utilisateurs. Vu que le graphe est assez complexe, le plus judicieux serait de trouver un moyen plus simplifié pour la résolution de ce graphe. A partir du graphe co-modal, un autre graphe spécifique appelé graphe de transfert va être déduit. Il s'agit d'un graphe constitué de différents composants et des arcs virtuels qui les relient. Tous les arcs appartenant au même service de transport sont groupés

dans un même composant. La recherche d'itinéraires co-modaux optimisés commence par une première optimisation en termes de temps appliquée sur chaque graphe de chaque composant. Afin de gagner le temps, nous avons profité de l'intelligence des systèmes multi-agents et nous avons utilisé la fonction de clonage. Le SupA crée un clone pour chaque composant. Chaque clone va exécuter un algorithme qui sera décrit ultérieurement pour la recherche des plus courts chemins dans chaque composant mono-service. Il est plus rapide et plus efficace de travailler en parallèle et sur des graphes mono-service qui décrivent un seul service de transport mono-opérateur ou multi-opérateurs. Une fois les réponses obtenues, le *SupA* construit un nouveau graphe de transfert avec les plus courts chemins calculés. Ceci nous donne des premières solutions optimisées en termes de temps mais sans aucune affectation de requêtes. En effet, le traitement se fait à l'ensemble des requêtes simultanément sans avoir recours aux requêtes des utilisateurs ni à leurs préférences ou à leurs critères d'optimisation.

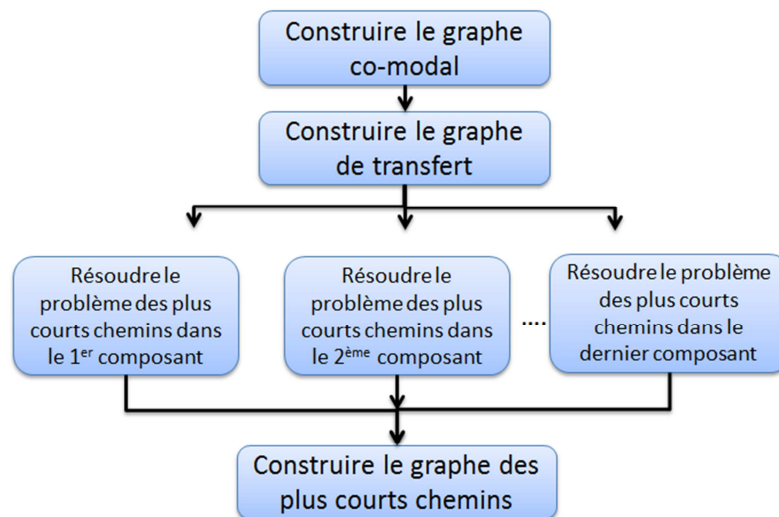


Figure III-8 Clonage du SupA pour la résolution du problème des plus courts chemins dans chaque composant

- Troisième Rôle : Génération des agents Routes *RA*.

A partir des *routes* identifiées après le calcul des premiers plus courts chemins en terme de temps et donc du dernier graphe construit, le *SupA* génère les agents Routes. Chaque arc représente une *route* pour se déplacer d'un point vers un autre dans une fenêtre de temps précise et qui peut être servie par différents véhicules appartenant à différents services de transport. Ces *RAs* prennent une forme de chromosome dans lesquels le *SupA* associe à la *route* les différents véhicules et les différentes personnes intéressées par cette *route* afin

d'affecter les solutions aux requêtes. Le but est de trouver la meilleure affectation de chaque chromosome et effectuer une combinaison de ces chromosomes pour former les solutions optimales en termes des trois critères.

Le fonctionnement global du *SupA* est décrit dans la figure suivante :

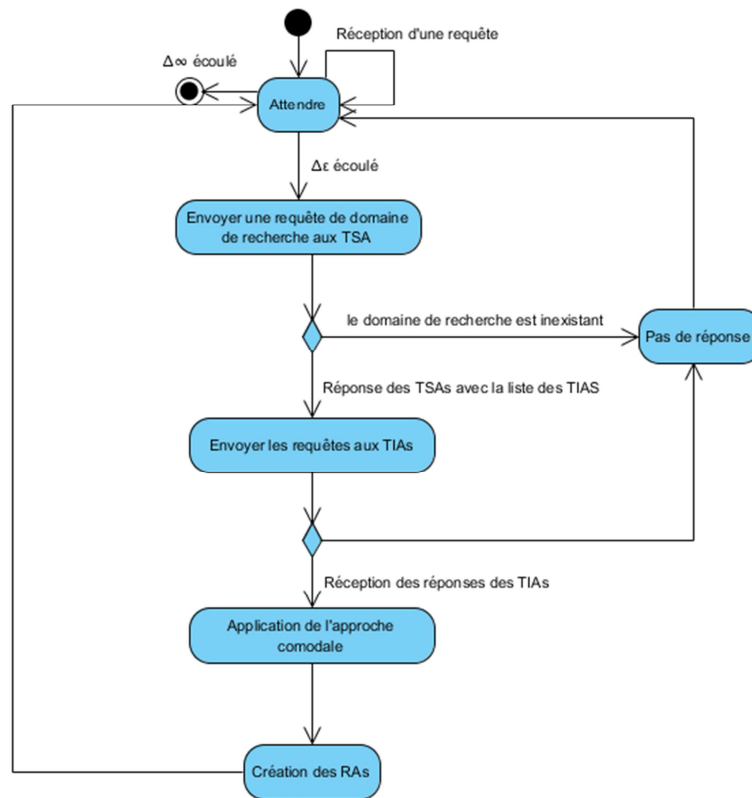


Figure III-9 Diagramme d'activité de l'agent SupA

III.4.1.3. Agent Service de Transport (TSA)

A chaque service de transport, nous associons un agent *TSA*. Ce service peut être par exemple un service de transport en commun, de covoiturage ou d'autopartage. Chaque *TSA* est responsable d'un ensemble de *TIAs*.

Le *TSA* présente deux scénarios de fonctionnement :

- Soit un nouveau *TIA* demande à s'inscrire au système. Il fournira alors un ensemble de données. La base de données des *TIAs* du système est alors mise à jour.
- Soit il reçoit d'un agent *SupA* une demande de domaine de recherche. Cette requête lui fournit les points de départ et d'arrivée. Il détermine alors les *TIAs* de départs et d'arrivée associés à ces points et sélectionne un ensemble de *TIAs* qui permet de relier les deux premiers. S'il arrive à constituer le domaine de recherche relatif à cette requête, il transmet le résultat au *SupA* sinon un refus est transmis au *SupA* qui

l'incitera à reformuler sa requête. Le comportement de l'agent *TSA* est illustré par le diagramme d'activité de la Figure III-10.

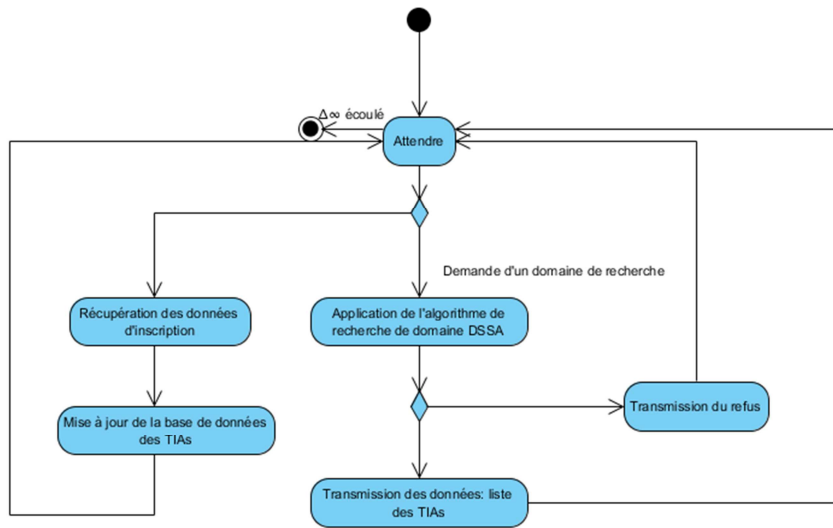


Figure III-10 Diagramme d'activité de l'agent TSA

III.4.1.4. Agent Information Transport (TIAgent)

Pour chaque opérateur de transport associé à un service de transport que ce soit un service de transport en commun, un service de voitures en libre service ou de covoiturage, nous associons un agent *TIA*. Chaque *TIA* est responsable d'un opérateur de transport ainsi que de son réseau. Lorsqu'il reçoit une requête du *SupA*, le *TIA* doit la transformer pour l'envoyer à son système d'information. Dans ce cas, il va transmettre au *SupA* soit la réponse à sa requête soit un refus dans le cas où l'information requise n'est pas fournie par le système d'information (Figure III-11).

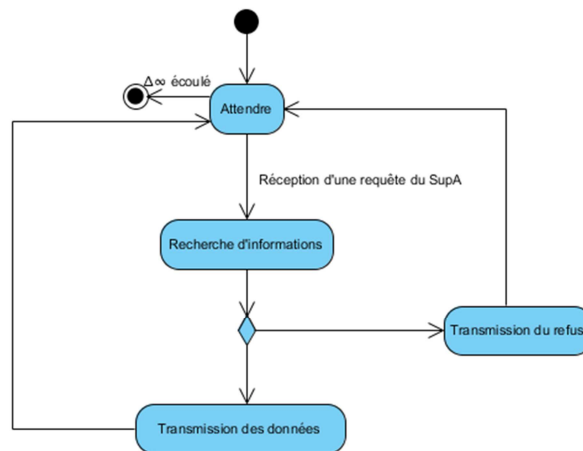


Figure III-11 Diagramme d'activité de l'agent TIA

III.4.1.5. Agent Route (RA)

En obtenant le graphe de transfert avec les plus courts chemins (SPTG), le *SupA* crée en temps réel un *RA* pour chaque *route* donnée $R_g(d_g, a_g, W_g)$ faisant partie du graphe SPTG. Cet agent est représenté par un chromosome généré pour une *route* disponible dans le but d'assigner les requêtes concernées aux véhicules capables d'assurer cette *route*. Après une première affectation, chaque *RA* va appliquer une méthode évolutionnaire afin d'obtenir la meilleure affectation et ainsi la meilleure solution pour chaque requête concernant ce tronçon ou bien ce bout d'itinéraire. Ensuite, des coalitions vont être entamées afin de regrouper tous les *RAs* qui peuvent construire une *combinaison de routes* possible pour un itinéraire. Nous pouvons avoir autant de coalitions que de combinaisons tout en sachant qu'un *RA* peut participer dans plusieurs coalitions. Ces coalitions apparaissent et disparaissent dynamiquement selon la réception des requêtes et des réponses. La liste des *combinaisons de routes* formées va être envoyée à l'agent *Evaluateur* pour la détermination de la meilleure solution optimisée pour chaque requête.

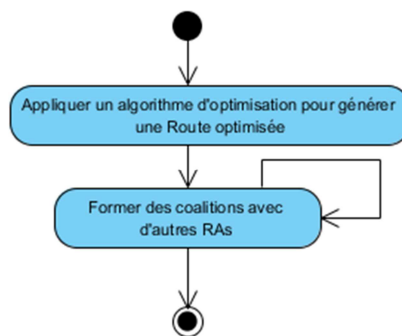


Figure III-12 Diagramme d'activité de l'agent RA

III.4.1.6. Agent Evaluateur (EA)

Cet agent calcule la meilleure combinaison de Routes pour chaque demande d'itinéraire et la transmet ensuite à l'agent *Interface* (*IA*). Le comportement de l'agent *EA* est illustré par le diagramme d'activité de la Figure III-13.

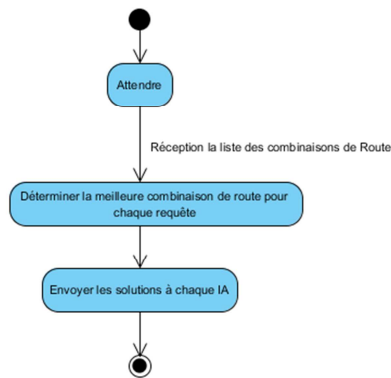


Figure III-13 Diagramme d'activité de l'agent Evaluateur (EA)

Le comportement d'une société, composé des agents IA, un agent TSA, des agents TIA, un agent SupA, un agent EA et des agents RA est illustré par le diagramme de séquence de la Figure III-14.

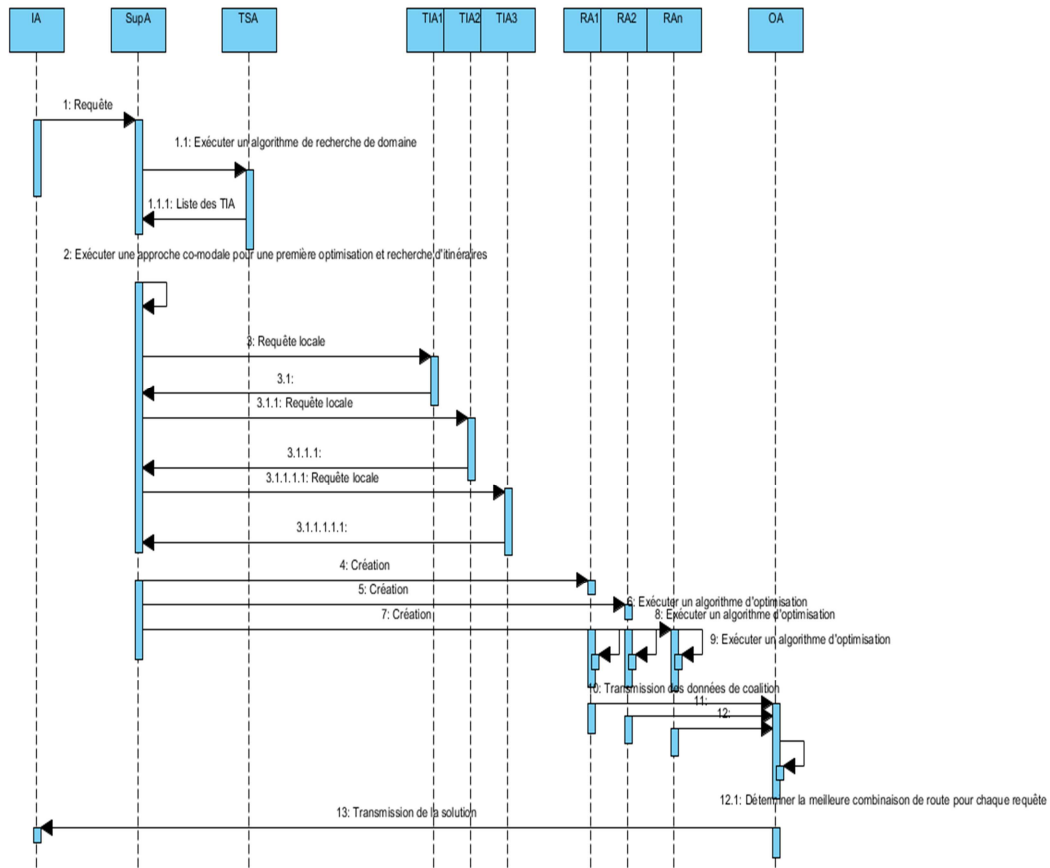


Figure III-14 Comportement du système SITCoMo

Ainsi, la recherche et combinaison des itinéraires optimisés est assurée sur deux niveaux. Un premier niveau concerne l'identification des routes qui peuvent faire partie des solutions que le système va proposer et la recherche des plus courts chemins dans un graphe co-modal. Le deuxième consiste en une approche évolutionnaire appliquée afin de déterminer des solutions

optimisées en termes de plusieurs critères selon les préférences et les contraintes des utilisateurs.

Nous allons dans ce qui suit présenter le premier niveau d'optimisation indispensable pour la résolution du problème.

III.5. Algorithmes de recherche et de composition d'itinéraires

Le but du système de SITCoMo est de satisfaire et répondre aux requêtes des utilisateurs avec des itinéraires co-modaux optimisés en termes de trois critères : le temps, le coût et les émissions des gaz à effet de serre. Vu la complexité du problème, l'identification des routes, la recherche et la composition des itinéraires, l'optimisation multicritère..., tout cela nous a poussé à adopter une double approche. Une hybridation entre les algorithmes de recherche des plus courts chemins et les algorithmes évolutionnaires nous a semblé indispensable pour fournir des résultats optimisés, pertinents et efficaces selon plusieurs critères.

La résolution du système commence alors par une première approche co-modale dans laquelle nous identifions les différentes routes et nous déterminons des premières solutions optimisées en termes de temps.

La conception de cette première partie nous a permis de déceler deux besoins d'optimisation. Une première optimisation qui permettrait par rapport à une requête d'utilisateur de limiter le domaine de recherche et de le définir par un ensemble d'agents TIAs. Il s'agit de trouver une chaîne minimale, ou des chaînes minimales de TIAs adjacents qui permet ou permettent de relier le TIA associé au point de départ à celui associé au pont d'arrivée de façon à pouvoir par la suite composer un itinéraire global en interrogeant les TIAs d'une chaîne

Une deuxième optimisation consiste à rechercher et à composer une première meilleure solution correspondant à la même requête en interrogeant les différents TIAs retenus. Cela revient à trouver un plus court chemin dans un environnement co-modal distribué.

III.6. Sélection du domaine de recherche

Etant donné que chaque agent de service de transport (TSA_i , $1 \leq i \leq K$, K le nombre total de TSAs) est responsable d'un ensemble d'agents d'information de transport ($TIA_{i,j}$, $1 \leq i \leq K$, $1 \leq j \leq K_i$, K_i le nombre total des opérateurs j du service de transport i), nous représentons l'environnement de chaque service de transport i par un graphe d'adjacence non orienté $Ga_i(Na_i, Ea_i)$.

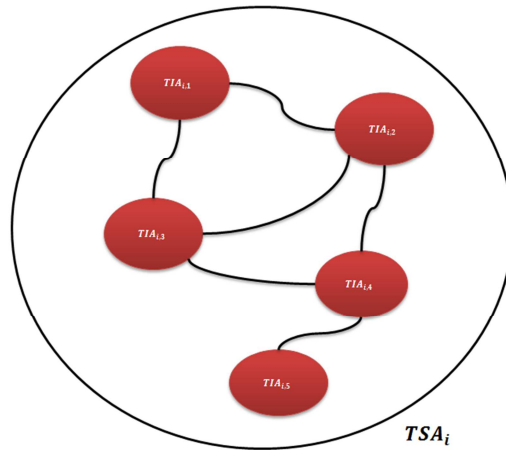


Figure III-15 Exemple d'un graphe d'adjacence pour un service de transport i

Comme le montre la figure III-15, les nœuds Na_i représentent l'ensemble des agents $TIA_{i,j}$, $1 \leq i \leq K$, $1 \leq j \leq K_i$ associés aux différents opérateurs de transport du service i . Les arcs Ea_i représentent les possibilités de passage du réseau d'un opérateur à un autre. Tous les arcs sont munis d'un poids unitaire.

Deux agents $TIA_{i,j}$ et $TIA_{i,k}$, $1 \leq j, k \leq K_i$ sont adjacents s'ils correspondent à des nœuds adjacents sur le graphe $Ga_i(Na_i, Ea_i)$.

On considère pour une requête $I_k(d_k, a_k, W_k)$, $TIA_{i,j}(d_k)$ correspond à l'agent Information de Transport associé au point d_k et $TIA_{i,j}(a_k)$ est celui associé au point d'arrivée a_k . Les différents chemins de Ga_i qui permettent de relier $TIA_{i,j}(d_k)$ et $TIA_{i,j}(a_k)$ constituent l'ensemble des séquences de réseaux de transport qu'un utilisateur pourrait emprunter pour aller de d_k à a_k pour un seul service de transport.

Pour trouver les différentes *routes*, il est inutile d'interroger des opérateurs de transport qui ne peuvent pas répondre aux requêtes. Ainsi, au lieu de transmettre les requêtes à tous les opérateurs existants, il faut minimiser le temps de réponse à toute requête de plus court chemin dans chaque composant. Pour cela, il est nécessaire de minimiser le nombre d'agents $TIA_{i,j}$, $1 \leq j \leq K_i$ à interroger pour participer dans la composition d'itinéraires et ainsi trouver le domaine de recherche indispensable.

Le domaine de recherche pour une requête est défini par le TIA associé au point de départ, celui associé au point d'arrivée et un ensemble de TIAs intermédiaires qui assurent le lien entre les deux premiers.

Pour cela, nous proposons un algorithme de sélection de domaine de recherche (Domain Search Selection Algorithm DSSA) appliqué par chaque agent de service TSA_i et dont les étapes sont les suivantes :

- Etape 1 : L'agent TSA_i identifie au départ les agents $TIA_{i,j}(d_k)$ et $TIA_{i,j}(a_k)$ correspondant respectivement aux points de départ et d'arrivée d_k et a_k . S'il y a différents agents $TIA_{i,j}(d_k)$ et $TIA_{i,j}(a_k)$, nous appliquons les prochaines étapes pour chaque couple $(TIA_{i,j}(d_k), TIA_{i,j}(a_k))$.
- Etape 2 : Tester l'adjacence de $TIA_{i,j}(d_k)$ et $TIA_{i,j}(a_k)$.
- Etape 3 : Si $TIA_{i,j}(d_k)$ et $TIA_{i,j}(a_k)$ sont adjacents alors l'agent TSA_i leur permet de coopérer et peuvent ainsi calculer par leurs propres ressources le plus court chemin.
- Etape 4 : Si $TIA_{i,j}(d_k)$ et $TIA_{i,j}(a_k)$ ne sont pas adjacents, le TSA_i utilise le graphe d'adjacence pour chercher les k plus courts chemins qui correspondent aux k plus courts domaines de recherche, en termes du nombre d'agents $TIA_{i,j}$ à impliquer pour répondre à la requête.

Pour trouver ces k plus courts chemins nous utilisons un algorithme dans la littérature appelé algorithme d'inondation (Flooding algorithm). Il s'agit d'un algorithme qui est très utilisé dans les problématiques de routage dans les réseaux informatiques et dans la découverte de routes entre deux nœuds dans les réseaux Ad Hoc. Le principe de cet algorithme est de partir d'un nœud d'origine $TIA_{i,j}(d_k)$, d'explorer ses successeurs et de sauvegarder en parallèle les chemins explorés, et ce en évitant d'explorer les nœuds déjà retenus dans chacun de ces chemins. L'algorithme récursif s'appelle lui-même pour chaque étape de l'exploration d'un nœud courant $TIA_{i,j}$. L'exploration est arrêtée quand le successeur d'un nœud $TIA_{i,j}$ est le nœud destination $TIA_{i,j}(a_k)$: le chemin obtenu dans ce cas est une solution à retenir. Nous ajoutons une autre condition d'arrêt lorsqu'on aura trouvé tous les chemins constitués au plus, de k arcs, ce qui correspond au nombre maximal de transbordement inter-opérateurs du service de transport.

L'algorithme DSSA est présenté ci-dessous :

| Domain Search Selection Algorithm (DSSA) |
|---|
| In : Nœuds de départ et d'arrivée $TIA_{i,j}(d_k)$ et $TIA_{i,j}(a_k)$ |
| Out : Vecteur chemin contenant la liste des $TIA_{i,j}$ |
| 1. Initialisation : |

- $TIA_{i,j}(d_k)$: Nœud de départ

- $TIA_{i,j}(a_k)$: Nœud d'arrivée

- Path est un chemin vide

2. *Flooding* ($TIA_{i,j}(d_k), TIA_{i,j}(a_k), \text{path}$)

- Ajouter $TIA_{i,j}(d_k)$ à Path

- **Si** ($TIA_{i,j}(d_k) = TIA_{i,j}(a_k)$) and (nombre d'arcs de Path $\leq k$) **Alors** Path est une solution.

- **Pour** $\forall j \in N^+(TIA_{i,j})$ **Faire**

Si $TIA_{i,j}$ n'est pas un nœud de Path

Flooding($TIA_{i,j}, TIA_{i,j}(a_k), \text{Path}$)

$N^+(TIA_{i,j})$: ensemble des nœuds successeurs de $TIA_{i,j}$.

III.7. Identifications des plus courts chemins

Après l'identification du domaine de recherche, les TIAs identifiés vont appliquer des recherches locales dans leurs propres systèmes d'information pour pouvoir déterminer les différents itinéraires globaux. Un itinéraire global est composé d'un ensemble de bouts de routes ou tronçons appelés *routes*. Ces *routes* vont constituer un graphe co-modal. Le but maintenant est de déterminer les plus courts chemins dans ce graphe. Or la résolution d'un tel graphe s'avère assez complexe, ce qui nous pousse à adopter une nouvelle approche appliquée sur un graphe spécifique que nous appelons graphe de transfert.

III.7.1. Définition des graphes

III.7.1.1. Généralités

Un graphe G est défini par $G = (N, E)$ avec :

- N : ensemble d'éléments appelés sommets ou nœuds.
- E : ensemble des éléments appelés « arcs ». Un arc est un couple $(i, j) \in E$ tel que $i, j \in N$.

Il existe deux types de graphes : les graphes non-orientés et les graphes orientés (figures III-16 et III-17)

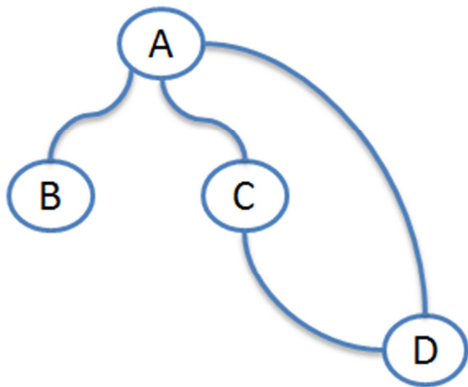


Figure III-16 Graphe non orienté

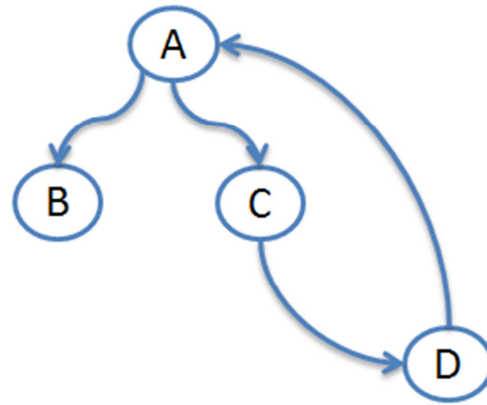


Figure III-17 Graphe orienté

Dans un graphe non-orienté, les deux arcs (i, j) et (j, i) sont équivalents. Par contre, dans un graphe orienté l'arc (i, j) est différent de l'arc (j, i) . Pour un arc (i, j) , i représente le sommet source et j le sommet destination.

Dans un graphe orienté, il est possible de connaître pour un nœud $x \in N$, la liste de ses nœuds prédécesseurs et celle de ses nœuds successeurs.

La liste des nœuds prédécesseurs d'un nœud $x \in N$ est formulée de la manière suivante :

$$N^-(x) = \{y \in N ; (y, x) \in E\}$$

De même, la liste des nœuds successeurs d'un nœud $x \in N$ est formulée de la manière suivante :

$$N^+(x) = \{y \in N ; (x, y) \in E\}$$

Dans un réseau de transport, il est très important de faire la différence entre source et destination et de connaître l'ordre de passage par les nœuds. Donc, nous travaillons toujours avec les graphes orientés.

III.7.1.2. Graphe co-modal

A l'instant t , $G = (N, E, M)$ définit un graphe co-modal ou bien un réseau co-modal avec $N = \{n_1, \dots, n_j\}$ est l'ensemble des nœuds, J est le nombre total de nœuds, $E = \{e_1, \dots, e_L\}$ est l'ensemble des arcs, L est le nombre total d'arcs et $M = \{m_1, \dots, m_K\}$, représente l'ensemble des services de transport (par exemple transport public, covoiturage, autopartage...), K est le nombre total de services. Un arc $e_l \in E$ avec $l \in \{1..L\}$ peut être identifié par $(n_p, n_q)_{m_r}$ où $n_p, n_q \in N$ et $m_r \in M$ avec $p, q \in \{1 \dots J\}$ et $r \in \{1..K\}$. L'arc e_l exprime qu'il est possible d'aller du nœud n_p vers le nœud n_q en utilisant le service de transport m_r . Une valeur $\omega_{e_l} = \omega(n_p, n_q)_{m_r}$ est associée à chaque arc e_l pour indiquer le poids ou le coût de l'arc.

Définition 1 : Un graphe $G = (N, E, M)$ est un graphe co-modal s'il existe au moins deux services de transport $m_i, m_j \in M$ qui assurent deux routes différentes dans le même graphe. On a alors $(n_p, n_q)_{m_i}, (n_{p'}, n_{q'})_{m_j} \in E, m_i \neq m_j$ avec $i, j \in \{1..K\}$ et $n_p, n_{p'}, n_q, n_{q'} \in N, p, p', q, q' \in \{1..J\}$. Dans le cas où $n_q = n_{p'}$, nous obtenons le cas où deux tronçons de routes successifs sont assurés par deux services de transport différents ou bien aussi $n_p = n_{p'}$ and $n_q = n_{q'}$ ($p, p', q, q' \in \{1..J\}$) où le même arc (n_p, n_q) est servi par deux services de transport m_i et m_j ($m_i \neq m_j$).

Le graphe devient un graphe uni-service si tous les arcs sont servis par un seul service de transport.

Un chemin ou une *combinaison de route* $RC_{n_1, n_l} = (n_1 \rightarrow n_l)$ d'un graphe $G = (N, E, M)$ est une succession d'arcs liant les nœuds n_1 et n_l . On note cette combinaison de route en listant successivement les nœuds qui le composent $RC_{n_1, n_l} = ((n_1, n_2)_{m_1}, \dots, (n_{l-1}, n_l)_{m_k})$ avec $\forall l \in \{1, \dots, J\}, n_j \in N, (n_j, n_{j+1})_{m_i} \in E, m_i \in M$ et $i \in \{1..K\}$. Chaque deux nœuds successifs représentent un arc de graphe.

Une *combinaison de route* $RC_{n_1, n_l} = (e_1, e_2, \dots, e_l)$ est dite co-modale s'il existe deux arcs $e_p, e_q \in E, e_p = (n_p, n_{p'})_{m_i}, e_q = (n_q, n_{q'})_{m_j}, m_i \neq m_j, i \neq j$ et $i, j \in \{1..L\}$ qui sont assurés par deux services de transport différents. Dans le cas où un seul service de transport est inclus dans la *combinaison de route*, elle est dite alors uni-service.

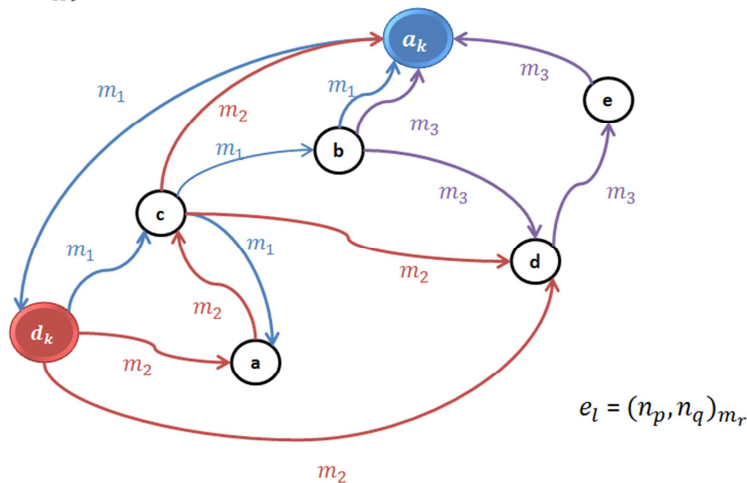


Figure III-18 Graphe co-modal

Définition 2 : Nous définissons pour une *combinaison de route* $RC_{n_i, n_j} = (n_i \rightarrow n_j)$ ou pour un arc $(n_i, n_j)_{m_k}$ dans un graphe co-modal, $i, j \in \{1..L\}$ and $k \in \{1..K\}$, un intervalle de

temps $[t_{n_i}, t_{n_j}]$ avec t_{n_i} est le temps de départ du nœud n_i et t_{n_j} le temps d'arrivée au nœud n_j .

