



HAL
open science

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

Hayfa Zgaya

► **To cite this version:**

Hayfa Zgaya. 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. Modélisation et simulation. Ecole Centrale de Lille, 2007. Français. NNT : . tel-00160802

HAL Id: tel-00160802

<https://theses.hal.science/tel-00160802>

Submitted on 8 Jul 2007

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 : 49

Ecole Centrale de Lille

THESE

Présentée en vue d'obtenir le grade de

DOCTEUR

Spécialité : Automatique et Informatique Industrielle

par

Hayfa ZGAYA

Doctorat délivré par l'Ecole Centrale de Lille

**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**

Soutenue publiquement le 6 juillet 2007 devant le jury :

Président :	Pierre BORNE	Professeur, Ecole Centrale de Lille
Rapporteurs :	Bernard DESCOTES-GENON	Professeur, Université Joseph Fourier de Grenoble
	Alain QUILLIOT	Professeur, Université Blaise Pascal de Clermont-Ferrand
Examineurs :	Francis BRETAEU	Directeur du Département Logistics Programs, EADS
	Arnaud FREVILLE	Professeur, Direction Recherche du CR-NPDC
	Antoine JARRY	Responsable d'exploitation TRANSVILLE
	René MANDIAU	Professeur, Université de Valenciennes
Directeur :	Slim HAMMADI	Professeur, Ecole Centrale de Lille

Thèse préparée dans le laboratoire LAGIS UMR 8146 à l'Ecole Centrale de Lille.

Résumé

Les travaux de recherche présentés dans cette thèse s'intègrent dans le cadre du projet national VIATIC.MOBILITE du pôle de compétitivité I-TRANS « Le ferroviaire au cœur des systèmes de transports innovants » (<http://www.i-trans.org/index.htm>). De nos jours, les informations liées au déplacement et à la mobilité dans un réseau de transport représentent sans aucun doute un potentiel important. En effet, on peut imaginer une infinité de services innovants liés à la mobilité, non seulement à destination du grand public, mais également à des entreprises, dans le conseil en mobilité pour leurs plans de déplacement. Le but de cette thèse est donc de fournir un système d'aide à la mobilité qui s'articule autour des motifs de déplacements quotidiens, occasionnels, de tourisme, de culture, etc. avec la possibilité de pouvoir bénéficier d'une information pertinente et exploitable.

Ces travaux de recherche visent à mettre en œuvre un Système d'Information de Transport Multimodal (SITM) pour optimiser la gestion de flux des requêtes utilisateurs qui peuvent être nombreuses et simultanées. Dans ce cas, le SITM doit pouvoir procéder par des processus de décomposition des requêtes simultanées en un ensemble de tâches indépendantes. Chaque tâche correspond à un service qui peut être proposé par plusieurs fournisseurs d'information, en concurrence, avec différents coûts, formats et temps de réponse. Un fournisseur d'information voulant proposer ses services via le SITM, doit d'abord y enregistrer son système d'information, en assumant la responsabilité des aspects juridiques et qualitatifs de ses données. Le SITM est donc lié à un Réseau informatique Étendu et distribué de Transport Multimodal (RETM) qui comporte plusieurs sources d'information hétérogènes des différents services proposés aux utilisateurs de transport.

L'aspect dynamique, distribué et ouvert du problème, nous a conduits à adopter une modélisation multi-agent pour assurer au système une évolution continue et une flexibilité pragmatique. Le système multi-agent proposé s'appuie sur les métaheuristiques pour la recherche et la composition des services; la recherche des services se base sur le paradigme Agent Mobile (AM) utilisant un algorithme d'optimisation dynamique de construction des Plans De Routes (PDR). Cette première étape d'optimisation prépare les parcours des AMs en prenant en considération l'état du RETM. La composition des services utilise les algorithmes évolutionnistes pour optimiser les réponses en termes de coût et de temps, sachant qu'une réponse à une requête utilisateur ne doit pas dépasser un temps maximum autorisé et qu'un utilisateur cherche toujours à avoir le meilleur rapport qualité prix pour les services qu'il demande.

Enfin, le SITM prend en considération les éventuelles perturbations sur le RETM (pannes, goulets d'étranglements, etc.) pour satisfaire les requêtes utilisateurs dans tous les cas de figure. Dans ce contexte, nous avons créé un protocole de négociation entre les agents mobiles et les agents responsables des choix des fournisseurs d'information pour les services demandés, ces agents sont appelés agents Ordonnanceurs. Le protocole proposé dépasse les limites d'une communication agent traditionnelle, ce qui nous a incités à associer au système une ontologie flexible qui permet d'automatiser les différents types d'échanges entre les agents grâce à un vocabulaire approprié.

Les résultats expérimentaux présentés dans cette thèse justifient l'utilisation du paradigme agent mobile dans notre système qui remplace parfaitement bien les paradigmes classiques telle que l'architecture client/serveur. Les simulations présentées montrent

différents scénarios de gestion d'un nombre des requêtes Δ_ε – simultanées plus ou moins important. En effet, quelque soit le nombre de requêtes utilisateurs formulées pendant un court laps de temps Δ_ε , le système se charge de leur décomposition, de l'identification des services demandés et des fournisseurs d'information susceptibles d'y répondre.

Mots-clés

Transport multimodal, optimisation distribuée, algorithmes évolutionnaires, systèmes multi-agents, négociation, agent mobile, ontologie.

Abstract

Research tasks presented in this thesis belongs to the French national project VIATIC.MOBILITE from the industrial cluster I-TRANS « France highlights leading-edge technology in rail systems and innovative transport » (<http://www.i-trans.org/index.htm>). Nowadays, information related on displacement and mobility in a transport network represents certainly a significant potential. Indeed, we can imagine infinity of innovating services related to mobility, not only intended for general public, but also for companies, to a relevant control of the mobility for their displacement plans. The main goal of our work is to provide a system with a mobility assistance, which is related to the daily or occasional traffic motivations, tourism an culture occupations, etc. with a possibility to profit from relevant and exploitable information.

This work aims to design, optimize and implement a Multimodal Transport Information System (MTIS) to optimize the management of the data flow of the users' requests, which can be simultaneous and numerous. That's why , the MTIS has firstly to decompose each set of simultaneous requests into a set of independent sub-requests called tasks. Each task corresponds to a service which can be proposed, differently, by several information providers, in competition, with different cost, response delay and different size and format of the data. An information provider, which aims to propose some services through the MTIS, has firstly to register its information system, by assuming the responsibility for the legal and qualitative aspects of the correspondent data. Thus, the MTIS is related to an Extended and distributed Transport Multimodal Network (ETMN) which contains several heterogeneous data sources including the different proposed services to transport users.

The dynamic and distributed aspects of the problem incite us to adopt a multi-agent approach to ensure a continual evolution and a pragmatic flexibility of the system. The proposed multi-agent system is based on metaheuristics for the research and the composition of the services; services research is based on the Mobile Agent paradigm (MA) using a dynamic optimisation algorithm for the MA Workplans design. This first step of optimisation prepares the MA routes, taking into account the network state. The services composition uses evolutionary algorithms to optimize the responses in terms of costs and delays, knowing that a response to a user request must respect a fixed due date with a reasonable cost.

Finally, the MTIS takes into account possible disturbance through the EMTN (crash, bottlenecks, etc.) in order to satisfy user requests in all the cases. In this context, we designed a negotiation protocol between mobile agents and optimiser agents of the system, called Scheduler Agents. The proposed protocol exceeds the limits of traditional communication agents what encouraged us to associate to the system a flexible ontology which makes it possible to automate the various types of exchanges between the agents thanks to a suitable vocabulary.

Experimental results presented in this theses, justify the usage of the MA paradigm in our system according to the classical paradigms such as the Client/Server one. The different experimental scenarios show a pertinent management of any amount of Δ_ϵ – simultaneous requests. Indeed, a great number of user requests through a short period of time Δ_ϵ , does not affect the system functioning which decompose them, identifies the required services and the possible information providers.

Key-words

Multimodal transport, distributed optimization, evolutionary algorithm, multi-agent system, negotiation, mobile agent, ontology.

REMERCIEMENTS

Je tiens à remercier en tout premier lieu mon directeur de thèse le Professeur Slim HAMMADI, Professeur à l'Ecole Centrale de Lille, de m'avoir accueillie, soutenue, encouragée et encadrée tout au long de cette thèse, de m'avoir permis de travailler sur un sujet passionnant et innovant, d'avoir cru en mes capacités et de m'avoir fait découvrir réellement et d'aimer sincèrement la recherche.

Je tiens également à exprimer ma profonde gratitude et mon plus grand respect au Professeur Pierre BORNE, Professeur à l'Ecole Centrale de Lille, pour m'avoir accueillie au sein de l'équipe ID du LAGIS et pour avoir accepté de présider le jury de soutenance de cette thèse.

Mes plus sincères remerciements s'adressent au Professeur Alain QUILLIOT, Directeur du LIMOS et de la fédération CNRS TIMS et Professeur d'Informatique à l'Université Blaise Pascal de Clermont-Ferrand et au Professeur Bernard DESCOTES-GENON, Professeur à l'Université Joseph Fourier de Grenoble, 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 remercie les autres membres du jury pour avoir accepté d'être les examinateurs de cette thèse ; j'adresse donc des remerciements tout à fait particuliers au Professeur Arnaud FREVILLE, Professeur et directeur de Recherche au Conseil Régional du Nord Pas De Calais (CR-NPDC). J'aimerais également adresser spécialement un grand MERCI au Professeur René MANDIAU, Professeur à l'Université de Valenciennes. Je remercie aussi vivement les industriels qui ont bien voulu examiner ce travail et participer à ce jury, à savoir Mr Antoine JARRY responsable d'exploitation TRANSVILLE et Mr Francis BRETAUDEAU, directeur du Département Logistics Programs à EADS.

Je ne pourrais oublier de remercier le personnel du LAGIS et de l'Ecole Centrale de Lille pour leur gentillesse, disponibilité et serviabilité.

Enfin, je remercie du plus profond de mon cœur ma famille qui m'a toujours encouragée et soutenue ; jamais je n'aurais pu réussir sans son amour qui m'a permis de tenir et de m'alimenter de courage et de détermination.

Table des matières

index des Figures.....	13
index des tableaux	14
index des tables	14
Introduction générale.....	15
CHAPITRE I. Le Transport Multimodal pour la mobilité urbaine: vers des Systèmes d'Information distribués.....	19
I.1 Introduction	21
I.2 La mobilité urbaine	22
I.2.1 Le comportement de mobilité	23
I.2.2 Choix et contraintes des pratiques modales	25
I.2.3 La nouvelle ère de l'information d'aide à la mobilité urbaine.....	25
I.3 Le Transport Multimodal	28
I.3.1 Définitions.....	28
I.3.2 Les réseaux de Transport public Multimodal.....	28
I.3.3 Les problèmes du Transport Multimodal	29
I.4 Les Systèmes d'Information de Transport Multimodal	34
I.4.1 Les systèmes existants.....	34
I.4.2 Intégration de l'information	40
I.5 Les pôles de compétitivités	44
I.5.1 Le projet « I-Trans » de pôle de compétitivité du Nord Pas-de-Calais.....	44
I.5.2 Le projet VIATIC du I-TRANS	45
I.6 Conclusion.....	45
CHAPITRE II. Systèmes multi-agents et optimisation dans le domaine du Transport : vers une architecture distribuée à base d'agents optimisateurs.....	47
II.1 Introduction	49
II.2 Les Systèmes Multi-Agents et les méthodes d'optimisation : pourquoi une telle alliance ?.....	49
II.3 Les Systèmes Multi-Agents (SMA).....	50
II.3.1 Agent : Définitions et particularités	50
II.3.2 Les Systèmes Multi-Agents	59
II.3.3 Systèmes d'information et agents logiciels.....	66
II.4 Les SMA dans le domaine du Transport.....	67
II.5 Optimisation	68
II.5.1 Définitions.....	68
II.5.2 Classification des problèmes d'optimisation.....	70
II.5.3 L'optimisation multiobjectif	71
II.5.4 La dominance	71
II.5.5 La ε – dominance.....	73
II.5.6 Les méthodes de résolution des problèmes d'optimisation multiobjectif.....	74
II.5.7 Les métaheuristiques	76
II.6 Les algorithmes évolutionnaires dans le domaine du transport	88
II.7 Vers une architecture distribuée à base d'agents optimisateurs.....	89
II.8 Conclusion.....	89
CHAPITRE III. Approche proposée d'optimisation distribuée de recherche et de compositions des services multi-sources au profit des clients de Transport.....	91
III.1 Introduction	93

III.2	Formulation du problème	93
III.3	Architecture multi-agents proposée	97
III.3.1	Les bases de données locales	101
III.3.2	Comportements des agents dans le système.....	102
III.4	Approches et algorithmes d'optimisation proposés	108
III.4.1	Elaboration des PDRIs des agents ACI.....	109
III.4.2	Optimisation et Composition des flux de services par une approche évolutionnaire : calcul des PDRDs.....	114
III.5	Modèle d'archivage dynamique des données (MADD).....	125
III.5.1	Classification des données et archivage	125
III.5.2	Fonctionnement.....	127
III.6	Conclusion.....	128
CHAPITRE IV. Protocole de négociation entre agents et conception d'une ontologie de transport		129
IV.1	Introduction	131
IV.2	Situation de perturbation	132
IV.3	La négociation : la forme d'interaction adoptée.....	134
IV.3.1	Structure de base	134
IV.3.2	Mécanismes de négociation	135
IV.3.3	Modèles de négociation.....	136
IV.4	Le protocole de négociation proposé	138
IV.4.1	L'initiateur.....	138
IV.4.2	Les participants	139
IV.4.3	Le protocole de négociation proposé	141
IV.5	Réattribution des services.....	147
IV.5.1	Algorithme de Reconstruction Dynamique des Chemins des Agents ACI (ARDyCA)	147
IV.5.2	L'algorithme ARDyCA.....	148
IV.6	Conception d'une ontologie de Transport.....	151
IV.6.1	Définition	151
IV.6.2	L'apport d'un support ontologique pour les processus de négociation.....	152
IV.6.3	Travaux antérieurs.....	152
IV.7	Une ontologie flexible pour le Transport	154
IV.7.1	Niveaux d'ontologies	154
IV.7.2	Les prédicats.....	155
IV.7.3	Les termes	156
IV.7.4	Utilisation des ontologies	160
IV.8	Conclusion.....	160
CHAPITRE V. simulations et Résultats.....		161
V.1	Introduction	163
V.2	Choix de la plateforme multi-agent.....	163
V.3	Outils et paramétrages.....	164
V.4	Justification du choix du paradigme Agent Mobile et son efficacité dans le SITM 166	
V.4.1	Frais de transmission de données : un modèle pseudo-formel.....	167
V.4.2	Résultats de simulation.....	168
V.5	L'apport du Modèle d'Archivage Dynamique des Données (MADD) dans le SITM 173	
V.6	L'approche évolutionnaire : choix de la meilleure solution.....	175
V.7	Exemple détaillé de mise en œuvre du processus de négociation.....	176

V.8 Conclusion.....	186
Conclusion GENERALE.....	189
Glossaire.....	191
Bibliographie.....	193
Annexes.....	201
Annexe A.....	203
Annexe B.....	221
Annexe C.....	233

INDEX DES FIGURES

Figure I.1- Evolution des transports des personnes en France.....	23
Figure I.2 - Service de composition de trajet sur le site Internet de Nord Pas-de-Calais.....	27
Figure I.3- Borne intermodale.....	27
Figure I.4- Processus classique de planification.....	31
Figure I.5- Principe de la régulation.....	32
Figure II.1 - Mécanisme de déplacement d'un agent mobile.....	56
Figure II.2– Comparaison des modèles Client/Serveur et AM.....	58
Figure II.3– Situation de conflit.....	61
Figure II.4– Le tableau noir.....	63
Figure II.5– OMA à structure hiérarchique.....	64
Figure II.6– OMA à structure de marché.....	65
Figure II.7– OMA à structure de communauté.....	65
Figure II.8– OMA à structure de société.....	66
Figure II.9– Les différents espaces de recherche.....	69
Figure II.10 – Les différents minima.....	70
Figure II.11– L'optimalité locale au sens de Pareto.....	72
Figure II.12– Le théorème du contact.....	73
Figure II.13– La dominance et l' ε – dominance.....	74
Figure II.14– Allure de la fonction objectif d'un problème d'optimisation difficile.....	76
Figure II.15– Exploration de l'ensemble des solutions réalisables par une approche de recherche locale.....	77
Figure II.16– Exploration de l'ensemble des solutions réalisables par une approche constructive.....	81
Figure II.17– Exploration de l'ensemble des solutions réalisables par une approche évolutive.....	82
Figure II.18– Fonctionnement générique d'un algorithme évolutionnaire.....	84
Figure II.19– Combinaisons de différentes approches.....	88
Figure III.1 - Identification des nœuds.....	94
Figure III.2 - Architecture du système.....	99
Figure III.3- Exemple de scénario de changement d'état d'une société.....	99
Figure III.4- Comportement global du SITM pendant une période de formulation.....	100
Figure III.5- Organisation du SITM.....	102
Figure III.6- Comportement de l'agent AI.....	103
Figure III.7- Comportement de l'agent AId.....	104
Figure III.8- Comportement de l'agent AO.....	105
Figure III.9- Comportement de l'agent ACI.....	106
Figure III.10- Comportement de l'agent AF.....	107
Figure III.11- Comportement d'une société d'agents.....	108
Figure III.12- Exemple de RETM.....	110
Figure III.13- Approche évolutionnaire.....	124
Figure III.14- Archivage dynamique de l'information.....	126
Figure III.15- Modèle de l'Archivage dynamique des données.....	128
Figure IV.1- Situation de perturbation.....	133
Figure IV.2– Exemple de chromosome modifié suite à une perturbation.....	133
Figure IV.3– Structure de base d'une négociation.....	135
Figure IV.4 – Le Protocole du Réseau Contractuel.....	137

Figure IV.5– Exemple de PDRD restant d’un agent participant.....	139
Figure IV.6– Exemple de priorités d’un agent participant.....	140
Figure IV.7– Exemple de préférences d’un agent participant.....	141
Figure IV.8– Exemple de contraintes d’un agent participant.....	141
Figure IV.9- Le protocole de négociation proposé	142
Figure IV.10- Système d’Ontologies Flexible pour de Transport (SOFT)	154
Figure IV.11- Prédicats	156
Figure IV.12- Concepts	158
Figure IV.13- Actions d’agents	159
Figure V.1– Communication des agents dans le SITM	165
Figure V.2– Interface de résultats	165
Figure V.3– Résultat de simulation.....	169
Figure V.4 – Comparaison des paradigmes AM et CS pour une seule instance de FeTAR ..	172
Figure V.5 - Variation des instances FeTAR	172
Figure V.6- Variation de la taille de données des réponses	173
Figure V.7– Exemple d’amélioration du SITM par le MADD	175
Figure V.8 - Fronts optimaux	176
Figure V.9– Application du protocole de négociation	186

INDEX DES TABLEAUX

Tableau II-1 – Classification des problèmes d’optimisation.....	70
Tableau III-1- Analogie entre le PJSF et le PRCS	97
Tableau III-2– Notations	110
Tableau III-3- Temps de parcours	111
Tableau III-4– Indicateur temporel et de mise à jour de classification de l’information.....	126
Tableau V-1– Exemple d’indicateurs.....	174

INDEX DES TABLES

Table III-1- Exemple de table de services.....	96
Table III-2- Exemple d’une instance FeTAR.....	116
Table III-3 - Exemple de vecteur d’évaluation	123
Table IV-1– Exemple de chromosome touché par une perturbation	132
Table V-1- Décomposition des requêtes	170
Table V-2- Chromosome CH ₁	171
Table V-3– Table des PDRDs.....	171
Table V-4 - Chromosome CH ₂	174
Table V-5- Chromosome CH ₃	179

INTRODUCTION GENERALE

Les travaux de recherche présentés dans cette thèse s'intègrent dans le cadre du projet national VIATIC.MOBILITE du pôle de compétitivité I-trans*, qui réunit les principaux acteurs français du ferroviaire et des systèmes de transports innovants et qui s'inscrit dans le cercle fermé des 15 pôles à vocation mondiale labellisés par le Comité Interministériel d'Aménagement du Territoire le 12 juillet 2005.

Ce projet vise à faciliter la mobilité urbaine des personnes en les aidant à bouger dans les meilleures conditions possibles depuis la préparation du déplacement jusqu'à l'arrivée au point de destination. Le but est de fournir toutes les informations nécessaires en temps réel ainsi que d'occuper le voyageur pendant son déplacement en lui proposant divers services, qui peuvent aussi bien correspondre à des services de transport (une portion de chemin, une zone géographique bien reconnue, le trafic routier, etc.) ou à des services connexes (événements culturels, prévisions météorologiques, jeux, etc.). L'information fournie doit être pertinente, instantanée et interactive en temps réel avant et au cours du déplacement et via n'importe quel dispositif de connexion à diffusion multi-canal de l'information (ordinateur portable, ordinateur de bureau, GSM, PDA, etc.).

L'objectif à atteindre représente un véritable défi par rapport à la croissance exponentielle des services disponibles sur les réseaux distribués à grande échelle. En effet, une information demandée correspond à un service, qui peut être proposé différemment, par plusieurs fournisseurs d'information, en concurrence, avec différents coûts, temps de réponse et taille de données. Sachant qu'un ensemble de services peut être sollicité simultanément par un nombre très important d'utilisateurs, des mesures de conception et d'optimisation spécifiques doivent être pris en considération pour faire face aux risques d'effondrement d'un tel système d'information.

Dans ce contexte, notre projet se situe dans la conception, la mise en œuvre et l'optimisation d'un système d'Information de Transport Multimodal qui soit ouvert, dynamique et susceptible de gérer des flux de données volumineux, distribués et hétérogènes. Le système en question doit donc être capable de fournir aux clients des réseaux de Transport

* www.i-trans.org

une information pertinente en temps réel avant et pendant leurs déplacements. Malheureusement, les applications réparties sur de tels réseaux ne sont pas faciles à réaliser à cause de leur aspect dynamique dans l'espace et dans le temps. Ce type d'application est également caractérisé par une hétérogénéité syntaxique et sémantique des données réparties et également par une probabilité d'incidence élevée (goulets d'étranglement, échecs, etc.) qui rend sa conception, sa mise en œuvre et son analyse très complexes. Le choix d'une architecture Multi-Agent pour le Système d'Information de Transport Multimodal est donc suffisamment justifié par son caractère dynamique, distribué, hétérogène et ouvert.

Pour accéder et partager correctement et de la meilleure manière possible les données distribuées situées dans un Réseau Étendu de Transport Multimodal (RETM), la technologie mobile [Pharm et al.,98] [Theilmann et al., 99] peut être complémentaire à l'intelligence artificielle puisqu'elle peut réduire considérablement le trafic réseau [Carzaniga et al., 97]. Doter un agent logiciel de mobilité lui permet d'émigrer vers n'importe quel nœud du réseau pouvant héberger des entités mobiles. Les nœuds qu'un Agent Mobile (AM) visite, correspondent à son itinéraire appelé Plan de Route (PDR).

La présente thèse est organisée en 5 chapitres comme suit :

1. Dans le premier chapitre, nous élaborons une étude générale sur l'information liée aux déplacements des personnes et à la mobilité urbaine. Nous exposons ainsi un ensemble non exhaustif de systèmes d'information pour le transport existants et les techniques possibles pour l'intégration d'une information multi-modale ;
2. Le chapitre 2 sera consacré à un état de l'art général sur les systèmes multi-agents et les méthodes d'optimisation ainsi que leur apport dans les systèmes d'information dédiés au transport ;
3. Dans le chapitre 3, nous présentons notre approche d'optimisation distribuée pour la mise en œuvre d'un système d'information multi-agent d'aide à la mobilité urbaine pour la recherche et la composition des services multi-sources au profit des clients de transport;
4. Le chapitre 4 sera consacré aux solutions proposées face à la carence du système présenté, à cause des aléas du réseau d'information du transport. En effet, nous proposons dans ce chapitre, un protocole de négociation entre les

agents mobiles et les agents stationnaires optimisateurs du système, ce qui nous incite à mettre en œuvre une ontologie flexible pour le transport ;

5. Dans le dernier chapitre, nous présentons des scénarios d'utilisation du système adopté pour démontrer l'efficacité des solutions proposées.

**CHAPITRE I. LE TRANSPORT MULTIMODAL POUR LA
MOBILITE URBAINE: VERS DES SYSTEMES D'INFORMATION
DISTRIBUES**

I.1 Introduction

La loi française d'Orientation sur les Transports Intérieurs (LOTI) du 30 décembre 1982 a affirmé un droit au transport devant permettre de se déplacer dans les meilleures conditions de qualité, de prix ainsi que de coûts pour la collectivité. D'autant plus que l'heure est à la recomposition du paysage de la mobilité tant du côté de l'offre que de la demande.

Du côté de la demande, la mobilité est plus dispersée sur les plans de la géographie, des modes, des fréquences et des destinations. La mobilité « domicile-travail » n'est plus dominante et les technologies personnelles ouvrent des possibilités en terme d'information sur les parcours, par leurs multiples usages pendant les temps de transport. Cette nouvelle mobilité s'enrichit donc d'activités conjointes aux déplacements.

Du côté de l'offre, ces nouvelles pratiques de transit conduisent à des accélérations brutales ; Des usages conjugués de la voiture et des TC (Transport en Commun) transforment les flux et impactent sur le paysage urbain et sur l'organisation de la vie quotidienne. En plus, l'amélioration de l'offre du transport renforce la dispersion géographique des déplacements et des modes "doux" – vélos, rollers, cheminements piétons, etc., ce qui affecte les pratiques quotidiennes.

De ce fait, les acteurs des technologies sont focalisés sur les dimensions *mobiles* du quotidien : mobile commerce, mobile office, info mobiles, etc. et les réponses multimédia à apporter, ce qui améliore la fluidité des déplacements par des outils d'informations, de repérage, de guidage. Ces initiatives débouchent sur des produits transversaux (la billettique, la monétique, l'accessibilité à des lieux publics, etc.).

Néanmoins, la croissance exponentielle des services disponibles sur les réseaux distribués à grande échelle représente une véritable rivalité pour répondre minutieusement aux besoins des clients du transport qui exigent des informations appropriées, interactives, instantanées et de meilleurs coûts avant et au cours de leurs déplacements.

Un client du transport est un voyageur qui sollicite une aide au déplacement pouvant être une combinaison de services de transport et de services connexes au transport. Un service de transport peut concerner différents modes de transport et différents opérateurs de transport et évoque essentiellement les itinéraires. Il peut être par exemple une description détaillée d'offres d'itinéraires, des horaires de départs et d'arrivées, des durées de trajets, des coûts de déplacements etc. Un service connexe au transport (information météorologique, événements culturels, renseignements sur des restaurants, des hôtels etc.) n'est pas nécessaire pour décrire

le déplacement en soi mais donne des informations complémentaires de proximité qui peuvent orienter le choix du voyageur dans l'ensemble de l'offre de mobilité.

Ce chapitre est organisé comme suit : Dans le paragraphe I.2, nous abordons les principes de la mobilité urbaine et son lien direct avec l'information du transport ; Le paragraphe I.3 présente le transport multimodal en général, ses principales caractéristiques ainsi que les problèmes évoqués dans ce domaine. Les travaux antérieurs en termes de systèmes d'information et les moyens techniques et technologiques pour leur élaboration sont présentés dans le paragraphe I.4. Enfin, dans le paragraphe I.5, nous présentons le projet « i-trans » et notamment la partie VIATIC.

1.2 La mobilité urbaine

La mobilité urbaine représente l'ensemble des déplacements quotidiens entrepris sur un espace urbain par différents modes de transport et pour différents motifs, d'une origine vers une destination. La notion de mobilité quotidienne dépasse la notion de déplacement car elle intègre également leurs causes et conséquences. L'étude de la mobilité urbaine au quotidien se base donc sur les habitudes et les raisons de déplacements quotidiens des usagers, elle intègre ainsi l'étude de plusieurs disciplines : l'anthropologie¹, la sociologie, la psychologie, la psychosociologie, la géographie, l'économie, la démographie, la socio économie et l'ingénierie. La mobilité quotidienne se base sur la localisation dans les réseaux de transport et la qualité de l'offre proposée, le nombre d'automobiles, le nombre de titulaires du permis de conduire pour le ménage et la possibilité d'utilisation d'un véhicule pour chaque personne. Une approximation du niveau de sociabilité peut être évaluée par le taux de personnes mobiles, le nombre de déplacements par personne et par jour et le nombre d'activités pratiquées à l'extérieur du domicile. Le nombre de déplacements par sortie (chaînage) donne une indication sur la complexité des programmes d'activités ou sur les capacités ou nécessités d'adaptation à des situations de mobilité difficile.

La mobilité quotidienne ne cesse d'augmenter depuis la révolution industrielle du 19^{ème} siècle suite à un développement technologique considérable dans le domaine du transport. Les nouvelles technologies continuent à progresser et offrent une plus grande liberté de se déplacer (Figure I.1).