III.7.1.3. Graphe de transfert

Certains chercheurs dans le domaine de transport tel que (Ayed, et al., 2010) ont proposé une nouvelle approche basée sur la notion de graphe de transfert pour résoudre le problème du transport multimodal et dépendant du temps. Dans notre cas, nous nous sommes intéressés à cette nouvelle approche en l'adaptant et en la modifiant pour des graphes co-modaux qui utilisent des services de transport (Jeribi, et al., 2011). Au lieu de diviser notre graphe en un ensemble de modes de transport, nous considérons plutôt un graphe de transfert composé d'un ensemble de réseaux uni-service et un ensemble d'arcs qui relie ces derniers. Le graphe de transfert est défini par $G_T = (C, TR)$ où $C = \{C_1, C_2, \dots, C_K\}$ est l'ensemble de réseaux de transport uni-service qu'on appellera composants et TR est l'ensemble des arcs virtuels qui relient les différents composants. Chaque composant est représenté par un graphe $C_i = (N_i, E_i, M_i, PTC_i), \forall i, j \in \{1, \dots, K\}, M_i \neq M_j$.

Nous avons ainsi $N = \bigcup_{i \in \{1, \dots, K\}} N_i$, $E = \bigcup_{i \in \{1, \dots, K\}} E_i$ et $M = \bigcup_{i \in \{1, \dots, K\}} M_i$. En plus, $TR = \{(n_i, n_j) \text{ tel que } n_i \in C_i, n_j \in C_j, n_i = n_j\}$ avec (n_i, n_j) représente un transfert d'un service de transport m_i vers un autre service m_j ($m_i, m_j \in M$) au *Point de Transfert Co-modal* n_i (ou n_j). $n_i \in PTC_i, n_j \in PTC_j, i, j \in \{1..K\}$ sont appelés *Points de Transfert Co-modal* et représentent la même localisation géographique. On a alors $PTC_i = \{n_i \in C_j \setminus \exists n_j \in C_j \text{ avec } n_i = n_j, i, j \in \{1, \dots, K\}\}$.

Chaque composant $C_i = (N_i, E_i, M_i, PTC_i)$ correspond à un seul service de transport. Dans notre système, nous proposons à nos utilisateurs trois services ce qui nous donne trois composants dans le graphe de transfert $C = \{C_1, C_2, C_3\}$. C_1 représente le service du transport en commun, C_2 correspond au service de covoiturage et C_3 représente les véhicules en libre-service (Vélopartage, autopartage...).

Ce modèle s'adapte avec la nature distribuée du monde réel de l'information du transport puisqu'il sépare les différents modes en différents réseaux uni-modaux ou bien uni-services. Chaque réseau uni-service est alors indépendant et peut être modifié ou mis à jour sans avoir aucun impact sur les autres réseaux.

La figure III-19 illustre un exemple de graphe de transfert avec trois composants C_1, C_2 et C_3 reliés par quatre transferts. a, c, b et d sont les points de transfert. Chaque composant comprend des arcs qui n'appartiennent qu'à un seul service de transport. Dans cet exemple,

nous pouvons aller de d_k à a_k en prenant seulement le transport en commun $RC_{d_k, a_k} = (d_k, c)_{C_1}, (c, b)_{C_1}, (b, a_k)_{C_1}$. Nous avons également une autre possibilité de *combinaison de routes* $RC_{d_k, a_k} = (d_k, d)_{C_2}, (d, e)_{C_3}, (e, a_k)_{C_3}$.

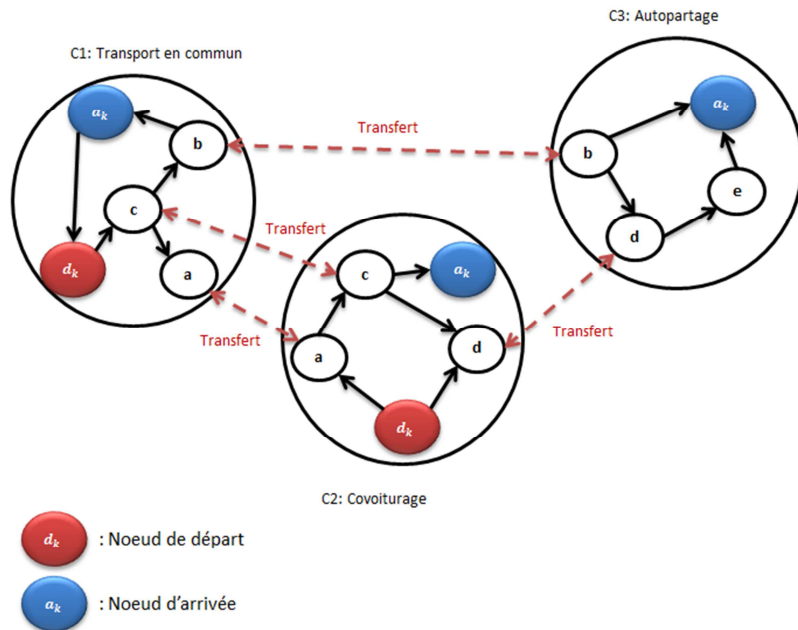


Figure III-19 Exemple de graphe de transfert

L'approche de résolution commence donc avec la construction du graphe co-modal après la localisation des points de départ et d'arrivée de chaque demande d'itinéraire $I_k \in I_t$ et l'identification grâce à l'algorithme DSSA de tous les opérateurs de chaque service de transport qui vont participer à la satisfaction des utilisateurs. Le graphe co-modal est construit à partir des différentes *routes* identifiées par les différents opérateurs qui peuvent répondre aux différentes requêtes I_t . Ensuite, le graphe de transfert est déduit du graphe co-modal.

Il s'agit maintenant de trouver les plus courts chemins dans ce graphe de transfert. Pour cela, nous attribuons un clone de *SupA* pour chaque composant, pour trouver les plus courts chemins uni-service. Cependant, en allant plus dans les détails de chaque composant, chaque service de transport peut être assuré par plusieurs opérateurs. Les bases de données des différents opérateurs de transport sont totalement éparpillées et isolées. Le système est distribué, co-modal et complexe. Nous présentons dans ce qui suit les algorithmes appliqués pour le calcul des plus courts chemins dans chaque composant.

III.7.2. Calcul du plus court chemin dans chaque composant C_i

Dans le graphe de transfert, nous distinguons deux types de chemins ou de *combinaisons de routes*. Des chemins inter-composants et intra-composants. Une combinaison de routes inter-

composants est considérée comme un chemin qui lie deux nœuds $x, y \in N$ où au moins deux arcs de ce chemin appartiennent à deux composants différents. Cependant, un chemin intra-composant inclus dans C_i est un chemin qui relie deux nœuds $x, y \in N_i$ et tous ses arcs sont inclus dans un seul composant C_i . Dans un composant C_i , il est possible d'avoir plusieurs possibilités de combinaisons de routes $RC_{x,y}^i$ qui lient x et y .

Nous distinguons quatre types de chemins intra-composants :

- RC_{d_k, a_k}^{*i} est le plus court chemin qui part du nœud d'origine d_k et arrive au nœud d'arrivée a_k dans la classe C_i .
- RC_{d_k, PTC_i}^{*i} est le plus court chemin qui part du nœud d'origine d_k et arrive à un nœud de transfert PTC_i dans la classe C_i .
- RC_{PTC_i, PTC_j}^{*i} est le plus court chemin qui part d'un nœud de transfert PTC_i et arrive à un autre nœud de transfert PTC_j dans la classe C_i .
- RC_{PTC_i, a_k}^{*i} est le plus court chemin qui part d'un nœud de transfert PTC_i et arrive au nœud d'arrivée a_k dans la classe C_i .

Selon un graphe de transfert $G_T = (C, TR)$, la première étape d'optimisation commence alors par le calcul de toutes les combinaisons de routes : RC_{d_k, a_k}^{*i} , RC_{d_k, PTC_i}^{*i} , RC_{PTC_i, PTC_j}^{*i} , RC_{PTC_i, a_k}^{*i} dans chaque composant C_i .

III.7.2.1. Algorithmes de plus court chemin dans le domaine du transport

Les algorithmes de plus court chemin sont énormément utilisés dans le domaine du transport spécialement pour deux applications du transport public :

- Analyse des performances des réseaux de transport. Une phase de modélisation du réseau et d'analyse de performances doit être établie avant toute conception de réseau de transport. Cette analyse permet de faire un choix sur l'emplacement des lignes de transport, des arrêts et des stations.
- Aide au déplacement et planification des itinéraires dans les systèmes d'information avancés ATIS (Advanced Traveler Information System). Ceci revient à implémenter des algorithmes de plus court chemin sur les systèmes d'information d'aide au déplacement SIAD qui permettent de répondre à des requêtes locales d'itinéraires provenant de différents utilisateurs.

Le problème de plus court chemin consiste à trouver le chemin qui minimise le coût global et ce en partant d'un nœud de départ x vers un nœud d'arrivée y dans un graphe $G = (N, E)$, $x, y \in N$ et tel que les poids des arcs représentent le coût de la course.

Dans la littérature, cette problématique a fait l'objet de nombreuses recherches. Dans ce cadre, plusieurs algorithmes ont été mis en place pour traiter les spécificités des différents types de graphes (graphes cycliques, graphes avec valeurs négatives ...). Ces algorithmes sont souvent classés en deux catégories : les algorithmes de correction d'étiquettes et les algorithmes de fixation d'étiquettes.

III.7.2.2. Les algorithmes de correction d'étiquettes (label Correction Algorithms)

Les algorithmes les plus connus parmi les algorithmes de correction d'étiquettes sont l'algorithme de Bellman-Ford (Bellman, 1958) (Ford, et al., 1962) et l'algorithme de Moore (Moore, 1959). Le principe dans ces algorithmes est d'associer un coût initial à chaque chemin puis de corriger la valeur de ce coût au fur et à mesure du parcours des nouveaux nœuds. La complexité de ces algorithmes est de $O(|N||E|)$.

III.7.2.3. Les algorithmes de fixation d'étiquettes (label Setting Algorithms)

Contrairement aux algorithmes de correction d'étiquettes, le principe de ce type d'algorithmes est un marquage définitif de chaque nœud parcouru. L'algorithme le plus cité dans la littérature de la recherche d'itinéraires est l'algorithme de Dijkstra (Dijkstra, 1959). En effet, ce dernier a été adapté à différents contextes et notamment aux réseaux à grande échelle tels que les réseaux de transport qui présentent un nombre important de nœuds et d'arcs.

Algorithme de Dijkstra

On note :

$\varphi_{d_k i}$: coût du chemin qui va de d_k vers i ; $\omega_{i,j}$: poids de l'arc (i,j)

$pred(i)$: nœud précédent de i ;

$N^+(i)$: ensemble des nœuds successeurs de i ; $N_{current}(i)$: ensemble des nœuds courants.

1. Initialisation :

$\varphi_{d_k i} = \infty$, $pred(i) = \infty \quad \forall i \in N - \{d_k\}$

$\varphi_{d_k d_k} = 0$, $pred(d_k) = d_k$ et $N_{current} = \{d_k\}$

2. Sélection d'un nœud :

Soit le nœud i avec $\min_{j \in N_{current}} (\varphi_{d_k j})$

$N_{current} = N_{current} \cup \{i\}$

3. Exploration des successeurs possibles

$\forall j \in N^+(i)$ Faire

Si $\varphi_{dkj} > \varphi_{dki} + \omega_{i,j}$ **alors Faire**
 $\varphi_{dkj} = \varphi_{dki} + \omega_{i,j}$
 $pred(j) = i$
Si $j \notin N_{current}$ **alors** $N_{current} = \{j\} \cup N_{current}$

2. Critère de fin

$N_{current} = \emptyset$ sinon retour à l'étape 2.

L'implémentation la plus simple de l'algorithme de Dijkstra stocke la liste des sommets dans une liste liée. La sélection du nœud i à la deuxième étape dans ce cas est un simple parcours linéaire de la liste. La complexité de cet algorithme est de $O(|N|^2)$.

Par ailleurs cet algorithme a été aussi adapté pour résoudre des problèmes de plus court chemin dans les graphes dynamiques où le coût $\omega_{i,j}(t)$ de l'arc (i,j) est une fonction du temps de départ t de la station i . Nous sommes plutôt intéressés par la résolution de ce type de graphes vu que le réseau de transport ne peut être représenté que par un graphe dynamique vu son changement et son évolution continuel dans le temps.

Plusieurs travaux de recherche se sont tournés vers la résolution du problème du plus court chemin dans les graphes dynamiques (Chabini, et al., 1997) (Chabini, 1997) (Pallottino, et al., 1997) (Sung, et al., 2000), d'autres plus spécialement se sont intéressés au problème du plus court chemin sur un graphe multimodal monocritère ou multicritères (Bousedjra, 2005) (Abbaspour, et al., 2009) (Zang, et al., 2008) (Zhu, et al., 2008) (Zidi, 2006).

Nous nous basons sur l'ensemble de ces travaux mais sans oublier un aspect très important pour la composition en temps réel d'un itinéraire co-modal global. En effet, en plus du dynamisme du graphe, il faut prendre en compte l'aspect distribué des données provenant des différents opérateurs des différents services de transport. Dans ce cas, les travaux les plus intéressants sont ceux de Wang et Kampke (Wang, et al., 2004) ainsi que (Kamoun, 2007) et (Feki, 2010). Ils ont développé des algorithmes qui permettent de composer un plus court chemin global en ligne à partir de plusieurs réseaux représentés par des graphes statiques. Nous adapterons ces algorithmes à notre problématique de plus court chemin dans chaque composant à partir de différents réseaux du service correspondant représentés par des graphes dynamiques.

III.7.3. Algorithme du plus court chemin sur un graphe dynamique et distribué

III.7.3.1. Modélisation du problème

Dans le domaine du transport co-modal, les réseaux sont dynamiques et dépendants du temps. En effet, que ce soit pour le transport en commun par exemple ou le covoiturage, plusieurs courses peuvent exister entre un point de départ et un point d'arrivée avec différents horaires. A la différence des réseaux de transports statiques, le coût du trajet devient dépendant du temps de départ. Nous nous intéressons par la suite au critère du durée de voyage.

Pour la résolution de cette problématique de recherche et de composition d'itinéraires dans un environnement distribué composé de différents réseaux de transports dynamiques, on cherche le chemin le plus rapide qui mène à destination. Dans ce type de problème, notre objectif est de trouver le temps d'arrivée « au plus tôt » à chaque nœud. Il existe dans la littérature plusieurs algorithmes proposés pour résoudre ce problème.

III.7.3.2. Réseaux dynamiques FIFO et non FIFO (First In First Out)

Nous distinguons deux types de graphes ou de réseaux dynamiques. Un réseau de transport est dit FIFO (First In First Out) si et seulement si pour tout arc (i, j) du graphe, nous vérifions la propriété suivante :

$$\forall t, t' \geq 0 \text{ et } t \leq t' ; t + \omega_{i,j}(t) \leq t' + \omega_{i,j}(t')$$

La figure suivante explique le principe des propriétés FIFO et non FIFO.

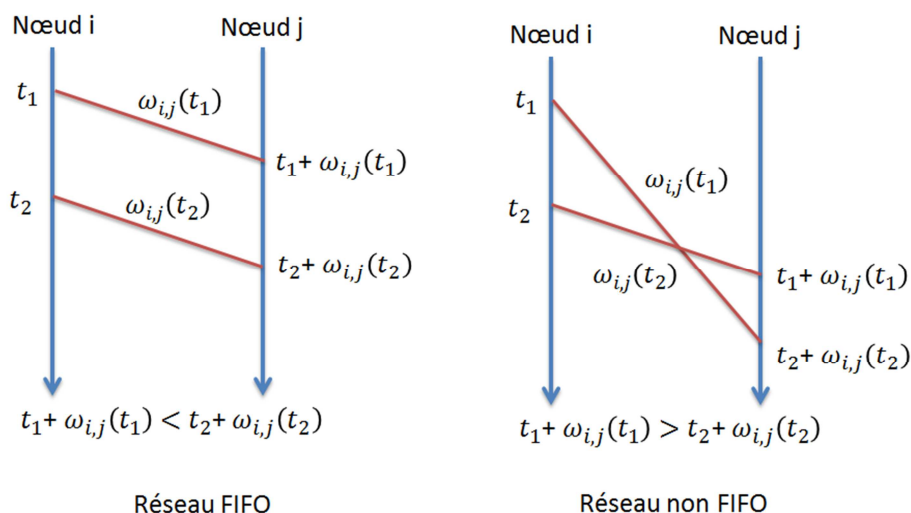


Figure III-20 Différence entre Réseau FIFO et non FIFO

Si la propriété est vérifiée par tous les arcs du graphe, il ne peut exister, à la différence des réseaux non FIFO, deux trajets de i vers j , tel que le premier partant de i , arrive en deuxième à

j. En effet dans les réseaux non FIFO, à une station donnée, il s'avère parfois nécessaire de laisser un bus ou un métro passer et prendre le prochain afin d'arriver au plus tôt à destination. C'est aussi le cas du covoiturage, lors des périodes de pointe, il est préférable de retarder le départ pour éviter les bouchons de circulation et arriver plus tôt à destination.

Pour la résolution des réseaux FIFO, (Chabini, et al., 1997) proposent une adaptation du plus court chemin classique Dijkstra, qui permet de calculer à partir d'un nœud de départ donné, l'ensemble des plus courts chemins vers tous les autres nœuds y compris le nœud d'arrivée souhaité. L'algorithme commence par initialiser les temps d'arrivée au plus tôt $\varphi_{d_k i}(td_k)$ des différents nœuds i ainsi que leurs prédécesseurs $pred_{td_k}(i)$ à la valeur ∞ . $\varphi_{d_k i}(td_k)$ est le temps d'arrivée au plus tôt au nœud i en partant du nœud de départ d_k à un temps de départ td_k .

On définit ensuite l'ensemble des nœuds courants à explorer par l'algorithme. Cet ensemble est initialisé au départ au seul élément nœud de départ $N_{current} = \{d_k\}$. On initialise le temps d'arrivée minimal à ce nœud, ainsi que son précédent, avec $\varphi_{d_k d_k}(td_k) = td_k$ et $pred_{td_k}(d_k) = d_k$.

A partir de l'ensemble $N_{current}$, l'algorithme sélectionne un nœud i à explorer. Le nœud sélectionné est celui qui offre la meilleure heure d'arrivée au plus tôt soit le min de $\varphi_{d_k i}(td_k)$ avec $i \in N_{current}$. Une fois le nœud i , sélectionné, il est retiré de l'ensemble des nœuds courants.

A partir du nœud sélectionné i , on explore tous les nœuds successeurs de i , $N^+(i)$. Pour chaque nœud j accessible à partir de i tel que $j \in N^+(i)$, on met à jour, si nécessaire, la valeur du temps d'arrivée au plus tôt $\varphi_{d_k j}(td_k)$ et du précédent $pred_{td_k}(j)$. En effet, si $\varphi_{d_k j}(td_k) > \varphi_{d_k i}(td_k) + \omega_{i,j}(\varphi_{d_k i}(td_k))$ alors $\varphi_{d_k j}(td_k) = \varphi_{d_k i}(td_k) + \omega_{i,j}(\varphi_{d_k i}(td_k))$ et $pred_{td_k}(j) = i$. Dans ce cas, le nœud j dont la valeur $\varphi_{d_k j}(td_k)$ a été améliorée, sera inséré, s'il n'existe pas déjà, dans l'ensemble $N_{current}$ des nœuds courants. Il peut être aussi sélectionné dans une étape ultérieure.

Ce procédé est répété tant qu'il existe encore des nœuds candidats $N_{current} \neq \emptyset$. Une fois l'algorithme exécuté, le plus court chemin vers le nœud a_k est retrouvé par chaînage arrière.

Concernant les graphes non FIFO, la propriété FIFO n'est plus valable, d'où il vient nécessaire d'avoir plusieurs étiquettes, étant donné que dans ces graphes, il est possible d'arriver plus tôt au nœud a_k , par un chemin qui n'est pas forcément composé de plus courts chemins. Les auteurs de (Chabini, et al., 1997) proposent de modifier l'algorithme précédent,

en insérant dans l'ensemble $N_{current}$ des couples nœuds, temps (i, t) qui permettent de sauvegarder chaque arrivée possible au nœud i . Il s'agit d'une utilisation implicite du modèle espace-temps.

III.7.3.3. Modélisation du graphe dynamique distribué

Un système distribué est composé de plusieurs sous-systèmes autonomes et d'un serveur de calcul central. Un sous-système gère sa propre base de données et il peut répondre aux demandes du serveur de calcul. Ce dernier n'a pas une vue globale des bases de données. Il ne peut qu'envoyer des requêtes aux différents sous-systèmes pour récupérer des informations. Les sous-systèmes peuvent présenter des chevauchements à certains endroits, on parle alors de points d'intersection entre les différents sous-systèmes

Pour modéliser le problème du plus court chemin distribué nous associons à chaque opérateur d'un service C_i , une classe $C_{i,j}$ représentant un opérateur j d'un service C_i avec $1 \leq i \leq K$ et $1 \leq j \leq K_i$; K et K_i sont respectivement le nombre total de services de transport et le nombre total d'opérateurs du service de transport i .

$C_{i,j}$ permet de résoudre le problème du plus court chemin dans un graphe $G_{i,j}(N_{i,j}, E_{i,j})$ où $N_{i,j}$ et $E_{i,j}$ sont respectivement l'ensemble des nœuds et l'ensemble des arcs relatifs à la classe $C_{i,j}$. Un nœud représente une station ou un point de rendez-vous relatifs au réseau de transport d'un opérateur. Un arc représente une course réalisée par un moyen de transport géré par le même opérateur.

Ainsi un arc $(x, y)_{i,j} \in E_{i,j}$ si et seulement si $x \in N_{i,j}$ et $y \in N_{i,j}$ et le transport entre x et y est assuré par un moyen de transport géré par $C_{i,j}$.

Pour un seul composant, le graphe $C_i = (N_i, E_i, M_i, PTC_i)$ est une agrégation des graphes $G_{i,j}$, $1 \leq i \leq K$ et $1 \leq j \leq K_i$.

$$C_i = (N_i, E_i, M_i, PTC_i) ; N_i = \bigcup_{j=1}^{K_i} N_{i,j} \text{ et } E_i = \bigcup_{j=1}^{K_i} E_{i,j}$$

Certes, comme les réseaux de transport qu'ils représentent, les $G_{i,j}$ présentent des nœuds d'intersection relatifs aux points d'échange et qui permettent au cours d'un déplacement de changer d'opérateur de transport.

Par exemple, si x et y sont des nœuds d'intersection des classes $C_{i,l}$ et $C_{i,j}$ telles que : $x \in N_{i,l} \cap N_{i,j}$, $y \in N_{i,l} \cap N_{i,j}$ avec $(x, y)_{i,l} \in E_{i,l}$ et $(x, y)_{i,j} \in E_{i,j}$ alors il y a deux arcs entre x et y dans le graphe $C_i = (N_i, E_i, M_i, PTC_i)$.

Concernant la valorisation des arcs, l'aspect distribué doit figurer. Noter $\omega(x, y)$ pour un arc (x, y) ne suffit pas. En effet, dans un graphe distribué, il est possible d'avoir plusieurs arcs entre deux nœuds x et y , chaque arc appartenant à une classe différente $C_{i,j}$. Il est indispensable alors de faire la différence entre deux poids : $\omega(x, y)_{i,j}$ pour l'arc $(x, y)_{i,j}$ et $\omega(x, y)_{i,l}$ pour l'arc $(x, y)_{i,l}$.

Dans un composant C_i , un itinéraire est défini par un chemin entre deux nœuds de C_i . Nous notons un itinéraire ou chemin entre deux nœuds x et $y \in N_i$, $RC_{x,y}^i$. Il s'agit d'une succession d'arcs liant x à y . Par contre dans un graphe distribué, il est possible d'avoir plusieurs chemins entre deux nœuds qui peuvent être soit des chemins intra-classe (appartenant à une même classe) ou inter-classes (appartenant à des classes différentes). Dans la figure III-22, le chemin (F,s,r,D) est un chemin local de la classe $C_{1,1}$ étant donné que les arcs $(F,s)_{1,1}$, $(s,r)_{1,1}$, $(r,D)_{1,1}$ appartiennent tous à la classe $C_{1,1}$. Par contre, le chemin (A,F,s,r,D,m) est un chemin interclasse vu que $(A,F)_{1,3} \in C_{1,3}$, les arcs $(F,s)_{1,1}$, $(s,r)_{1,1}$, $(r,D)_{1,1} \in C_{1,1}$ et $(D,m)_{1,2} \in C_{1,2}$.

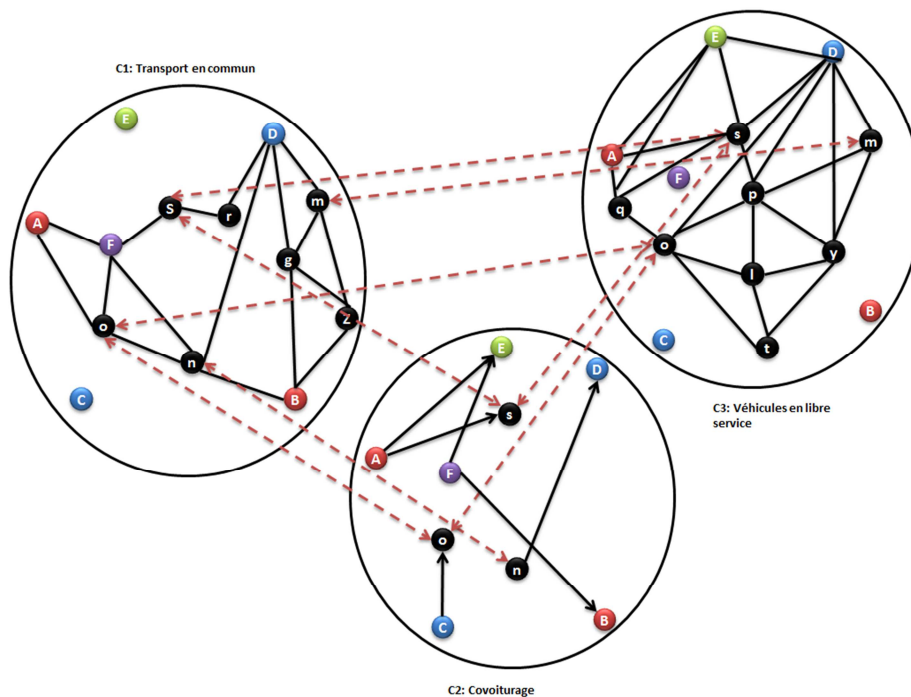


Figure III-21 Graphe de transfert avec trois composants

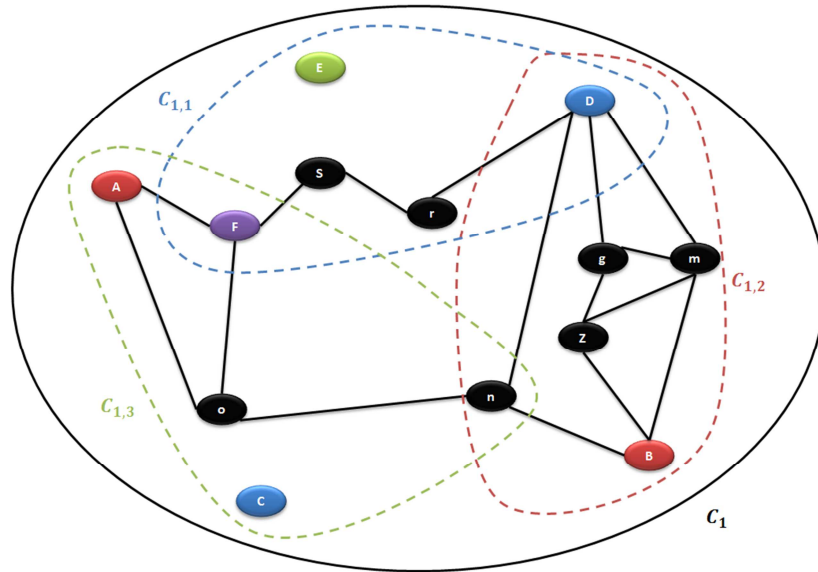


Figure III-22 Graphe distribué d'un composant

Si le graphe $C_i = (N_i, E_i, M_i, PTC_i)$ était centralisé ou connu à l'avance, pour calculer le plus court chemin d'une requête, il suffirait d'exécuter un algorithme de plus court chemin classique tel que Dijkstra sur ce graphe par exemple. Cependant, nous sommes dans le cas de système distribué. Dans ce cas, nous adaptons l'approche des auteurs cités auparavant (Wang, et al., 2004) (Kamoun, 2007) (Feki, 2010) qui se base sur la construction d'un graphe d'intersection complet noté G_{cint} . Ce graphe ne considère que les nœuds d'intersection entre les classes $C_{i,j}$ et les meilleurs itinéraires locaux qui permettent de les relier. L'algorithme de Dijkstra sera par la suite exécuté sur le graphe G_{cint} . Dans la partie suivante, nous allons détailler l'approche et l'algorithme du plus court chemin adopté pour calculer les plus courts chemins à l'intérieur de chaque composant C_i .

III.7.3.4. Algorithme du plus court chemin distribué dans un composant C_i

Le problème consiste à rechercher dans un composant C_i le chemin ayant le coût minimum liant un nœud x à un nœud y tel que $x, y \in N_i$. Ce chemin peut être inter-classes si les arcs qui le composent n'appartiennent pas à une même classe $C_{i,j}$.

Pour la résolution du problème du plus court chemin dans un système distribué, Wang et Kaempke (Wang, et al., 2004) ainsi que Kamoun (Kamoun, 2007) et Feki (Feki, 2010), ont proposé un algorithme qui garantit l'optimalité de la solution trouvée.

Dans leur algorithme, les auteurs font la différence entre un plus court chemin local et un plus court chemin inter-classes. Nous notons $RC_{x,y}^{*i,j}$ le plus court chemin local dans une classe $C_{i,j}$ et $RC_{x,y}^{*i}$ le plus court chemin dans le composant C_i pour aller de x à y .

L'approche se base sur la construction de nouveaux graphes : graphe complet d'intersection et graphe d'intersection étendu. Nous proposons dans la suite d'adapter cette approche pour résoudre le problème de plus court chemin distribué.

III.7.3.5. Le graphe d'intersection complet G_{cint}

Le graphe complet d'intersection $G_{cint}(N_{cint}, E_{cint})$ correspondant au graphe d'un service de transport $C_i = (N_i, E_i, M_i, PTC_i)$ est défini comme suit :

- Les nœuds de G_{cint} sont les nœuds d'intersection des K_i classes de $C_i = (N_i, E_i, M_i, PTC_i)$:

$$N_{cint} = \bigcup_{l=1, j=1}^{K_i} N_{i,j} \cap N_{i,l}$$

- (x, y) est un arc de E_{cint} si et seulement si $x \in N_{cint}$ et $y \in N_{cint}$ et $\{x, y\} \subset N_{i,j}$ ensemble des nœuds d'une classe $C_{i,j}$. Dans ce cas, le coût $\omega(x, y) = \omega(RC_{m,n}^{*i,j})$. Cela veut dire qu'on associe à l'arc (x, y) la meilleure offre locale qui peut être fournie par la classe $C_{i,j}$ pour aller de x vers y . Dans le cas où plusieurs classes contiennent le couple $\{x, y\}$, on note $I(x, y) = \{j \mid x, y \in C_{i,j}\}$ l'ensemble des index de ces classes. Le poids associé à l'arc (x, y) sera la meilleure offre parmi l'ensemble des meilleures offres locales des différents $C_{i,j}$

$$RC_{x,y}^{*i} = \min_{j \in I(x,y)} (RC_{x,y}^{*i,j})$$

Une deuxième étiquette *rec* sera associée à l'arc (x, y) afin de marquer à quelle classe $C_{i,j}$ il est associé. On note :

$$rec(x, y) = l \text{ si } RC_{x,y}^{*i} = RC_{x,y}^{*i,l} \text{ avec } l \in I(x, y).$$

Pour illustrer ce qui a été présenté ci-dessus, nous considérons le graphe distribué de la figure III-22.