¹ Science humaine qui étudie l'être humain sous tous ses aspects : sociaux, culturels et physiques

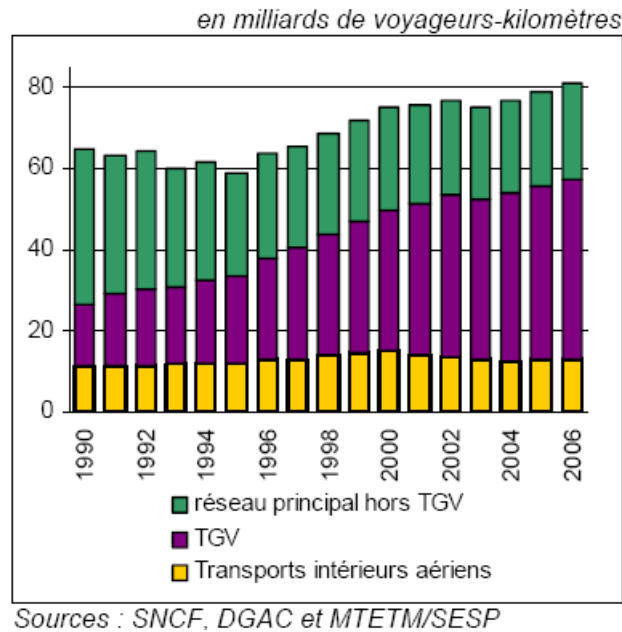


Figure I.1- Evolution des transports des personnes en France

I.2.1 Le comportement de mobilité

L'organisation des déplacements pour le déroulement des activités d'un individu dans sa vie quotidienne résume son comportement de mobilité qui dépend de nombreuses composantes déterministes tels que la décision de partir à un endroit ou de le quitter, le choix de la destination, le choix de l'itinéraire et le choix du mode de déplacement. Par conséquent, le choix de mobilité adhère à un embrouillement décisionnel relativement complexe. En revanche, une aide à la mobilité peut lui procurer les informations nécessaires et suffisantes pour l'aider à se décider et aller de l'avant avec satisfaction. Cette aide à la mobilité peut être liée à la proposition de services, classés en trois catégories :

Un service adapté à une personne non familière au système de la métropole

Quelles informations à procurer à un nouvel arrivant selon sa situation et ses attentes ? Quelles sont les possibilités TC de la Métropole pour son quotidien ? Le plan du réseau est-il intéressant et suffisant ? Quel « kit virtuel de bienvenue » adéquat sa situation ? Existe-il un service spécifique adapté à sa situation ?

Encadrement et interaction

La possibilité de garder le contact avec le client qui veut s'exprimer grâce aux nouvelles technologies, ouvre de nouvelles formes de dialogues plus spontanées et plus directes. Pourquoi le monde des transports en est encore absent ? Une relation

plus étroite peut s'installer entre l'exploitant et son client. La GRC (Gestion Relation Client) ou (CRM Customer Relationship Management) peut-elle s'appliquer aux transports collectifs ? Il faut donc aider à la mise en œuvre des comités de lignes réclamés par les acteurs des transports régionaux. Créer un outil simple, direct et convivial capable de recueillir les avis, les conseils, les réclamations, les félicitations, etc. des voyageurs afin d'améliorer le service rendu, tout en connaissant mieux le client, en particulier ses demandes de déplacement, ses besoins en information (multimodale).

Contraintes temporelles et spatiales

Ce sont des services d'aide à la mobilité liés à un motif de déplacement :

Flâner : proposer par exemple une ballade urbaine ou un parcours culturel : Lille 2004, une course plaisir, une rencontre tribale, etc., en proposant les transports collectifs ;

Apprendre : Comparer en équivalent polluant la consommation pour se rendre d'un point A à un point B de la Métropole, en automobile, en TC, aux heures de pointe, etc. ;

Y aller : Proposer en temps réel le meilleur moyen de se rendre tout de suite à un endroit donné comme à une manifestation incontournable.

Pour les services personnalisés, l'idée proposée ici est de mettre à disposition (services gratuits et payants) des assistants personnels (PDA) auprès de 100 à 200 (peut être plus) ou d'autres dispositifs communs qui pourraient être réservés aux abonnés des transports collectifs.

En général, un individu peut se trouver dans une situation théorique du choix modal, c'est-à-dire qu'il est décidé d'avance sur sa destination et qu'il est face à un choix d'un ou de plusieurs modes de transport, parmi différentes options possibles. Le comportement de mobilité peut également dépendre de la situation d'opportunité modale c'est-à-dire qu'un individu oriente son programme d'activité en fonction de la disponibilité exceptionnelle d'un certain moyen de transport. Dans ce cas, c'est le choix modal qui détermine la planification éventuelle de l'enchaînement des différentes activités. D'un autre côté, il est possible qu'un individu privilégie un certain mode de déplacement, ce qui peut influencer sur le choix de ses lieux d'activités. Dans ce cas, le comportement de mobilité de l'individu dépend du choix

modal, qui est déterminé avant le choix de l'itinéraire et même avant le choix de la destination.

I.2.2 Choix et contraintes des pratiques modales

C'est le lien entre la demande et l'offre de transport, en se basant sur les besoins des individus, leurs attentes par rapport à la qualité de déplacement, leurs motivations et leurs choix fixés avant l'acte de déplacement. Les préférences des pratiques modales tiennent compte de la fiabilité d'un moyen de transport, la minimisation des coûts de déplacement, la valorisation du temps de déplacement, la garantie d'autonomie d'un moyen de déplacement et pourquoi pas la volonté de mise en œuvre d'une sensibilité écologique avec le réchauffement maintenant lisible et remarquable de la planète.

Pour faire des choix il faut s'informer et pour s'informer il faut disposer des moyens nécessaires pour le faire, des moyens adaptés à la mobilité urbaine des personnes, des moyens qui ne sont pas fonctionnels seulement avant et après le déplacement mais surtout pendant le déplacement, moment où nous avons tous plus besoin d'information, d'orientation et d'encadrement. Le but de cette thèse est d'offrir les moyens techniques pour l'aide à la mobilité urbaine en mettant à disposition des personnes, un Système d'Information de Transport Multimodal (SITM) pour la recherche et la composition des services liés au transport.

I.2.3 La nouvelle ère de l'information d'aide à la mobilité urbaine

Le Wi-Fi en test à bord du TGV

Le domaine du service à bord du TGV continue sa progression par la création d'un portail multimédia d'accès à Internet. Les clients TGV (1^{ère} et 2^{nde} classe) pourront ainsi s'y connecter par Wi-Fi en utilisant leur propre matériel et via des liaisons sol-bord satellitaires.

Le Wi-Fi (« Wireless Fidelity ») est un commercial pour la technologie IEEE 802.11x de réseau local Ethernet sans fil (WLAN), il est composé de plusieurs normes qui opèrent sur des fréquences radios différentes : la norme « 802.11b » pour les réseaux sans fil opérant sur la fréquence 2,4 GHz avec une bande passante de 11 Mbps, la norme « 802.11a » pour les réseaux sans fil opérant sur la fréquence 5 GHz avec une bande passante de 54 Mbps et la norme « 802.11g » (combinaison des deux premières) pour les réseaux sans fil opérant sur la fréquence 2,4 GHz avec une bande passante de 54 Mbps.

A partir de l'automne 2007, cinq rames de la flotte TGV EST EUROPEEN seront progressivement équipées pour proposer à la clientèle une première version gratuite de l'offre qui inclut l'accès à Internet, des informations SNCF et générales, thématiques voyage et divertissements. Néanmoins, la nouvelle version ultérieure de l'offre va également inclure des services payants tels que des services pratiques d'achat en ligne, des films et même des accès aux intranets des entreprises des clients.

La phase de test de cette innovation technologique continue jusqu'à l'année prochaine (2008) au terme de laquelle la SNCF décidera ou pas de déployer son offre sur les 52 rames du TGV EST EUROPEEN. Cette décision dépendra de la performance atteinte et surtout de la satisfaction des clients.

Usages touristique

Les services de la mobilité urbaine pour le tourisme n'est pas encore au cœur des innovations technologiques mais déjà en cours de débats² intégrant la disponibilité des guides touristiques en ligne et les services pratiques en continu sur n'importe quel support mobile tel que le téléphone portable ou le PDA. Dans ce contexte, les opérateurs de tourisme doivent aujourd'hui être présents sur le Net pour offrir en ligne leurs services de tourisme (hôtels, locations, restaurants, événements culturels, campings, etc.).

Aide au déplacement

Pour un déplacement d'une origine à une destination, une personne peut avoir une multitude de possibilités de moyens de transport, d'opérateurs de transport, de trajets, de niveaux de confort et de coût, etc. La recherche d'un itinéraire peut être influencée par exemple par l'envie de faire une marche à pied ou le besoin de procurer des services spécialisés pour les personnes à mobilité réduite. Ce domaine dans le transport est appelé ***transport multimodal*** et sera défini et présenté en détail dans le paragraphe I.3.

Dans ce contexte, des services de composition d'itinéraire commencent à apparaître, nous pouvons citer l'exemple du site Internet³ du Nord Pas-de-Calais qui offre sur sa page

² Les Rendez-vous de la stratégie : Innovations technologiques et usages touristiques vendredi 11 mai 2007, organisée par la direction du Tourisme en partenariat avec OSEO.

³ www.ter-sncf.com/nord_pas_de_calais

d'accueil (Figure I.2) un service de composition de trajet à la carte et de porte à porte grâce à un calculateur d'itinéraire en ayant la possibilité de combiner TER, métro, tram ou bus.

Mon trajet

Bienvenue sur le nouveau service de recherche d'itinéraires et d'horaires en Région Nord Pas de Calais.



[aide](#)

Mon point de départ
arrêt, rue, place ou monument commune de départ

Mon point d'arrivée
arrêt, rue, place ou monument commune d'arrivée

Date de mon trajet 04 Mai 2007
Partir vers 20 h 55

Mes préférences de voyage
Arriver le plus rapidement

informations :  

Lancer la recherche

Figure I.2 - Service de composition de trajet sur le site Internet de Nord Pas-de-Calais

D'ailleurs, le petit dernier de la SNCF (mai 2007) est une borne intermodale (§I.3.1), qui va probablement exister un peu partout dans les espaces intermodaux dans les gares et les stations pour fournir des informations sur les horaires et les correspondances entre bus, tram, métro et train (Figure I.3).



Figure I.3- Borne intermodale

I.3 Le Transport Multimodal

I.3.1 Définitions

Un mode de transport peut être assuré par plusieurs opérateurs, ainsi un itinéraire donné peut être intra ou multi opérateur et donc exploité de différentes manières par un seul ou par plusieurs opérateurs. De ce fait découle le concept de la *multimodalité* qui se définit par la possibilité de plusieurs modes de transport (bus, métro, tramway, TGV, etc.) pour un même déplacement entre une origine et une destination. Ceci explique pourquoi il existe plusieurs combinaisons de modes de transport pour tracer un seul trajet entre une origine et une destination ; une seule instance de ces combinaisons représente l'*intermodalité* et se définit donc par l'enchaînement de plusieurs moyens de transport pour un seul déplacement. 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 peut le renseigner sur les modalités (intermodalité et multimodalité) et les conditions de son déplacement et elle peut également l'encadrer et l'orienter pendant son parcours. Enfin, *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, panneaux lumineux, 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.).

I.3.2 Les réseaux de Transport public Multimodal

I.3.2.1 Organisation du transport public en France et en Europe

L'organisation des transports publics en France est contrôlée par les composantes représentatives de l'état c'est-à-dire les régions qui sont composées de départements qui sont eux même composés de communes. Ces autorités organisatrices gèrent la réglementation des activités de transport, l'initiation des projets dans ce domaine ainsi que le contrôle de leurs applications. Ces autorités restent propriétaires des équipements et des infrastructures de transport (parcs de dépôts, lignes de métro, etc.) et peuvent gérer directement leur réseau de transport. Mais en général, ce sont les exploitants, la plupart des sociétés privées, qui gèrent l'exploitation des services. Cette alliance se base sur des conventions de délégation des services publics qui précisent les engagements et les responsabilités des deux parties.

En Europe, et à la différence du cas Français, ce sont les opérateurs privés qui s'organisent pour tirer profit des points intéressants sur le marché du transport selon les avantages observés sur les périmètres. L'état intervient indirectement au niveau de l'organisation comme le financement des lignes, jugées déficitaires, de bus ou de métro ou l'allocation des aides tarifaires pour une classe d'utilisateurs comme les étudiants et les seniors.

I.3.2.2 Configuration d'un réseau de transport multimodal

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 font 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.

I.3.3 Les problèmes du Transport Multimodal

Les études liées au transport multimodal peuvent être classées en deux catégories: le transport côté régulateur et le transport côté client. 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. Le transport côté client représente un axe de recherche pour l'aide à l'information des voyageurs par la modélisation, la mise en œuvre et l'optimisation des systèmes d'information de transport multimodal pour servir au mieux les clients de transport, avant et pendant leurs déplacements.

I.3.3.1 Le transport côté régulateur

Dans l'ensemble des réseaux du transport urbain, l'exploitation d'une ligne se décompose en deux phases distinctes. La première est une phase de conception et d'élaboration du programme de production, c'est à dire, le Tableau de Marche de base (TM) et la seconde correspond à la phase d'adaptation du programme de production aux conditions

réelles d'exploitation. La première phase correspond à la planification, qui se fait en temps anticipé avant la mise en exploitation, et nécessite non seulement l'ordonnancement des véhicules en leur attribuant des courses mais aussi l'ordonnancement du personnel pour leur attribuer des services. La deuxième phase représente la régulation et se déroule en temps réel. Cette phase vise à satisfaire différents objectifs telles que la ponctualité, la régularité et la correspondance. D'ailleurs, la production du transport collectif opère dans un univers essentiellement aléatoire. Les perturbations majeures qui peuvent l'affecter sont:

- les aléas de circulation (congestion) ;
- les perturbations liées au matériel roulant ;
- les problèmes liés au personnel machiniste (absence, apprentissage du métier, etc.) ;
- les variations dans la configuration de la demande ;
- les modifications dans les conditions de circulation et par conséquent la variation des temps de parcours ;
- les erreurs dans l'application des TM.

L'irrégularité apparaît comme la conséquence de l'instabilité qui se développe à la suite de l'apparition des perturbations ou d'une mauvaise prévision des plans de marche. Elle se traduit par le non respect des horaires prévus, l'apparition de trains de bus et de lacunes, la non réalisation d'une correspondance, etc. Elle participe à une détérioration sensible de l'efficacité de la production à travers la dégradation de la qualité de service offert à l'utilisateur (accroissement du temps de déplacement en ligne, de l'inconfort et du temps d'attente aux arrêts) et à travers les surcoûts qu'elle engendre pour l'exploitant. Les coûts impliqués sont extrêmement importants et font de la régulation un enjeu essentiel pour améliorer la performance et l'efficacité du transport en commun.

I.3.3.1.1 Processus de planification

L'affectation et la planification du trafic dans un réseau multimodal sont plus difficiles par rapport au cas monomodal à cause de la complexité des déplacements. L'objectif de ce processus est d'établir une offre prévisionnelle de transport qui s'ajusterait à la demande ; Il s'agit d'un processus d'ordonnancement des véhicules et du personnel pour leur attribuer respectivement des courses et des services. La complexité de ce problème de planification a fait l'objet de diverses études liées au domaine de l'ordonnancement et des tournées de véhicules.

Le processus classique de planification est illustré par la Figure I.4 [Huisman et al.,01]. La configuration initiale des lignes du réseau et leurs fréquences ainsi que les durées des trajets entre les différents points du réseau sont fixés comme paramètres d'entrée du processus, qui génère en premier lieu des tableaux d'horaires correspondant aux temps et lieux d'arrivée et de départ des véhicules, appelés *courses*. L'ensemble des courses à affecter à chaque véhicule représente une tâche d'ordonnancement statique pour générer des *blocs de véhicules*. Ensuite, l'ordonnancement du personnel fixe les fiches horaires journalières de service des conducteurs et des régulateurs, ainsi que les différentes relèves.