Les poids en minutes de chaque arc du graphe sont présentés dans le tableau suivant :

| | |
|-------|--|
| C_1 | $\omega(F, s)_{1,1} = 18, \omega(s, r)_{1,1} = 5, \omega(r, D)_{1,1} = 20$ |
| C_2 | $\omega(n, D)_{1,2} = 30, \omega(n, B)_{1,2} = 20, \omega(B, z)_{1,2} = 5, \omega(z, g)_{1,2} = 8, \omega(z, m)_{1,2} = 15, \omega(m, D)_{1,2} = 12,$ $\omega(m, g)_{1,2} = 5, \omega(g, D)_{1,2} = 15, \omega(B, m)_{1,2} = 5$ |
| C_3 | $\omega(A, F)_{1,3} = 20, \omega(A, o)_{1,3} = 25, \omega(o, n)_{1,3} = 15, \omega(n, B)_{1,3} = 25, \omega(o, F)_{1,3} = 10$ |

Le graphe d'intersection issu du graphe distribué de la figure III-22 est présenté dans la figure suivante.

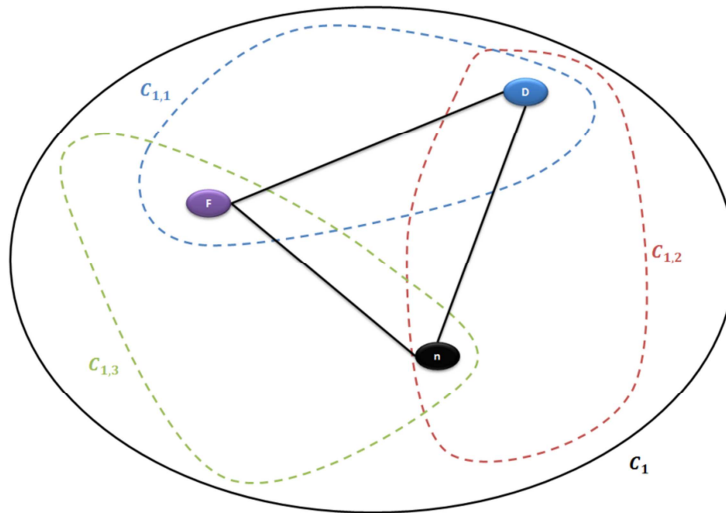


Figure III-23 Graphe d'intersection

Nous calculons les plus courts chemins qui constituent le graphe d'intersection :

$$RC_{F,D}^{*1} = \omega(F, s)_{1,1} + \omega(s, r)_{1,1} + \omega(r, D)_{1,1} = 18 + 5 + 20 = 43$$

$$RC_{F,n}^{*1} = \omega(F, o)_{1,3} + \omega(o, n)_{1,3} = 10 + 15 = 25$$

$$RC_{n,D}^{*1} = \omega(n, D)_{1,2} = 30.$$

Nous passons alors à l'étape de construction du graphe d'intersection étendu.

III.7.3.6. Le graphe d'intersection étendu G_{veint}

Un graphe étendu d'intersection doit être défini pour calculer le plus court chemin global entre deux points de départ et d'arrivée, d_k et a_k . Ce graphe est défini par :

- Un ensemble des nœuds d'intersection des différentes classes $C_{i,j}$ ajouté aux nœuds associés respectivement au nœud de départ d_k et au nœud d'arrivée a_k .
- Un ensemble d'arcs du graphe G_{cint} auxquels on ajoute l'ensemble des arcs relatifs aux plus courts chemins locaux qui permettent de relier d_k et a_k aux nœuds d'intersection des classes relatives à points de départ et d'arrivée.

Dans le but de ne pas avoir une quantité énorme de données, on choisit de construire un graphe virtuel étendu pour chaque requête c'est-à-dire pour chaque couple de nœuds (d_k et a_k). Ce graphe est appelé $G_{veint}(d_k, a_k, W_k)$.

Reprenons l'exemple précédent et calculons le plus court chemin entre les deux points A et B. Le graphe virtuel étendu est représenté dans la figure :

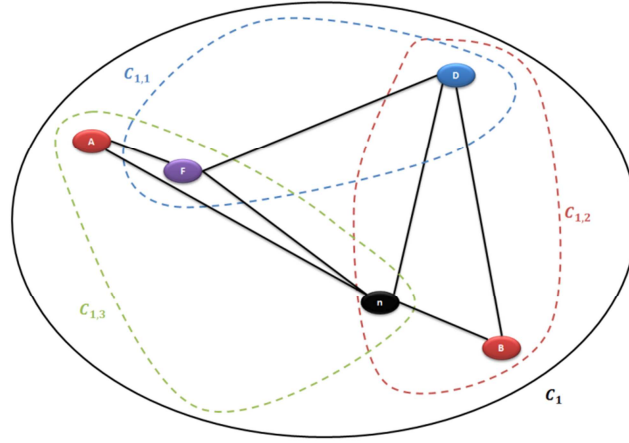


Figure III-24 Graphe étendu virtuel $G_{veint}(A, B, W_{AB})$

Les valeurs des poids des arcs ajoutés par rapport au graphe d'intersection sont calculées de la manière suivante :

$$RC_{A,F}^{*1} = \omega(A, F)_{1,3} = 20$$

$$RC_{A,n}^{*1} = \omega(A, o)_{1,3} + \omega(o, n)_{1,3} = 35 + 15 = 50$$

$$RC_{n,B}^{*1} = D(n, B)_{1,2} = 25$$

$$RC_{B,D}^{*1} = (B, m)_{1,2} + D(m, D)_{1,2} = 5 + 12 = 17$$

Kamoun (Kamoun, 2007) a démontré que tout arc du graphe $G_{veint}(d_k, a_k, W_k)$ vérifie la propriété FIFO par la suite le graphe $G_{veint}(d_k, a_k, W_k)$ est un graphe FIFO. Pour trouver le plus court chemin dans ce graphe il suffit alors d'exécuter un algorithme du plus court chemin FIFO noté SRA (Shortest Route Algorithm).

Algorithme du plus court chemin(SRA)

On note :

$\varphi_{d_k i}$: coût du chemin qui va de d_k vers i ; $\omega_{i,j}(t)$: poids de l'arc (i,j) à l'instant t

$pred(i)$: nœud précédent de i ;

$N^+(i)$: ensemble des nœuds successeurs de i ; $N_{current}(i)$: ensemble des nœuds courants.

3. Initialisation :

$$\varphi_{d_k i}(td_k) = \infty, pred_{td_k}(i) = \infty \text{ et } index(i) = \infty, \forall i \in N - \{d_k\}$$

$$\varphi_{d_k d_k}(td_k) = td_k, pred_{td_k}(d_k) = d_k \text{ et } N_{current} = \{d_k\}$$

4. Sélection d'un nœud :

Soit le nœud i avec $\min_{j \in N_{current}}(\varphi_{d_k j}(td_k))$

5. Exploration des successeurs possibles

$\forall j \in N^+(i)$ Faire

Si $\varphi_{d_k j}(td_k) > \varphi_{d_k i}(td_k) + \omega_{i,j}(\varphi_{d_k i}(td_k))$ et $\varphi_{d_k i}(td_k)$ alors Faire

$$\varphi_{d_k j}(td_k) = \varphi_{d_k i}(td_k) + \omega_{i,j}(\varphi_{d_k i}(td_k))$$

$$pred_{td_k}(j) = i = i$$

$$index(j) = \text{Class Index}$$

$$\text{Si } j \notin N_{current} \text{ alors } N_{current} = \{j\} \cup N_{current}$$

Fin Si

6. Critère de fin

$N_{current} = \emptyset$ sinon retour à l'étape 2.

L'algorithme global qui calcule le plus court chemin distribué dans un composant et qui fait appel à l'algorithme SRA est décrit dans ce qui suit.

Algorithme distribué du plus court chemin(DSRA)

Étape 1 : Construire le graphe complet d'intersection G_{cint} .

Étape 2 : Construire le graphe étendu virtuel $G_{veint}(d_k, a_k, W_k)$

Étape 3 : Calculer le plus court chemin $RC_{d_k, a_k}^* = (d_k = n_0, n_1, \dots, n_n = a_k)$ en utilisant l'algorithme du plus court chemin (SRA).

Étape 4 : Pour chaque pair de nœuds du plus court chemin $RC_{d_k, a_k}^* = (d_k = n_0, n_1, \dots, n_n = a_k)$ retrouver la classe relative à l'arc (n_j, n_{j+1}) afin de déterminer en détail les *routes* inter-classes associées.

Ainsi pour notre exemple, nous déterminons la classe d'appartenance et le détail correspondant et nous retrouvons que pour la requête de A et B, le résultat est le chemin $RC_{A,B}^{*1} = (A, F, n, B)$ (Figure)

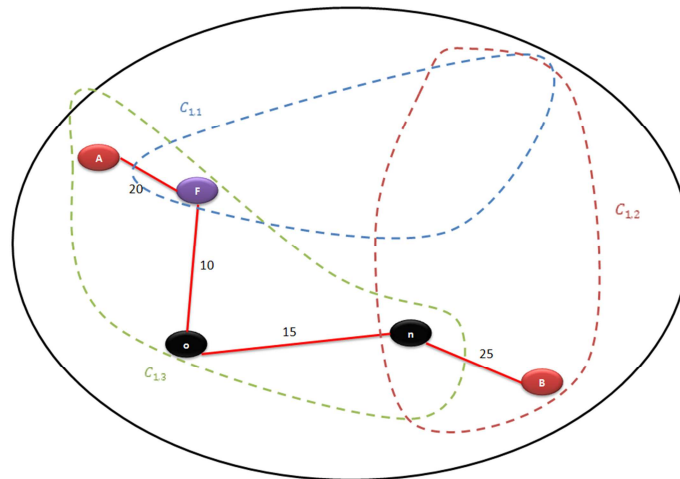


Figure III-25 Le plus court chemin entre A et B dans C_1

III.7.4. Graphe de Transfert de plus court chemin (SPTG)

Comme décrit auparavant, calculer le plus court chemin doit être effectué dans chaque composant. Il faut non seulement trouver les plus courts chemins entre les nœuds de départ et d'arrivée de chaque requête (RC_{d_k, a_k}^{*i}) mais aussi ceux entre les nœuds de départ et

d'arrivée et les points de Transfert co-modaux $(RC_{d_k, PTC_i}^{*i}, RC_{PTC_i, a_k}^{*i})$. Les plus courts chemins reliant ces derniers doivent être calculés aussi (RC_{PTC_i, PTC_j}^{*i}) .

Définition 4 : D'après le graphe de transfert $G_T = (C, TR)$, on définit un graphe de transfert des plus courts chemins (SPTG) $G_F = (N_f, E_f)$ avec N_f est l'ensemble des Points de transfert co-modal PTC_i dans chaque composant ainsi que tous les nœuds de départ et d'arrivée d_k et a_k de toutes les requêtes $N_f = \cup_{C_i \in C} PTC_i \cup \{d_k, a_k\}$. L'ensemble E_f rassemble tous les plus courts chemins identifiés dans chaque composant C_i $E_f = \cup RC_{d_k, a_k}^{*i} \cup RC_{d_k, PTC_i}^{*i} \cup RC_{PTC_i, PTC_j}^{*i} \cup RC_{PTC_i, a_k}^{*i} \forall C_i \in C$.

Ainsi, à partir d'un graphe co-modal complexe qui contient des arcs de services et d'opérateurs différent, nous obtenons un graphe plus léger. Cependant, la solution n'est pas finale, étant donné que jusqu'à cette étape aucun itinéraire n'est affecté à aucune requête et les autres critères demandés par les utilisateurs ne sont pas pris en compte.

Nous allons par la suite continuer la composition des itinéraires en transformant chaque arc constituant le graphe SPTG en un chromosome géré par un agent RA. L'application d'une méthode évolutionnaire entre en jeu pour effectuer toutes les affectations possibles et déterminer les meilleures *routes* en termes de trois critères : le temps, le coût et les émissions des gaz à effet de serre. Une coalition de ces RAs permet de déterminer les meilleures *combinaisons de routes* pour chaque requête. Toute cette approche sera détaillée dans le chapitre IV.

III.8. Conclusion

Dans ce chapitre, nous avons présenté les deux premiers niveaux d'optimisation proposés. Ces deux approches ont été intégrées dans un système d'information multi-agent dont le comportement vise à satisfaire les utilisateurs du système en termes de temps, coût et émissions des gaz à effet de serre. Le premier niveau est appliqué par l'agent Service de Transport (TSA) pour l'optimisation du domaine de recherche et la définition de la liste des agents Information de Transport (TIA). Ces derniers vont coopérer pour fournir toutes les données indispensables au système. Une deuxième approche d'optimisation appliquée par le Super Agent (SA) dans le but de déterminer les plus courts chemins sur un graphe spécial appelé graphe de transfert.

L'application de l'algorithme DSRA ne suffit pas pour la détermination des solutions finales optimisées en termes de trois critères. Pour cela, nous procédons au troisième niveau

d'optimisation en adoptant une approche évolutionnaire. Chaque RA créé représente un chromosome dans lequel nous devons affecter une requête à un véhicule assurant cette *route*. Chaque RA exécute un algorithme afin de déterminer son meilleur chromosome. La combinaison des itinéraires finaux pour chaque requête se fait grâce à la coalition de ces RAs optimisés.

IV. Chapitre IV : Approche évolutionnaire pour la recherche et la composition des itinéraires co-modaux

IV.1. Introduction

Dans ce chapitre, nous présentons le troisième niveau d'optimisation de notre solution. En effet, comme expliqué précédemment, notre approche globale s'applique sur trois niveaux d'optimisations. Pour la recherche et combinaisons des itinéraires co-modaux optimisés, il nous a fallu partager l'optimisation en deux étapes. Une première optimisation monocritère que le *SupA* applique sur un graphe de Transfert et qui donne comme résultats un graphe de transfert des plus courts chemins. Ce graphe SPTG est constitué d'un ensemble de *routes*. Pour poursuivre la formation des itinéraires, le *SupA* donne une forme spécifique à chaque *route* gérée par un agent Route. Chaque agent représente une *route* avec ses points de départ et d'arrivée, ses dates de départ et d'arrivée ainsi que les véhicules qui la desservent et les personnes qui peuvent être intéressées par cette *route*. La meilleure affectation de chaque *route* va être déterminée par chaque agent en appliquant une approche évolutionnaire pour commencer ensuite à former les différentes *combinaisons de routes* pour chaque requête. La meilleure combinaison sera ensuite déterminée par l'agent évaluateur (EA).

IV.2. D'une optimisation monocritère vers une optimisation multicritère

Dans le chapitre précédent en décrivant le comportement de Super Agent (*SupA*), nous avons décrit ses trois différents rôles. Le premier rôle consiste à définir le domaine de recherche, le deuxième permet d'identifier un ensemble de *routes* et de calculer les plus courts chemins en termes de temps.

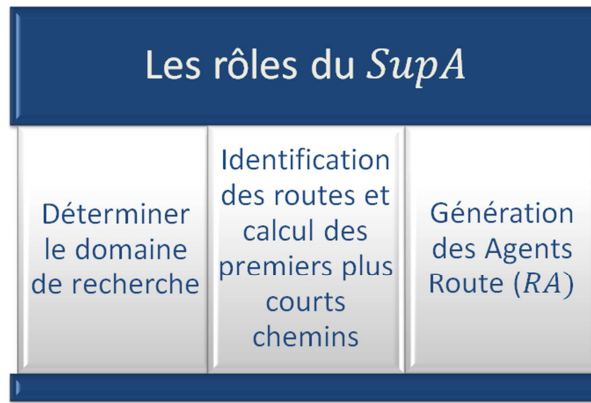


Figure IV-1 Les différents rôles du SupA

Le *SupA* construit un graphe final avec tous les plus courts chemins ou plutôt les plus rapides. Ces chemins sont composés de différentes *routes* ou tronçons. Ils sont considérés comme des solutions préliminaires mais pas des solutions finales. En effet, le but du système est de satisfaire tous les utilisateurs selon différents critères en dépendant de leurs préférences et leurs contraintes. Nous sommes alors face à un problème d'optimisation combinatoire multicritère. Pour le résoudre, nous avons opté pour les algorithmes évolutionnaires qui n'ont cessé de prouver leur performance et leur pertinence pour la résolution des problèmes d'optimisation NP-difficile mono-objectif et multiobjectif. L'approche basée sur l'algorithme évolutionnaire sera alliée évidemment à l'intelligence et l'autonomie des agents. En effet, l'algorithme évolutionnaire sera appliqué sur chaque *route* pour trouver la meilleure affectation en sachant que chaque *route* est représentée par un agent (RA). Une coalition de ces agents permet de former les différentes combinaisons de *route* possibles pour déterminer à la fin la meilleure solution pour chaque requête. L'optimisation multicritère commence alors par la création des RAs par le *SupA* sous une forme spéciale qui sera présentée dans le paragraphe IV.5.1.

Avant de décrire notre approche évolutionnaire, nous présentons dans les paragraphes IV.3 et IV.4 le principe des algorithmes évolutionnaires ainsi que les différentes méthodes de résolution des problèmes d'optimisation multicritère existantes.

IV.3. Les algorithmes évolutionnaires

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, 1992).

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 (Figure IV-2)

- Initialisation de la population π_0 en choisissant P individus dans un espace e recherche Ω , généralement par tirage aléatoire avec une probabilité uniforme sur Ω .
- Evaluation des individus de π_0 : calculer la valeur de la fonction de coût pour chaque individu.
- La génération i construit la population π_i à partir de la population π_{i-1} :
 - Sélection des individus les plus performants de π_{i-1} pour la formation de la nouvelle génération.
 - Application (avec une probabilité donnée) des opérateurs de variation aux parents sélectionnés, ce qui génère de nouveaux individus, les enfants ; on parlera de mutation pour les opérateurs unaires et de croisement pour les opérateurs binaires (ou n-aires).
 - Evaluation des enfants.
 - Remplacement de la population π_{i-1} par une nouvelle population créée à partir des enfants et/ou des vieux parents de la population π_{i-1} au moyen d'une sélection darwinienne (les plus adaptés survivent)
- L'évolution stoppe quand le niveau de performance souhaité est atteint, ou qu'un nombre fixé de générations s'est écoulé sans améliorer l'individu le plus performant.

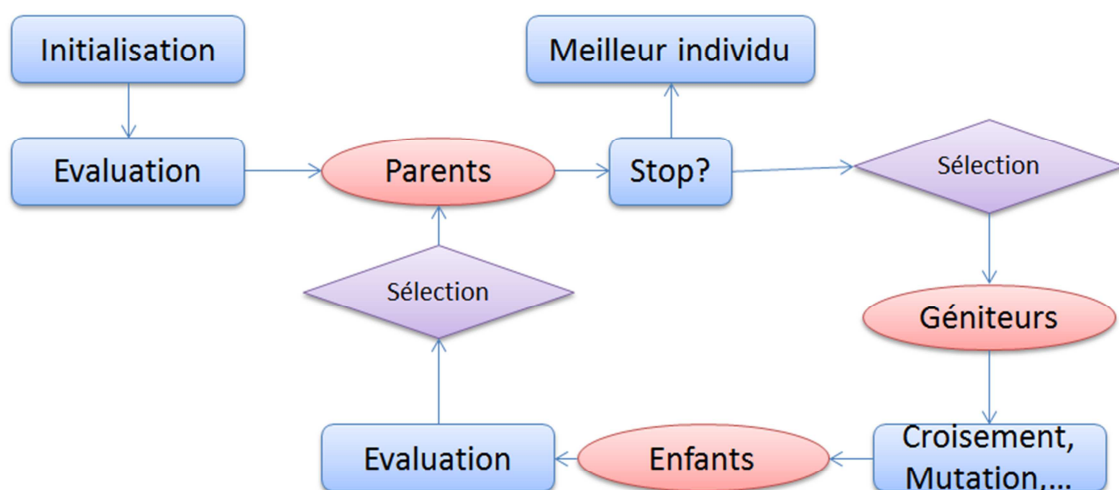


Figure IV-2 Etapes d'un algorithme évolutionnaire

Ainsi, 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. Nous détaillons ces étapes dans ce qui suit.

IV.3.1.Choix d'un codage

Il s'agit de la modélisation d'une solution d'un problème quelconque en un chromosome : une séquence de gènes. 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, et al., 2003). Un génotype représente simplement l'ensemble des valeurs des gènes du chromosome et un phénotype représente la solution d'un problème traduisant ces valeurs contenues dans un génotype (Kacem, 2003). Il existe principalement deux types de codage : le codage binaire (représentation sous forme de chaîne binaire) (figure IV-3) et le codage réel (représentation directe des valeurs réelles de la variable).

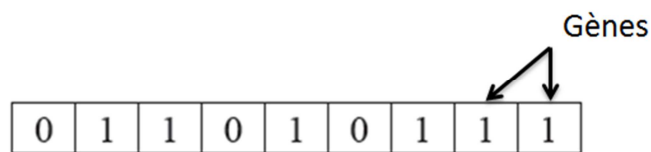


Figure IV-3 Exemple d'un codage binaire

Le codage peut être direct, quand il y a une correspondance bijective entre l'ensemble des chromosomes et celui des solutions qui leur correspondent. Il peut être indirect, quand un générateur doit être utilisé pour définir la solution associée à un chromosome. Il peut également être mixte, lorsqu'il combine les deux codages (Aloulou, 2002).

IV.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, 1998) :

- La sélection déterministe consiste, par exemple, à garder les meilleurs individus au sens de leurs coûts et à rejeter le reste. Si plus de quelques individus doivent être

sélectionnés, cela suppose un tri de l'ensemble de la population – mais cela ne pose un problème de temps calcul que pour des très grosses tailles de population. Les individus les moins performants sont totalement éliminés de la population et le meilleur individu est toujours sélectionné. On dit que cette sélection est élitiste.

- Une sélection stochastique : Il s'agit toujours de favoriser les meilleurs individus, mais ici de manière stochastique, ce qui laisse une chance aux individus moins performants. Par contre, il peut arriver que le meilleur individu ne soit pas sélectionné, et qu'aucun des enfants n'atteigne une performance aussi bonne que celle du meilleur parent . . .
- Une sélection stochastique peut être réalisée par la technique de la roulette pondérée (Goldberg, 1989), 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 $\frac{f_i}{\sum_j f_j}$.

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 la sélection par tournoi (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.

IV.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 recombines 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** : On choisit, 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

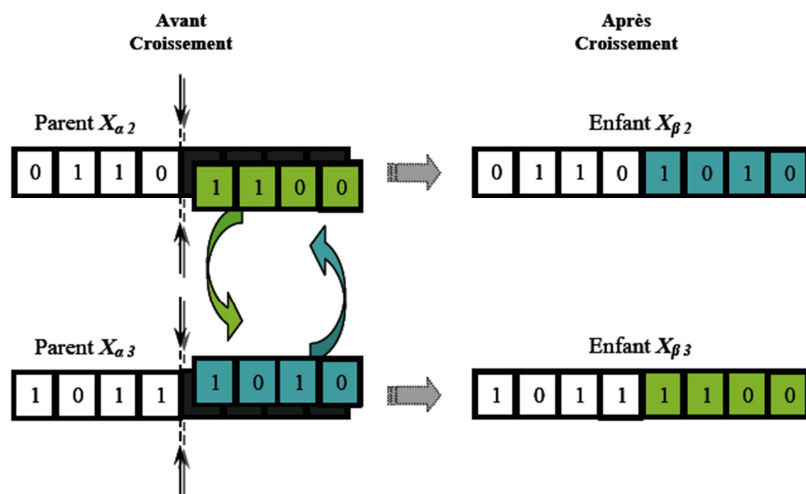


Figure IV-4 Exemple de croisement à un seul point

- **Croisement multi-points** : plusieurs points de croisement sont sélectionnés et il y a un échange des différentes parties d'allèles cernées par ces points, entre les parents.

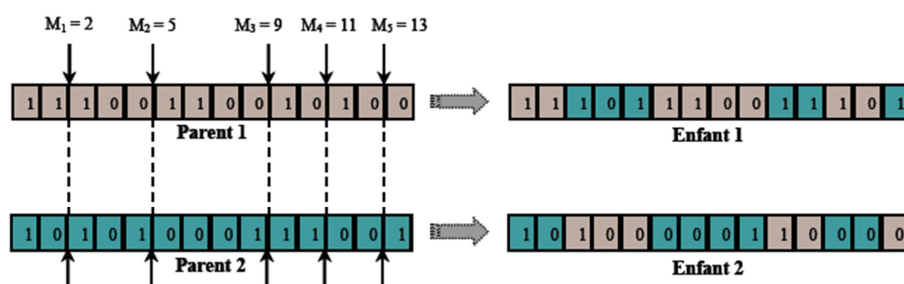


Figure IV-5 Exemple de croisement multi-points

- **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.

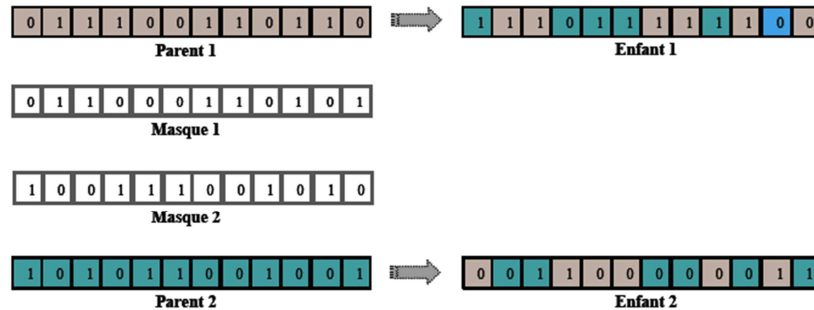


Figure IV-6 Exemple de croisement uniforme

IV.3.4. Opérateur de mutation

La mutation est définie comme étant la modification aléatoire de la valeur d'un gène dans un chromosome. Elle n'apporte pas forcément d'améliorations néanmoins, elle permet de faire évoluer les générations des individus et de les diversifier. Cette diversification a pour but d'éviter la stagnation de la population et la convergence prématurée vers un minimum local.

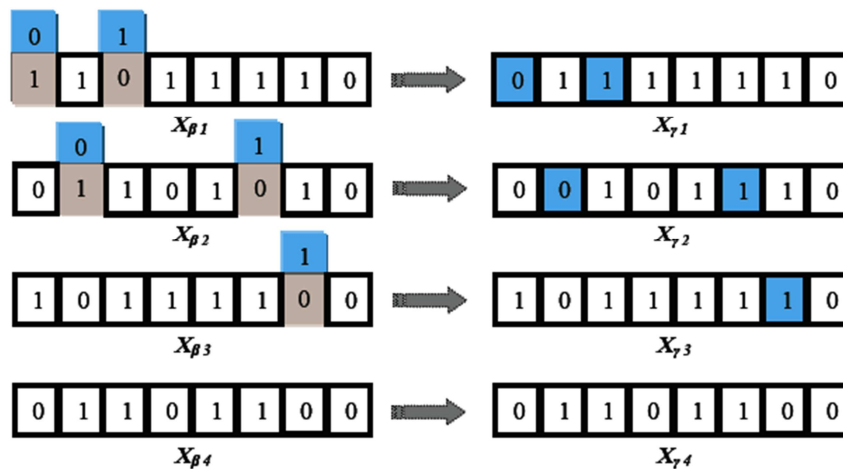


Figure IV-7 Exemple de mutation

IV.3.5. Convergence des algorithmes évolutionnaires

Les algorithmes évolutionnaires présentent l'avantage d'être modulaire étant donné que leur application ne nécessite pas une connaissance de la particularité du problème étudié. Cependant, le choix des paramètres tels que l'effectif de la population, le nombre maximal de générations, la probabilité de croisement p_c et la probabilité de mutation p_m .

L'effectif de la population et le nombre maximal de générations dépendent de la complexité du problème. Il est clair que choisir une population de faible effectif conduira à l'obtention d'un minimum local et donc à une convergence prématurée. Par contre, une grande population engendrera un temps de calcul excessif. Il s'ensuit que le bon choix se fait par la détermination d'un bon compromis. De même, le nombre maximal des générations doit représenter un compromis de la qualité des solutions et du temps de calcul.

Dans la littérature, une règle générale a été proposée pour la détermination des deux probabilités p_c et p_m en attribuant une petite probabilité de l'opérateur de mutation (exemple $p_m = 0.05$ et $p_c = 0.95$). En effet, l'étude de l'influence de ces taux montre qu'avec une probabilité très élevée, la convergence est perturbée en induisant une oscillation de la valeur moyenne du critère optimisé (Mesghouni, 1999). En revanche, un faible taux de mutation permet d'assurer une bonne exploration de l'espace de recherche sans d'autant perturber la convergence.

IV.4. L'optimisation multicritère et les algorithmes évolutionnaires

D'après un grand nombre de travaux, les algorithmes évolutionnaires et plus précisément les algorithmes génétiques sont très bien adaptés au traitement des problèmes d'optimisation multiobjectif. La difficulté d'un problème multiobjectif est qu'il n'existe pas de définition d'une solution optimale. Le décideur exprime seulement qu'une solution est préférable, mais en général il est difficile d'identifier une solution meilleure que toutes les autres.

Dans la notion de dominance, une solution est jugée meilleure qu'une autre si les valeurs obtenues selon tous les critères sont meilleures. Une solution qui domine une autre par rapport à quelques objectifs et qui est dominée par rapport aux autres, n'est pas comparable à celle-ci. Une variante de cette approche considère qu'une solution est meilleure qu'une autre si le nombre d'objectifs où elle la domine est supérieur au nombre d'objectifs où elle est dominée par celle-ci. Cette notion de dominance est illustrée dans la figure suivante :

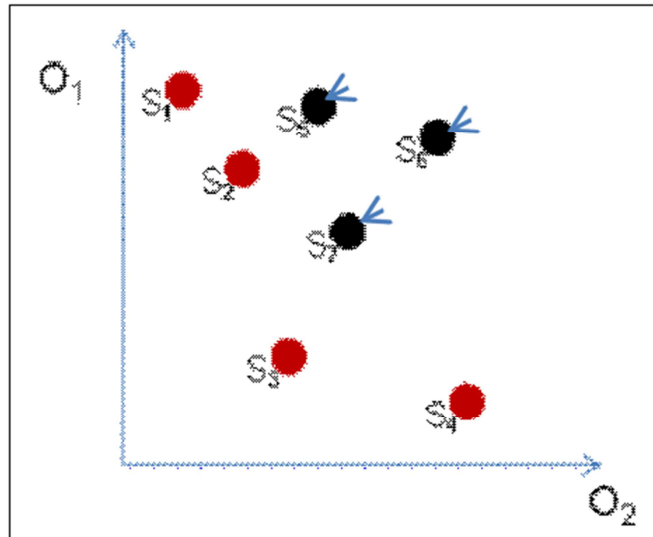


Figure IV-8 Illustration d'un ensemble de solutions dans l'espace des objectifs

Dans cette figure, les solutions 1, 2, 3 et 4 dominent les solutions 5, 6 et 7. Les solutions de l'ensemble dominant n'étant pas comparables entre elles représentent des optima et constituent le front de Pareto.

Ainsi, une relation de dominance est formulée mathématiquement de la manière suivante :

Le vecteur \vec{x}_1 domine un autre vecteur \vec{x}_2 si les deux conditions suivantes sont satisfaites :

- \vec{x}_1 est au moins aussi bon que \vec{x}_2 dans tous les objectifs,
- \vec{x}_1 est strictement meilleur que \vec{x}_2 dans au moins un objectif.

Nous allons dans ce qui suit nous intéresser aux diverses méthodes qui ont été mises en œuvre pour le traitement de problèmes d'optimisation multiobjectif.

Deux approches de résolution d'un problème d'optimisation multiobjectif peuvent être distinguées dans la littérature (Colette, et al., 2002) (Roy, et al., 1993) :

- Première approche : consiste à ramener le problème à un problème d'optimisation mono-objectif.
- Deuxième approche : consiste à proposer des solutions ne tenant compte de l'ensemble des critères.

Nous résumons les principales méthodes de résolution dans la figure suivante :

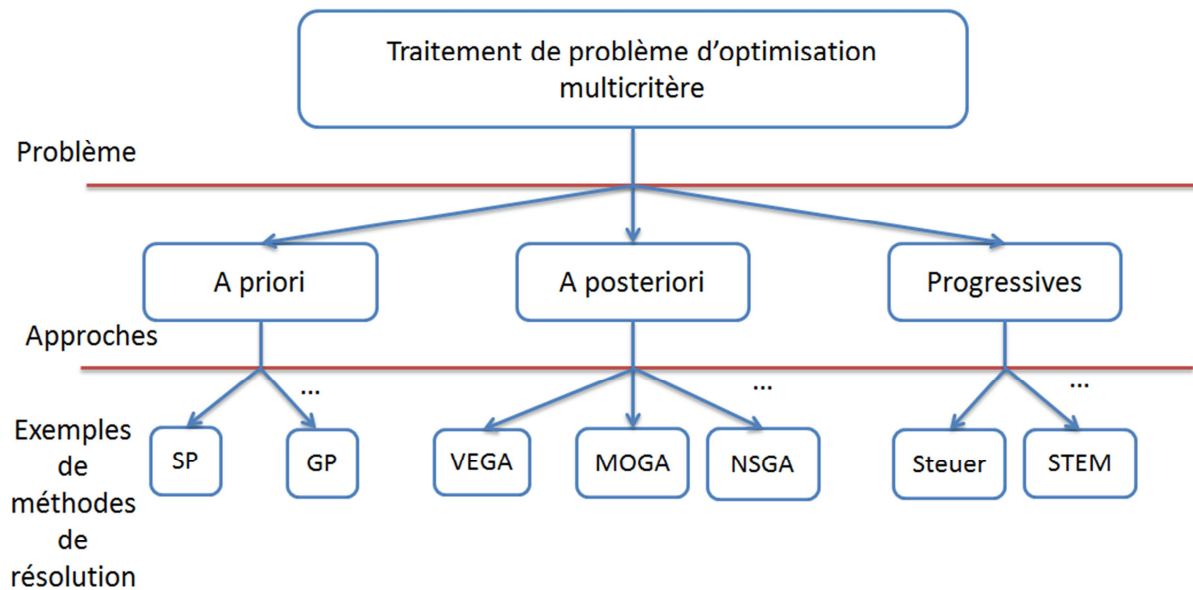


Figure IV-9 Traitement de problème d'optimisation multicritère : Approches et exemples de méthodes de résolution

IV.4.1. Approche de résolution à priori

L'utilisateur définit le compromis qu'il désire réaliser avant de lancer la méthode d'optimisation. Dans cette famille on retrouve la plus part des méthodes par agrégation où les fonctions objectif sont fusionnées en une seule. Parmi les méthodes à priori les plus connues, nous citons la méthode somme pondérée, la méthode par but, la méthode ε -contrainte et la méthode lexicographique.

IV.4.2. La méthode somme pondérée (SP)

Répondue et assez simple à mettre en œuvre, le principe de l'approche somme pondérée est d'affecter aux fonctions objectifs des poids relatifs à l'importance de chacun des objectifs. Une difficulté de cette méthode réside dans le choix des poids en fonction des préférences des experts. Plusieurs travaux issus de la recherche opérationnelle, et notamment la méthode MachBeth (Clivillé, et al., 2007), proposent des méthodes et des outils permettant de déterminer automatiquement ces poids en fonction du classement, par ordre de préférence des experts, de certaines solutions ou bien d'objectifs.

L'interprétation des poids est en général délicate car elle intègre à la fois la notion d'importance relative des poids et le facteur de normalisation des échelles des critères. La logique d'agrégation est compensatoire ce qui peut masquer une forte insuffisance sur un critère donné. De plus, de faibles variations des valeurs de poids peuvent conduire à des solutions radicalement différentes.

IV.4.2.1. La méthode par but

Un ensemble de buts que nous espérons atteindre est défini pour chaque fonction objectif. L'algorithme tente de minimiser l'écart entre la solution courante et ses buts. La méthode détermine la meilleure solution qui s'approche d'une solution dite utopique S^* et qui est fixée au départ par le décideur.

IV.4.2.2. La méthode ε -contrainte

Cette méthode est aussi appelée méthode du compromis (Miettinen, 1998). Elle transforme un problème d'optimisation multiobjectif en un problème d'optimisation mono-objectif de la façon suivante :

- Choisir un objectif à optimiser prioritairement,
- Choisir un vecteur de contraintes initiales ($\vec{\varepsilon}$),
- transformer le problème en gardant l'objectif prioritaire et en transformant les autres objectifs en contraintes d'inégalités comme suit :

Minimiser $\vec{f}(\vec{x})$ avec $\vec{x} \in S_r$ revient à Minimiser $\vec{f}_1(\vec{x})$ avec $\vec{f}_2(\vec{x}) \leq \varepsilon_2, \dots, \vec{f}_n(\vec{x}) \leq \varepsilon_n$ et $\vec{x} \in S_r$.

IV.4.2.3. La méthode lexicographique

Il s'agit de classifier les objectifs par ordre de préférence, de considérer à la suite les fonctions objectif les unes après les autres suivant l'ordre et de minimiser à chaque fois un problème mono-objectif, en complétant au fur et à mesure l'ensemble des contraintes [122].

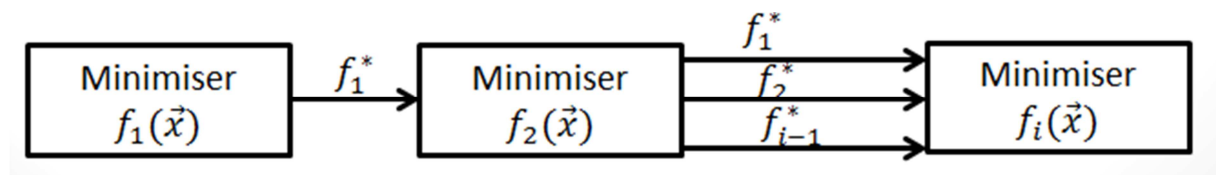


Figure IV-10 Méthode lexicographique

Cette méthode procède en i étapes :

- Etape 1 : Minimiser $f_1(\vec{x})$ avec $\vec{x} \in S_r$
Nous notons f_1^* la solution de ce problème.
- Etape 2 : Minimiser $f_2(\vec{x})$
 $f_1(\vec{x}) = f_1^*$ et $\vec{x} \in S_r$
- ...
- Etape i : Minimiser $f_i(\vec{x})$
 $f_1(\vec{x}) = f_1^*$; $f_2(\vec{x}) = f_2^*$; ... ; $f_{i-1}(\vec{x}) = f_{i-1}^*$

$$\vec{x} \in S_r$$

IV.4.3.Approche de résolution à postériori

Avec une approche à postériori, l'expression des préférences et la pondération des objectifs se fait a posteriori. Le processus d'optimisation détermine un ensemble de solutions candidates et sélectionne la meilleure solution à la fin du processus.

Les approches a posteriori les plus utilisées sont celles basées sur les algorithmes génétiques. Différentes formulations de ces approches, comme VEGA (Vector Evaluated Genetic Algorithm), MOGA (Multiple Objective Genetic Algorithm), NSGA (Non-dominated Sorting Genetic Algorithm) peuvent être trouvées dans la littérature (Colette, et al., 2002).

IV.4.3.1. La méthode Vector Evaluated Genetic Algorithm (VEGA)

Comme elle est décrite dans (Coello, 2000), cette méthode permet de traiter un problème d'optimisation multiobjectif sans avoir à agréger les fonctions objectif en une seule fonction. Elle opère de la manière suivante :

- Etape 1 : Itération i : initialisation d'une population de taille N.
- Etape 2 : Création de k groupes composé chacun de N/k individus. K étant le nombre de fonctions objectif.
- Etape 3 : Calcul des efficacités puis mélange des individus
- Etape 4 : on applique l'algorithme génétique classique (croisement –mutation sélection) puis on passe à l'itération i+1

La figure décrit le principe de l'algorithme VEGA ;

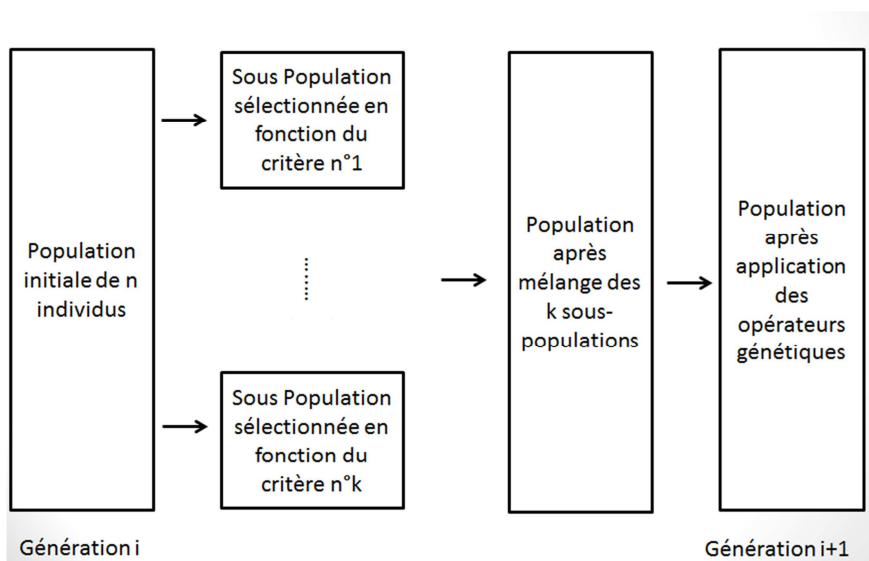


Figure IV-11 Principe de l'algorithme VEGA

IV.4.3.2. La méthode Multiple Objective Genetic Algorithm (MOGA)

Cette méthode est basée sur la dominance au sens de Pareto [123]. Le rang d'un individu qui permet de le classer par rapport aux autres, est donné par le nombre d'individus qui dominent l'individu considéré.

Par exemple, le rang d'un individu x_k à la génération i qui est dominé par $p_k^{(i)}$ est donné par :

$$rang(x_k, i) = 1 + p_k^{(i)}$$

L'algorithme MOGA est décrit ci-dessous :

| Algorithme MOGA |
|--|
| 1. Initialisation de la population |
| 2. Evaluation des fonctions objectif |
| 3. Assignment d'un rang basé sur la dominance |
| 4. Assignment d'une efficacité à partir du rang |
| 5. For $i=1$ to G |
| Sélection aléatoire proportionnelle à l'efficacité |
| Croisement |
| Mutation |
| Evaluation des fonctions objectif |
| Assignment d'un rang basé sur la dominance |
| Assignment d'une efficacité à partir du rang |
| End For |

IV.4.3.3. La méthode Non dominated Sorting Genetic Algorithm (NSGA)

Cette méthode est basée sur une classification en plusieurs niveaux des individus. Dans un premier temps, avant de procéder à la sélection, on affecte à chaque individu de la population un rang (en utilisant le rang de Pareto). Tous les individus non dominés sont classés dans une catégorie, à laquelle on affecte une efficacité factice, qui est inversement proportionnelle au rang de Pareto de la catégorie considérée.

IV.4.4. Approche de résolution progressive

Le principe des méthodes de l'approche progressive est de guider l'exploration tout au long du processus d'optimisation. Le but est d'alterner entre le processus de recherche et le

processus de décision défini par un classement de solutions, ou bien un choix des poids de pondération des objectifs. A chaque étape, l'ensemble des solutions est analysé afin d'orienter les futures itérations vers les zones les plus intéressantes de l'espace de recherche. C'est pour cela qu'elles sont également appelées méthodes interactives. Les méthodes les plus connues de cette classe sont la méthode STEM, aussi appelée STEP et celle de Steuer (Coelho, et al., 2009) (Jones, et al., 2002).

IV.4.4.1. La méthode STEM

La méthode STEM peut être résumée comme étant une suite successive de formulations Min-Max qui minimise la distance entre une solution candidate et une solution théorique S^* . En effet, la prise de décision progressive permet de réduire successivement l'espace des solutions en recentrant les intervalles d'acceptation ainsi que les poids des objectifs. Après une première itération, le décideur peut être satisfait par la solution trouvée. Si ce n'est pas le cas, il peut relaxer certaines contraintes du problème, modifier les poids ou les objectifs eux-mêmes, ou bien rajouter des contraintes supplémentaires au problème en fonction de son analyse de la (ou des) solution(s) retenue(s).

IV.4.4.2. La méthode Steuer

A chaque itération, un ensemble de solutions est proposé à l'expert qui doit en choisir une ou bien les classer par ordre de préférence. Les poids sont alors automatiquement recalculés afin de minimiser la distance entre l'objectif global et la (ou les) solutions choisies. Cette méthode représente une alternative assez intéressante mais coûteuse en temps de calcul et nécessite des outils très spécifiques et une expertise élevée du problème traité. En effet, n'ayant pas la possibilité de revenir sur les décisions précédentes, un changement de préférences ou bien de décideur impliquerait une relance depuis le début de tout le processus d'optimisation.

IV.5. Approche évolutionnaire proposée pour une optimisation multicritère

Dans cette partie, nous considérons les importantes caractéristiques des algorithmes évolutionnaires et leur pertinence à résoudre les problèmes NP-difficiles. Nous présentons quelques points essentiels pour l'approche de résolution adoptée :

- Une représentation génétique (un codage) appropriée à la problématique pour déterminer les solutions possibles du problème d'optimisation ;
- Des opérateurs génétiques qui transforment la composition des enfants lors de la reproduction. Nous rappelons qu'une *route* ou un tronçon d'itinéraire pour une

personne ou une requête doit être servie par un seul véhicule sélectionné à partir de l'ensemble des véhicules qui peuvent desservir cette *route*. Ainsi, nous choisissons de corriger les solutions générées par un autre opérateur : afin de respecter cette contrainte ;

- Les parents sont sélectionnés au hasard à partir de la population courante pour le croisement et la mutation avec une probabilité de croisement p_c ($0 < p_c < 1$) et une probabilité de mutation p_m ($0 < p_m < 1$) ;
- Nous adoptons une technique de remplacement non élitiste pour construire la nouvelle population ;

IV.5.1. Génération des agents Route

Comme expliqué auparavant, chaque arc du graphe final déduit après calcul des plus courts chemins en termes de temps est appelé *route* $R_g(d_g, a_g, W_g)$ et il est représenté par un agent RA. Le *SupA* crée autant de RAs que de *routes* dans le graphe final.

Mais la première question qui se pose : quel sera le choix d'une représentation appropriée à une *route* possible ? Ceci est considéré fondamental pour le succès des applications des algorithmes évolutionnaires. Dans des travaux antérieurs dans notre équipe, (Zgaya, 2007) a proposé une représentation flexible du chromosome appelée Représentation Flexible d'Assignment des Tâches «Flexible Tasks Assignment Representation » (FeTAR). Nous adaptons cette représentation à notre problème et nous proposons un codage efficace pour une *route*, en respectant les contraintes du problème traité. Notre représentation est appelée VeSAR « Vehicle Sharing Assignment Representation » (Jeribi, et al., 2011).

Le chromosome est représenté par une matrice dont les lignes représentent l'ensemble des personnes qui composent globalement les requêtes simultanées I_t et les colonnes représentent les différents véhicules identifiés $V_h^{R_g}$ ($1 \leq h \leq H$) pour servir la *route* $R_g(d_g, a_g, W_g)$. Chaque élément de la matrice indique l'attribution d'une personne $P_p \in P_t$ à un véhicule $V_h^{R_g}$, $1 \leq h \leq H$ comme suit :

$$CH[p, h] = \begin{cases} 1 : & \text{Si } P_p \text{ est affecté à } V_h \\ * : & \text{Si } P_p \text{ peut être affecté à } V_h \\ X : & \text{Si } P_p \text{ ne peut pas être affecté à } V_h \end{cases}$$

Une personne ne peut pas être affectée plus d'une fois aux véhicules et ne peut pas être affectée à un véhicule si ses préférences ou contraintes excluent cette affectation. Par exemple, lorsqu'une personne ne peut pas conduire une voiture d'autopartage, nous prenons

en compte cette contrainte dans le processus d'affectation : nous mettons un **x** pour l'autopartage pour cette personne.

Nous prenons l'exemple de trois requêtes simultanées à $t= 9h15$. Après la génération des routes disponibles, nous considérons que la route $R(x, y, W_{x,y})$ appartient à une combinaison de routes possible pour au moins l'un des trois utilisateurs dans la fenêtre de temps $W_{x,y}=[9h30,10h45]$. Pour cette route correspond un chromosome VeSAR où les lignes correspondent aux personnes concernées par cette route dans la même fenêtre de temps et les colonnes correspondent aux véhicules disponibles pour aller du point de départ vers le point d'arrivée de cette route et dans la même fenêtre de temps.

- User 1 (P_6) : n'aime pas le covoiturage,
- User 2 (P_3) : n'aime pas le transport public,
- User 3 (P_{25}) : ne peut pas conduire un vélo.

Nous présentons ci-dessous une instance d'un chromosome VeSAR. La vie du chromosome commence par une phase de pré- affectation :

Table IV-1 Exemple de chromosome VeSar

| $R(x, y, [9h30, 10h45])$ | $V_{13}(2, 3)$ | Vélo(12) | $V_4(1, 4)$ | Bus ₉ | Autolib(8) |
|--------------------------|----------------|----------|-------------|------------------|------------|
| P_6 | x | * | X | * | * |
| P_3 | * | * | * | x | * |
| P_{25} | * | x | * | * | * |

Une affectation possible est décrite ci- dessous

Table IV-2 Instance de VeSAR

| $R(x, y, [9h30, 10h45])$ | $V_{13}(2, 3)$ | Vlib(12) | $V_4(1, 4)$ | Bus ₉ | Autolib(8) |
|--------------------------|----------------|----------|-------------|------------------|------------|
| P_6 | x | * | X | 1 | * |
| P_3 | 1 | * | * | x | * |
| P_{25} | 1 | x | * | * | * |

A chaque affectation, chaque RA se met à jour et s'auto-évalue en termes de valeurs de critères. La mise à jour concerne essentiellement le nombre de passagers dans les véhicules de covoiturage ainsi que le nombre des véhicules en libre service disponibles (Jeribi, et al., 2009).

IV.5.2.Fonctions d'évaluation du chromosome VeSAR

Un chromosome doit être évalué par trois fonctions vu que le but de l'approche proposée est de résoudre le problème d'optimisation multicritère. Les critères pris en considération sont le

coût du trajet en Euros, le temps total du trajet en minutes et les émissions des gaz à effet de serre en gramme équivalent Carbone.

Pour formuler le coût de chaque *route*, nous définissons tout d'abord la variable binaire :

$X_{V_h}^{R_g}$ = transport de la personne en utilisant le véhicule V_h / $1 \leq h \leq H$ pour la *route* $R_g(d_g, a_g, W_g)$.

$$X_{V_h}^{R_g} = \begin{cases} 1, & \text{si le vehicle } V_h \text{ est utilisé pour assurer la route } R_g \\ 0, & \text{sinon} \end{cases}$$

Soit $C_{V_h}^{1R_g}$, $C_{V_h}^{2R_g}$, $C_{V_h}^{3R_g}$ le coût du transport respectivement en terme de coût, temps et émission des gazs à effet de serre si le véhicule V_h / $1 \leq h \leq H$ est utilisé pour assurer la *route* R_g .

Soit $C_{R_g}^1, C_{R_g}^2, C_{R_g}^3$ sont les différents coûts pour une *route* $R_g(d_g, a_g, W_g)$ qui correspondent à un chromosome CH ; Nous obtenons alors les formules suivantes :

$$C_{R_g}^1 = \sum_{h=1}^H C_{V_h}^{1R_g} \cdot X_{V_h}^{R_g}$$

$$C_{R_g}^2 = \sum_{h=1}^H C_{V_h}^{2R_g} \cdot X_{V_h}^{R_g}$$

$$C_{R_g}^3 = \sum_{h=1}^H C_{V_h}^{3R_g} \cdot X_{V_h}^{R_g}$$

Définissons tout d'abord la formulation de ces différents critères.

IV.5.2.1. Critère coût $C_{R_g}^1$ pour une route R_g

Le coût d'une *route* dépend du véhicule choisi pour desservir cette *route*. En effet, ce critère dépend du type du véhicule décrit dans la section III.2 du chapitre III.

Pour le cas du transport public, le tarif des trajets est unique et connu à l'avance selon l'opérateur de transport. Cela ne nécessite alors aucun calcul.

Pour le cas du covoiturage et de l'autopartage, le tarif d'un trajet ou bien même d'une partie du trajet dépend de la distance parcourue. En effet, comme présentés dans le chapitre I, les coûts des services liés à la notion de véhicule partagé sont souvent calculés en fonction du nombre de Km parcourus.

Ainsi, si nous considérons un véhicule $V_h^{R_g}$ de type A (covoiturage) ou B (véhicules en libre service) alors

$$C_{V_h}^{1R_g} = C_{km} * D(d_g, a_g)$$

Avec C_{km} est le coût de parcours d'un km.

$D(d_g, a_g)$ est la distance entre le point de départ et le point d'arrivée de la *route* en Km.

IV.5.2.2. Critère temps $C_{R_g}^2$ pour une *route* R_g

Le deuxième critère à optimiser est le temps. Dans un système de transport, chercher à se déplacer rapidement est le souci majeur de chaque voyageur. Nous définissons la formulation du temps en fonction du véhicule comme suit :

Soit $T(d_g, a_g)$: le temps de parcours du point de départ d_g au point d'arrivée a_g d'une *route* avec un véhicule $V_h^{R_g}$.

$$T(d_g, a_g) = \frac{D(d_g, a_g)}{V_m(V_h^{R_g})}$$

$D(d_g, a_g)$: la distance entre les deux points d_g et a_g en sachant qu'ils sont définis par leur coordonnées géographiques (longitude et latitude) .

$$D(d_g, a_g) = R * \arccos[(\sin(lat_{d_g}) * \sin(lat_{a_g})) + \cos(lat_{d_g}) * \cos(lat_{a_g}) * \cos(long_{a_g} - long_{d_g})]$$

Avec $R = 6378 \text{ km}$, R est le rayon de la terre. lat_{d_g} et lat_{a_g} sont les latitudes des points d_g et a_g . $long_{a_g}$ et $long_{d_g}$ sont les longitudes des points d_g et a_g .

$V_m(V_h^{R_g})$: Vitesse moyenne du véhicule $V_h^{R_g}$ parcourant la *route* R_g .

Pour pouvoir calculer le temps que prend chaque véhicule $V_h^{R_g}$ pour desservir une *route* R_g , nous allons fixer au préalable la vitesse moyenne de chaque type de véhicule.

Tableau IV-1 Vitesse moyenne de différents véhicules

| Véhicule | Vitesse Moyenne |
|------------------------|-----------------|
| Bus | 40 |
| Métro | 44 |
| Train | 45 |
| Vélo | 10 |
| Voiture de covoiturage | 50 |
| Voiture d'autopartage | 30 |

IV.5.2.3. Critère émission des gaz à effet de Serre (GES) $C_{R_g}^3$ pour une route

R_g

Les transports sont une source de GES du fait :

- du gaz carbonique provenant de la combustion des carburants (pétrole, gaz, GPL, etc.),
- des fuites liées à la climatisation le cas échéant, qui engendrent des émissions d'halocarbures
- des polluants locaux divers, qui peuvent être directement des gaz à effet de serre (oxydes d'azote), ou être des précurseurs de l'ozone, qui est lui-même un gaz à effet de serre (l'ozone des basses couches, encore appelé ozone troposphérique, est responsable d'environ 15% de la perturbation humaine du système climatique).

Le calcul du taux des émissions des GES se fait aussi en fonction de la distance parcourue. Nous formulons ce critère de la manière suivante :

$$C_{V_h}^3 = D(d_g, a_g) * Fe(V_h^{R_g})$$

$Fe(V_h^{R_g})$: Facteur d'émission du véhicule par Km

Le facteur d'émission permet d'évaluer les émissions des GES. Il peut être calculé selon plusieurs unités (Mazoue, 2007) :

- Au véhicule.kilomètre (g eq CO2 / VK) : représente l'émission d'un véhicule parcourant 1 km

- A la place.kilomètre offerte (g eq CO₂ / PKO) : représente l'émission d'une place disponible, vide ou occupée, sur 1 km. Le PKO représente l'offre de transport.
- Au voyageur.kilomètre (g eq CO₂ / voy.km) représente l'émission d'un voyageur sur un trajet d'1 km. Le voy.km représente la fréquentation.

L'ADEME propose dans son dernier bilan Carbone de 2010 (ADEME, 2010), plusieurs facteurs d'émissions calculés selon le véhicule de transport que ce soit véhicule de transport public ou bien une voiture particulière.

Pour pouvoir calculer notre critère des émissions des GES, nous présentons les facteurs d'émissions que nous allons prendre en compte.

Commençons par la voiture particulière. Le facteur d'émission prend en considération :

- Emissions liées à la construction de la voiture
- Emissions liées à l'emploi de l'énergie dans les secteurs industriels amont (les équipementiers)
- Emissions liées à la fabrication des matériaux utilisés pour construire une voiture.
- Emissions liées à la combustion.

Le tableau résume les facteurs d'émission pour une voiture par Km et en fonction du parcours.

Tableau IV-2 Facteurs d'émission par voiture et par Km en fonction du parcours

| Type de parcours | Extra Urbain | Mixte | Urbain | Urbain heure de pointe |
|---|--------------|-------|--------|------------------------|
| Emissions par Km parcouru en grammes équivalent Carbone | 59,1 | 69,9 | 88,9 | 97,8 |

Nous passons au transport public, nous présentons également les différents facteurs d'émission des GES pour différents types d'autobus (tableau IV-3), métro, train et tramway (tableau IV-4 et tableau IV-5).

Tableau IV-3 Facteurs d'émission par véhicule par km pour différents types d'autobus

| Type de bus | Fabrication g équi C/ km | Amont g équi C par V.km | Combustion G équi C par V.km | Total g équi C par V. Km |
|-------------------------|--------------------------------|-------------------------------|------------------------------------|-----------------------------|
| Minibus | 17,7 | 11 | 109 | 137,7 |
| Autobus urbain ldf | 16,7 | 41 | 389 | 446,7 |
| Autobus urbain province | 16,7 | 25 | 233 | 274,7 |
| Autocar interurbain | 15,1 | 29 | 277 | 321,1 |

Tableau IV-4 Facteurs d'émission par voyageur pour différents modes de transport public (données RATP 2005)

| Mode de transport | g équi C (Voy.km) |
|-------------------|-------------------|
| Métro | 1,6 |
| RER | 1,6 |
| Tramway | 1,6 |

Tableau IV-5 Facteurs d'émission par voyageur par km par type de train (SNCF 2009)

| Type de train | g équi C (Voy.km) |
|---------------|-------------------|
| TGV | 6,1 |
| TRN | 9,0 |
| TER | 16,3 |
| Moyenne | 6,9 |

Ainsi, à chaque itération un chromosome VeSAR est évalué par un vecteur d'évaluation englobant les trois critères. Notons que dans le cas de covoiturage, certains critères dépendent aussi du nombre de passagers dans la voiture. En effet, le critère des émissions des GES et celui du coût du trajet pour une requête ne sont plus associés au véhicule. Ils doivent être divisés par le nombre de passagers dans la voiture après la mise à jour.

IV.5.3.L'opérateur de croisement

IV.5.3.1. Algorithme de croisement

Le croisement est une combinaison de gènes appartenant à des chromosomes parents donnant naissance à un ou plusieurs chromosomes enfants. Nous proposons un opérateur de croisement qui effectue le croisement de deux chromosomes parents pour donner naissance à deux chromosomes enfants (Algorithme de croisement)

Algorithme de croisement

1. Sélectionner aléatoirement deux chromosomes parents P_1 et P_2 et un véhicule $V_h^{Rg} \in V^{Rg}$
2. Effectuer les affectations des nœuds des chromosomes pour $\forall V_m^{Rg} \in V^{Rg}$
 - Si $h = m$ alors faire**
 - Les affectations des personnes de V_h^{Rg} dans E_1 (respectivement E_2) doivent correspondre aux mêmes affectations de V_h^{Rg} dans P_1 (respectivement P_2)
 - Sinon**
 - Les affectations des personnes de V_m^{Rg} dans E_1 (respectivement E_2) doivent correspondre aux mêmes affectations de V_m^{Rg} dans P_2 (respectivement P_1)

IV.5.3.2. Exemple d'application de l'opérateur de croisement

Nous illustrons l'algorithme que nous venons de présenter par l'exemple suivant.

Nous considérons alors deux chromosomes parents P_1 et P_2 et un véhicule $V_2(3, 4)$ choisis aléatoirement. $V_2(3, 4)$ correspond à la troisième colonne de la matrice qui représente le chromosome. Le chromosome ne manque pas d'être mis à jour par rapport au nombre de places vacants dans la voiture par exemple, ou bien au niveau du nombre des véhicules en libre service disponibles après toute affectation.

| R(x,y, [9h30, 10h45]) | Vélo(12) | V ₆ (1, 3) | V ₂ (2, 4) | Bus ₉ | Auto(7) |
|-----------------------|----------|-----------------------|-----------------------|------------------|---------|
| P ₂ | x | x | x | 1 | * |
| P ₃ | * | 1 | * | x | * |
| P ₆ | * | x | x | * | 1 |
| P ₁₅ | x | * | 1 | x | * |

Parent 1

| R(x,y, [9h30, 10h45]) | Vélo(12) | V ₆ (2, 3) | V ₂ (1, 4) | Bus ₉ | Auto(7) |
|-----------------------|----------|-----------------------|-----------------------|------------------|---------|
| P ₂ | x | x | x | * | 1 |
| P ₃ | * | * | 1 | x | * |
| P ₆ | * | x | x | 1 | * |
| P ₁₅ | x | * | 1 | x | * |

Parent 2

La deuxième étape de l'algorithme de croisement permet de reproduire les affectations de P_1 (respectivement P_2) dans E_1 (respectivement E_2).

| R(x,y, [9h30, 10h45]) | Vélo(12) | V ₆ (1, 3) | V ₂ (2, 4) | Bus ₉ | Auto(7) |
|-----------------------|----------|-----------------------|-----------------------|------------------|---------|
| P ₂ | | | x | | |
| P ₃ | | | * | | |
| P ₆ | | | x | | |
| P ₁₅ | | | 1 | | |

Enfant 1

| R(x,y, [9h30, 10h45]) | Vélo(12) | V ₆ (2, 3) | V ₂ (1, 4) | Bus ₉ | Auto(7) |
|-----------------------|----------|-----------------------|-----------------------|------------------|---------|
| P ₂ | | | x | | |
| P ₃ | | | 1 | | |
| P ₆ | | | x | | |
| P ₁₅ | | | 1 | | |

Enfant 2

Les affectations des autres véhicules Vélo(12), V₆(2,3), Bus₉ et Auto(8) dans E_1 (respectivement E_2) doivent correspondre aux mêmes affectations de Vélo(12), V₆(2,3), Bus₉ et Auto(8) dans P_2 (respectivement P_1).

| R(x,y, [9h30, 10h45]) | Vélo(12) | V ₆ (2, 3) | V ₂ (2, 4) | Bus ₉ | Auto(7) |
|-----------------------|----------|-----------------------|-----------------------|------------------|---------|
| P ₂ | x | x | x | * | 1 |
| P ₃ | * | * | * | x | * |
| P ₆ | * | x | x | 1 | * |
| P ₁₅ | x | * | 1 | x | * |

Enfant 1

| R(x,y, [9h30, 10h45]) | Vélo(12) | V ₆ (1, 3) | V ₂ (1, 4) | Bus ₉ | Auto(7) |
|-----------------------|----------|-----------------------|-----------------------|------------------|---------|
| P ₂ | x | x | x | 1 | * |
| P ₃ | * | 1 | 1 | x | * |
| P ₆ | * | x | x | * | 1 |
| P ₁₅ | x | * | 1 | x | * |

Enfant 2

Après l'application de l'opérateur de croisement, nous devons vérifier si le chromosome obtenu est viable. Un chromosome est viable s'il affecte un seul véhicule à chaque personne. Dans notre exemple, E_2 n'est pas viable puisqu'il affecte deux véhicules V₆ et V₂ à la requête de la personne P₃. Il faut alors remédier à ce problème et trouver une solution pour rendre ce chromosome viable.