Ce problème est généralement traité par les compagnies de transport par les bus. L'outil le plus fréquemment utilisé est HASTUS, développé dans les années 80 [Rousseau et al., 85].

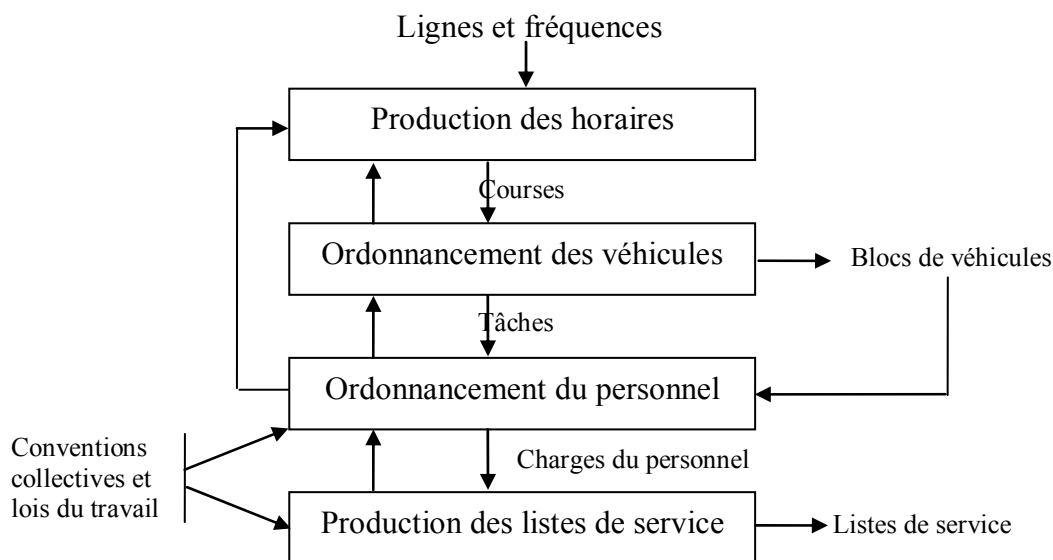


Figure I.4- Processus classique de planification

I.3.3.1.2 Processus de régulation

Le processus de planification est réalisé en temps anticipé (en général une fois par an) et ne s'appuie que sur des estimations de la demande et des durées des trajets entre les différents arrêts du réseau. La moindre variation de ces paramètres génère, en temps réel, des retards des véhicules, provoquant de longues attentes des passagers.

La gestion du trafic dans un réseau de transport ne s'arrête donc pas à l'étape de la planification. Cette dernière doit être suivie par un processus de régulation (Figure I.5), réalisé

en temps réel afin de gérer les écarts entre les horaires théoriques planifiés et les horaires réels de passage. Ce lien constitue un autre axe de recherche dans le domaine du transport.

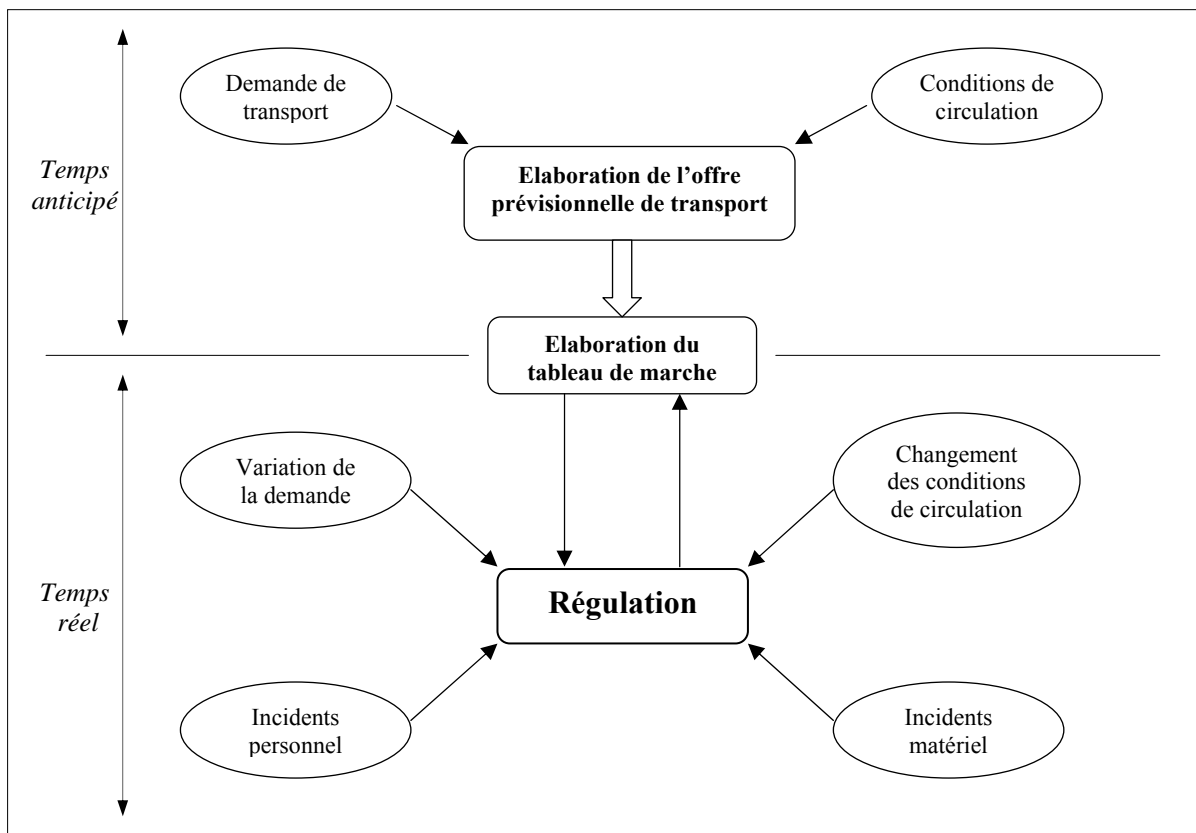


Figure I.5- Principe de la régulation

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

Dans ce contexte, *Ould Sidi* [OuldSidi, 06] a développé un système d'aide à la décision, à partir des informations fournies par le SAE (Système d'Aide à l'Exploitation) pour élaborer, évaluer et proposer des solutions efficaces aux régulateurs, en tenant compte de leurs préférences. Le travail du régulateur est donc mieux cerné et orienté, ce qui rend la régulation de meilleure qualité. [OuldSidi, 06] 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.

Fayech [Fayech et al., 02] 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 *Fayech* 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.

I.3.3.2 Le transport côté client

Les études menées dans ce domaine visent à fournir au client du transport 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 l'information dont il a besoin. Le but est alors d'épargner au voyageur la peine de rechercher une information de plus en plus difficile à trouver, à composer et à optimiser vu la croissance du volume des données sur des réseaux d'information de plus en plus étendus, ce qui engendre une concurrence commerciale de l'information.

Les travaux actuels dans le domaine du transport côté client visent à encadrer le client du transport de son point de départ jusqu'à son point de destination en lui proposant des services riches et variés 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 ce sens, des travaux ont été réalisés pour l'optimisation des itinéraires [Kamoun, 07] [Zidi, 06] sans viser la conception et l'optimisation d'un système d'information pour

l'aide à la mobilité urbaine dans tout le sens du terme, c'est-à-dire, l'aide au déplacement grâce à des informations génériques optimisées. C'est dans ce cadre que cette thèse apporte des solutions.

1.4 Les Systèmes d'Information de Transport Multimodal

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 vise à répondre à la chaîne de l'information qui se décompose en cinq objectifs : le recueil des données ; le traitement des données ; l'élaboration de l'information ; l'usage de l'information et la diffusion de l'information. Le dernier point inclut le problème de la représentation de l'information aux usagers indépendamment du support utilisé : écran graphique, écran Wap, téléphone, etc. On parle dans ce cas de *plasticité* de l'information.

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. Cette dernière fonctionnalité est très importante parce qu'elle vise à assister les clients de transport à consulter l'information dont ils ont besoin pour les aider à se déplacer. De ce fait, l'aide à l'information des voyageurs correspond à un module particulier d'un SIAE, appelé, Système d'Information d'Aide au Déplacement (*SIAD*).

1.4.1 Les systèmes existants

La plupart des systèmes existants sont soit monomodaux qui concernent un seul mode de transport, soit multimodaux mais concernent le même opérateur. Généralement, ce sont des sites Web qui constituent des compléments aux moyens conventionnels d'information (horaires, carte du réseau, etc.). Les améliorations attendues sont généralement les suivantes : fournir de l'information "temps réel" à l'utilisateur, proposer des solutions multimodales, rendre l'information accessible par différents moyens, et offrir des informations "personnalisées" à travers des systèmes interactifs.

I.4.1.1 Systèmes monomodaux

Ce sont des systèmes à un seul exploitant qui ne concernent qu'un seul mode de transport. Nous pouvons citer le système InfoBus⁴ 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 pouvons également parler des systèmes des deux opérateurs Still⁵ et Fibleu⁶ qui sont des systèmes pour les réseaux des bus correspondants aux deux villes Lens et Tours.

I.4.1.2 Systèmes multimodaux à agglomération limitée

Ces systèmes offrent une information portant sur plusieurs modes de transport et ne fournissant qu'une information statique avant le voyage. Généralement, un seul opérateur est à la source de cette information multimodale. Nous pouvons citer, par exemple, le système de Transpole⁷ de Lille, le système SIT de Madrid⁸, RATP⁹ de l'Ile-de-France, le système de Transvilles¹⁰ de Valenciennes ou encore LePilote¹¹ de Marseille. Chacun de ces systèmes n'assume que l'information sur une région en particulier.

I.4.1.3 Systèmes multimodaux à agglomération étendue

Afin d'étendre l'information multimodale sur un plus grand nombre de régions, il est nécessaire de juxtaposer les différents systèmes existants pour en faire un « super » système d'information multimodal multiopérateur, qui concerne plusieurs modes de transport et plusieurs opérateurs en même temps. Il s'agit donc de faire coopérer différents systèmes et d'intégrer ainsi des données issues de plusieurs sources différentes et distantes.

Quelques projets européens et internationaux ont été développés pour intégrer et fournir l'information multimodale, certains d'entre eux ont opté pour une architecture distribuée, où chaque opérateur est responsable de son système d'information, de la mise à jour et de la pertinence de ses données. Les autres, disposant de territoires moins étendus, ont opté pour une architecture centralisée.

⁴ www.mairie-metz.fr

⁵ www.still.fr

⁶ www.fibleu.fr

⁷ www.transpole.fr

⁸ www.ctm-madrid.es

⁹ www.ratp.fr

¹⁰ www.transvilles.com

¹¹ www.lepilote.fr

I.4.1.3.1 Architecture répartie

Le caractère multi-opérateur de l'information multimodale nécessite l'intégration de données issues de plusieurs sources réparties et distantes appartenant à différents fournisseurs d'information qui représentent les différents exploitants du transport. Le caractère distribué de l'information multimodale a incité l'idée d'une architecture répartie pour un SITM. Les avantages d'une telle architecture sont :

- La rationalité: les sources d'information distantes et réparties sont interrogées à distance seulement en cas de demande ;
- La confidentialité: les fournisseurs d'information ne sont pas obligés de fournir l'intégralité de leurs données. L'information nécessaire n'est donc accessible qu'en cas de demande contrôlée pour des raisons commerciales ;
- La responsabilité: les différents fournisseurs d'information détiennent la responsabilité de l'exactitude, la maintenance et la mise à jour de leurs données respectives ;
- La flexibilité: laisser le marché des systèmes d'information et de recherche d'itinéraire ouvert aux systèmes existants ainsi qu'à de nouveaux partenaires, fournisseurs de systèmes ou fournisseurs d'information.

Il s'agit d'architectures réparties via une intégration d'application. Les deux systèmes les plus connus sont DELFI¹²(Allemagne) et TransportDirect¹³(Angleterre) qui, mise à part leur multimodalité, ils couvrent l'ensemble de leur territoire national respectif.

DELFI (Durchgängige Elektronische FahrplanInformation): Information horaire électronique continue

Initié en 1996 par le Ministère Fédéral des Transports allemand, DELFI est un système d'information multimodale spécifique à l'organisation du transport en Allemagne, conçu pour réunir les différents systèmes existants afin de mettre en place un système d'information multimodale étendu à l'ensemble du territoire allemand. Avec un tel projet, l'Allemagne est

¹² www.delfi.de

¹³ www.transportdirect.info

considérée comme un précurseur dans la conception de systèmes d'information de porte à porte sur de longues distances.

Ce système permet la recherche d'itinéraire de porte à porte sur l'ensemble du territoire national en ne consultant qu'une seule source d'information sans que l'utilisateur ait à consulter plusieurs sources différentes. En plus, DELFI laisse le marché des systèmes d'information et de recherche d'itinéraire ouvert aux systèmes existants ainsi qu'aux nouveaux partenaires, fournisseurs de systèmes ou fournisseurs d'information. Tous les concepteurs des systèmes d'information et de recherche d'itinéraire déjà existants en Allemagne ont participé au développement de DELFI.

Ainsi, ce projet a engendré l'élaboration d'un système d'information distribué de transport avec intégration d'applications ; DELFI utilise CORBA (Common Object Request Broker Architecture) qui est une architecture commune de systèmes objet distribués proposée par l'OMG. CORBA permet l'interopération entre objets clients et objets serveurs distribués via des API (Application and Programming Interface) qui sont des interfaces standardisées pour la programmation d'applications.

L'architecture de DELFI est essentiellement constituée d'un module d'identification des origines-destinations, d'un noyau local qui permet de réaliser les calculs d'itinéraires locaux et d'un super mécanisme, sous forme d'un compositeur principal, qui permet les traitements répartis.

Une requête voyageur est traitée en temps réel en identifiant d'abord les serveurs responsables des recherches en début et fin de voyage. Ensuite, le compositeur principal identifie, vérifie et sollicite les différents points de transition à ces différents serveurs et enfin il compose l'itinéraire final le mieux adapté aux besoins de l'utilisateur. Mise à part les traitements en temps réels pour répondre aux requêtes des utilisateurs, le fonctionnement de DELFI nécessite des traitements en temps différé qui correspondent à la mise à jour des données communes nécessaires pour le fonctionnement du système. Ces données communes sont localisées dans une méta-base de données centrale qui est copiée sur les différents systèmes locaux à chaque mise à jour. La méta-base contient les tables de transformation, de traduction et de responsabilité. Les tables de transformation et de traduction jouent le rôle de dictionnaire de traduction des différents "langages" utilisés par les systèmes qui composent DELFI et les tables de responsabilité permettent aux sites qui composent DELFI d'identifier les serveurs responsables de la recherche.