IV.5.3.3. Opérateur de correction de l'opérateur de croisement

L'opérateur de correction est nécessaire pour transformer une solution non viable en une solution viable. Une solution non viable n'affecte pas un véhicule donné ou l'affecte plusieurs fois par différents véhicules. Pour un chromosome CH , nous proposons un opérateur de correction décrit par l'algorithme de **Correction_croisement** suivant :

| Algorithme de Correction_croisement |
|--|
| <p>Pour $\forall P_p \in P_t$ Faire Initialiser à 0 le tableau TabIndices de dimension maximale H Initialiser l'indice j à 0 $\forall V_h^{Rg} \in V^{Rg}$ Faire Si $CH[p, h]$ est égale à 1 alors Faire Incrémenter j</p> |

```

TabIndices[j]= p

Si j = 0 alors Faire
  Tirer au hasard un indice l, 1 ≤ l ≤ H/ CH[p, l] =*
  CH[p, l] = 1
Si j > 1 alors Faire
  Tirer au hasard un indice l, 1 ≤ l ≤ H/ CH[p, l] = 1
  Pour  $V_y^{Rg} \in V^{Rg}$  faire :
    Si CH[p, y] = 1 et y ≠ l alors faire
      CH[p, y] =*

```

IV.5.3.4. Exemple d'application pour l'opérateur de correction de croisement

Dans l'exemple précédent nous avons obtenu un enfant E_2 non viable puisque les véhicules V_6 et V_2 sont tous les deux affectés à P_3 .

| R(x,y, [9h30, 10h45]) | Vélo(12) | $V_6(1, 3)$ | $V_2(1, 4)$ | Bus ₉ | Auto(7) |
|-----------------------|----------|-------------|-------------|------------------|---------|
| P ₂ | x | x | x | 1 | * |
| P ₃ | * | 1 | 1 | x | * |
| P ₆ | * | x | x | * | 1 |
| P ₁₅ | x | * | 1 | x | * |

Enfant 2

L'application de l'algorithme Correction_croisement à l'enfant E_2 donne comme résultat deux enfants E_{21} ou E_{22} .

| R(x,y, [9h30, 10h45]) | Vélo(12) | $V_6(2, 3)$ | $V_2(1, 4)$ | Bus ₉ | Auto(7) |
|-----------------------|----------|-------------|-------------|------------------|---------|
| P ₂ | x | x | x | 1 | * |
| P ₃ | * | * | 1 | x | * |
| P ₆ | * | x | x | * | 1 |
| P ₁₅ | x | * | 1 | x | * |

Enfant E_{21}

| R(x,y, [9h30, 10h45]) | Vélo(12) | $V_6(1, 3)$ | $V_2(2, 4)$ | Bus ₉ | Auto(7) |
|-----------------------|----------|-------------|-------------|------------------|---------|
| P ₂ | x | x | x | 1 | * |
| P ₃ | * | 1 | * | x | * |
| P ₆ | * | x | x | * | 1 |
| P ₁₅ | x | * | 1 | x | * |

Enfant E_{22}

IV.5.4.L'opérateur de mutation

IV.5.4.1. Algorithme de mutation

La mutation est un autre opérateur génétique important vu qu'il apporte aux algorithmes génétiques la propriété d'ergodicité de parcours d'espace. Cette propriété indique que l'algorithme génétique sera susceptible d'atteindre tous les points de l'espace d'état, sans pour autant les parcourir tous dans le processus de résolution. L'opérateur de mutation introduit

une certaine variation dans les gènes des individus de la population pour créer un nouveau chromosome modifié.

Nous proposons l'algorithme de mutation suivant :

| Algorithme de Mutation | |
|-------------------------------|--|
| 1. | Sélectionner aléatoirement un chromosome, une personne $\forall P_p \in P_t$ et un véhicule $\forall V_h^{Rg} \in V^{Rg}$ |
| 2. | Si $CH[p, h] = *$ alors Faire Trouver $j / 1 \leq j \leq H$ et $CH[c_p, c_j] = 1$ $CH[p, j] = *$ $CH[p, h] = 1$ Sinon Si $CH[p, h] = 1$ et $\exists j / 1 \leq j \leq H$ et $CH[p, j] = *$ alors Faire $CH[p, j] = 1$ $CH[p, j] = *$ |

IV.5.4.2. Exemple d'application pour l'opérateur de mutation

Nous illustrons l'algorithme précédent par l'exemple suivant :

Nous choisissons aléatoirement un chromosome CH , une personne P_6 et un véhicule Auto :

| R(x,y, [9h30, 10h45]) | Vélo(12) | V ₆ (1, 3) | V ₂ (2, 4) | Bus ₉ | Auto(7) |
|-----------------------|----------|-----------------------|-----------------------|------------------|----------|
| P ₂ | x | x | x | 1 | * |
| P ₃ | * | 1 | * | x | * |
| P ₆ | * | x | x | * | 1 |
| P ₁₅ | x | * | 1 | x | * |

Pour appliquer la mutation nous avons :

$CH[c_3, c_5] = 1$ et $\exists j$ (1 ou 4) avec $CH[c_3, c_j] = *$, il faut alors modifier et affecter $CH[c_3, c_5] = *$ et choisir au hasard j (1 par exemple) et affecter 1 à $CH[c_3, c_1]$. Nous obtenons alors un nouveau chromosome :

| R(x,y, [9h30, 10h45]) | Vélo(11) | V ₆ (1, 3) | V ₂ (2, 4) | Bus ₉ | Auto(8) |
|-----------------------|----------|-----------------------|-----------------------|------------------|---------|
| P ₂ | x | x | x | 1 | * |
| P ₃ | * | 1 | * | x | * |
| P ₆ | 1 | x | x | * | * |
| P ₁₅ | x | * | 1 | x | * |

IV.5.5. Choix du meilleur chromosome

Etant donné que le but primaire de notre système est de satisfaire au mieux les utilisateurs, ces derniers sont invités dès l'expression de leur requête à associer une pondération $\alpha_{k,i}$ pour chaque critère i . Ceci donne en quelque sorte un ordre de priorité des critères qu'ils souhaitent optimiser.

Pour déterminer le meilleur chromosome, nous adoptons une méthode agrégative tel que le coût d'un chromosome est exprimé de la manière suivante :

$$C_{R_g} = \sum_{i=1}^{CR} \alpha_{k,i} \cdot C_{R_g}^i$$

Avec $1 \leq i \leq CR$ et CR est le nombre total de critères à considérer. Dans notre cas, $CR = 3$.

Cependant, les trois critères ont des unités différentes, la relation précédente ne peut pas avoir lieu puisqu'il n'y a aucun sens à sommer le temps, le coût et les émissions des gaz à effet de serre. Une normalisation de critères est alors indispensable.

Pour cela, nous considérons qu'après calcul de chaque critère pour chaque requête, nous pouvons définir la valeur maximale obtenue pour chaque critère. Les valeurs maximales sont notées $C_{R_{gmax}}^1$, $C_{R_{gmax}}^2$ et $C_{R_{gmax}}^3$.

Le coût de chaque chromosome devient ainsi :

$$C_{R_g} = \sum_{i=1}^C \alpha_{k,i} \cdot \frac{C_{R_g}^i}{C_{R_{gmax}}^i}$$

Le meilleur chromosome pour une *route* donné est celui qui minimise C_{R_g} .

Une fois que chaque RA a trouvé son meilleur chromosome, nous obtenons une liste de chromosomes optimisés représentés par des agents. Pour continuer le processus de composition d'itinéraires co-modaux optimisés, tous ces agents vont coopérer et former un ensemble de coalitions pour fournir les solutions finales aux utilisateurs.

IV.6. Coalition des RAs pour la formation des combinaisons de routes

La formation de coalition dans un système multi-agents fait partie des différentes approches qui supportent la notion de collaboration des agents. Les coalitions permettent aux agents de satisfaire des besoins qui requièrent la synergie de compétences de différents agents, comme par exemple, dans le cadre de la résolution de tâches complexes pour lesquelles des agents agissant seuls seraient incapables ou moins efficaces.

Une coalition peut être définie comme une organisation à court terme basée sur des engagements spécifiques et contextuels, ce qui permet aux agents de bénéficier de leurs compétences respectives.

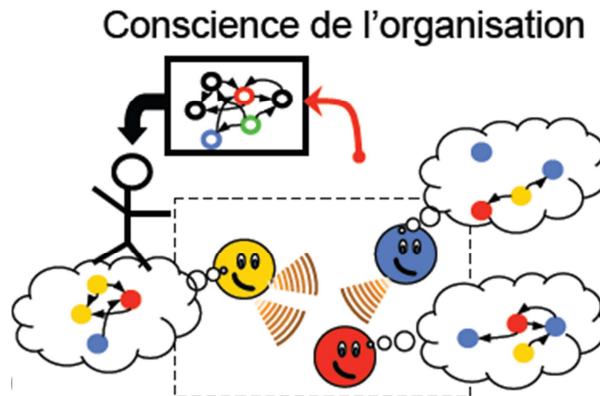


Figure IV-12 Coalition des agents

Pour notre système, nous allons procéder à l'identification des *combinaisons de routes* grâce à la coalition des RAs. Nous présentons dans notre travail, une forme spéciale de coalition puisque les agents vont coopérer ensemble et former des *combinaisons de routes* qui vont être présentées comme solutions aux différents utilisateurs.

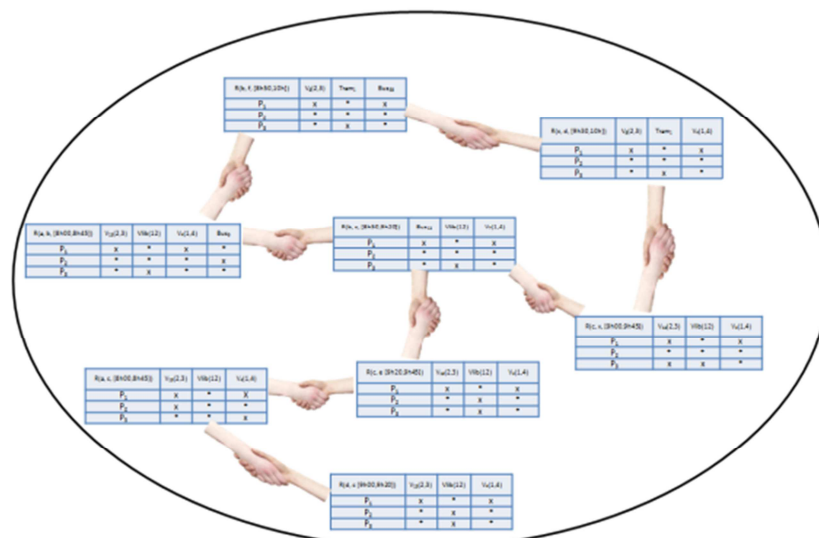


Figure IV-13 Coalition des RAs

IV.6.1. Formulation de la coalition des RAs

A un instant t , nous considérons un ensemble A : ensemble de n RAs, $A = \{RA_1, RA_2, \dots, RA_n\}$. Un ensemble de requêtes sont proposés $I_t = \{I_{1,t}, I_{2,t}, \dots, I_{N,t}\}$ sont proposés à ces agents. Le but de chaque RA est de participer à une ou plusieurs coalitions.

Représentant une *route*, chaque agent RA_i , $1 \leq i \leq n$ est défini par un ensemble d'attributs $R_{RA_i} = \{d_{g_i}, a_{g_i}, td_{g_i}, ta_{g_i}\}$ avec d_{g_i} et a_{g_i} sont respectivement les points de départ et d'arrivée de la *route* correspondante R_{g_i} . td_{g_i} et ta_{g_i} sont également les temps de départ au plus tôt et celui d'arrivée au plus tard de la *route* R_{g_i} .

En plus, comme défini précédemment dans le chapitre II, une demande d'itinéraire est I_k , $1 \leq k \leq N$ est définie par $I_k = \{d_k, a_k, td_k, ta_k\}$.

Un agent RA_i avec $R_{RA_i} = \{d_{g_i}, a_{g_i}, td_{g_i}, ta_{g_i}\}$ peut être une solution complète à une requête $I_k = \{d_k, a_k, td_k, ta_k\}$ si et seulement si :

$$\begin{cases} d_{g_i} \cong d_k, a_{g_i} \cong a_k \\ td_{g_i} \geq td_k, ta_{g_i} \leq ta_k \end{cases}$$

Le terme $d_{g_i} \cong d_k$ veut dire que le point d_{g_i} peut être exactement le point d_k ($d_{g_i} = d_k$) ou bien un point d_{g_i} est un point très proche géographiquement de d_k . De même pour les points d'arrivée a_{g_i} et a_k .

Par contre, si les conditions précédentes ne sont pas valables, l'agent RA_i ne peut pas être, lui seul, une solution à une requête. Il va alors participer à une formation de coalition pour faire partie d'une solution.

Dans ce cas, pour répondre à une demande d'itinéraire I_k , un ensemble d'agents RAs noté $a = \{RA_1, RA_2, \dots, RA_J\}$, $1 \leq J \leq n$ vont coopérer pour former une combinaison de *route* si et seulement si :

$$\begin{cases} d_{g_1} \cong d_k, d_{g_1} \geq td_k \\ a_{g_j} \cong a_k, ta_{g_j} \leq ta_k \end{cases}$$

Définition : Une coalition RC_k est un ensemble de RAs noté a qui forment une combinaison de *route* et donc une solution possible à une requête I_k . Cette coalition est noté $RC_k = \langle a, I_k \rangle$.

IV.6.2. Coalition : protocole d'interaction

Afin de décrire le processus de formation de coalition et la communication entre les agents RAs, il faut absolument définir un protocole d'interaction (Genin, et al., 2010) (Müller, et al., 2006). Nous proposons alors des stratégies de communication qui permettent aux agents de former des coalitions. Chaque coalition forme alors une combinaison de *route* qui peut être une solution possible pour une requête I_k . Un agent peut être l'initiateur et demander aux autres agents de former avec lui une coalition, comme il peut aussi recevoir l'invitation d'autres agents pour former une coalition.

En se référant aux travaux de (Müller, et al., 2006), nous considérons qu'un agent RA peut remplir différents rôles comme décrit dans ce qui suit :

- *Candidat* (Applicant role) : lorsque l'agent RA n'a encore aucun partenaire et attend de former de participer à des coalitions.
- *Initiateur* (Initiator role) : lorsque l'agent RA décide de commencer une coalition et envoie des demandes aux différents agents pour répondre à une requête.
- *Sollicité* (Sollicited role) : lorsque l'agent RA reçoit une proposition de coalition de la part d'un autre agent.
- *Membre* (Member role) : lorsque l'agent RA accepte de participer à la coalition d'un agent initiateur.

Le but de tous les agents RAs est de trouver toutes les combinaisons de *route* possibles pour répondre à un ensemble de requêtes $I_t = \{I_{1,t}, I_{2,t}, \dots, I_{N,t}\}$. Ils adoptent alors le rôle *Candidat* au début du processus, en attente d'entamer ou de recevoir des propositions de coalition.

Le processus de coalition commence par identifier les agents initiateurs. Un agent *Candidat* prend le rôle *Initiateur* si et seulement si son point de départ correspond au point de départ d'au moins une requête. Son temps de départ doit être égal à celui d'une requête ou avec un petit retard noté Δ que nous pouvons tolérer :

$$\begin{cases} d_{g_i} \cong d_k \\ td_{g_i} = td_k + \Delta \end{cases}$$

Dans le cas où $a_{g_i} \cong a_k$ et $ta_{g_i} \leq ta_k$, l'agent *Initiateur* est déjà lui seul une solution à la requête I_k . Il n'a pas alors besoin de chercher à former des coalitions avec d'autres agents pour cette requête.

Dans le cas contraire, l'agent *Initiateur* doit inviter un ensemble de RAs noté a pour former une coalition pour répondre à une requête I_k . $a = \{RA_1, RA_2, RA_J\}$ avec $1 \leq J \leq n$ et J est le nombre d'agents qui vont participer dans cette coalition et qui forment une combinaison de *route* pour la requête.

L'*Initiateur* doit envoyer une proposition de coalition aux autres RAs *Candidats*. Pour optimiser le nombre de messages envoyés, l'agent *Initiateur* demande au SupA la liste des agents intéressants pour cette coalition. En effet, étant donné que c'est l'agent SupA qui a créé les RAs d'après le graphe de transfert de plus court chemin calculé, il maîtrise forcément les relations de voisinage de chaque *route*. Ceci évitera un grand nombre de messages envoyés entre tous les RAs.

Un agent *Candidat* devient *Sollicité* lorsqu'il reçoit une proposition de coalition de la part d'un agent *Initiateur*. La communication entre un agent *Initiateur* et un agent *Candidat* est traduite par le schéma suivant (Jeribi, et al., 2012) :

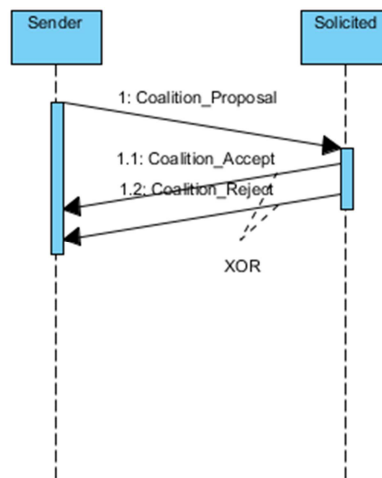


Figure IV-14 Proposition de coalition

L'agent *Sollicité* doit avoir un comportement décisionnel qui lui permet de décider s'il est intéressé par la participation à la coalition proposition ou non.

Un agent *Sollicité* RA_i accepte la proposition de coalition de l'agent RA_j si et seulement si

$$d_{g_i} \cong a_{g_j} \text{ et } td_{g_i} \geq ta_{g_j}$$

Si l'agent *Sollicité* accepte la proposition, il adopte le rôle de *Membre* dans cette coalition. Sinon, au cas où il refuse, il retourne à l'état *Candidat*.

Si l'agent *Initiateur* RA_j reçoit m acceptations de la part de différents RA_i , $1 \leq i \leq m$, la proposition de coalition RC_k sera alors dupliquée en m $RC_{k,p}$, $1 \leq p \leq m$. Ainsi, chaque nouvel agent *Membre* va être ajouté à une coalition $RC_{k,p}$.

Maintenant, pour chaque $RC_{k,p}$, si le point d'arrivée a_k n'est pas encore atteint alors la formation de la coalition doit être poursuivie.

L'Initiateur demande alors au dernier Membre ajouté à l'ensemble a de trouver les *routes* qui doivent compléter la *combinaison de routes*. Ainsi, à son tour, pour chaque $RC_{k,p}$, le dernier membre va envoyer une proposition de coalition et attend la réponse. S'il obtient au moins une réponse positive de la part de RA_j , il informe l'Initiateur de l'acceptation. RA_j devient alors *Membre* de cette coalition et l'Initiateur informe les autres membres de son adhésion.

La figure suivante explique le principe de communication entre Initiateur, Sollicité et Membre

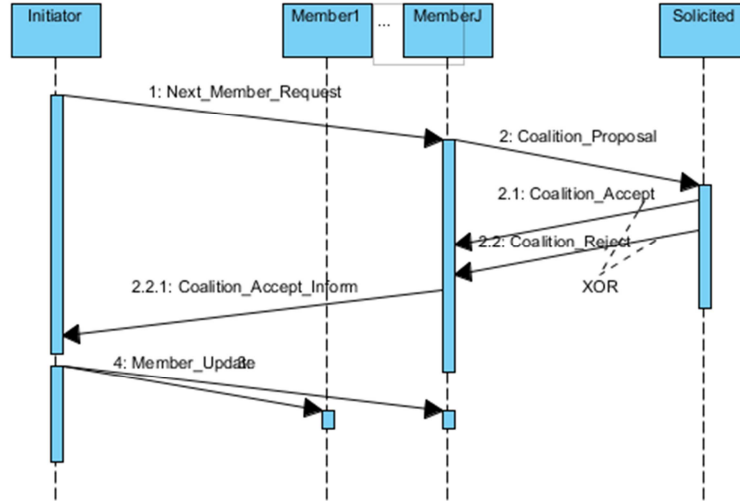


Figure IV-15 Formation de coalition

Le processus continue de la même manière jusqu'à obtenir les différentes *combinaisons de routes*. Une *combinaison de route* est terminée lorsque le point d'arrivée d'une requête est atteint.

Après la coalition des RAs et l'identification des différentes *combinaisons de route* $RC_{k,p}$ pour chaque requête $I_k(d_k, a_k, W_k) \in I_t$, l'EA (agent Evalueur) doit calculer la meilleure.

IV.7. Meilleure combinaison de routes pour chaque requête

L'agent Evalueur (EA) a le rôle de calculer la meilleure *combinaison de routes* pour chaque requête selon l'ordre de préférences donné par l'utilisateur.

Nous définissons tout d'abord la variable binaire :

Y_{R_g} = l'itinéraire incluant R_g pour répondre à la requête $I_k(d_k, a_k, W_k)$.

$$Y_{R_g} = \begin{cases} 1, & \text{si la Route } R_g \text{ est sélectionnée pour faire partie} \\ & \text{de la combinaison de Route } RC_{k,p} \\ 0, & \text{sinon} \end{cases}$$

Etant donné que notre système fournit des itinéraires co-modaux optimisés en termes de trois critères. Soit les coûts $CR_{k,p}^1$, $CR_{k,p}^2$ et $CR_{k,p}^3$ de chaque combinaison de routes relatifs respectivement au coût, temps et émissions des GES.

Le coût $CR_{k,p}$ de chaque *combinaison de routes* $RC_{k,p}$ est calculé de la manière suivante :

$$CR_{k,p}^1 = \sum_{R_g} C_{R_g}^1 \cdot Y_{R_g}$$

$$CR_{k,p}^2 = \sum_{R_g} C_{R_g}^2 \cdot Y_{R_g}$$

$$CR_{k,p}^2 = \sum_{R_g} C_{R_g}^2 \cdot Y_{R_g}$$

L'EA considère que la meilleure *combinaison de routes* pour chaque requête est celle qui optimise au mieux le critère préféré de l'utilisateur en se basant sur son ordre de priorité exprimé au début lors de la formulation de sa demande d'itinéraire.

IV.8. Conclusion

Dans ce chapitre, nous avons présenté une approche d'optimisation multicritère pour la combinaison des itinéraires co-modaux en termes de coût, temps et émissions des GES. L'approche adoptée est intégrée dans un système d'information multi-agent. Encore une fois, nous avons profité de faire une alliance entre les systèmes multi-agents et les méthodes d'optimisation pour un système plus performant et plus efficace. L'approche évolutionnaire adoptée est appliquée par différents agents en même temps, chaque agent cherche à trouver sa meilleure valeur. Ensuite, la coalition de ces agents permet de construire les différentes *combinaisons de routes* et obtenir ainsi des itinéraires co-modaux optimisés selon les préférences des utilisateurs.

La formulation du problème définie et nos algorithmes fixés, la concrétisation de nos objectifs est complétée par une implémentation de ceux-ci. Nous présentons alors dans le chapitre suivant les détails relatifs aux outils utilisés pour le développement du système SITCoMo. Différents jeux de tests ainsi que des détails sur le déploiement y sont fournis aussi.

V. Chapitre V : Implémentation et Scénarios du système SITCoMo

V.1. Introduction

Dans ce dernier chapitre, nous présentons les différents résultats issus de nos approches d'optimisation, intégrées dans l'architecture distribuée proposée. Nous proposons alors d'exposer des scénarios de simulation de ces approches qui détaillent le fonctionnement du système SITCoMo face à différentes situations.

Nous commençons tout d'abord par argumenter le choix de l'outil informatique ainsi que la plateforme multi-agents utilisés pour l'implémentation du système. Ensuite, nous décrivons l'aspect graphique du système au niveau de l'interface utilisateur et l'interface de présentation des solutions. Enfin, la dernière partie de ce chapitre est consacrée à l'implémentation et simulation des algorithmes de système SITCoMo.

V.2. Choix de l'outil informatique

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 subtiles 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 obtenue, 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*).

V.3. Choix de la plateforme Multi-agents

La composition d'itinéraires multi-opérateurs sous-entend la disponibilité des données de chaque opérateur ce qui n'est pas toujours le cas. L'approche que nous proposons vise principalement à surmonter cette contrainte en intégrant non pas 'les données de chaque opérateur' mais 'un accès vers son système d'information'. Une telle stratégie nous permet de récupérer les données nécessaires non pas à partir de la base de l'opérateur, mais à partir de son SIAD. Une telle intégration nous permet de profiter des calculateurs de chaque opérateur pour des déplacements locaux. De plus cette stratégie met la responsabilité de mise à jour des horaires et l'indication des perturbations sur les opérateurs eux-mêmes. En effet, en mettant à jour en temps réel 'les horaires', 'les coupures' et 'les retards', notre système répond directement avec la meilleure offre disponible.

De ce fait, le problème de recherche d'itinéraires entre deux stations appartenant à des opérateurs différents est ramené à un problème de recherche d'un ensemble d'opérateurs séparant les opérateurs de départ et d'arrivée. En effet, si nous pouvions déterminer l'ensemble des opérateurs impliqués dans le déplacement ainsi que les pôles d'échange (i.e. nœuds d'intersection) entre ces différents opérateurs, la composition d'itinéraire multi-opérateur ou multi-services sera traduite par une concaténation des itinéraires calculés par les processeurs locaux des opérateurs de cet ensemble.

Par ailleurs, l'itinéraire fourni par chaque SIAD local peut ne pas être unique. Ainsi, en faisant varier ces itinéraires locaux, nous obtenons des chemins différents avec des durées différentes, mais passant par un même ensemble d'opérateurs selon un même ordre. Nous utilisons donc les algorithmes décrits dans le Chapitre III pour trouver le plus court chemin dans cet ensemble.

Ces algorithmes sont mis en place sous forme d'une communication entre agents. Ceux-ci sont plus adaptés à ce type de problème : en effet, en comparant la notion d'agent à la notion d'objet nous remarquons rapidement que les agents englobent les spécificités des objets et les complètent par plusieurs autres caractéristiques comme le niveau de communication, la richesse des interactions entre les agents (protocole, typologie et ontologie) et la prise autonome de décision. Or, la mise en place d'un tel système nécessite un environnement particulier de développement et des outils bien précis. En recherchant dans les outils disponibles, nous pouvons trouver une variété de plateformes qui permettent la mise en place et le suivi du processus de développement. La plupart de ces outils sont développés avec Java. Nous donnons dans ce qui suit une brève présentation des outils les plus cités et utilisés.

V.3.1. Zeus

ZEUS est un outil développé par British Telecom Intelligent System Research Lab. Il permet le développement des systèmes collaboratifs selon les quatre phases : analyse, conception réalisation et support à l'exécution. La première phase commence par la description des rôles, ensuite l'organisation et enfin la coordination. Cependant, cet outil est assez complexe et difficile à maîtriser.

V.3.2. MadKit

MadKit¹⁶ 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, 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).

V.3.3. Jade

Avant de décrire la plateforme Jade, définissons la norme FIPA¹⁷ qui est un projet de normalisation dans le domaine multi-agent. En effet, vu la diversité des travaux effectués dans la recherche multi-agents, le problème de la normalisation est devenu urgent. Plusieurs projets

¹⁶ <http://www.madkit.org>

¹⁷ <http://www.fipa.org/>

se sont intéressés à ce sujet dans le but unique : faciliter l'interopérabilité entre agents et entre concepteurs.

Les projets les plus connus sont Agent UML (AUML) et FIPA. Nous nous intéressons dans la suite à la norme FIPA (Foundation for Intelligent Physical Agents). Fondée en 1996, la FIPA offre des moyens standardisés pour une meilleure communication entre agents tout en respectant leur sens initial.

JADE¹⁸ (Java Agent Development Framework) a été développé à l'Université de Parme en France. C'est une plateforme JAVA de développement des systèmes multi-agents répondant aux normes FIPA. Jade utilise pour la communication entre agents le langage FIPA ACL. De plus, cette plateforme possède trois modules nécessaires à la conformité pour la norme FIPA.

- Le Director Facilitator (DF) est un module qui fournit un service de pages jaunes à la plateforme.
- L'Agent Communication Channel (ACC) gère les différentes communications entre les agents.
- L'Agent Management System (AMS) supervise l'enregistrement des agents, leur authentification, leur accès et l'utilisation du système.

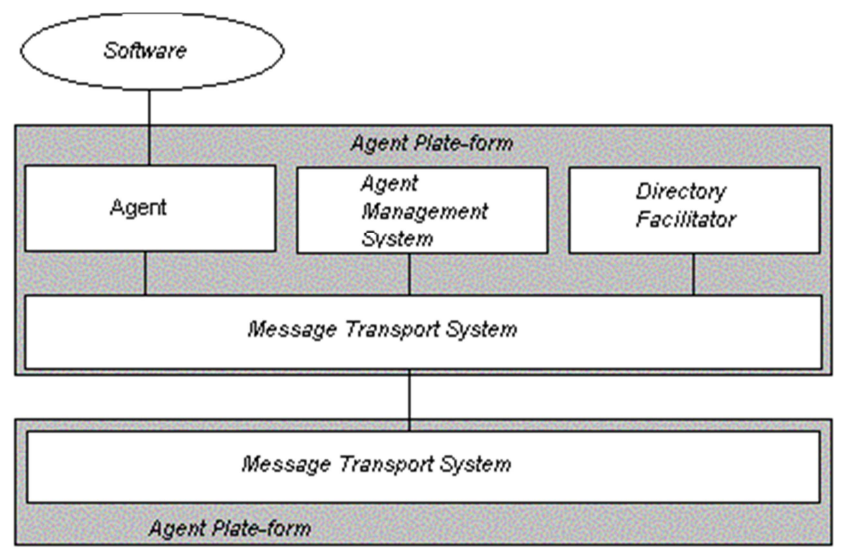


Figure V-1 Architecture logicielle de la paletform JADE

Notre objectif premier est d'assurer une bonne portabilité et une interopérabilité optimale. En effet, au sein de notre laboratoire (LAGIS) nous travaillons dans le but de développer une plateforme générique qui regroupe plusieurs services autour du transport.

La portabilité pourrait être assurée en choisissant une plateforme développée en langage Java. En effet, le langage Java est portable ce qui facilite la portabilité de la plateforme. Or la

¹⁸ <http://jade.tilab.com/>

plupart des plateformes multi-agent sont à base de java, la portabilité est donc assurée dans la majorité des cas.

Nous nous intéressons alors à l'interopérabilité. Pour cela, le choix le plus judicieux doit se conformer à une norme de développement multi-agent. En choisissant la plateforme Jade, nous nous conformons aux normes FIPA. De plus, étant donné que plusieurs travaux dans notre équipe ont adopté cette technologie, nous le choisissons afin de mieux s'intégrer dans l'existant.