Bien que DELFI représente un Système d'information distribué de Transport multimodal à l'échelon national, il reste limité sur de nombreux points:

- L'architecture de DELFI n'est pas complètement distribuée à cause de l'aspect centralisé de la base commune ;
- DELFI ne couvre que l'information multimodale "avant le voyage". L'information qui est nécessaire "pendant" la réalisation du déplacement est absente du projet ;
- DELFI ne permet pas d'offrir certains services essentiels comme : le prix d'un déplacement sur un itinéraire complet, la comparaison entre le transport individuel (VP) et le transport public (TC), des informations pertinentes et exhaustives sur les perturbations (grèves, accidents, etc), etc.

DELFY fait intervenir plusieurs fournisseurs de systèmes d'information et fournisseurs de services. De ce fait, le choix d'une architecture centralisée aurait causé :

- Des problèmes de fiabilité de la base dont les mises à jour des données dépendent d'un grand nombre d'intervenants,
- Des difficultés d'ouverture aux nouveaux partenaires, services ou autres systèmes.

Ces raisons justifient le choix d'une architecture répartie pour conserver l'autonomie de chaque fournisseur et faciliter l'intégration de nouveaux partenaires.

EU-Spirit¹⁴ : l'exemple européen

C'est une extension de DELFI à l'échelle européenne. Ce système concerne à part l'Allemagne, le Danemark et la Suède.

PREDIM¹⁵ : l'exemple français

Créé par le Programme de Recherche d'Expérimentation et de l'Innovation dans les Transports Terrestres¹⁶ (PREDIT), le PREDIM est une Plate-forme de Recherche et d'Expérimentation pour le Développement de l'Information Multimodale qui regroupe

¹⁴ www.eu-spirit.com

¹⁵ www.predim.org

¹⁶ www.predit.prd.fr

plusieurs projets régionaux pour générer l'information multimodale par l'intégration de l'information issue de différentes sources.

TransportDirect : l'exemple anglais

Initié en 2000 par le ministère de transport public anglais, le projet TransportDirect vise à réaliser un système d'information national qui gère l'information relative à l'intégralité du transport en commun. Les systèmes d'information des différents exploitants du transport communiquent via le protocole JourneyWeb¹⁷, mis en place dans le cadre de ce projet. Grâce à ce protocole, les différents opérateurs communiquent pour offrir le meilleur service multimodal multiopérateur à leurs clients, en échangeant des données au format XML, conformément au standard local TransXchange¹⁸.

Pour son fonctionnement, TransportDirect utilise deux bases de données nationales :

-La « National Public Transport Gazetteer » (NPTG) : qui catalogue les différents pôles d'échange, régions, villes et villages au niveau national ;

-La « National Public Transport Access Node » (NaPTAN) : qui catalogue, pour chaque opérateur local et pour les différentes régions, l'ensemble des nœuds, correspondants aux différents arrêts et stations.

Dans le cadre de ce projet, chaque calculateur local détient une vue globale sur l'ensemble de l'information correspondante au réseau national. Chaque calculateur est donc capable de rechercher un itinéraire d'un point de départ, se trouvant dans sa zone de couverture, vers un pôle d'échange national répertorié et identifié par la NPTG. Ce pôle d'échange permet donc l'accès au réseau de l'opérateur distant, correspondant à la destination en question.

Un utilisateur voulant se procurer une information multimodale, consulte le SI concerné qui, à son tour, consulte les deux bases de données du système afin de déterminer le calculateur local distant correspondant à la destination ainsi que les pôles d'échange qui permettent d'y accéder. Ensuite, le calculateur local de l'origine compose l'itinéraire global en concaténant son offre, à son homologue distant.

¹⁷ www.journeyweb.org.uk

¹⁸ www.transxchange.dft.gov.uk

I.4.1.3.2 Architecture centralisée

Le choix d'une architecture centralisée ne peut être intéressant que dans le cas de territoires peu étendus qui optent pour une intégration de données, comme c'est le cas de la Suisse et des Pays-Bas avec leur système respectif GOFAS et 9292.

GOFAS : l'exemple Suisse,

C'est un système composé d'un calculateur d'itinéraire appelé Infopool, utilisant une base de donnée centrale, appelée Internet-GIS qui importe les horaires et les données géographiques à partir des sources de données des différents opérateurs pour en faire une base de données centrale couvrant le territoire national.

9292¹⁹ : l'exemple des Pays-Bas,

C'est un système qui fournit une information multimodale de déplacement et de perturbation. L'importation des données de ce système centralisé des différentes sources (correspondant aux différents opérateurs) se fait par le biais de convertisseurs spécifiques.

I.4.2 Intégration de l'information

Le problème de l'intégration des données et des applications s'est toujours posé au cœur des entreprises qui possèdent souvent des systèmes hétérogènes composés de programmes développés à l'interne ou procurés de l'extérieur. Cependant, l'intégration des données et des programmes dans les systèmes d'information sur les réseaux à grande échelle représente un enjeu économique important dans les entreprises d'aujourd'hui. L'intégration dans les systèmes d'information est caractérisée par sa complexité à cause de l'augmentation croissante des sources de données et des applications hétérogènes et distribuées. Le problème d'intégration représente donc un domaine très dynamique de recherche en informatique spécifié par différentes approches qui visent à lui fournir une réponse conviviale et efficace. Les solutions à l'intégration se classifient selon l'investissement technologique des entreprises et l'intégration de l'information devient aujourd'hui une intégration de systèmes d'information qui doit passer par une intégration d'application, appelée ***Intégration des Applications d'Entreprise*** (EAI).

¹⁹ www.9292ov.nl

I.4.2.1 Les solutions d'intégration

Il existe trois types d'intégration : l'intégration au niveau utilisateur, l'intégration de données et celle des applications. Le premier type d'intégration se fait au niveau de l'interface utilisateur via les portails web, permettant l'identification « logging » des utilisateurs, simultanément, à différentes applications. L'intégration de données permet d'intégrer et réunir les bases de données d'une manière distribuée ou d'une manière centralisée pour permettre l'importation des données des autres bases, en adaptant les différents modèles de données correspondants. Enfin, l'intégration des applications se base sur une infrastructure utilisant la technique d'envoi et de réception des messages de données intra applications, avec des fonctions de transformation et de routage de données. La gamme concernée de logiciels-médiateurs²⁰ (« middleware ») supporte un ensemble d'API (Application Programmatic Interface) permettant l'accès aux différentes applications à intégrer, via des bibliothèques prédéfinies.

I.4.2.2 Les standards d'intégration

XML (eXtensible Markup Language) : le langage de balisage extensible

Créé en 1996, XML est un meta-langage extensible permettant de décrire les structures et les contenus des données en différenciant, contrairement à HTML, les balises de contenu et de mise en forme, ce qui facilite la transformation des données.

XML permet aujourd'hui d'assurer le transport et l'échange des flux de données entre différentes applications d'un système d'information à travers une description antérieure et formalisée du contenu. Ainsi, grâce à son caractère auto-descriptif, il permet la portabilité des données et leur partage entre applications très diverses, sachant que le contenu décrit également la structure des données qu'il transporte.

Depuis 1998, XML a été normalisé par l'association mondiale de l'internet (World Wide Web Consortium), ce qui le transforme en un outil standard pour les nombreux projets d'intégration d'application.

Les services web (« web services »)

Un service Web est un ensemble de protocoles et de normes utilisables sur le Web par des applications ou par d'autres services pour échanger des données entre les applications.

²⁰ Permet la communication entre des clients et des serveurs ayant des structures et une implémentation différentes et l'échange d'informations dans tous les cas et pour toutes les architectures.

C'est le successeur des technologies précédentes (CORBA²¹ de l'OMG, COM²² de Microsoft et EJB²³ de Sun) qui permettent l'invocation d'une méthode ou d'un traitement à distance, à travers le réseau, dans une architecture distribuée orientée objet. Ces technologies, contrairement aux services Web, ne permettent pas de construire une architecture répartie et hétérogène, composée d'objets de technologies différentes (objets CORBA, COM et EJB). En revanche, les services web permettent l'intégration des composants applicatifs hétérogènes en standardisant les interfaces (API) et les protocoles de communication via Internet.

Lorsqu'un client demande un service (service web) à un représentant, ce dernier lui communique le site où est implanté le service, par le biais d'annuaire. Un service web est construit sur la base de trois standards technologiques: Le **WSDL** (Web Service Description Language) qui est une application XML de description et d'utilisation de services; le **UDDI** (Universal Description Discovery and Integration) qui est un standard de recherche de services et **SOAP** (Simple Object Access protocol) qui est un protocole utilisé pour échanger ces informations.

1.4.2.3 L'intégration de l'information dans le domaine du transport

Les données nécessaires pour la génération d'une information multimodale peuvent être statiques ou dynamiques. Les données statiques permettent de définir le réseau de transport d'un opérateur comme la description des données et leurs géolocalisations ainsi que les offres proposées des services de transport. Les données dynamiques représentent les données temps réel (les modifications des durées de trajets théoriques, les perturbations, etc.).

Dans le but de produire l'information multimodale issue de toutes ces données, la majorité des opérateurs de transport disposent d'un Système d'Information d'Aide au déplacement (SIAD) qui interfère avec les autres SIADs des différents opérateurs concernés, pour procéder à une intégration d'information. La stratégie d'intégration adoptée peut être une intégration de données, localisées dans les différentes bases de données des différents opérateurs afin de constituer une base de données globale et centralisée. La deuxième stratégie d'intégration est une intégration d'applications par la production et la composition de l'information multimodale à partir des résultats de plusieurs SIADs existant, interrogés automatiquement.

²¹ www.corba.org

²² www.microsoft.com/com

²³ <http://java.sun.com/products/ejb>

Dans le but d'échanger et d'intégrer les informations issues de différents systèmes distants, de nombreux travaux s'intéressent actuellement à la standardisation des données à échanger et leur modèle conceptuel ainsi que le format d'échange à adopter. Quelques normes existent telles que :

TRANSMODEL²⁴

C'est une norme définissant un modèle conceptuel de données, initiée en France en 2001 et transformée ensuite en une norme européenne. TRANSMODEL permet de décrire le transport public gérant toutes les données relatives aux activités des opérateurs du transport public, incluant les activités de planification, de régulation et d'exploitation. La norme est définie par un ensemble de diagrammes UML (Unified Modeling Language) et Merise détaillés, elle est actuellement utilisée par la France, la Belgique, l'Italie, le Royaume Uni, les pays Bas et le Danemark.

TRIDENT²⁵ (Transport Intermodality Data Sharing and Exchange Networks)

TRIDENT décrit un ensemble de schémas XML qui permettent de normaliser l'échange des données entre exploitants du VP (Voiture Privée) et Tc (transport en Commun) en modes normal et perturbé. Ces caractéristiques font de TRIDENT la meilleure base pour l'élaboration d'une normalisation.

SIRI²⁶ (Service Interface for Real Time Information)

SIRI a été initié en 2004 et standardisé par le Comité Européen de normalisation, à l'issue d'un projet européen intégrant la France, l'Allemagne et le Royaume Uni. C'est un protocole d'échange de données de transport en temps réel entre différents serveurs correspondants à des opérateurs de transport, en se basant sur TRANSMODEL comme modèle conceptuel de données.

CHOUETTE²⁷ (Création d'Horaires avec un OUtil d'Echange de données TC selon le format TRIDENT Européen)

²⁴ www.transmodel.org

²⁵ www.ertico.com/en/activities/activities

²⁶ www.kizoom.com/standards/siri

²⁷ <http://chouette.adullact.net>

L'outil se limite aux échanges d'information concernant la description statique de l'offre TC (réseau, horaires), et ne couvre donc pas les messages de perturbation en temps réel. Les objectifs de CHOUETTE consistent à permettre :

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

1.5 Les pôles de compétitivités

Un pôle de compétitivité vise à concentrer autour de quelques entreprises, laboratoires (publics et privés) de recherche et organismes de formation, un ou quelques projets de développement économique durable, tournés vers l'international et s'appuyant sur le triptyque industrie/recherche/formation. Il doit s'inscrire dans un territoire précis qu'il a vocation à conforter. Ce qui est visé par l'appel d'offre lancé par le Gouvernement fin 2004 est une démarche de mobilisation et de mise en synergie de ressources industrielles, scientifiques et de formation, sur des thématiques ciblées, dans des territoires précis et sur des objectifs de développement partagés et à valeur ajoutée économique et sociale.

1.5.1 Le projet « I-Trans » de pôle de compétitivité du Nord Pas-de-Calais

Le titre du projet qui intéresse le secteur transport est « Le ferroviaire au cœur des systèmes de transport innovants », fortement inspiré d'une réflexion menée antérieurement aux pôles de compétitivité au sein de l'association TTP (Transports Terrestres Promotion). Ce projet, baptisé « I-trans » est à rapprocher par ailleurs d'un autre travail lancé dans la même période par la Direction Recherche de la Région sur la valorisation de la recherche transport. Y. Ravalard est l'initiateur du projet. Différents groupes de travail ont été constitués :

- GT1, groupe de pilotage (cohérence, stratégie) ;

- GT2, inventaire de la recherche mobilisable ;
- GT3, groupe thématique sur l'interopérabilité ferroviaire ;
- GT4, groupe thématique sur l'interopérabilité routière ;
- GT5, groupe thématique sur l'intermodalité de fret ;
- GT6, groupe thématique sur l'intermodalité voyageur ;
- GT7, inventaire des formations.

I.5.2 Le projet VIATIC du I-TRANS

Notre équipe de recherche participe au groupe de travail GT6 dédié à l'intermodalité voyageurs qui repose essentiellement sur des travaux partenariaux menés, depuis de nombreuses années, dans le cadre de la recherche nationale PREDIT et de la recherche régionale GRRT. L'acronyme de notre projet VIATIC, piloté par G. Uster de l'INRTS/ESTAS, repose en effet sur le concept d'accompagnement du citoyen souhaitant se déplacer sans contrainte, libre dans sa mobilité et dans le choix de ses modes de transports. Le terme anglo-saxon de « seamless mobility » résume très bien cette idée : une mobilité sans couture, sans sensation de rupture de charge, sans souci de billet, sans piège tarifaire, lisse et naturelle, etc. Cependant, nous ajoutons une dimension supplémentaire dans l'accompagnement. Nous considérons que le citoyen doit retrouver dans sa mobilité quotidienne un même niveau de qualité dans l'accès à ses services de base fixe chez lui ainsi que sur son lieu de travail et d'étude. Dans les systèmes de transports collectifs, le temps passé est assez souvent vécu négativement et mériterait d'être mieux valorisé. Dans les pôles d'échanges, lieux de transit, il conviendrait d'adoucir les ruptures de charge en proposant des services. Le concept de continuité servicielle intégré dans un système d'aide à la mobilité est l'objectif de nos travaux de recherche. Nous allons donc développer et évaluer dans les chapitres suivants un système d'aide à la mobilité qui tend vers un objectif de mobilité durable.

I.6 Conclusion

La mobilité urbaine des personnes au quotidien peut être parfaitement bien encadrée par des informations de transport multimodal. Dans ce contexte, les systèmes d'information d'aide au déplacement urbains sont d'une grande utilité s'ils offrent toute information utile au

transport. Cependant, la plupart des systèmes d'information actuels sont seulement dotés d'algorithmes de calcul d'itinéraires pour composer la meilleure offre de transport selon les préférences de l'utilisateur. C'est déjà un point d'avancement important mais la mobilité urbaine des personnes dépasse la simple idée de déplacement ! Nous proposons donc dans cette thèse un système d'information d'aide à la mobilité urbaine qui permet d'offrir n'importe quel service lié à la mobilité. Le système proposé se base sur une approche multi-agent pour la recherche et la composition des services liés aux transports, que nous concevons et optimisons. Dans le chapitre suivant, nous présentons un état de l'art sur les systèmes multi-agents et les méthodes d'optimisation dans le domaine du Transport.

**CHAPITRE II. SYSTEMES MULTI-AGENTS ET
OPTIMISATION DANS LE DOMAINE DU TRANSPORT : VERS
UNE ARCHITECTURE DISTRIBUEE A BASE D'AGENTS
OPTIMISATEURS**

II.1 Introduction

Dans le chapitre précédent, nous avons évoqué un état de l'art sur les systèmes d'information de transport multimodal existants. Néanmoins, un système d'information pour le transport multimodal 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-agent de modélisation ainsi que les techniques d'optimisation existantes dans le domaine du transport.

Le reste de ce chapitre est organisé comme suit :

- Dans le deuxième paragraphe, nous expliquons la raison pour laquelle nous faisons le lien entre les systèmes multi-agents et les méthodes d'optimisation, en évoquant ces deux domaines complètement différents dans le même chapitre ;
- Dans le troisième paragraphe, nous procédons par un état de l'art sur les systèmes multi-agent : définitions, caractéristiques, typologies, classifications en démontrant les avantages de ce paradigme pour les systèmes d'information ;
- Dans le quatrième paragraphe, nous énumérons quelques cas d'utilisation des systèmes multi-agents dans le domaine du transport ;
- Le cinquième paragraphe est dédié aux techniques d'optimisation existantes et notamment les algorithmes évolutionnaires ;
- Le sixième paragraphe énumère quelques cas d'utilisation des algorithmes évolutionnaires dans le domaine du transport.

II.2 Les Systèmes Multi-Agents et les méthodes d'optimisation : pourquoi une telle alliance ?

Une modélisation multi-agent d'un problème permet de le répartir en des compétences et des connaissances vers des entités autonomes appelées agents, de fournir un outil de recombinaison, de liens et de dialogue entre ces entités par des moyens de communication et d'interaction et enfin de préciser les règles de recombinaison par l'élaboration des organisations.

Dans un système multi-agent, le changement d'état de certains objets de l'environnement d'évolution des agents peut affecter leur comportement et prises de décision. Ainsi, pour optimiser leurs choix et pour orienter leurs prises de décision, ces agents peuvent être dotés d'approches d'optimisation adaptées à leurs compétences et connaissances. Les objets sont généralement caractérisés par de nombreux critères d'évaluation, qui rend la prise

de décision combinatoire et multi-objective. D'où l'alliance entre les systèmes multi-agents et les méthodes d'optimisation ; ces deux approches sont donc différentes mais parfaitement complémentaires.

II.3 Les Systèmes Multi-Agents (SMA)

La programmation classique est fermée, elle répond de moins en moins aux besoins des nouveaux systèmes d'information distribués, hybrides et hétérogènes, à caractère dynamique et à sémantique riche, variée et évolutive. Ce besoin d'ouverture a incité les chercheurs à se retourner vers le domaine de l'Intelligence Artificielle (IA) qui 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. C'est le domaine de l'Intelligence Artificielle Distribuée (IAD), sous domaine de l'IA. Dans ce qui suit, nous présentons la notion d'agent et ses caractéristiques avant d'entamer les SMA.

II.3.1 Agent : Définitions et particularités

II.3.1.1 Définitions

Plusieurs travaux ont porté sur la notion d'agent, ce qui a engendré des définitions aussi riches que variées. Pour résumer toutes ces explications évoquées dans la littérature, nous retenons celle de *Jennings* [Jennings et al, 96] qui définit un agent en tant qu'entité *autonome* et celle de *Florez* [Florez, 99] qui définit un agent en tant qu'entité *interactive*.

- Un agent est un programme *autonome*, capable de contrôler ses faits et gestes, il se base sur la perception qu'il a de son environnement et il est à la poursuite d'un ou de plusieurs objectifs.

La définition retenue d'un agent en tant qu'entité interactive, établie par *Florez* [Florez, 99] est assez exhaustive et explicite, elle se base sur des définitions préalablement établies par, entre autres, *Woodridge* [Woodridge et al, 95] :

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

Une définition plus élaborée de la notion d'agent, qui regroupe les deux définitions précédentes, a été constituée par *Ferber* dans son ouvrage [Ferber, 95] là où il définit un agent comme suit :

- Un agent est une entité physique ou virtuelle :
 1. qui est capable d'agir dans un environnement,
 2. qui peut communiquer directement avec d'autres agents,
 3. qui est mue par un ensemble de tendances (sous la forme d'objectifs individuels ou d'une fonction de satisfaction, voire de survie, qu'elle cherche à optimiser),
 4. qui possède des ressources propres,
 5. qui est capable de percevoir (mais de manière limitée) son environnement,
 6. qui ne dispose que d'une représentation partielle de cet environnement (et éventuellement aucune),
 7. qui possède des compétences et des offres de services,
 8. qui peut éventuellement se reproduire,
 9. 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.

II.3.1.2 Caractéristiques

Un agent est particulièrement caractérisé par 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. Un agent autonome doit également avoir un certain degré de contrôle non trivial sur ses actions et sur ses états internes.
- la *réactivité* : c'est la capacité de l'agent de percevoir son environnement et de réagir automatiquement en fonction des changements discernés.

- la **proactivité** : Un agent proactif ne se contente pas de répondre aux événements. Il possède, en plus de ses attributs et méthodes, 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, avec son plein gré, d'interagir avec d'autres agents pour atteindre leurs buts respectifs.

D'autres caractéristiques peuvent être mentionnées dans la littérature telles que la **continuité** qui correspond à la capacité d'un agent d'être continuellement actif ou à sa capacité à maintenir une identité et un état sur une longue période [Florez, 99], l'**adaptabilité** qui correspond à la capacité d'apprendre et de s'accorder aux changements de l'environnement en s'améliorant avec l'expérience, la **collaboration** qui correspond à la capacité de travailler avec d'autres agents pour la réalisation d'un objectif commun etc.

II.3.1.3 Typologie des agents

Les agents peuvent être classés selon différentes propriétés, par exemple : la granularité (degré d'intelligence), le rôle, la mobilité, la capacité de coopérer, etc.

II.3.1.3.1 Classification des agents selon la granularité

L'intelligence d'un agent se base sur sa capacité de raisonner, d'apprendre, de comprendre et de planifier. Cette propriété est également liée à son autonomie et sa flexibilité dans un environnement dynamique. En général, trois degrés d'intelligence sont distingués, nous les classons ici du moins important au plus important:

- les agents **réactifs** : ce sont des agents passifs qui réagissent seulement à un stimulus. Ce type d'agent ne dispose pas de module de raisonnement interne ;
- les agents **pro-actifs** : ce sont des agents dynamiques qui entreprennent car ils possèdent, en plus de leurs attributs et méthodes, des processus internes qui leur permettent de prendre des initiatives pour réaliser leurs buts. Un agent pro-actif est donc un agent dirigé buts ;
- les agents **cognitifs** : un agent cognitif 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.

Le courant cognitif prédispose les agents à être dotés d'un outil d'abstraction pour les décrire comme des systèmes intentionnels. Plusieurs modèles existent, les plus connus sont : le modèle BDI (Believe, Desire, Intention) et le modèle BUC (Believe, Uncertainty, Choice).

II.3.1.3.2 Classification des agents selon la mobilité

Un agent peut être stationnaire ou mobile :

- 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 ;
- Contrairement à un agent stationnaire, 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. La notion de la mobilité est évoquée plus en détail dans II.3.1.5.

II.3.1.3.3 Classification des agents selon la fonction ou le rôle

Une telle classification se base sur le rôle joué par l'agent dans le système. Les agents les plus connus sont :

- Les agents d'*information* (agents Internet): la fonction principale de ces agents est de gérer un grand volume d'information à travers le web via les moteurs de recherche. Ils peuvent être stationnaires ou mobiles ; stationnaires comme les agents « gestionnaires de courrier » ou « secrétaires virtuelles » ou mobiles pour pouvoir naviguer sur la toile pour chercher les informations et les amener vers leur destination ;
- Les agents *bases de données* : ces agents jouent un rôle important dans la répartition des données sur plusieurs serveurs. Ils sont souvent utilisés dans l'exploration des données²⁸ (« datamining »), la collecte des données, la recherche de l'information, le traitement parallèle des requêtes, etc.

²⁸ Ensemble de méthodes et de techniques permettant d'extraire des informations à partir d'une grande masse de données.

- Les agents de **détection d'intrusions** : ce sont des agents conçus pour la sécurité informatique afin de repérer les tentatives de nuisance au matériel ou au logiciel via des Systèmes de Détection d'Intrusions (IDS). La détection d'intrusions consiste à scruter le trafic réseau, collecter tous les événements, les analyser et générer des alarmes en cas d'identification de tentatives malveillantes.
- Les agents de **commerce** : ces agents facilitent les opérations commerciales effectuées en ligne telles que les visites des galeries ou magasins virtuels et la recherche des offres commerciales. Deux types d'agents de commerce sont distingués: les agents contrôlés par les clients et ceux contrôlés par les producteurs. Le premier type d'agents est également appelé « shopping agents » et agit pour le client qui spécifie les caractéristiques de l'offre désirée, l'agent doit alors prospecter sur Internet pour trouver la meilleure offre. Le deuxième type d'agents de commerce agit au profit des producteurs, ils sont donc orientés vers la prospection de marchés pour connaître les goûts et les besoins des consommateurs.

II.3.1.4 Agent vs Objet

La programmation orientée agent peut être considérée comme une couche au dessus de la programmation orientée objet. Contrairement à un objet, un agent est autonome, persistant et évolue dans le temps dans un environnement dynamique. L'objet, quand à lui, est rigide et passif, il ne peut pas choisir de répondre ou pas à des appels à ses méthodes ou à ses compétences.

II.3.1.5 L'agent dans les applications réparties : le caractère mobile

II.3.1.5.1 Schémas d'organisation distribuée

Les systèmes distribués ont fait leurs apparitions dans les années 70 pour permettre un échange d'information distant. De nombreux schémas d'organisation ont émergé pour concevoir ces systèmes :

- Le schéma d'organisation à mémoire distribuée partagée : ce schéma est particulièrement adapté aux applications où le partage est important et consiste en des processus distribués qui communiquent à travers une mémoire de grande taille ;

- Le schéma d'organisation à événement : permet une communication entre plusieurs processus via un gestionnaire d'évènements qui fait correspondre chaque événement à ses demandeurs ;
- Le schéma d'organisation pair à pair (« peer-to-peer ») : son principe se base sur une communication unidirectionnelle entre deux processus voisins. Dans ce schéma d'organisation, tous les acteurs d'une application interagissent par paire et sont tous équivalents. La relation entre client et serveur est donc symétrique, ils sont ainsi désignés par le même terme *servent* (*serveur et client*). L'avantage de ce type de modèle est la simplicité mais les inconvénients majeurs sont la lourde charge du serveur qui dépend intégralement de l'entité cliente.
- Le schéma d'organisation client/serveur : ce schéma reste le plus largement répandu. Il permet de faire cohabiter dans un même système, des machines aux performances (système et matériel) très différentes [Rouvrais, 02]. Ce schéma a d'abord été mis en œuvre par le mécanisme **d'appel de procédure à distance**, puis par le mécanisme **d'invocation de méthode** avec l'apparition de l'orienté objet. Enfin, par le mécanisme **d'invocation de service** web pour s'adapter aux environnements à grande échelle.

II.3.1.5.2 Apparition de la technologie mobile

L'apparition des réseaux sans fil a engendré de nombreux problèmes dans les systèmes et les applications répartis qui n'étaient pas conçus pour le concept de l'informatique mobile. Ces problèmes sont dus à trois facteurs [Rouvrais, 02] :

- La mobilité des machines nécessite des mécanismes étendus de nommage ou de routage ;
- Les réseaux sans fil ont généralement une bande passante très réduite et très variable, il est donc indispensable de gérer la disponibilité des données malgré les **déconnexions**. Par conséquent, l'**asynchronisme** des interactions entre les deux sites devient nécessaire ;
- Les ressources des machines mobiles telles que l'espace de stockage ou la durée de vie des batteries sont limitées.

La technologie mobile offre de nombreux atouts par rapport aux mécanismes d'interaction classique. Sa première motivation est la minimisation des communications

distantes car il est en général moins coûteux de migrer le code de traitement que de déplacer les données à traiter, qui peuvent être beaucoup plus volumineuses. L'objectif de base du code mobile est donc de naviguer en fonction de l'application, d'étendre des fonctionnalités ou de supporter des opérations déconnectées. Quand le code mobile active une opération distante, il déconnecte le client et le reconnecte ultérieurement pour récupérer le résultat [Bernard, 99].

II.3.1.5.3 Le paradigme Agent Mobile

Un Agent Mobile (AM) est composé de code, de données et d'un état d'exécution. Il dispose de la capacité de migrer d'un nœud à un autre dans un réseau pour s'y exécuter et donc accomplir un certain nombre de tâches en utilisant les ressources des nœuds visités (Figure II.1). Les schémas à base de code mobile peuvent être distingués en considérant la localisation des différentes entités du système à savoir [Rouvrais, 02] :

1. Les **ressources** : ce sont les **données** passives ou les **unités physiques** tel que le processeur ;
2. Le **code** du service demandé par le client : ce sont les **opérations à effectuer**, c'est le programme (c'est-à-dire l'algorithme) réalisant le service ;
3. L'**unité d'exécution** : c'est la machine qui encapsule l'**état** d'exécution en cours tels qu'une pile, un tas ou un compteur ordinal.