V.4. Présentation et aspect graphique

V.4.1. Interface

A sa connexion, chaque utilisateur utilise l'interface des itinéraires pour formuler sa demande. (Figure V-2)

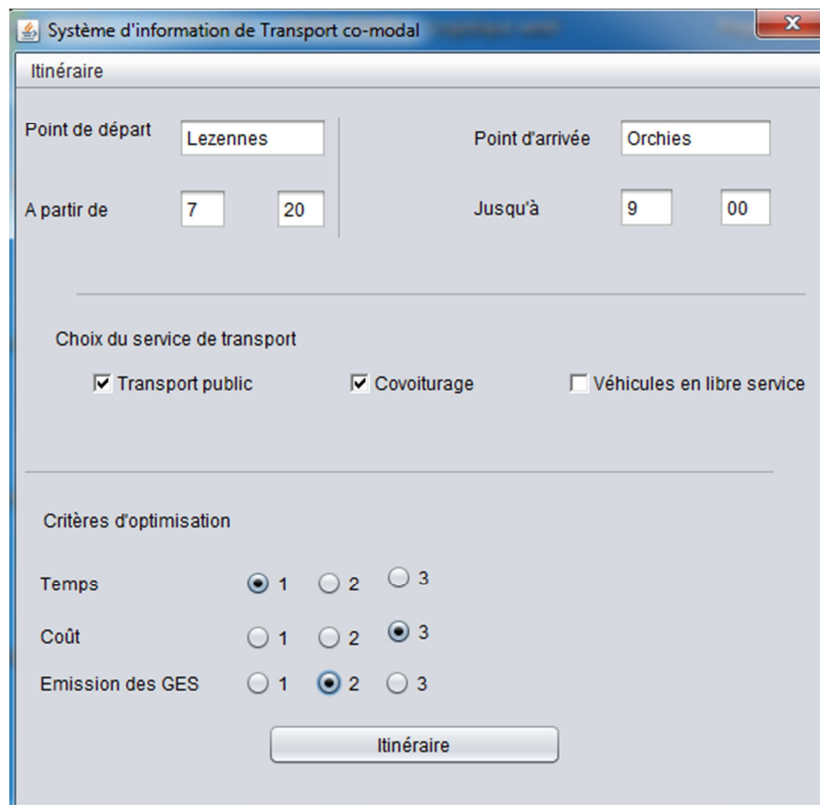


Figure V-2 Interface utilisateur

Dans cette interface, l'utilisateur doit mentionner

- Son point de départ
- Son point d'arrivée
- Sa date de départ au plus tôt souhaitée
- Sa date d'arrivée au plus tard souhaitée

Ensuite, il a le choix de choisir les services de transport qu'il souhaite prendre durant le trajet.

La troisième partie de l'interface permet de sélectionner les critères à optimiser par ordre de préférence.

V.4.2. Google Maps

En plus de l'itinéraire fourni comme solution, il est souhaitable que le système soit capable de dessiner l'itinéraire co-modal sur une cartographie. Pour cela, nous avons eu recours à Google Maps. Pour visualiser les itinéraires sur Google Maps, il faut utiliser l'API de Google Maps. Nous avons créé un JeditorPane qui exécute un code HTML et affichant l'image sur la carte.

Dans la figure IV-2, nous pouvons visualiser un exemple d'itinéraire visualisé sur Google Maps.

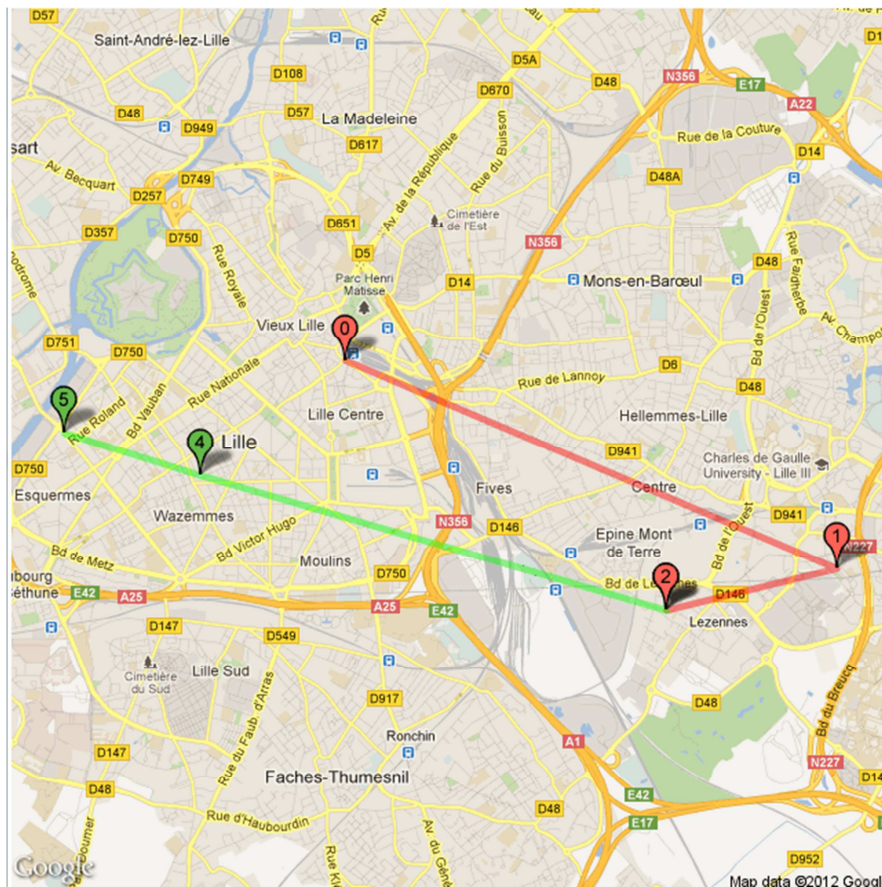


Figure V-3 Exemple d'itinéraire sur Google Maps

L'URL qui doit être envoyé à Google Maps ne doit pas dépasser 2048 caractères ce qui nous permet à peine de dessiner les marqueurs représentant les nœuds qui constituent le chemin et les routes en couleurs différentes. Nous ne pouvons rajouter aucune autre information. En

plus, comme le montre la figure, le chemin est dessiné comme étant un ensemble de lignes droites reliant les différents nœuds sans trop voir le trajet réel de l'utilisateur.

Pour remédier à ce problème, nous avons eu recours à un progiciel appelé Cartocom.

V.4.3. Cartocom

Cartocom est un progiciel développé par la société Bayo pour subvenir spécialement aux besoins de notre équipe en matière d'information géographique.

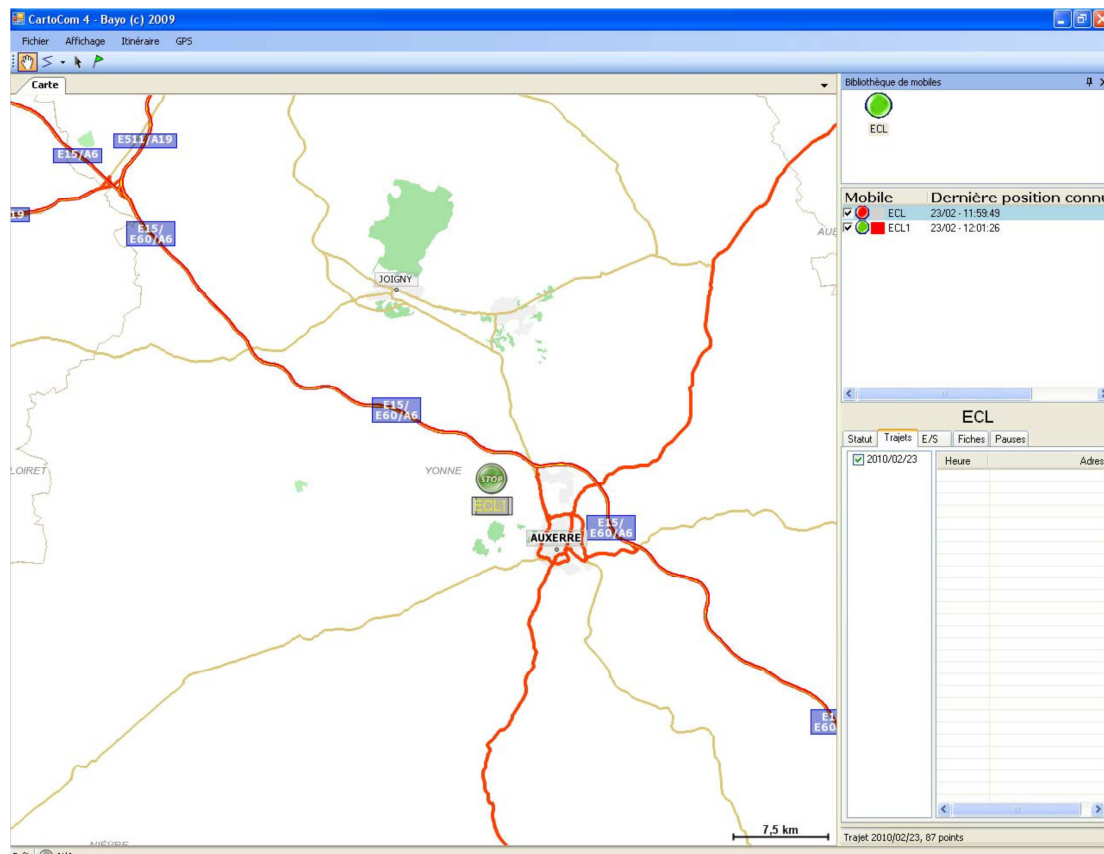


Figure V-4 Interface Cartocom

Il s'agit d'une version communicante d'un logiciel de cartographie. Il reprend toutes les fonctions de traitement de la cartographie, dont voici les principales :

- Outils de création de routes, de points, de tracés personnalisables.
- Transferts de tracés, de routes et de waypoints depuis ou vers des récepteurs GPS de marque Garmin, MLR, Magellan ou SILVA.
- Recherche d'adresse (par numéro, nom de rue et commune) sur fond vectoriel et calcul d'itinéraire.

Il aussi d'autres fonctionnalités spécifiques comme :

- Le référencement géographique de clients,

- La localisation et suivi en temps réel d'un véhicule,
- La récupération et visualisation en temps réel ou en différé des trajets effectués par un véhicule sur les fonds cartographiques. Tous les déplacements sont automatiquement archivés et consultables ultérieurement.

Etant munie d'outils GPS, cette plateforme acquise par l'équipe, facilite en effet l'acquisition des données géographiques et assure la localisation continue des usagers connectés au système.

Cartocom a ainsi été ancré dans la plateforme de covoiturage dynamique de (Sghaier, 2011) comme un module GIS responsable de récupérer les données transit des usagers de leurs origines jusqu'à leurs destinations en offrant en parallèle de multiples services.

Nous allons aussi profiter de la performance et de l'efficacité de ce progiciel et nous allons l'utiliser pour notre simulation et le connecter à notre système SITCoMo pour afficher directement les itinéraires co-modaux calculés.

V.5. Implémentation et simulation des algorithmes du système SITCoMo

Notre première motivation dans ce travail est de montrer la faisabilité d'un calcul d'itinéraire co-modal et distribué pour les services de transport français. Or, il est encore plus difficile de montrer cette faisabilité quand on n'a pas accès aux différents systèmes d'information et d'aide aux déplacements différents associés aux opérateurs des différents services de transport. Ainsi, pour nos travaux de recherche nous avons basé la simulation sur des exemples et des données théoriques, qui restent conforme au contexte réel du transport.

V.5.1. Données utilisées

Pour valider le fonctionnement de notre système et les algorithmes de recherche et d'optimisation d'itinéraires, nous avons utilisé certaines données réelles récupérées à partir de certains opérateurs de transport de différents services et d'autres données pour d'autres services, générées aléatoirement mais qui se rapprochent de la réalité.

Nous considérons que notre système offre aux utilisateurs trois services de transport.

- Service du transport public : nous avons obtenu des données réelles pour un petit réseau de transport qui inclut trois opérateurs : Transpole, SNCF et BCD Ligne.

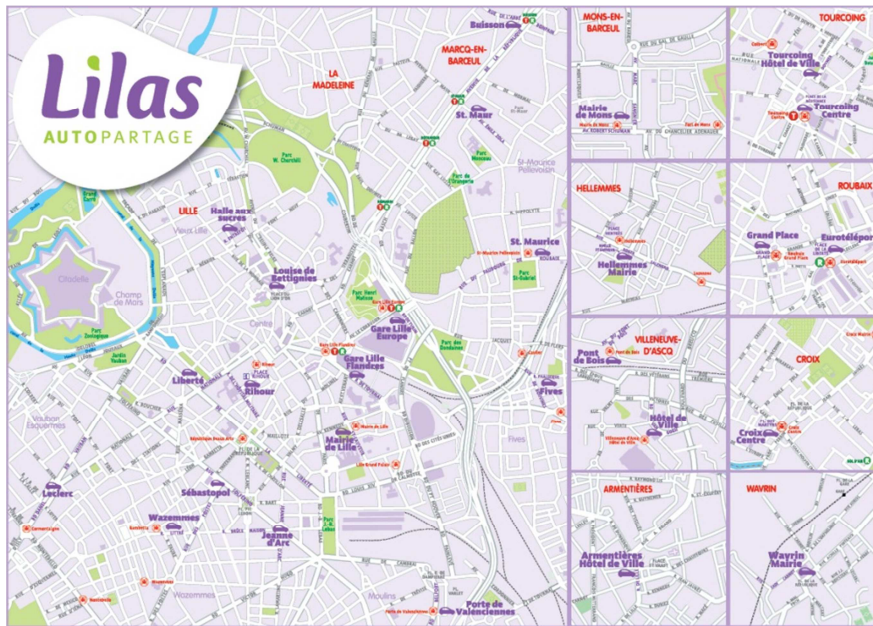


Figure V-6 Stations de voitures d'autopartage

Concernant les vélos en libre service, nous avons repris les données des stations des vélos Vville. Ces stations sont éparpillées par tout et leur utilisation est de plus en plus fréquente. Nous retrouvons les stations de Vville dans la figure V-7.

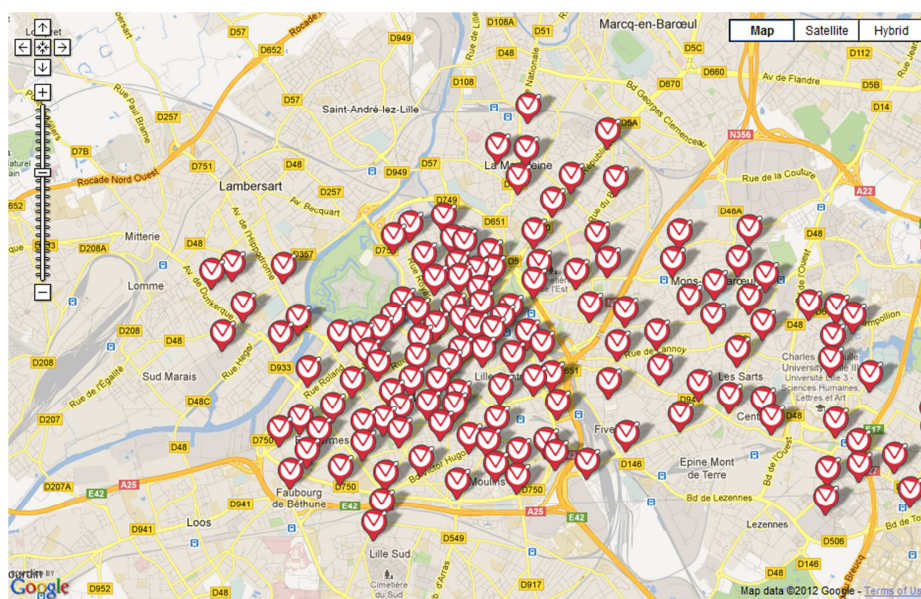


Figure V-7 Stations de Vville

- Service de covoiturage : nous considérons des données aléatoires qui se rapprochent de la réalité. Une liste de voitures qui se déplacent d'un point à l'autre dans des horaires bien précis.

V.5.2. Tests et scénarios

Pour illustrer nos approches adoptées et illustrer notre système, nous proposons différents scénarios pour bien expliquer l'exécution des algorithmes proposés.

Comme décrit dans 5.5.1, les services et les opérateurs de transport sont :

- TSA_1 : Agent pour le service de transport en commun.
 - ✓ $TIA_{1,1}$: Opérateur Transpole
 - ✓ $TIA_{1,2}$: Opérateur SNCF
 - ✓ $TIA_{1,3}$: Opérateur Ligne BCD
- TSA_2 : Agent pour le service de véhicules en libre-service
 - ✓ $TIA_{2,1}$ = Opérateur Lilas
 - ✓ $TIA_{2,2}$ = Opérateur Vville
- TSA_3 : Agent pour le service de covoiturage
 - ✓ $TIA_{3,1}$: Opérateur de covoiturage

Les différents scénarios que nous allons considérer sont les suivants :

- **Scénario1** : Nous considérons une seule requête I_1 à $t= 7h45$ allant de Dunkerque à Lezennes dans la fenêtre de temps $[8, 9 :15]$. L'utilisateur n'a aucune préférence pour les modes de transport et veut optimiser en premier lieu le temps, puis l'émission des GES et enfin le coût du trajet.
- **Scénario2** : Nous considérons 4 requêtes I_1, I_2, I_3 et I_4 exprimées par 4 personnes $P_t = \{P_1, P_2, P_3, P_4\}$:

| Requête I_k | Point de départ d_k | Point d'arrivée a_k | Intervalle de temps W_k | Préférences | Priorités des critères |
|---------------|-----------------------|----------------------------------|---------------------------|-------------------------------------|--|
| I_1 | Dunkerque | Villeneuve d'ascq Hotel de Ville | [7h30, 9h30] | Transport public ; covoiturage | 1) Temps 2) Coût 3) Emission des GES |
| I_2 | Dunkerque | Lezennes | [7h20, 9h15] | Transport public ; covoiturage ; | 1) Temps 2) Emission des GES 3) coût |
| I_3 | Dunkerque | Pont de Bois | [7h30, 9h] | Covoiturage | 1) Emission des |

| | | | | | |
|-------|-----------|--------------------|------------|------------------|--|
| | | | | | GES 2) Temps 3) coût |
| I_4 | Dunkerque | Gare Lille Europe. | [7h30, 9h] | Transport public | 1) Temps 2) Coût 3) Emissions des GES. |

- **Scénario 3** : Nous proposons de traiter à l'instant $t = 7h$; 6 requêtes simultanées $I_t = \{I_1, I_2, I_3, I_4, I_5, I_6\}$ exprimées par 6 personnes différentes $P_t = \{P_1, P_2, P_3, P_4, P_5, P_6\}$. Les requêtes sont décrites dans le tableau suivant :

| Requête I_k | Point de départ d_k | Point d'arrivée a_k | Intervalle de temps W_k | Préférences | Priorités des critères |
|---------------|-----------------------|----------------------------------|---------------------------|---|--|
| I_1 | Dunkerque | Villeneuve d'ascq Hotel de Ville | [7h30, 9h30] | Transport public ; covoiturage | 4) Temps 5) Coût 6) Emission des GES |
| I_2 | C.H.R.B Calmette | Orchies | [7h20, 10h30] | Transport public ; Autopartage ; | 4) Temps 5) Emission des GES 6) coût |
| I_3 | Dunkerque | Ch Dron | [7h30, 9h] | Covoiturage, transport public, auopartage | 4) Emission des GES 5) Temps 6) coût |
| I_4 | Cormontaigne | Ascq_Village | [8h45, 10h] | Transport public, vélo, covoiturage | 4) Temps 5) Coût 6) Emissions des GES. |
| I_5 | Boulogne Ville | Port de douai | [6h, 9h15] | Transport public | 1) Temps 2) Coût |

| | | | | | |
|-------|----------|-------------------|----------|--------------------------------|--|
| | | | | | 3) Emissions des GES. |
| I_6 | Lezennes | CHR Oscar Lambret | [7h, 9h] | Transport public, covoiturage, | 1) Emission des GES 2) Temps 3) coût |

V.5.3. Identification du domaine de recherche : Application du DSSA

Après l'expression des requêtes de la part de tous les utilisateurs, le *SupA* les reçoit et commence à rechercher les routes. Le premier rôle du *SupA* est de déterminer le domaine de recherche en envoyant une requête à chaque TSA_i , $1 \leq i \leq 3$. Après l'application de l'algorithme de DSSA, chaque TSA_i envoie la liste des opérateurs qui vont répondre aux requêtes. Nous observons dans les figures, les résultats de la liste des opérateurs de transport envoyés par chaque TSA_i pour chaque requête concernant les scénarios 1 et 3.

```

MainProject (3) [Java Application] C:\Program Files (x86)\Java\jre7_32\bin\javaw.exe (1:
Requête de domaine de recherche pour la requête: 0
Covoiturage || CovoituragerK || 7:30 || 9:30 || 0
VeloPartage || VLille || 7:30 || 9:30 || 0
Bus_44_1 || Transpole || 7:30 || 9:30 || 0
Metro2_vers_CH_Dron || Transpole || 7:30 || 9:30 || 0
Ter1_Dunkerque_LilleEurope || SNCF || 7:30 || 9:30 || 0
Metro2_vers_St_Philbert || Transpole || 7:30 || 9:30 || 0
Bus_44_2 || Transpole || 7:30 || 9:30 || 0
Bus_43_1 || Transpole || 7:30 || 9:30 || 0
Metro1_vers_4_Cantons || Transpole || 7:30 || 9:30 || 0
Metro1_vers_CHRB_Calmette || Transpole || 7:30 || 9:30 || 0
Ter2_LilleFlandres_Orchies || SNCF || 7:30 || 9:30 || 0
Ter2_Orchies_LilleFlandres || SNCF || 7:30 || 9:30 || 0
Ter3_Valenciennes_LilleFlandres || SNCF || 7:30 || 9:30 || 0
Ter3_LilleFlandres_Valenciennes || SNCF || 7:30 || 9:30 || 0
Bus_43_2 || Transpole || 7:30 || 9:30 || 0
Ter1_LilleEurope_Dunkerque || SNCF || 7:30 || 9:30 || 0
.....

```

Figure V-8 Liste des opérateurs pour chaque requête pour le scénario 1

```

<terminated> MainProject (1) [Java Application] C:\Program Files (x86)\Java\jre7_32\bin\javaw.
Requête de domaine de recherche pour la requête: 0
Covoiturage || CovoiturageFR || 7:30 || 9:30 || 0
Metro2_vers_CH_Dron || Transpole || 7:30 || 9:30 || 0
Ter1_Dunkerque_LilleEurope || SNCF || 7:30 || 9:30 || 0
Metro2_vers_St_Philbert || Transpole || 7:30 || 9:30 || 0
Bus_BoulogneVille_Dunkerque || LigneBCD || 7:30 || 9:30 || 0
Metro1_vers_4_Cantons || Transpole || 7:30 || 9:30 || 0
Metro1_vers_CHRB_Calmette || Transpole || 7:30 || 9:30 || 0
Ter2_Orchies_LilleFlandres || SNCF || 7:30 || 9:30 || 0
Ter3_Valenciennes_LilleFlandres || SNCF || 7:30 || 9:30 || 0
Bus_Dunkerque_BoulogneVille || LigneBCD || 7:30 || 9:30 || 0
Ter2_LilleFlandres_Orchies || SNCF || 7:30 || 9:30 || 0
Ter3_LilleFlandres_Valenciennes || SNCF || 7:30 || 9:30 || 0
Ter1_LilleEurope_Dunkerque || SNCF || 7:30 || 9:30 || 0
*****
Requête de domaine de recherche pour la requête: 1
Ter2_LilleFlandres_Orchies || SNCF || 7:20 || 10:30 || 1
Metro1_vers_4_Cantons || Transpole || 7:20 || 10:30 || 1
Ter2_Orchies_LilleFlandres || SNCF || 7:20 || 10:30 || 1
Metro2_vers_St_Philbert || Transpole || 7:20 || 10:30 || 1
Metro2_vers_CH_Dron || Transpole || 7:20 || 10:30 || 1
Ter3_LilleFlandres_Valenciennes || SNCF || 7:20 || 10:30 || 1
Ter2_Valenciennes_LilleFlandres || SNCF || 7:20 || 10:30 || 1
Metro1_vers_CHRB_Calmette || Transpole || 7:20 || 10:30 || 1
*****
Requête de domaine de recherche pour la requête: 2
Covoiturage || CovoiturageFR || 7:30 || 9:30 || 2
Metro2_vers_CH_Dron || Transpole || 7:30 || 9:30 || 2
Ter1_Dunkerque_LilleEurope || SNCF || 7:30 || 9:30 || 2
Bus_BoulogneVille_Dunkerque || LigneBCD || 7:30 || 9:30 || 2
Metro2_vers_St_Philbert || Transpole || 7:30 || 9:30 || 2
Bus_Dunkerque_BoulogneVille || LigneBCD || 7:30 || 9:30 || 2
Ter1_LilleEurope_Dunkerque || SNCF || 7:30 || 9:30 || 2
*****
Requête de domaine de recherche pour la requête: 3
VeloPartage || VLille || 8:45 || 10:0 || 3
Bus_43_1 || Transpole || 8:45 || 10:0 || 3
Metro2_vers_CH_Dron || Transpole || 8:45 || 10:0 || 3
Bus_43_2 || Transpole || 8:45 || 10:0 || 3
Bus_44_2 || Transpole || 8:45 || 10:0 || 3
Metro1_vers_4_Cantons || Transpole || 8:45 || 10:0 || 3
Metro1_vers_CHRB_Calmette || Transpole || 8:45 || 10:0 || 3
Metro2_vers_St_Philbert || Transpole || 8:45 || 10:0 || 3
*****
Requête de domaine de recherche pour la requête: 4
Metro2_vers_CH_Dron || Transpole || 6:0 || 9:15 || 4
Ter1_Dunkerque_LilleEurope || SNCF || 6:0 || 9:15 || 4
Bus_BoulogneVille_Dunkerque || LigneBCD || 6:0 || 9:15 || 4
Metro2_vers_St_Philbert || Transpole || 6:0 || 9:15 || 4
Bus_Dunkerque_BoulogneVille || LigneBCD || 6:0 || 9:15 || 4
*****
Requête de domaine de recherche pour la requête: 5
Metro1_vers_4_Cantons || Transpole || 7:0 || 9:0 || 5
Metro1_vers_CHRB_Calmette || Transpole || 7:0 || 9:0 || 5
Metro2_vers_CH_Dron || Transpole || 7:0 || 9:0 || 5
Metro2_vers_St_Philbert || Transpole || 7:0 || 9:0 || 5
*****

```

Figure V-9 Liste des opérateurs pour chaque requête pour le scénario 3

Nous pouvons voir par exemple pour le scénario 3 que pour la première requête, le TSA_1 donne Transpole, SNCF et Ligne BCD comme opérateurs pour répondre à la requête alors que pour la deuxième, on ne trouve que Transpole et SNCF.

V.5.3.1. Calcul des premiers plus courts chemins : construction des graphes et application du DSRA.

Une fois, le $SupA$ a reçu la liste des opérateurs de la part de tous les TSA_i , il va envoyer les requêtes à l'ensemble de ces $TIA_{i,j}$, $1 \leq i \leq 3$; $1 \leq j \leq K_i$. Il reçoit alors les réponses pour construire le graphe co-modal constitué de tous les arcs envoyés. Cette communication est traduite par la figure V-10.

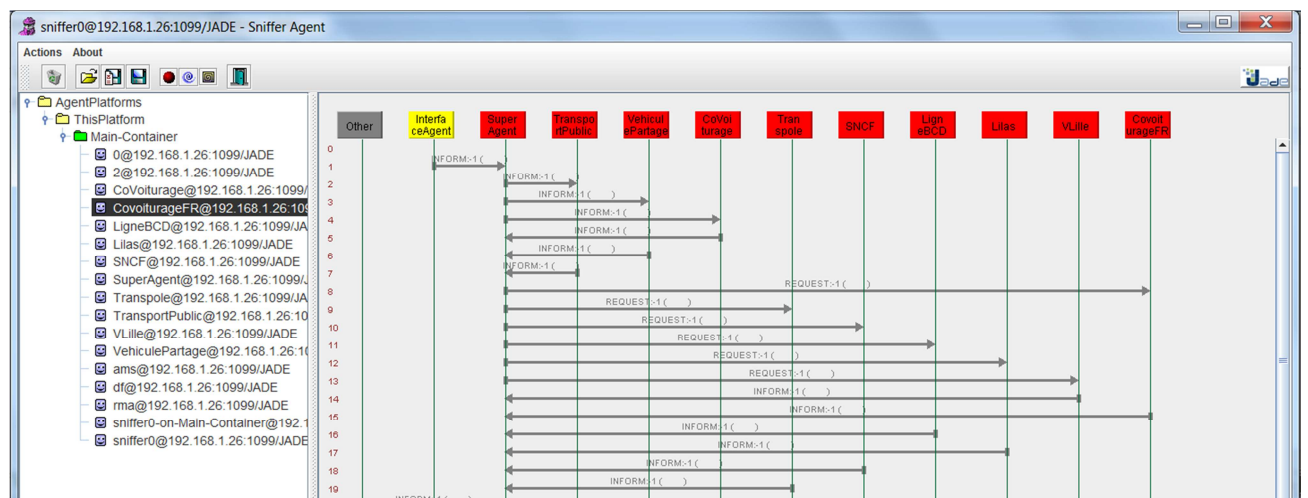


Figure V-10 Communication entre les agents $supA$, TSA_i et $TIA_{i,j}$

Pour mieux illustrer l'exécution de l'algorithme DSRA appliqué sur le graphe de transfert, nous nous concentrons sur le scénario 1. Pour cela nous considérons la partie du réseau suivante :

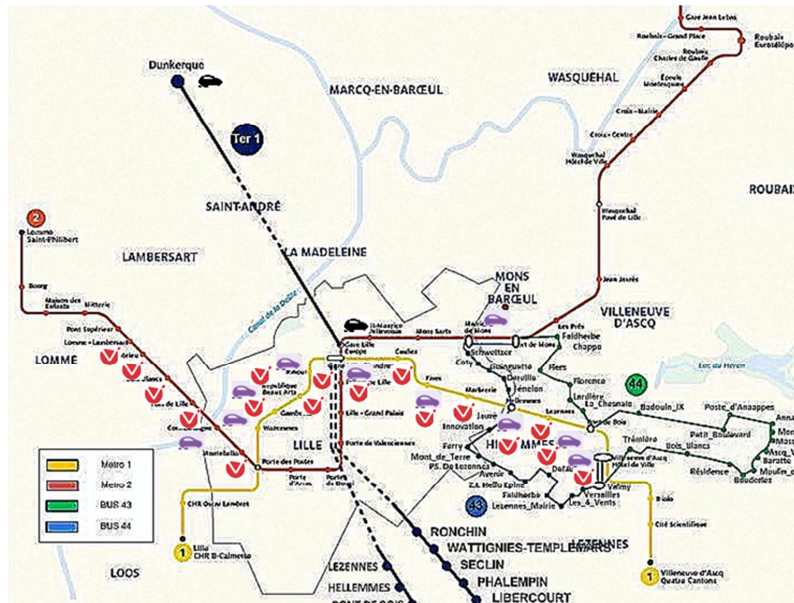


Figure V-11 Réseau de transport pour le scénario 1

Le graphe de transfert construit par le *SupA* à partir du graphe co-modal est représenté par la figure V-12

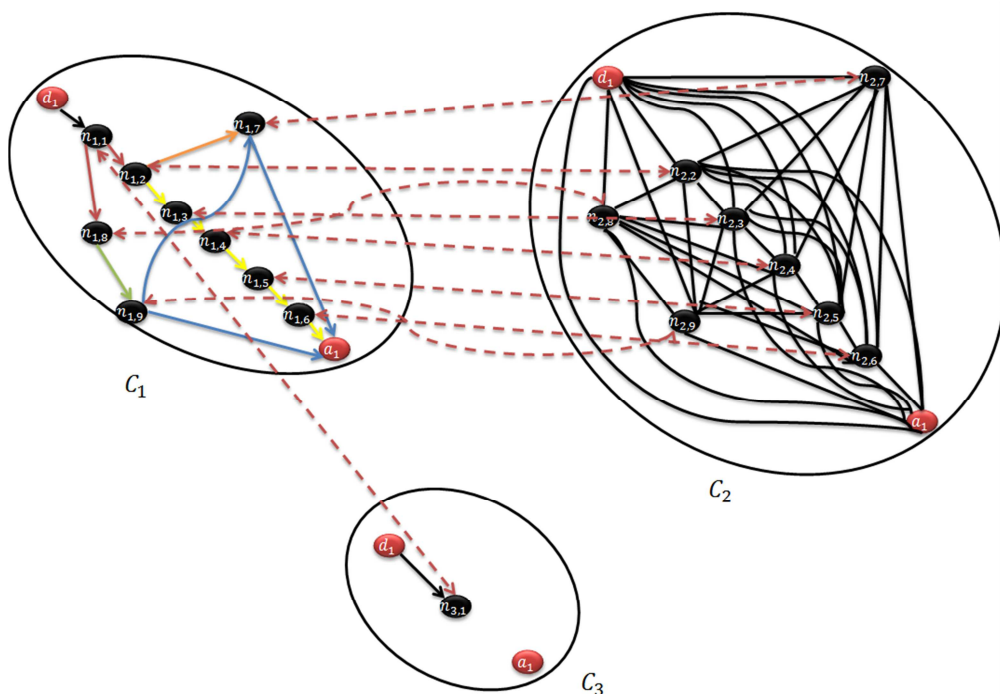


Figure V-12 Graphe de transfert pour le scénario 1

Pour trouver les plus courts chemins entre le point de départ $d_1 = \text{Dunkerque}$ et le point d'arrivée $a_1 = \text{Lezennes}$, nous obtenons trois graphes correspondants aux trois composants C_1 ,

C_2 et C_3 . Nous notons C_1 , C_2 et C_3 les composants relatifs respectivement aux services de transport en commun, véhicules en libre service et covoiturage.