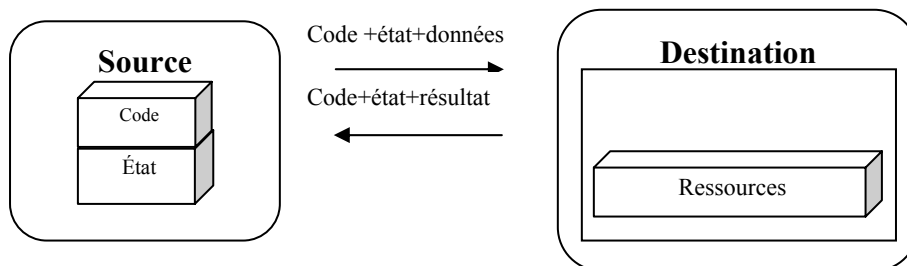


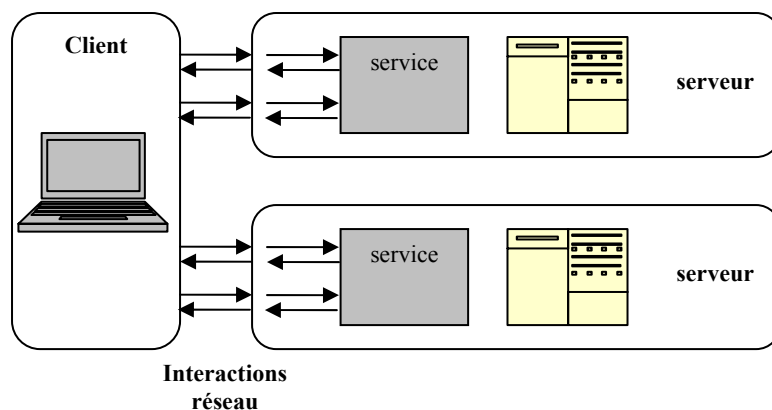
Figure II.1 - Mécanisme de déplacement d'un agent mobile

Un AM est créé par un nœud initiateur, appelé **nœud hôte**, puis envoyé sur une séquence de nœuds accepteurs, capables d'héberger des entités mobiles pour exécuter un certain nombre de tâches puis, retourne à son nœud initiateur pour lui procurer le résultat de son travail. La séquence ordonnée des nœuds accepteurs visités par un AM constitue son chemin, appelé aussi itinéraire (« Workplan »). Un nœud accepteur est la machine qui accueille l'agent mobile et l'autorise à communiquer avec elle et avec les autres agents qu'elle héberge. Cette machine doit être capable de négocier l'échange des agents, transmettre les

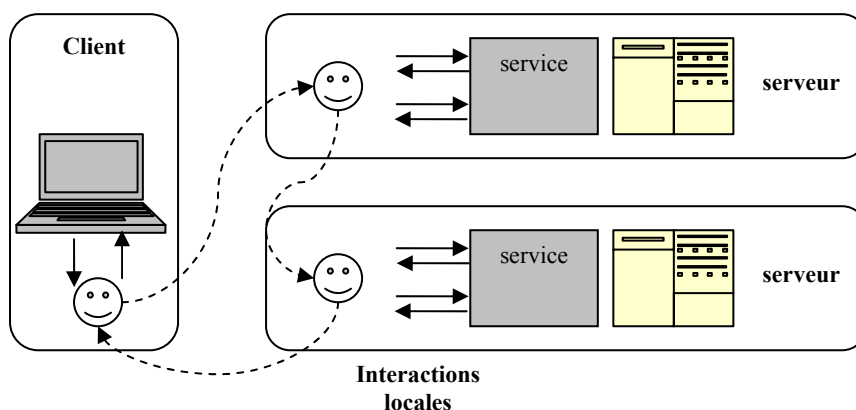
agents en cours d'exécution à d'autres hôtes, et recevoir un agent en reprenant le traitement qu'il effectuait. Dans une même machine, les agents interagissent entre eux à travers une interface communicante.

Lorsqu'un agent mobile migre sur un site distant, il devient une entité autonome, qui peut, par exemple, suivant l'exécution du code, se déplacer sur un autre site, accumuler des résultats, communiquer avec d'autres agents et signaler un événement, sans qu'une connexion permanente soit maintenue avec le site initial. L'objectif est de remplacer des interactions distantes entre agents par des interactions locales afin de diminuer le volume des échanges sur le réseau (Figure II.2). L'utilisation du paradigme AM permet donc d'améliorer la flexibilité des applications. En contrepartie, les agents eux-mêmes doivent être flexibles et s'adapter aux conditions d'exécution. Deux types de mobilité existent suivant la disposition de l'agent mobile :

- La mobilité est à son plus haut degré si elle est **forte**. Elle est appelée ainsi quand l'agent migre en totalité (code+état+données), c'est la capacité de capture et de restauration de l'état d'exécution d'un agent qui est d'abord capturé puis **transféré** (*migration par déplacement*) dans le site distant. Quand l'agent est reçu, son état est automatiquement rétabli. Ce schéma (la capture, le transfert et la restauration) ne demande aucune interaction avec l'utilisateur. Cette transparence est fort intéressante mais pas toujours facile à mettre en œuvre car la capture de l'état d'un agent nécessite des fonctions spécifiques qui ne sont pas fournies par la plupart des langages. Enfin, la migration forte est de plus en plus coûteuse à réaliser d'autant que l'état de l'agent mobile est important. Une autre variante de la mobilité forte revient à dupliquer l'agent (après sa capture) sur le site distant sans le déplacer (*migration par répllication*) ;
- La mobilité est dite **faible** si l'agent migre sans son unité d'exécution (l'état du processus), c'est la capacité pour un système de déplacer le code de l'agent accompagné seulement de données d'initialisation et non de l'état complet. Dans ce cas les informations sur l'état, qui permettent la poursuite logique de l'exécution, sont préservées dans **les données**. C'est au programmeur de gérer lui-même le mécanisme de reprise en ré exécutant l'agent dans le site distant.



a - modèle Client/Serveur



b - modèle AM

Figure II.2– Comparaison des modèles Client/Serveur et AM

En conclusion, le paradigme AM est innovant et concerne principalement les applications largement distribuées afin de remplacer efficacement les paradigmes classiques en améliorant l'exécution des appels des procédures à distance (RMC : Remote Procedure Call). Cette amélioration s'explique par la réduction de l'usage de la bande passante des réseaux en favorisant l'accès local aux données et aux services [Mandiau et al., 02b]. La Figure II.2 compare le paradigme AM au paradigme client/serveur.

Les AMs peuvent être dotés d'un certain degré d'intelligence en coopérant ou même en négociant [Zgaya et al., 07a] avec d'autres entités du système global. Dans ce cas, ils sont définis comme des AMs intelligents. Enfin, un AM peut être complètement autonome travaillant à son compte ou il peut dépendre d'un autre agent.

Nous nous sommes basés sur ce paradigme innovateur pour le fonctionnement du SITM car il s'adapte parfaitement aux contraintes du système en convolant une architecture robuste, ouverte et distribuée. Nous avons donc justifié l'utilisation de ce paradigme au sein

du SITM (§V.4), sur la base des deux niveaux d'optimisation [Zgaya et al., 07c] et le modèle d'archivage dynamique des données.

II.3.2 Les Systèmes Multi-Agents

Un Système Multi-Agent (SMA) est un ensemble d'entités (agents logiciels ou humains) qui interagissent dans un environnement pour résoudre des problèmes qui dépassent les capacités ou les connaissances individuelles de chaque agent. L'interaction peut aussi bien être une coopération qu'une compétition et l'environnement représente un espace commun d'interaction pour tous les agents. Cet environnement dépend de la nature de l'application considérée, il peut être par exemple le monde physique, les données d'un problème, une collection d'agents etc. Nous présentons dans ce qui suit une définition plus élaborée d'un SMA, récapitulant celle donnée par *Ferber* [Ferber, 95] :

Un SMA 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). L'application des actions des agents sur le monde et la réaction de ce monde à ces actions sont représentées par des opérateurs qui sont un ensemble de lois de l'environnement, appelées *lois universelles*. Ces lois peuvent être perçues, manipulées, transformées, consommées et produites par les agents du système par un ensemble d'opérations ou de compétences.

II.3.2.1 L'environnement dans un SMA

II.3.2.1.1 Présentation

Dans un SMA, l'environnement symbolise le monde dans lequel les agents évoluent. D'une manière générale, l'environnement est un champ d'interactions : signaux, traces, qui peut être doté d'une loi physique. Un environnement peut représenter :

- Un lieu où des actions individuelles ou collectives sont réalisées et où des réactions sont perçues ;
- Un espace de déplacement : grille, position des agents, etc. ;
- Un moyen de structuration des agents : relations de proximité, définition des topologies spatiales ou temporelles, etc. ;
- Une source de données pour le système ;
- Un lieu où des ressources sont disponibles.

II.3.2.1.2 Capacités d'un agent dans un environnement

Dans un environnement, un agent dispose des deux capacités de *perception* et d'*action*:

- la capacité de *perception* qui représente la capacité à reconnaître les objets (position, relation entre objets) ;
- la capacité d'*action* qui permet de transformer l'état du système en modifiant les positions et les relations qui existent entre les objets.

II.3.2.1.3 Caractéristiques

Un environnement peut être *statique* ou *dynamique*, *déterministe* ou *non déterministe*, *discret* ou *continu*, *accessible* ou *non accessible* [Hanon, 06] :

L'environnement d'un système est *statique* (par opposition à *dynamique*) si son état ne dépend que de ses états antérieurs et des actions réalisées par le système. Ainsi, l'environnement physique d'un agent, vivant en société, est généralement qualifié de dynamique car d'autres agents y agissent. Cet état dynamique est imprévisible car il ne peut pas être deviné par le système.

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.

Un environnement est *discret* (par opposition à *continu*) si l'ensemble des perceptions et l'ensemble des actions possibles sur cet environnement sont finis. Le monde réel est évidemment continu.

L'environnement d'un système est *accessible* si ce système peut être perçu, à chaque instant, d'une manière complète et précise. A titre d'exemple, l'environnement d'un robot n'est pas complètement accessible car ses capteurs possèdent une précision et une portée limitées.

II.3.2.2 Interaction

L'interaction représente l'une des plus importantes facettes d'un SMA. Elle est définie par *Doniec* [Doniec, 06] comme étant une relation dynamique instaurée entre plusieurs agents du fait de leurs actions combinées et réciproques. L'interaction peut avoir plusieurs formes : la coopération, la coordination, la négociation, la collaboration, etc. Toutefois, elle est essentiellement représentée par la coordination, qui fait référence à tous les processus utilisés dans la prise de décision et au comportement global d'un ensemble d'agents.

II.3.2.2.1 Interaction et coordination

La coordination est définie au sens général par *Doniec* [Doniec, 06], comme étant l'ensemble des actions permettant une production et/ou une organisation en vue d'atteindre un objectif déterminé. *Weiss* [Weiss, 00] définit la coordination en faisant référence aux ressources partagées car les agents doivent coordonner leurs actions dans l'environnement qu'ils partagent afin d'éviter les conflits de ressources (Figure II.3).

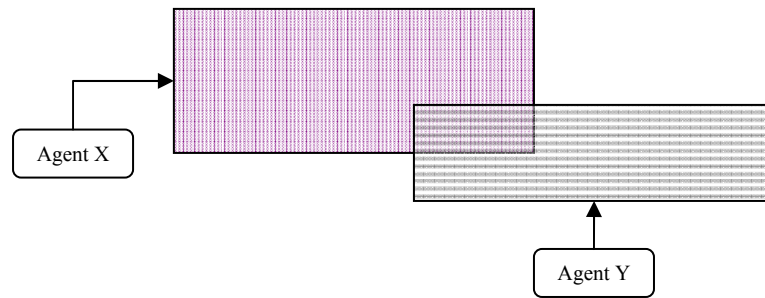


Figure II.3– Situation de conflit

Dans ce sens, soit les agents s'entraident et donc coopèrent pour essayer d'atteindre l'objectif global en évitant les situations de conflits, soit ils se retrouvent dans une situation de compétition qui implique une phase de négociation qui peut être parfois chaotique pour le système :

- **La Coopération** : les agents coopèrent lorsqu'ils sont motivés collectivement pour atteindre le but global (agents *non antagonistes*). Dans ce cas, les agents ne sont pas dans une situation de concurrence, ils essaient donc de s'accommoder sans se déranger. Pour coopérer, les agents collaborent pour partager les tâches et les ressources. La *collaboration* est donc une méthode de coopération qui permet la répartition du travail ;
- **La Négociation** : c'est le cas où 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 (agents *antagonistes*). *Verrons* [Verrons, 04] définit la négociation comme étant un processus grâce auquel plusieurs parties aboutissent successivement à une décision commune, après avoir diffusé initialement, leurs demandes respectives. 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, 81]. Nous avons adopté cette forme d'interaction au sein du système proposé entre les agents optimisateurs stationnaires et les agents mobiles collecteurs d'information (§IV.3).

II.3.2.2.2 Interaction et communication

En général, il faut communiquer pour interagir, mais dans certains cas, un agent peut inférer les plans des autres agents sans avoir besoin de communiquer avec eux [Tubbs, 84] [Rosenschein et al., 86]. La communication exprime l'échange de connaissances et de compétences entre les agents en exploitant divers moyens tels que la diffusion des signaux et l'échange des requêtes. Une communication peut être sélective sur un nombre restreint d'agents ou bien elle peut concerner l'ensemble des agents. A l'origine, la communication se fait indirectement à travers des structures de données partagées appelées *tableau noir* (« *BlackBoard* ») là où les agents accèdent alternativement pour déposer l'information qu'ils veulent diffuser ou récupérer les données dont ils ont besoin (Figure II.4). La communication peut être *directe* ou *indirecte* :

- **La communication indirecte** : Une communication indirecte peut avoir lieu par l'intermédiaire d'agents médiateurs mais généralement, les agents communiquent indirectement via l'environnement qu'ils perçoivent ou qu'ils modifient. Ainsi, l'agent qui désire diffuser une information, porte les modifications nécessaires sur son environnement. Ces modifications seront donc perçues par d'autres agents qui interprètent le message pour récupérer les informations nécessaires,
- **La communication directe** : se fait par envoi de message et correspond au modèle de communication actuel dans les SMA. Le message est encodé par l'agent émetteur par un langage avant sa transmission à l'agent récepteur, ce dernier décode le message reçu pour saisir son contenu.

Dans le contexte de la communication directe, des standards (langages et protocoles) ont été introduits par des organismes telle que la FIPA²⁹ (Foundation of Intelligent Physical Agents) pour une plus grande interopérabilité entre plateformes multi-agents. FIPA a proposé et spécifié le standard ACL comme de langage de communication (Agent Communication

²⁹ www.fipa.org

Language). Un autre langage courant est le KQML³⁰ (Knowledge Query and Manipulation Language) qui est le résultat des efforts de la DARPA³¹ (Defense Advanced Research Project Agency).

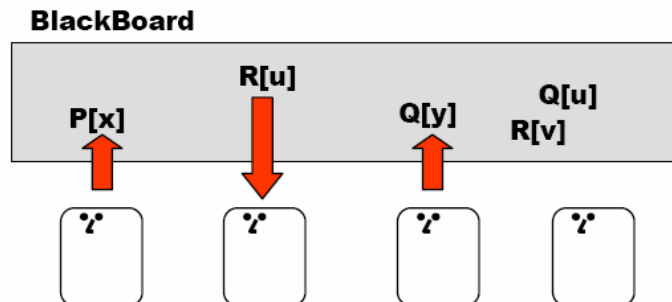


Figure II.4– Le tableau noir

II.3.2.3 Organisation Multi-Agent (OMA)

Un SMA peut être qualifié d'OMA si les agents qui le composent possèdent une connaissance des autres agents du système et suivent les règles fixées par le système global tout en gardant leurs capacités d'autonomie [Mandiau et al., 99]. Une OMA 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.