Dans ce graphe de transfert, nous obtenons neuf points de transfert co-modaux (PTC). La liste des PTCs est présentée dans le tableau :

Tableau V-1 Liste des PTC

| Numéro | PTC |
|---------------------|---------------------|
| $n_{1,1} = n_{3,1}$ | Gare Lille Europe |
| $n_{1,2} = n_{2,2}$ | Gare Lille Flandres |
| $n_{1,3} = n_{2,3}$ | Caulier |
| $n_{1,4} = n_{2,4}$ | Fives |
| $n_{1,5} = n_{2,5}$ | Marbrerie |
| $n_{1,6} = n_{2,6}$ | Hellemmes |
| $n_{1,7} = n_{2,7}$ | Mont de Terre |
| $n_{1,8} = n_{2,8}$ | Fort de Mons |
| $n_{1,9} = n_{2,9}$ | Faidherbe |

Nous remarquons que le graphe du composant C_3 est compliqué vu qu'il représente les véhicules en libre service auto et vélo. En effet, les nœuds de ce graphe sont les stations d'autopartage et de vélos en libre service. A partir de chaque nœud, il est possible d'aller vers n'importe quel autre nœud, cela explique tous les arcs inclus dans le graphe. Sur ce graphe, il n'est pas nécessaire d'appliquer DSRA vu sa complexité et en plus nous considérons qu'entre deux stations de véhicules en libre service le plus court chemin est celui qui relie ces deux nœuds directement.

Il est aussi inutile d'appliquer DSRA sur le graphe du covoiturage vu qu'il est composé d'un seul arc.

Nous allons alors voir l'application du DSRA sur le premier composant C_1 relatif au transport public. Pour mieux expliquer l'exécution de l'algorithme, nous considérons que chaque ligne de transport est un opérateur de transport. Le composant C_1 devient alors composé de 6 classes (Figure V-13). Ce graphe contient 6 points d'intersections : $n_{1,1}, n_{1,2}, n_{1,8}, n_{1,9}, n_{1,7}, n_{1,9}$ et a_1 .

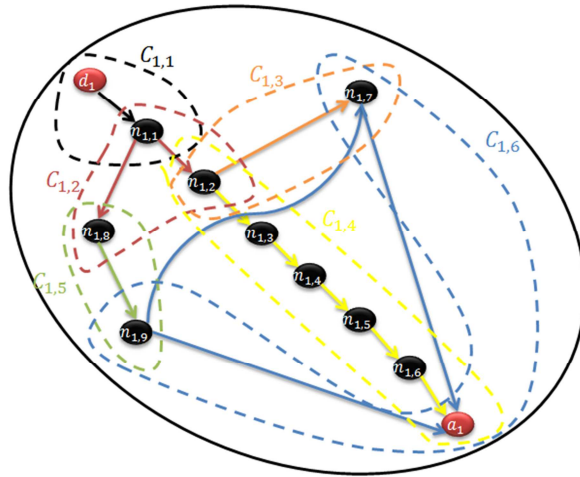


Figure V-13 Classes du composant C_1

A partir de ce graphe, un graphe d'intersection est construit (figure V-14) et les plus courts chemins entre les points de départ et d'arrivée ainsi qu'entre les points de transfert co-modaux vont être déterminés en exécutant l'algorithme DSRA.

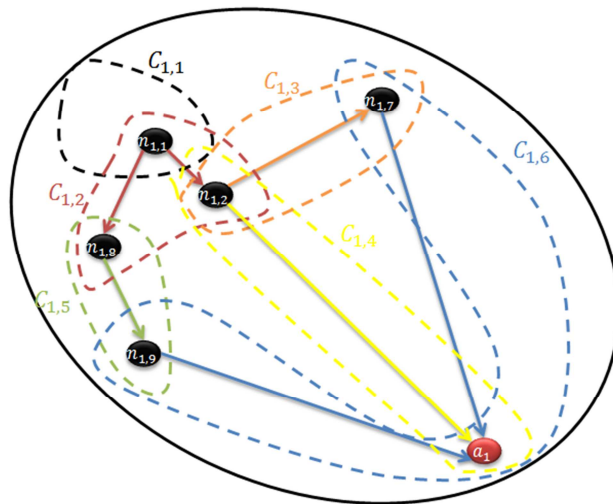


Figure V-14 Graphe d'intersection

Dans cet exemple, le plus court chemin dans le composant C_1 obtenu pour aller de d_1 vers a_1 est : $(d_1 \rightarrow n_{1,1} \rightarrow n_{1,2} \rightarrow a_1)$.

Pour prouver l'apport de notre approche et l'impact de l'algorithme DSRA pour une première optimisation, nous donnons dans ce qui suit les résultats obtenus en simulant le troisième scénario.

En construisant le graphe de transfert, le nombre de nœuds, le nombre de points PTC et le nombre d'arcs dans chaque composant sont représentés dans le tableau V-2.

Tableau V-2 Résultats du graphe de transfert pour le scénario 3

| Composant | C_1 | C_2 | C_3 |
|--|-------|-------|-------|
| Nombre des nœuds | 125 | 60 | 13 |
| Nombre des Points de Transfert Co-modaux (PTC) | 14 | 12 | 3 |
| Nombre d'arcs | 248 | 3540 | 2 |

En cherchant les plus courts chemins dans chaque composant, nous obtenons le graphe de Transfert des Plus Courts chemins (SPTG). Les caractéristiques de ce graphe sont décrites dans le tableau suivant :

Tableau V-3 Tableau 2 Résultats du graphe SPTG pour le scénario 3

| Composant | C_1 | C_2 | C_3 |
|--|-------|-------|-------|
| Nombre des nœuds | 30 | 21 | 13 |
| Nombre des Points de Transfert Co-modaux (PTC) | 14 | 12 | 3 |
| Nombre d'arcs | 88 | 66 | 2 |

D'après ces résultats, nous pouvons faire une comparaison entre le nombre des nœuds, celui des points de transferts co-modaux et le nombre d'arcs de chaque composant dans le graphe de transfert et le graphe SPTG (Figure V-15).

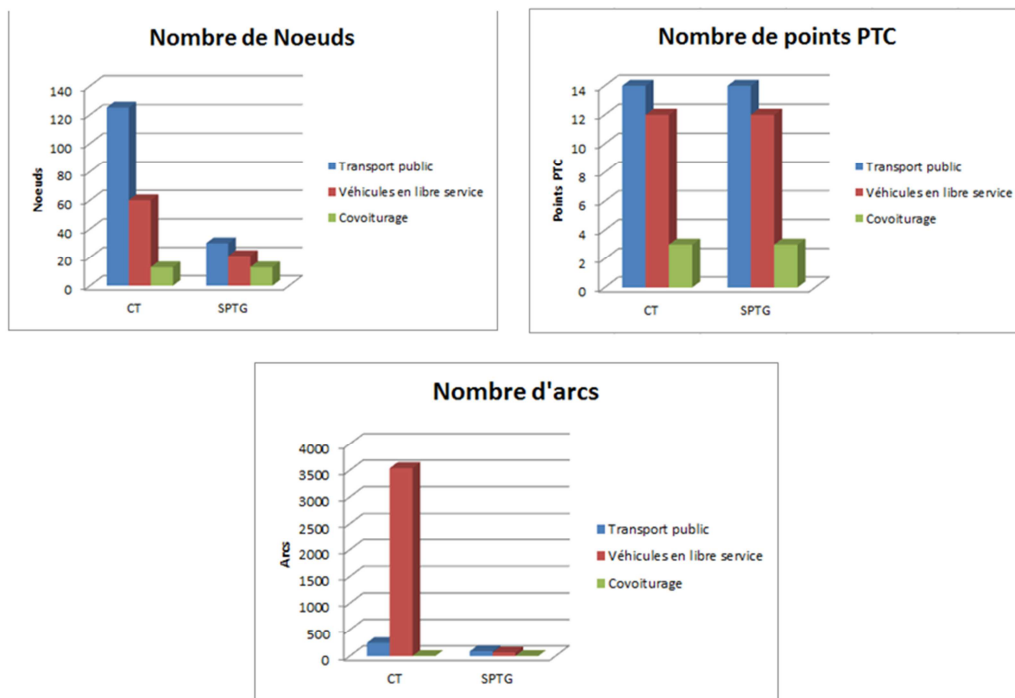


Figure V-15 Comparaison entre le graphe de transfert C_T et SPTG

Nous pouvons remarquer que l’algorithme DSRA a bien permis de réduire le nombre de nœuds et d’arcs de chaque composant et par suite il a réduit le nombre de routes à intégrer dans les solutions finales. Le nombre de points de transferts co-modaux reste le même étant donné que nous cherchons les plus courts chemins entre ces points et les points de départs et d’arrivée.

A partir du graphe SPTG, le *SupA* va pouvoir créer les agents routes (RA) sous forme de chromosomes. Une optimisation multicritère de ces agents et leur coopération sous forme de coalition permet de construire les solutions finales optimisées pour chaque requête.

V.5.3.2. Optimisation, coalition des RAs et construction des solutions finales

Chaque RA crée par le *SupA* cherche à trouver la meilleure route pour chaque requête en respectant ses préférences de modes de transport et ses priorités des critères. En effet, l’utilisateur classe ses critères qu’il veut optimiser par ordre de priorité. Nous considérons alors les pondérations $\alpha_{k,i}$, $1 \leq k \leq N$ et $1 \leq i \leq 3$, de la fonction objectif globale de la route de la manière suivante :

- $\alpha_{k,i} = 0.6$ si la priorité du critère i est 1
- $\alpha_{k,i} = 0.3$ si la priorité du critère i est 2
- $\alpha_{k,i} = 0.1$ si la priorité du critère i est 3

Nous prenons par exemple un chromosome relatif à une route générée après le calcul des plus courts chemins pour le deuxième scénario :

Table V-1 Chromosome R(Dunkerque, Gare Lille Europe, [7h30 ; 8h55])

| R(Dunkerque, Gare Lille Europe, [7h30 ; 8h55]) | $V_1(3,4)$ | TER ₁ (8h21) | $V_2(2,4)$ |
|--|------------|-------------------------|------------|
| I_1 | * | * | 1 |
| I_2 | * | 1 | * |
| I_3 | 1 | x | * |
| I_4 | X | 1 | x |

Dès que chaque RA calcule sa meilleure valeur, il commence à former des coalitions avec les autres RAs afin de former toutes combinaisons de routes possibles pour chaque requête. L’itinéraire co-modal optimisé pour chaque requête est calculé par l’agent évaluateur (EA). La figure suivante montre la communication entre différents RAs pour la formation des coalitions :

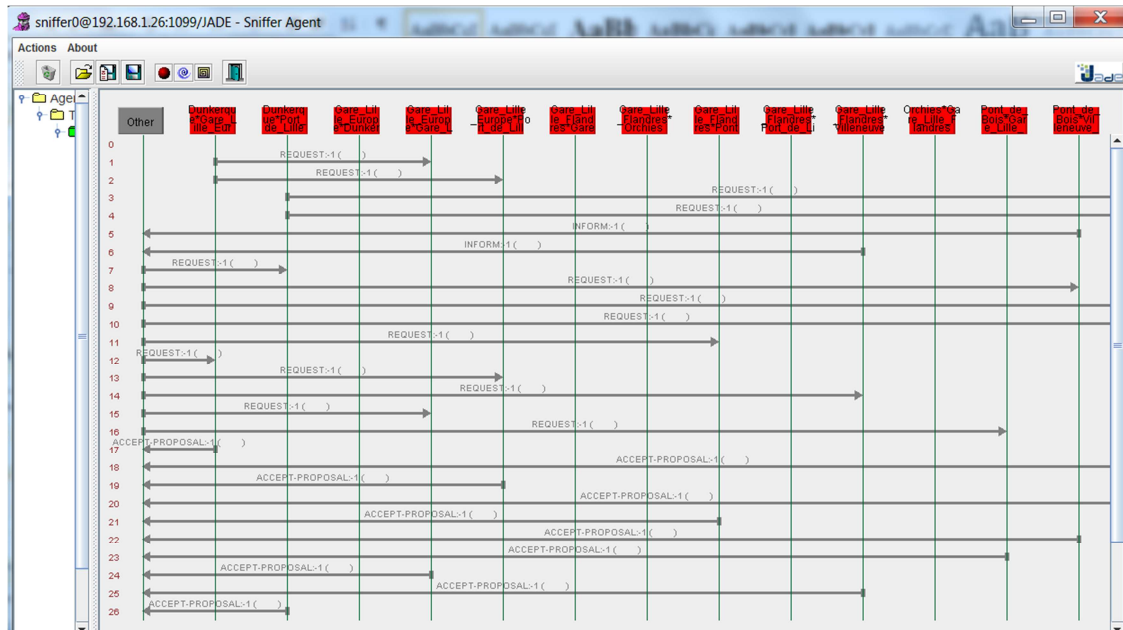


Figure V-16 Communication entre les RA pour former des coalitions

Nous continuons avec le scénario 3 et nous obtenons les solutions suivantes (Figure V-17)

```

Dunkerque est représenté par le point: 0
Port_de_Lille est représenté par le point: 1
Gare_Lille_Flandres est représenté par le point: 2
Villeneuve_d_Ascq_Hotel_de_Ville est représenté par le point: 3
Dunkerque|0-->Port_de_Lille|1 || 50 || Covoiturage || CovoiturageFR || 7:45/8:35
Port_de_Lille|1-->Gare_Lille_Flandres|2 || 12 || TransportPublic || Transpole || Metro2_vers_C_H_Dron || 8:35/8:47
Gare_Lille_Flandres|2-->Villeneuve_d_Ascq_Hotel_de_Ville|3 || 10 || TransportPublic || Transpole || Metro1_vers_4_Cantons || 8:48/8:58
http://maps.googleapis.com/maps/api/staticmap?size=640x640&maptype=roadmap&markers=red%7Clabel:0%7C51.034917,2.374763&
Le moyen: Covoiturage Dunkerque_Port_de_Lille est représenté par la couleur: black
Le moyen: Metro2_vers_St_Philbert est représenté par la couleur: brown
Le moyen: Metro1_vers_4_Cantons est représenté par la couleur: green

C.H.R.B_Calmette est représenté par le point: 0
Gare_Lille_Flandres est représenté par le point: 1
Orchies est représenté par le point: 2
C.H.R.B_Calmette|0-->Gare_Lille_Flandres|1 || 8 || TransportPublic || Transpole || Metro1_vers_4_Cantons || 7:22/7:30
Gare_Lille_Flandres|1-->Orchies|2 || 22 || TransportPublic || SNCF || Ter3_LilleFlandres_Valenciennes || 7:37/7:59
http://maps.googleapis.com/maps/api/staticmap?size=640x640&maptype=roadmap&markers=red%7Clabel:0%7C50.608027,3.039007&markers=red
Le moyen: Metro1_vers_4_Cantons est représenté par la couleur: black
Le moyen: Ter3_LilleFlandres_Valenciennes est représenté par la couleur: brown

Dunkerque est représenté par le point: 0
Port_de_Lille est représenté par le point: 1
C.H_Dron est représenté par le point: 2
Dunkerque|0-->Port_de_Lille|1 || 50 || Covoiturage || CovoiturageFR || 7:45/8:35
Port_de_Lille|1-->C.H_Dron|2 || 35 || TransportPublic || Transpole || Metro2_vers_CH_Dron || 8:36/9:11
http://maps.googleapis.com/maps/api/staticmap?size=640x640&maptype=roadmap&markers=red%7Clabel:0%7C51.034917,2.374763&markers=red
Le moyen: Covoiturage Dunkerque_Port_de_Lille est représenté par la couleur: black
Le moyen: Metro2_vers_CH_Dron est représenté par la couleur: brown

Cormontaigne est représenté par le point: 0
Massena est représenté par le point: 1
Ascq_Village est représenté par le point: 2
Cormontaigne|0-->Massena|1 || 7 || Velopartage || VehiculePartage || 8:45/8:52
Massena|1-->Ascq_Village|2 || 2 || TransportPublic || Transpole || Bus_43_2 || 8:57/8:59
http://maps.googleapis.com/maps/api/staticmap?size=640x640&maptype=roadmap&markers=red%7Clabel:0%7C50.625795,3.03993&markers=red
Le moyen: Velopartage est représenté par la couleur: black
Le moyen: Bus_43_2 est représenté par la couleur: brown

Boulogne_Ville est représenté par le point: 0
Dunkerque est représenté par le point: 1
Gare_Lille_Europe est représenté par le point: 2
Port_de_Douai est représenté par le point: 3
Boulogne_Ville|0-->Dunkerque|1 || 80 || TransportPublic || LigneBCD || Bus_BoulogneVille_Dunkerque || 6:15/7:35
Dunkerque|1-->Gare_Lille_Europe|2 || 34 || TransportPublic || SNCF || Ter1_Dunkerque_LilleEurope || 8:21/8:55
Gare_Lille_Europe|2-->Port_de_Douai|3 || 5 || TransportPublic || Transpole || Metro2_vers_St_Philbert || 8:55/9:0
http://maps.googleapis.com/maps/api/staticmap?size=640x640&maptype=roadmap&markers=red%7Clabel:0%7C50.727057,1.009008&markers=red
Le moyen: Bus_BoulogneVille_Dunkerque est représenté par la couleur: black
Le moyen: Ter1_Dunkerque_LilleEurope est représenté par la couleur: brown
Le moyen: Metro2_vers_St_Philbert est représenté par la couleur: green

Lezennes est représenté par le point: 0
C.H.R_Oscar_Lambret est représenté par le point: 1
Lezennes|0-->C.H.R_Oscar_Lambret|1 || 14 || TransportPublic || Transpole || Metro1_vers_CHRB_Calmette || 7:7/7:21
http://maps.googleapis.com/maps/api/staticmap?size=640x640&maptype=roadmap&markers=red%7Clabel:0%7C50.625375,3.115047&markers=red
Le moyen: Metro1_vers_CHRB_Calmette est représenté par la couleur: black

```

Figure V-17 Solutions obtenues pour l'exemple

Pour les six requêtes $I_1, I_2, I_3, I_4, I_5, I_6$, nous avons obtenu les combinaisons de routes suivantes :

- Pour I_1 : l'itinéraire est co-modal avec comme service le covoiturage pour la première route (Dunkerque, Port de Lille, [7 :45, 8 :35]), la ligne 2 du métro pour la deuxième route (Port de Lille, Gare Lille Flandres, [8 :35, 8 :47]) et la ligne 1 du métro pour la dernière route (Gare Lille Flandres, Villeneuve d'Ascq Hôtel de Ville, [8 :48, 8 :58]).
- Pour I_2 : l'itinéraire est multimodal avec un seul service de transport : le transport public. La première route de l'itinéraire (CHRB Calmette, Gare Lille Flandres, [7 :22, 7 :30]) est

assurée par la ligne 1 du métro et la deuxième route (Gare Lille Flandres, Orchies, [7 :37, 7 :59]) est desservie par le TER3 de l'opérateur de transport SNCF.

- Pour I_3 l'itinéraire est aussi co-modal avec covoiturage et transport public. Le trajet se décompose en deux routes. Une première (Dunkerque, Port de Lille, [7 :45, 8 :35]) est assurée par une voiture de covoiturage et une deuxième (Port de Lille, CH Dron, [8 :36, 9 :11]) assurée par le métro 2.
- Pour I_4 : l'itinéraire est co-modal avec deux services de transport : le vélo et le transport public. (Cormontaigne, Massena, [8 :45, 8 :52]) est desservie par un vélo en libre service et le chemin continue avec le bus 43 pour la route (Massena, Ascq_Village, [8 :57, 8 :59]).
- Pour I_5 , nous obtenons un itinéraire mono-service (transport public) mais multimodal avec trois opérateurs différents. Le premier opérateur est Ligne BCD pour parcourir (Boulogne Ville, Dunkerque, [6 :15, 7 :35]), vient ensuite le deuxième opérateur SNCF avec le TER1 pour la route (Dunkerque, Gare Lille Europe, [8 :21, 8 :55]). Le troisième opérateur à intervenir est Transpole pour le dernier tronçon de l'itinéraire (Gare Lille Europe, Port de douai, [8 :55, 9 :00]).
- Pour I_6 , l'itinéraire est monomodal avec une seule route desservie par la ligne 1 du métro (Lezennes, CHR Oscar Lambret, [7 :07, 7 :21])

Le système schématise ces solutions finales sur une carte de Google Maps. Les itinéraires obtenus dans cet exemple sont décrits sur la figure V-18.



Figure V-18 Solutions obtenues sur Google Maps

Comme décrit dans la section V.4.3, nous avons choisi de fournir les solutions sur le logiciel Cartocom pour avoir une meilleure visibilité des itinéraires (Figure V-19, V-20).

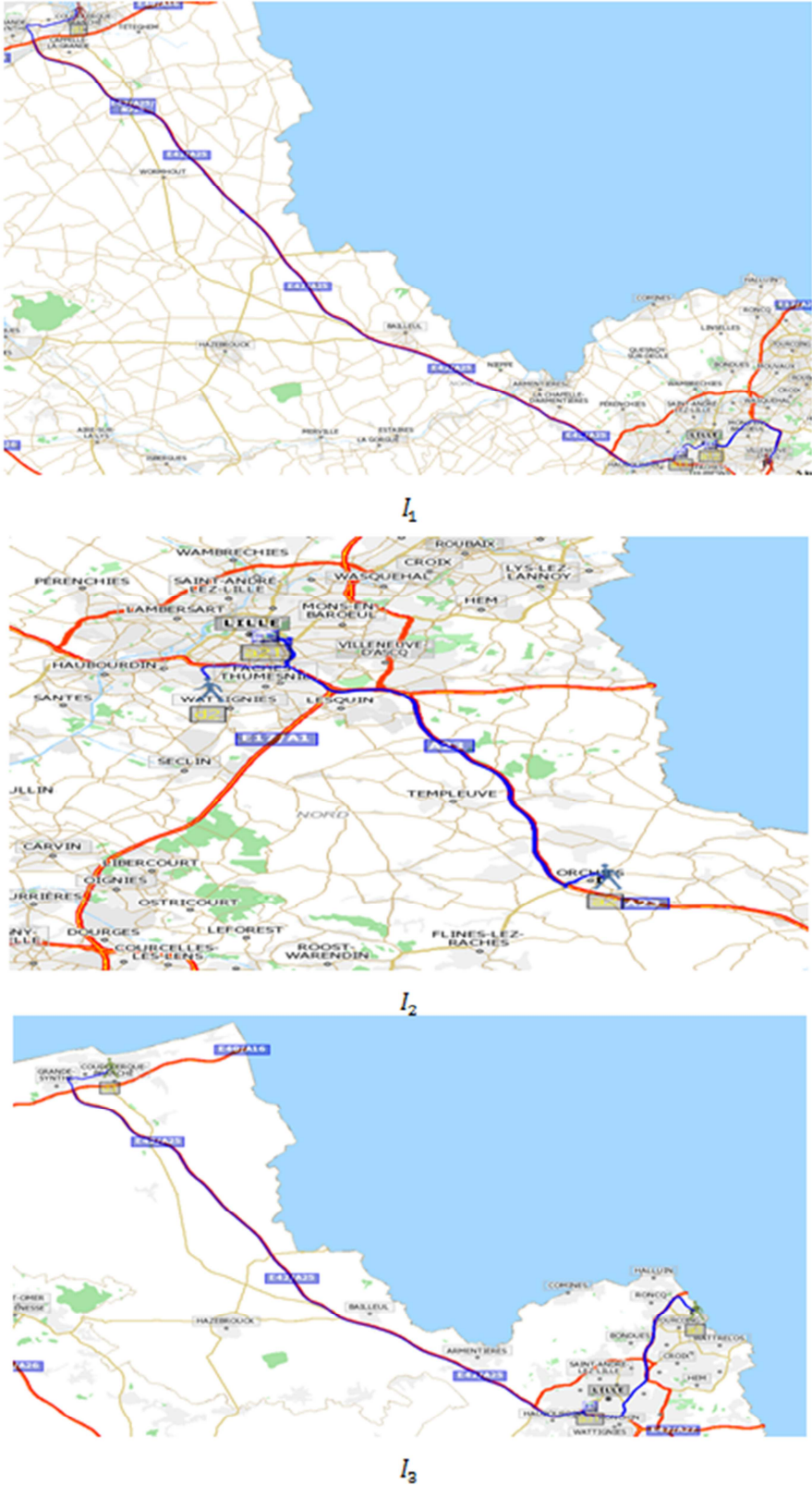


Figure V-19 Résultats obtenus avec CARTOCOM pour I_1 , I_2 et I_3



I_4



I_5



I_6

I_5

Figure V-20 Résultats obtenus avec CARTOCOM pour I_4 , I_5 et I_6

Nous remarquons dans cet exemple que les utilisateurs P_1 et P_3 partagent la même route, c'est-à-dire un même tronçon. En effet, partant de Dunkerque tous les deux, ils prennent la même voiture pour aller jusqu'à Port de Lille . A partir de ce dernier point, chacun emprunte sa propre route pour aller vers sa destination.

Si nous effectuons quelques changements au niveau des données, nous pouvons constater que le système offre d'autres itinéraires selon les préférences des utilisateurs.

Par exemple, nous changeons les priorités des critères de I_1 pour avoir l'ordre suivant :

- 1) Emission des GES
- 2) Temps
- 3) Coût

Pour ce changement, nous obtenons un autre itinéraire multimodal pour la requête I_1 .

```
Dunkerque est représenté par le point: 0
Gare_Lille_Europe est représenté par le point: 1
Gare_Lille_Flandres est représenté par le point: 2
Villeneuve_d_Ascq_Hotel_de_Ville est représenté par le point: 3
Dunkerque|0->Gare_Lille_Europe|1 || 32 || TransportPublic || SNCF || Ter1_Dunkerque_LilleEurope || 8:21/8:53
Gare_Lille_Europe|1->Gare_Lille_Flandres|2 || 1 || TransportPublic || Transpole || Metro2_vers_St_Philbert || 8:53/8:54
Gare_Lille_Flandres|2->Villeneuve_d_Ascq_Hotel_de_Ville|3 || 10 || TransportPublic || Transpole || Metro1_vers_4_Cantons || 8:54/9:4
http://maps.googleapis.com/maps/api/staticmap?size=640x640&maptype=roadmap&markers=red%7Clabel:0%7C51.034917,2.374763&markers=col:rec
Le moyen: Ter1_Dunkerque_LilleEurope est représenté par la couleur: black
Le moyen: Metro2_vers_St_Philbert est représenté par la couleur: brown
Le moyen: Metro1_vers_4_Cantons est représenté par la couleur: green
```

Figure V-21 Nouvel itinéraire pour I_1

La solution est composée de trois routes : la première assurée par un train TER1, la deuxième et la troisième par deux différents métros.

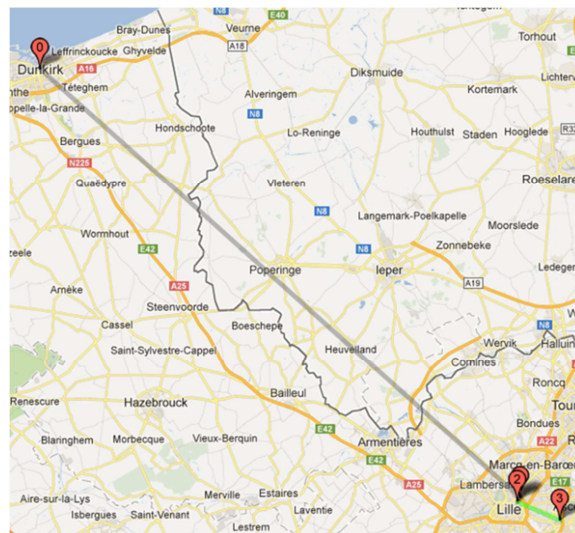


Figure V-22 Nouvel itinéraire pour I_1 sur Google Maps

V.6. Impact de la variabilité des services et des requêtes

Selon les différents scénarios que nous avons appliqués, nous remarquons que les solutions et la complexité du problème dépendent du nombre de requêtes simultanées ainsi que le nombre de services à inclure.

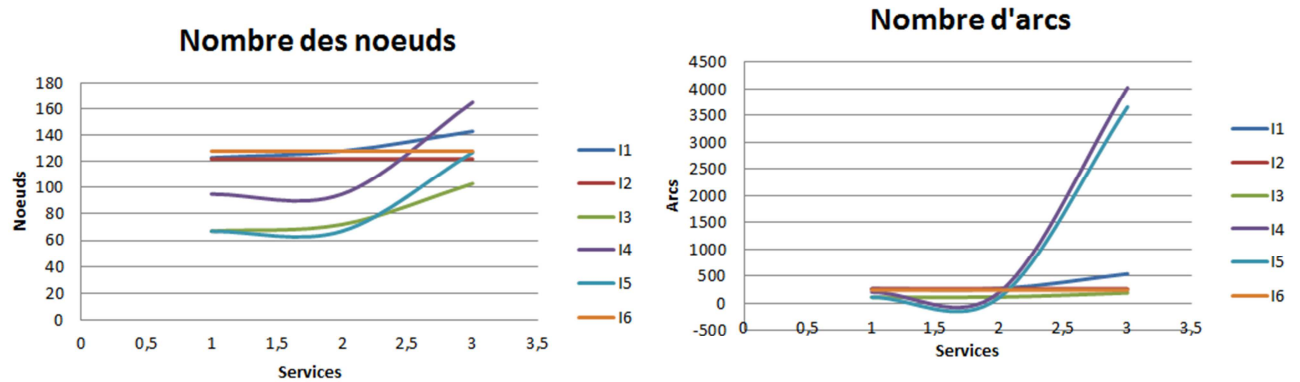


Figure V-23 Evolution du nombre des nœuds et du nombre d'arcs pour chaque requête en fonction du nombre des services

Dans la figure V-23, nous constatons qu'en augmentant le nombre des services à utiliser pour trouver les solutions pour chaque requête, le nombre de nœuds ainsi que le nombre d'arcs n'arrêtent pas d'augmenter. L'évolution du nombre d'arcs est plutôt stable pour les requêtes I_1 , I_3 et I_6 car pour cet exemple le système ne trouve pas d'autres routes même en ajoutant d'autres services.

Nous continuons à tester l'impact des services et des requêtes sur le système en ajoutant à chaque fois les requêtes et en variant aussi les services (Figure V-24). Il est normal que le nombre des nœuds ou le nombre d'arcs augmente en ajoutant les requêtes si celles-ci sont différentes et présentent des points variés et distants géographiquement. Cette évolution peut être progressive si les points de départs et d'arrivée sont assez proches.

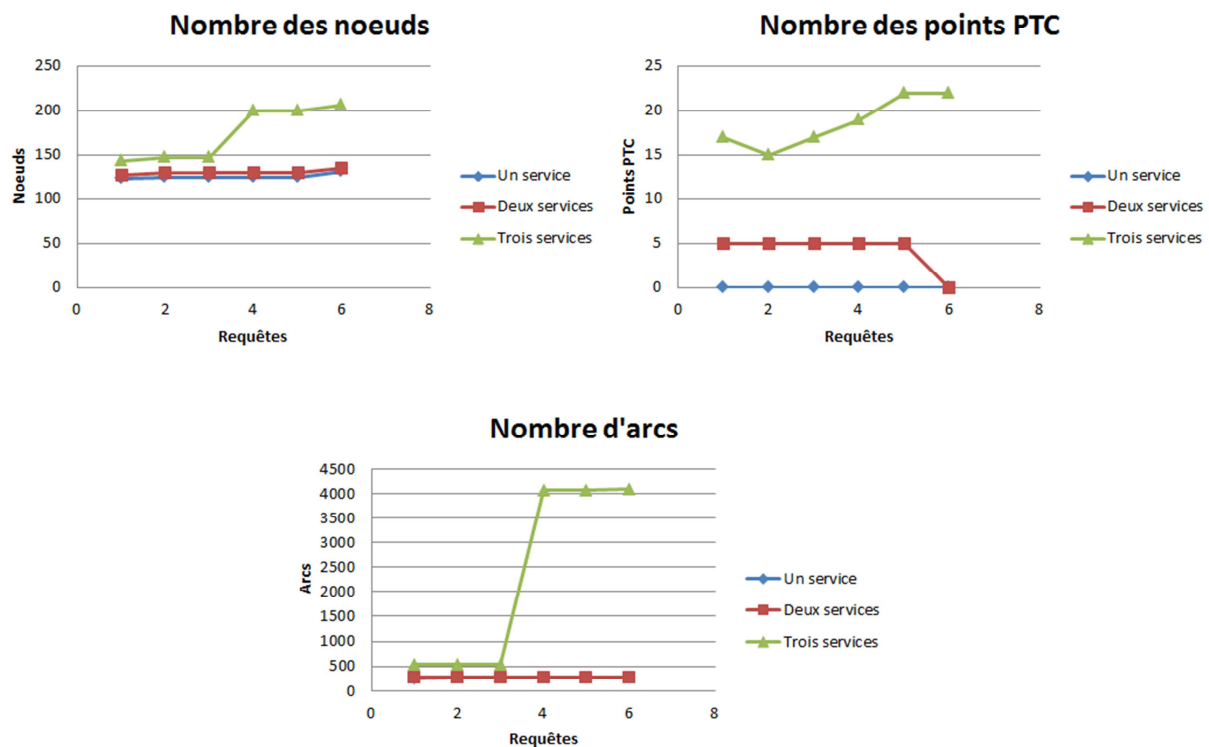


Figure V-24 Evolution du nombre des nœuds, nombre des points PTC et nombre des arcs en fonction des requêtes.

Concernant le nombre des points PTC, lorsqu'un seul service est inclus dans le système, ce paramètre doit être nul étant donné qu'aucun transfert co-modal n'est effectué. Nous trouvons seulement les points de transfert multimodaux entre les différents opérateurs du même service de transport.

L'évolution brusque du nombre d'arcs constatée sur la courbe est due à la participation du service des véhicules en libre service pour la recherche des solutions. En effet, comme nous avons vu dans V.5.2.2, ce service fait appel à un grand nombre de stations qui doivent toutes être connectées et ainsi nous obtenons un nombre d'arcs énorme qui peut atteindre dans l'exemple 4093 arcs. En effet, les stations des vélos spécialement sont un peu partout et aujourd'hui toutes les politiques vont dans le sens d'inclure le vélo comme mode journalier des voyageurs. Cependant, un si grand nombre de nœuds et d'arcs peut paralléliser le système et peut avoir de mauvaises conséquences sur les solutions fournies. Nous préférons alors laisser le service de vélos comme un service optionnel au cas où le système ne trouve pas de solutions.

V.7. Conclusion

Dans ce chapitre, nous avons montré le fonctionnement des approches d'optimisation adoptées dans le SITCoMo. Pour tester le fonctionnement global du système, nous avons implémenté les différents agents sur la plateforme JADE.

Les résultats et scénarios de simulation montrent la robustesse du système face à un flux de requêtes et un ensemble de services de transport. Nous avons donc démontré l'intérêt d'optimiser le nombre des opérateurs de transport à inclure dans les solutions ainsi que le premier niveau d'optimisation grâce à l'approche co-modale et l'exécution de l'algorithme. En effet, nous avons pu réduire le nombre énorme de nœuds à traiter pour pouvoir construire les itinéraires. L'application de l'approche évolutionnaire est très pratique et efficace lorsque le système reçoit un nombre de requêtes élevé. Ensuite, nous avons pu constater l'intérêt de la coalition des agents optimisés pour la formation des combinaisons de routes co-modales selon les préférences de chaque utilisateur.

Conclusion générale

Dans ce travail de recherche, nous avons proposé un système d'information de transport co-modal (SITCoMo) qui se base sur une approche multi-agent en intégrant des algorithmes d'optimisation. L'architecture du SITCoMo est ouverte, dynamique et distribuée, composée de sociétés d'agents dont les cycles de vies dépendent de l'utilisation du système, c'est-à-dire que la raison d'être même de ces agents au sein du système dépend catégoriquement de son utilisation, ce qui réduit considérablement la consommation des ressources.

Le système SITCoMo permet d'offrir à ses utilisateurs des réponses à leurs requêtes sous forme d'itinéraires co-modaux incluant leurs services ou modes de transport préférés et optimisés selon leurs critères. Les critères d'optimisation intégrés dans le système sont le temps total du trajet, le coût et les Emissions des GES. Ceci rend le problème abordé, un problème d'optimisation multicritère.

La solution proposée se déroule en trois étapes essentielles d'optimisation :

- La première étape concerne l'identification du domaine de recherche. Elle permet de limiter le nombre d'opérateurs de transport, quel que soit leur service, indispensables pour fournir toutes les données et toutes les routes possibles qui peuvent construire les solutions globales.
- La deuxième étape consiste à rechercher les différentes *routes* qui vont constituer les solutions finales. Pour la composition des itinéraires optimisés, nous avons commencé par une optimisation monocritère en termes de temps en exécutant un algorithme Dijkstra distribué appliqué sur un graphe de transfert. Le graphe de transfert est un graphe spécifique qui est composé d'un ensemble de composants et un ensemble d'arcs virtuels qui les relient. Ainsi, nous obtenons des plus courts chemins dans chaque composant, ce qui nous permet de générer un ensemble d'agents Route (RA) pour la troisième étape d'optimisation.
- La troisième consiste à appliquer une approche évolutionnaire sur chaque RA pour déterminer les meilleures routes pour chaque requête en termes des trois critères. Suite à cette approche, une communication entre les RAs à travers la coalition permet de

composer les itinéraires co-modaux optimisés. L'agent EA calcule en dernier lieu la meilleure combinaison de routes pour chaque requête et la transmet à l'utilisateur.

Les différentes approches d'optimisation que nous avons utilisées alliées à la technologie des systèmes multi-agents nous a permis d'exploiter les profits de l'un et l'autre dans une combinaison parfaite et harmonieuse faisant régner un contexte d'intelligence artificielle distribuée.

Dans nos travaux futurs, il est toujours possible d'étendre nos recherches et de pratiquer de nouvelles pratiques et techniques en ajoutant d'autres fonctionnalités supplémentaires envisageables pour étendre encore plus le champ d'application de notre système.

En effet, la flexibilité de notre système nous permet d'ajouter d'autres modules comme le module de covoiturage dynamique. Il est possible aussi de rajouter d'autres services pour avoir une plateforme complète. L'intégration de différentes applications touristiques tel que réservations d'hôtels, taxis à la demande pour les personnes à mobilité réduite...

La recherche peut être étendue en ajoutant un module d'anticipation et de régulation de perturbations. En effet, l'intégration des différents services de transports dans notre système reste toujours une solution aux problèmes de perturbations.

Il est possible également d'avoir une vision probabiliste (graphe stochastique) sur les itinéraires, surtout en cas de perturbation. Une extension de ce travail pourrait donc être constituée par la recherche des chemins les plus rapides dans un environnement distribué dynamique stochastique.

Il est intéressant aussi d'étudier la mise en place du système multi-agent proposé dans un environnement formé uniquement des nouveaux appareils émergents type Webphone (capable d'accueillir un ou plusieurs agents vu la puissance de leurs processeurs) et communiquant uniquement via le réseau GPRS.

Parmi les perspectives, le développement d'une interface Homme-machine (IHM) plastique est envisageable. La plasticité est définie comme étant la capacité d'adaptation d'une IHM à son contexte d'usage dans le respect de son utilisabilité. Il s'agit d'une propriété des systèmes interactifs qui fut introduite en réponse à la diversité des plates-formes. Comme à l'évidence l'IHM ne peut être la même sur grand et petit écran, l'idée était de régler, par l'adaptation, les

coûts de développement et de maintenance ainsi que les incohérences ergonomiques résultant de développements cloisonnés entre les versions petit et grand écran. Très vite, l'environnement est considéré, l'utilisateur ensuite, pour enfin revenir à la plate-forme comprenant que les IHM n'étaient plus seulement centralisées et sédentaires mais pouvaient se redistribuer au gré du contexte d'usage en termes d'utilisateur, de plate-forme et d'environnement.

Bibliographie

Abbaspour R.A. et Samadzadegan F. A solution for Time Dependent Multimodal Shortest Path Problem // Journal of Applied Sciences. - 2009. - Vol. 9. - pp. 3804-3812..

Adam Emmanuel Modèle d'organisation multi-agent pour l'aide au travail coopératif dans les processus d'entreprise : application aux systèmes administratifs complexes : Thèse de doctorat de L'université de Valenciennes et du Hainaut-Cambrésis. - 2000.

ADEME Calcul des facteurs d'émissions et sources bibliographiques utilisées : Bilan Carbone 2010, Guide des facteurs d'Emissions Version 6.1. - 2010.

Aloulou Mohamed Ali Structure flexible d'ordonnancements à performances contrôlées pour le pilotage d'atelier en présence de perturbations : Thèse de doctorat de l'Institut National Polytechnique de Lorraine, Nancy. - 2002.

Anli Abdouroihmane et Abed Mourad PerSyst : Un Système de Personnalisation de l'information transport multimodale // E-STA (Sciences et Technologies pour l'Automatique). - 2006. - Vol. 3. - pp. 1-6.

ARENE Intermodalité, transport collectif et vélo: exemple de deux pôles vélo [Rapport]. : Fiche d'opération, 2002.

Ayed H. [et al.] Solving time-dependent multimodal transport problems using a transfer graph model // Computers & Industrial Engineering. - Septembre 2010. - 2 : Vol. 61. - pp. 391-401.

Bai Rendong et Singhal Makesh Carpooling in Mobile Ad Hoc Networks: the Case of Multiple-Target Route Discovery // 5th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks and Workshops. - 16-20 Avril 2007. - pp. 1-9.

Ballet Jean Christophe et Clavel Robert Le covoiturage en France et en Europe: Etats des lieux et perspectives : Rapport d'Etude du CERTU, 2007.

application aux problèmes d'ordonnement de type flow-shop : Thèse de doctorat de l'Université Lille1 - Sciences et Technologies . - 2005.

Belgueliel Youcef et Maamri Ramdane Système d'information voyageur à base d'agents pour la recherche d'itinéraires multimodaux // eTI, Revue électronique en Technologies de l'Information. - 2012. - 6.

Bellemans Tom [et al.] An Agent-Based Model to Evaluate Carpooling at Large Manufacturing Plants // Procedia Computer Science. - 2012. - Vol. 10. - pp. 1221–1227.

Bellman Richard On a routing problem // Quarterly Applied Mathematics. - 1958. - Vol. 16. - pp. 87-90.

Blum Christian et Roli Andrea Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison // ACM Computing Surveys (CSUR). - 2003. - 3 : Vol. 35. - pp. 268 - 308 .

Booch G. Conception orientée objets et applications: Addison-Wesley, 1992.

Bounsaythip Catherine Algorithmes Heuristiques et Evolutionnistes : Application à la Résolution du Problème de Placement de Formes Irrégulière : Thèse de doctorat de l'Université de Lille 1. - 1998.

Bousedjra M. Contribution à la résolution du problème de plus court chemin multiobjectif par algorithmes évolutionnistes : application aux systèmes de transport intermodal : Thèse de doctorat de l'Université de Franche Comté et de l'Université de Technologie de Belfort-Montbéliard. - 2005.

Burmeister Birgit, Haddadi Afsaneh et Matylis Guido Applications of multi agent systems in traffic and transportation // IEE Proceeding on software Engineering. - 1997. - 1 : Vol. 144. - pp. 51 - 60 .

Calvo Roberto Wolfler [et al.] A distributed geographic information system for the daily carpooling problem // Computers and Operations Research. - Novembre 2004. - Vol. 31. - pp. 2263-2278.

Carabelea C. et Berger M. Agent negotiation in ad-hoc networks // The Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, 2005. MobiQuitous 2005. . - 2005. - pp. 482 - 484 .

Chabini Ismail [et al.] Reoptimization Algorithms for Minimum-Time Path Problems in Dynamic Networks - 1997.

Chabini Ismail Discrete Dynamic Shortest Path problems in transportation Applications: complexity and Algorithms with optimal run time // Transportation Research Records. - 1997.

Chaib-draa B. Industrial Applications of Distributed AI // Communications of the ACM. - 1995. - 11 : Vol. 38. - pp. 49-53.

Chaib-draa B. Interaction between Agents in routine, familiar and unfamiliar situations // International Journal of Intelligent and Cooperative Information Systems. - 1996. - 1 : Vol. 5. - pp. 1-25.

Chardigny Sylvain [et al.] Extraction d'Architecture à Base de Composants d'un Système Orienté Objet // INFORSID. - Mai 2007. - pp. 487-502.

Clavel Robert, Mariotto Muriel et Arzac Benjamin L'autopartage en France et en Europe en 2008: Etats des lieux et perspectives : Rapport d'étude du Certu. - 2008.

Clivillé Vincent, Berrah Lamia et Mauris Gilles Quantitative expression and aggregation of performance measurements based on the MACBETH multi-criteria method // International Journal of Production Economics. - 2007. - pp. 171–189.

Coelho Rajan Filomeno et Breikopf Piotr Optimisation multidisciplinaire en mécanique: Hermès - Lavoisier, 2009. - Vol. 1.

Coello Carlos A. An updated survey of GA-based multiobjective optimization techniques // ACM Computing Surveys (CSUR). - 2000. - 2 : Vol. 32. - pp. 109-143.

Colette Yves et Siarry Patrick Optimisation multiobjectif : Eyrolles, 2002.

Commisariat général du développement durable Les comptes du transport en 2010 [Rapport] : 48^{ème} rapport à la commission des comptes des transports de la nation. - 2011.

Commission Européenne European transport policy for 2010: Time to decide.: White paper , 2001.

Commission Européenne Examen à mi-parcours du Livre blanc sur les transports publié en 2001 par la Commission européenne : Rapport d'information sur la communication de la Commission au Conseil et au Parlement européen : Pour une Europe en mouvement Mobilité durable pour notre continent, 2007.

Deloitte Efficacités énergétiques et environnementales des modes de transport : Etude réalisée pour le compte de l'ADEME. - 2008.

Deneubourg Jean-Louis et Goss S Collective patterns and decision-making // Ethology, Ecology & Evolution. - 1989. - Vol. 1. - pp. 295-311.

Dessouky Maged M. et Diana Marco A new regret insertion heuristic for solving large-scale dial and-ride problems with time windows // Transportation research. Part B : methodological. - 2004. - 6 : Vol. 38. - pp. 539-557.

Dijkstra E. W. A Note on Two Problems in Connexion with Graphs // Numerische Mathematlk. - 1959. - pp. 269 - 271.

Doniec Arnaud Prise en compte des comportements anticipatifs dans la coordination multi-agent: Application à la simulation de trafic en carrefour : Thèse de doctorat de l'Université de Valenciennes et du Hainaut Cambrésis. - 2006.

Dréo Johann [et al.] Métaheuristiques pour l'optimisation difficile : Eyrolles, 2003.

- Durfee E.H. et V.R.Lesser** Partial global planning: a coordination framework for distributed hypothesis formation // IEEE Transactions on Systems, Man and Cybernetics. - 1991. - 5 : Vol. 21. - pp. 1167 - 1183 .
- Fayech Basma** Régulation des réseaux de transport multimodal : systèmes multiagents et algorithmes évolutionnistes: Thèse de doctorat, l'Université des Sciences et Technologies de Lille et l'Ecole Centrale de Lille. - France : 2003.
- Feki Mohamed Firas** Optimisation distribuée pour la recherche des itinéraires multiopérateurs dans un réseau de transport co-modal: Thèse de doctorat de l'Ecole Centrale de Lille. - France , 2010.
- Ferber Jacques** Les Systèmes multi-agents: Vers une intelligence collective : InterEditions, 1995.
- Fischer Klaus [et al.]** A model for cooperative transportation scheduling // In the proceedings of the 1st International Conference on MAS. - 1999. - pp. 109-116.
- Flores-Mendez Roberto A.** Towards a standardization of multi-agent system framework [Revue] // ACM Magazine. - 1999. - 4 : Vol. 5. - pp. 18 - 24 .
- Ford L. R. [et al.]** Flows in Networks : Princeton University, 1962.
- Genin Thomas et Aknine Samir** Coalition Formation Strategies for Self-Interested Agents in Task Oriented Domains // Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology. - 2010. - Vol. 2. - pp. 205-212 .
- Giannopoulos G.A.** The Application of Co-modality in Greece: a Critical Appraisal of Progress in the Development of Co-modal Freight Centres and Logistics Services [Revue] // Transport, Infrastructures, Logistic. - 2008. - 2 : Vol. 15. - pp. 289-301.
- Gille, Alain** La co-modalité outil du développement durable // Transports. - 2006. - 436. - pp. 73-82.
- Giorgini Paolo, Abdel-Naby Sameh et Sottini Francesco** ANDIAMO: A Multiagent System to Provide a Mobile-based Rideshare Service: Rapport technique DIT-06-097. - 2006.
- Goldberg David E.** Genetic algorithms in search, optimisation and machine learning: Addison Wesley, 1989.
- Green and ITS** An overview of innovations for a sustainable transport system in Stockholm: Sweco 2009. - 2009.

Hanon David Modèle décisionnel orienté comportement fondé sur le vote : Application à la navigation d'agents autonomes en environnement simulé: Thèse de doctorat de l'Université de Valenciennes et du Hainaut-Cambrésis. - 2006.

Holland John H. Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence : MIT Press, 1992.

Horn M.E.T Multi-modal and demand-responsive passenger transport systems: a modelling framework with embedded control systems // Transportation Research Part A: Policy and Practice. - 2002. - 2 : Vol. 36. - pp. 167–188.

Jeribi Karama, Hinda Mejri, Hayfa Zgaya et Slim Hammadi Multi-agent Based Evolutionary Method to Optimize Vehicle sharing Services// International Conference WSEAS, Decembre 14-16, 2009, Tenerife, Canary Islands, Spain.

Jeribi Karama, Hinda Mejri, Hayfa Zgaya et Slim Hammadi Combination of an Evolutionary Approach and Multi-agent Coalition in a co-modal Transport System // 10th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 2012). - 2012.

Jeribi Karama, Hinda Mejri, Hayfa Zgaya et Slim Hammadi Distributed Graphs for solving Co-modal Transport problems // 14th International IEEE Conference on Intelligent Transportation Systems (IEEE ITSC 2011). - Washington, District of Columbia. : [s.n.], 5-7 Octobre 2011.

Jeribi Karama, Hinda Mejri, Hayfa Zgaya et Slim Hammadi Vehicle Sharing Services Optimization Based on Multi-Agent Approach // 18th World Congress of the International Federation of Automatic Control. - 2011.

Jeribi Karama, Hinda Mejri, Hayfa Zgaya et Slim Hammadi Distributed Architecture for a Co-modal Transport System // IEEE International Conference on Systems, Man, and Cybernetics (IEEE SMC 2011), Octobre 9-12, 2011

Jones D.F, Mirrazavi S.K et Tamiz M Multi-objective meta-heuristics: An overview of the current state-of-the-art // European Journal of Operational Research. - 2002. - 1 : Vol. 137. - pp. 1–9.

Kacem Imed Ordonnancement multicritère des jobs-shops flexibles: formulation, bornes inférieures et approche évolutionniste coopérative : Thèse de doctorat de l'Ecole Centrale de Lille et de l'Université de Lille 1. - 2003.

Kamoun Mohamed Amine Conception d'un système d'information pour l'aide au déplacement multimodal: Une approche multi-agent pour la recherche et la composition des

itinéraires en ligne : Thèse de doctorat, Ecole Centrale de Lille et l'Université des Sciences et Technologies de Lille. - France , 2007.

Khalifa Ismahène Hadj Approches de modélisation et d'optimisation pour la conception d'un système interactif d'aide au déplacement dans un hypermarché : Thèse de doctorat de l'Ecole Centrale de Lille. - 2011.

Kirkpatrick S., Gelatt C. D. et Vecchi M. P. Optimization by Simulated Annealing [Revue] // Science. - 1983. - 4598 : Vol. 220. - pp. 671-681.

Kothari Amit B. Genghis - A Multiagent Carpooling System [Revue] // B.Sc. in Computer Science. - 2004.

Lacomme P., Prins C. et Tanguy A. Optimisation par colonies de fourmis pour les tournées sur arcs // 4^{ème} Conférence Francophone de Modélisation et Simulation MOSIM03. - 2003.

Legrand Philippe et Clavel Robert Le covoiturage dynamique: étude préalable avant expérimentation: Rapport d'étude du CERTU. - 2009.

Leriche Sébastien architectures à composants et agents pour la conception d'applications réparties adaptables : Thèse de doctorat de l'Université Toulouse III. - Toulouse , 2006.

Ljungberg Magnus et Lucas Andrew The OASIS Air Traffic Management System [Revue] // In proceedings of the 2nd Pacific Rim International Conference on Artificial Intelligence, PRICAI'92. - Seoul, Korea :, 1992.

Lopez Pierre et Roubellat François Ordonnancement de la production : Hermes Science Publications, 2000.

Ma Tai-Yu Modèle dynamique de transport basé sur les activités : Thèse de doctorat de l'Ecole Nationale des Ponts et Chaussées. - 2007.

Mandiau René et Strugeon Emmanuelle Grislin Le Systèmes multiagents: Techniques de l'Ingénieur S7216 traité informatique Industrielle. - 2002.

MAZOUÉ Sophie, Calcul de facteurs d'émission des modes de transports en commun urbains, Predit 2002-2007, 18 Septembre 2007.

Meijler Theo Dirk et Nierstrasz Oscar Beyond Objects: Components - 1998.

Mesghouni Khaled Application des algorithmes évolutionnistes dans les problèmes d'optimisation en ordonnancement de production : Thèse de doctorat de l'Université de Lille 1. - 1999.

Michalewicz Zbigniew Evolutionary computation techniques for nonlinear programming problems // International Transactions in Operational Research. - 1994. - 2 : Vol. 1. - pp. 223–240.

- Miettinen Kaisa** Nonlinear Multiobjective Optimization : Springer 1st edition, 1998.
- Modesti Paola et Sciomachen Anna** A utility measure for finding multiobjective shortest paths in urban multimodal transportation networks // European Journal of Operational Research. - Décembre 1998. - 3 : Vol. 111. - pp. 495-508.
- Moore E.F.** The Shortest Path Through a Maze // International Symposium on the Theory of Switching. - 1959. - pp. 285-292.
- Moulin Bernard et Chaib-draa Brahim** An overview of distributed artificial intelligence : John Wiley & Sons, 1996.
- Müller I. et R. Kowalczyk P. Braun** Towards Agent-Based Coalition Formation for Service Composition // IEEE/WIC/ACM International Conference on Intelligent Agent Technology . - 2006. - pp. 73 - 80 .
- Nawaz Muhammad, Jr E Emory Enscore et Ham Inyong** A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem // its journal Omega. - 1983. - 1 : Vol. 11. - pp. 91-95.
- Othmani Imed** Optimisation multicritère: fondements et concepts : Thèse de doctorat de l'Université Joseph Fourier Grenoble 1. - 1998.
- OuldSidi Mohamed Mahmoud** Contribution à l'amélioration des systèmes d'aide à la décision : Thèse de doctorat de l'Ecole Centrale de Lille et l'Institut National de Recherche sur les Transports et leur Sécurité. - France :, 2006.
- Pallottino S. et Scutella M.G** Shortest Path Algorithms in Transportation models: classical and innovative aspects : Rapport Technique TR 97-06. - 1997.
- Pruitt Dean G.** Negotiation behavior : Academic Press, 1981.
- R.Mandiau, E.G.L.Strugeon et G.Agimont** Study of the influence of organizational structure on the efficiency of a multi-agent system // Networking and information systems journal. - 1999. - 2 : Vol. 2. - pp. 153-179.
- Rehrl Karl, Brunsch Stefan et Mentz Hans-Joachim** Assisting Multimodal Travelers: Design and Prototypical Implementation of a Personal Travel Companion // IEEE Transactions on Transportation Systems. - Mars 2007. - 1 : Vol. 8. - pp. 31 - 42.
- Rouillard J., Vantroys T. et Chevrin V.** Les architectures orientées service : Une approche pragmatique des SOA : Vuibert, 2007.
- Roy Bernard et Bouyssou Denis** Aide multicritère à la décision : méthodes et cas : Economica, 1993.

Serugendo Giovanna Di Marzo [et al.] Engineering Self-Organising Systems: Nature-Inspired Approaches to Software Engineering : Volume 2977 de Lecture Notes in Computer Science: Lecture Notes in Artificial Intelligence, 2004.

Sghaier Manel Combinaison des techniques d'optimisation et de l'intelligence artificielle distribuée pour la mise en place d'un système de covoiturage dynamique : Thèse de doctorat de l'Ecole Centrale de Lille. - 2011.

Solnon Christine Contributions à la résolution pratique de problèmes combinatoires : des fourmis et des graphes : Mémoire pour l'obtention de l'Habilitation à Diriger des Recherches. - 2005.

Stach Christoph Saving time, money and the environment - vHike a dynamic ride-sharing service for mobile devices // IEEE International Conference on Pervasive Computing and Communications (PERCOM Workshops). - 21-25 Mars 2011. - pp. 352-355.

Sung Kiseok [et al.] Shortest paths in a network with time-dependent flow speeds // European Journal of Operational Research. - 2000. - 1 : Vol. 121. - pp. 32-39.

Szypersky C. Component Software-Beyond Object-Oriented Programming: Addison-Wesley, 1998.

Vallée Thomas et Yildizoglu Murat Présentation Des Algorithmes Génétiques et de Leurs Applications En Économie // Revue d'Economie Politique. - 2003. - pp. 114:711-745. .

Veldhuizen David Allen Van Multiobjective evolutionary algorithms: classifications, analyses, and new innovations: Thèse de doctorat de Air Force Institute of Technology Wright Patterson AFB, OH, USA. - 1999.

Verrons Marie-Hélène GeNCA : un modèle général de négociation de contrats entre agents: Thèse de doctorat de l'Université des Sciences et Technologies de Lille. - 2004.

Viroli Mirko, Ricci Alessandro et Omicini Andrea Operating instructions for intelligent agent coordination // The Knowledge Engineering Review. - Mars 2006. - 1 : Vol. 21. - pp. 49 - 69 .

Wang Jinchang et Kaempke Thomas Shortest route computation in distributed systems // Computers & Operations Research. - 2004. - 10 : Vol. 31. - pp. 1621-1633.

Weiss Gerhard Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence : MIT Press, 1999.

Yan Shangyao et Chen Chun-Ying A model and a solution algorithm for the carpooling problem with pre-matching information // Computers & Industrial Engineering. - Octobre 2011. - 3 : Vol. 61. - pp. 512-524.

Zang Zihui et Cai Wenxue A dynamic shortest path algorithm based on real-time traffic information in the urban public transit network // IEEE International Conference on Service Operations and Logistics, and Informatics. - 2008. - pp. 1500 - 1504 .

Zgaya Hayfa Conception et optimisation distribuée d'un système d'information d'aide à la mobilité urbaine : Une approche multi-agent pour la recherche et la composition des services liés au transport : Thèse de doctorat de l'Ecole Centrale de Lille. - France , 2007.

Zhang Jianwei [et al.] A multimodal transport network model for advanced traveler information systems // Proceedings of the 1st ARTIFACT Workshop within the Proceedings of the 2nd International Conference on Ambient Systems, Networks and Technologies. - Ontario, Canada , 19-21 Septembre 2011. - pp. 912-919.

Zhu Tongyu et Xiang Wang Towards Optimized Routing Approach for Dynamic Shortest Path Selection in Traffic Networks // International Conference on Advanced Computer Theory and Engineering. - 2008. - pp. 543 - 547 .

Zidi Issam Modélisation et Optimisation d'un Système de Transport à la Demande Multicritère et Dynamique : Thèse de doctorat de l'Ecole Centrale de Lille et l'Ecole Nationale des Sciences de l'Informatique de Tunisie . - 2012.

Zidi Kamel Système Interactif d'Aide au Déplacement Multimodal (SIADM) [Rapport] : Thèse de doctorat de l'Université des Sciences et Technologies de Lille et l'Ecole Centrale de Lille. - 2006.

Titre : Conception et réalisation d'un système de gestion de véhicules partagés : de la multimodalité vers la co-modalité

Résumé : De nos jours, l'intérêt porté à la préservation de l'environnement à travers la réduction des émissions de gaz à effet de serre prend de plus en plus d'ampleur. Depuis 2006, la politique multimodale a évolué vers une politique co-modale qui n'oppose plus la voiture au transport public mais encourage une combinaison de tous les modes de transport sans favoritisme dans le but d'une optimisation du service. Placés dans ce cadre, le but de cette thèse est de mettre en œuvre un système de gestion de véhicules partagés qui recouvre tous les services de transports existants tel que le transport public, le covoiturage, les véhicules en libre service et qui capable de satisfaire les demandes des utilisateurs en leur fournissant des itinéraires co-modaux optimisés en terme de temps, coût et émission des gaz à effet de serre tout en respectant leurs préférences et priorités. En recevant plusieurs requêtes simultanées en un court laps de temps, le système doit être capable à la fois de décomposer les solutions en tronçons que nous appelons Routes, en respectant toutes les similarités entre les différentes demandes et de regrouper les informations de manière cohérente pour déterminer les combinaisons de Routes possibles. Vu l'aspect dynamique et distribué du problème, une stratégie de résolution efficace mettant à profit une mixture de concepts ; à savoir les systèmes multi-agents et l'optimisation a été mise en place. Les résultats expérimentaux présentés dans cette thèse justifient l'importance de la co-modalité et la nécessité de mettre à profit la complémentarité entre les véhicules partagés et les autres moyens de transport à travers un système intelligent et global.

Mots-clés : Transport co-modal, Véhicules partagés, Système multi-agent, Optimisation, Algorithmes évolutionnaires, Graphe de transfert.

Title: Design and Implementation of shared vehicles system : From multimodality to co-modality

Abstract: Nowadays, the protection of the environment through the reduction of greenhouse gases is becoming more and more important. In order to resolve environmental problems, a multimodal policy is firstly adopted in order to encourage the use of public transport. Since 2006, a new notion: the co-modality is introduced and it consists on developing infrastructures and taking measures and actions that will ensure optimum combination of individual and public transport modes. In this context, the purpose of this thesis is to implement co-modal transport system that covers all the existing transport services such as the public transport, the carpooling or the free use vehicles (bikes, cars). In order to satisfy the user's requests, the system offers optimized co-modal itineraries in terms of three criteria: total time, total cost and greenhouse gases emission taking into account their preferences and constraints. In a short time interval, many transport users can formulate simultaneously a set of requests. So the system should find feasible decompositions in terms of independent sub-itineraries called Routes recognizing similarities and recognize the different possibilities of Routes Combinations to compose each itinerary demand. Considering the dynamic and distributed aspect of the problem, an effective strategy combining different concepts like multi-agent system and optimization methods is applied. The experimental results presented in this thesis justify the importance of co-modality and the necessity of taking advantage of the complementarity between the shared vehicles and other means of transportation through an intelligent and global system.

Key words: Co-modal transport, Shared vehicles, Multi-agent system, Optimization, Evolutionary algorithms, Transfer graph