II.3.2.3.1 Analyse des organisations

Un SMA se trouve être une organisation qui peut être analysée essentiellement de deux manières :

- L'*analyse fonctionnelle* : qui a pour but de déterminer les principales activités et rôles des agents et ce selon plusieurs dimensions : physiques, sociales, relationnelles, environnementales, etc. Par exemple, un agent peut avoir le rôle d'un médiateur, d'un planificateur, d'un ordonnanceur, de décideur, de fournisseur etc. ;
- L'*analyse structurelle* : qui a pour but de déterminer les éventuelles interactions entre les agents ainsi que leur ordre et leur évolution dans le temps. Par exemple, nous

³⁰ www.cs.umbc.edu/kqml

³¹ www-ksl.stanford.edu/knowledge-sharing

pouvons citer la relation d'accointance (connaissances qu'à l'agent sur les autres agents), de communication, de subordination (dépendance), d'information, de conflit, de compétition, etc.

II.3.2.3.2 Typologie des organisations

Nous pouvons classer les différentes typologies des organisations dans les SMAs en deux catégories : typologie relationnelle et typologie structurelle. Cette classification reprend les deux classifications différentes élaborées par *Adam* [Adam, 00] et *Mandiau* [Mandiau et al., 02a]:

- Typologie *relationnelle* : qui dépend de la typologie des interactions. Cette classification correspond à celle d'*Adam* [Adam, 00] et distingue les OMA à structure hiérarchique de celles à structure hétérarchique³² :

- Les *OMA à structure hiérarchique* : dans ce type de structure, le contrôle est centralisé sur un agent de niveau supérieur analogue à un « chef d'orchestre », qui 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.5) ;

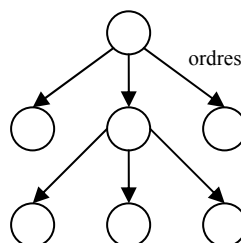


Figure II.5– OMA à structure hiérarchique

- Les *OMA à structure hétérarchique* : ce type de structure se décompose, à son tour, en trois catégories de types *marché*, *communauté* et *société* :
 - Les organisations de type *marché* : c'est une structure faiblement décentralisée car le contrôle est partagé sur plusieurs agents

³² Une structure hétérarchique est un type d'organisation dans lequel il n'y a pas de supérieur hiérarchique, où chaque entité prend ses propres décisions suivant un ensemble de critères et d'objectifs qui lui sont propres.

coordinateurs hiérarchiques qui visent des objectifs opérationnels différents (Figure II.6) ;

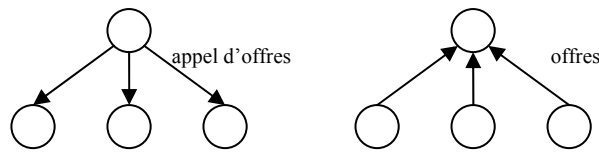


Figure II.6– OMA à structure de marché

- 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.7) ;

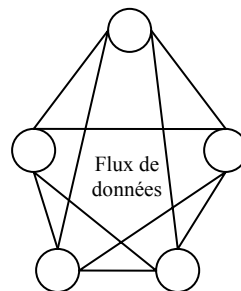


Figure II.7– OMA à structure de communauté

- Les organisations de type **société** : ce type d'organisation est un croisement entre les organisations hiérarchiques et les organisations de communauté, en définissant précisément les règles d'interaction. C'est une structure décentralisée là où le contrôle est partagé par plusieurs unités qui n'ont pas nécessairement les mêmes capacités. *Adam* [Adam, 00] parle d'ajustement mutuel entre les différentes unités par des principes de négociation (Figure II.8).

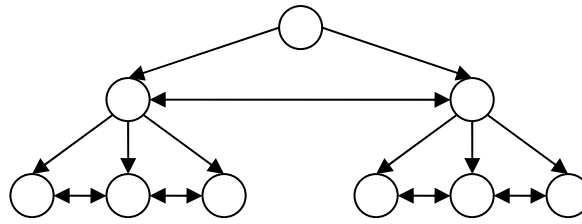


Figure II.8– OMA à structure de société

- Typologie *structurelle* : les structures d'une organisation multi-agent peuvent être soit prédéfinies (tout est connu à l'avance), soit découvertes à postériori (structure émergente) résultant des interactions entre agents. Cette classification rejoint celle de *Mandiau* [Mandiau et al., 02a] qui classe les organisation selon leurs processus de formation et distingue alors deux modèles d'organisation :

- Organisation *émergente* : cette organisation est à découvrir à postériori, c'est-à-dire qu'elle est définie progressivement avec l'évolution du système à travers les interactions inter-agents ;
- Organisation *support d'activités* : cette organisation est définie d'avance c'est-à-dire dès la conception du système. Elle impose le fonctionnement du contrôle global du système qui peut être soit centralisé, c'est-à-dire qu'il y a un monopole du contrôle correspondant à un agent coordinateur, ou décentralisé sur différents agents. Cette structure organisationnelle précise également pour chaque agent, sa fonction dans l'organisation, ses compétences ainsi que les possibilités d'interactions le concernant.

II.3.3 Systèmes d'information et agents logiciels

Une description globale d'un système d'information (SI) citée par [Mandiau et al., 02b] et proposée par [Barron et al., 99], représente un SI comme une structure qui comporte trois composants principaux :

1. Le composant responsable de la collecte d'information qui inclut des capteurs et observateurs chargés d'extraire les données et les observations qui concernent le domaine d'application du système ;

2. Le composant responsable du prétraitement des données si elles ont besoin de transformation (filtrage, regroupement etc.) avant leur utilisation ;

3. Le composant responsable du traitement de ces données afin de fournir de l'information à l'utilisateur du système. Ce composant représente le SI proprement dit.

D'après [Mandiau et al., 02b], chaque composant est susceptible d'être géré par des agents logiciels vu l'apport des agents aux SIs, qui se base sur deux aspects principaux [Strugeon, 98] :

1. la prise en compte de la distribution de l'information c'est-à-dire l'accès à distance à une source d'information, l'extraction des données à partir de bases hétérogènes et enfin l'intégration des informations récupérées de ces différentes sources ;

2. la personnalisation de l'information qui s'adapte au profil de l'utilisateur. Cet aspect se base sur la flexibilité et la richesse du SI et/ou sur sa capacité d'apprentissage.

De ce fait, l'architecture d'un SI doit au moins combler une partie des points suivants :

- le recueil des données distribuées en réseau en considérant son état, cette tâche est susceptible d'être le travail des Agents Mobiles (§II.3.1.5) ;
- l'intégration des données hétérogènes pour fournir à l'utilisateur final le résultat espéré;
- la personnalisation de l'information qui doit s'adapter aux besoins et au profil de chaque utilisateur;
- la pertinence de l'information reçue en temps réel.

II.4 Les SMA dans le domaine du Transport

Dans le domaine du transport, plusieurs modèles se basent sur les SMAs pour la résolution des problèmes :

- la simulation de trafic automobile [Mandiau et al., 02b] ;
- la régulation des réseaux de transport multimodal [Fayech, 03] [OuldSidi, 06];

- l'optimisation, la recherche et la composition des itinéraires multimodaux [Kamoun, 07] [Zidi, 06] ;
- la surveillance des véhicules automatisés [Durfée et al.,91] ;
- le contrôle du trafic aérien [Ljungberg et al.,92] ;
- le transport de marchandises [Fischer et al., 95] ;
- systèmes d'aide au pilotage des avions militaires [Chaib-draa, 95] ;
- la gestion du trafic urbain [Chaib-draa, 96] ;

Nous avons déjà intégré, au sein de l'équipe, des méthodes d'optimisation dans des SMA pour résoudre des problèmes liés au transport tel que les algorithmes de colonies de fourmis pour les problèmes de reconfiguration des réseaux de transport [Zidi et al.,06] ou les algorithmes génétiques pour la régulation de trafic [Fayech, 03], etc .

II.5 Optimisation

II.5.1 Définitions

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.

La recherche de l'optimum de la fonction f se fait par la modification d'une composition de variables, appelées *variables de décision*.

Mathématiquement parlant, *un problème d'optimisation* revient à minimiser une fonction $f(\vec{x})$ telle que $\vec{g}(\vec{x}) \leq 0$ et $\vec{h}(\vec{x}) = 0$ avec :

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

Les contraintes du problème d'optimisation délimitent un espace restreint de recherche de la solution optimale. Deux types de contraintes d'inégalité sont discernés:

- Des contraintes de type $B_{i\text{inf}} \leq x_i \leq B_{i\text{sup}}$: les valeurs de \bar{x} qui vérifient ces contraintes définissent un **espace de recherche**. Cet espace est illustré par la Figure II.9-a pour $n=2$;
- Des contraintes de type $c(\bar{x}) \leq 0$ ou $c(\bar{x}) \geq 0$: les valeurs de \bar{x} qui vérifient ces contraintes définissent un **espace des valeurs réalisables**. Cet espace est illustré par la Figure II.9-b pour $n=2$.

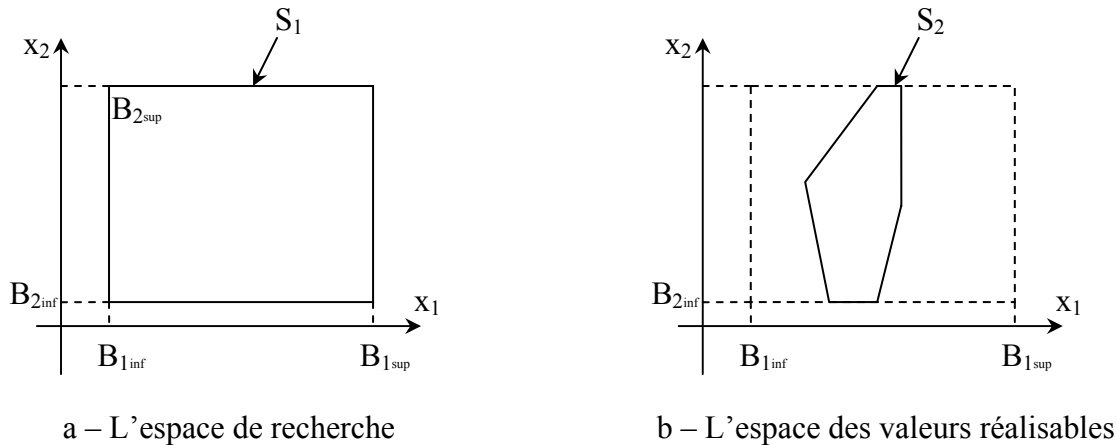


Figure II.9– Les différents espaces de recherche

Si la fonction objectif f est à minimiser alors un point \bar{x}^* est:

- un **minimum global** de la fonction f si $f(\bar{x}^*) < f(\bar{x}) \forall \bar{x}$ tel que $\bar{x}^* \neq \bar{x}$. Le point M_3 de la Figure II.10, correspond à cette définition.
- un **minimum local fort** de la fonction f si $f(\bar{x}^*) < f(\bar{x}) \forall \bar{x} \in V(\bar{x}^*)$ tel que $\bar{x}^* \neq \bar{x}$ et $V(\bar{x}^*)$ définit un voisinage de \bar{x}^* . Les points M_2 et M_4 de la Figure II.10, correspondent à cette définition.

- un *minimum local faible* de la fonction f si $f(\bar{x}^*) \leq f(\bar{x}) \forall \bar{x} \in V(\bar{x}^*)$ tel que $\bar{x}^* \neq \bar{x}$ et $V(\bar{x}^*)$ définit un voisinage de \bar{x}^* . Le point M_1 de la Figure II.10, correspond à cette définition.

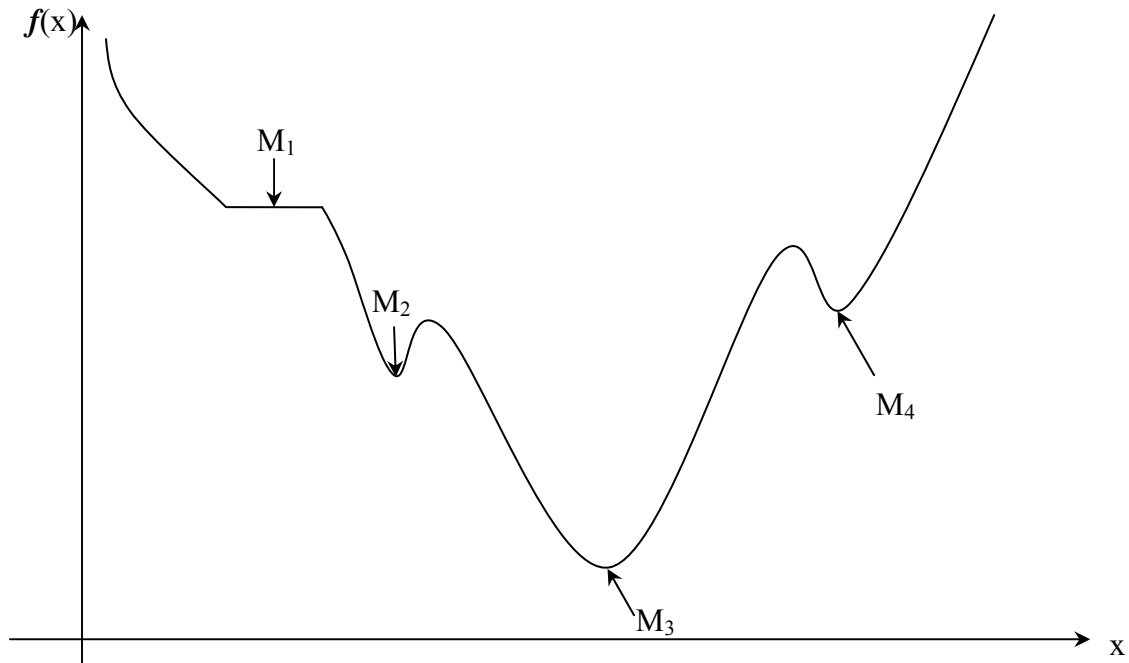


Figure II.10 – Les différents minima

II.5.2 Classification des problèmes d'optimisation

Caractéristiques du problème			Type du problème
Variable de décision	Nombre	1	monovariable
		>1	multivariable
	Type	Nombre réel continu	continu
Nombre entier		entier ou discret	
Permutation sur un ensemble fini de nombres		combinatoire	
Fonction objectif	Type	Fonction linéaire des variables de décision	linéaire
		Fonction quadratique des variables de décision	quadratique
		Fonction non linéaire des variables de décision	non linéaire
Formulation du problème	Type	Avec des contraintes	contraint
		Sans contraintes	non contraint

Tableau II-1 – Classification des problèmes d'optimisation

Les problèmes d'optimisation varient en fonction de leurs caractéristiques ; selon le nombre et le type de la variable de décision, le type de la fonction objectif et le type de la formulation du problème. Cette classification est illustrée par le Tableau II-1.

II.5.3 L'optimisation multiobjectif

Lorsque la modélisation d'un problème nécessite la satisfaction de plusieurs objectifs, il s'agit d'un *problème d'optimisation multiobjectif*, appelé aussi *problème d'optimisation multicritère*.

Mathématiquement parlant, *un problème d'optimisation multiobjectif* revient à minimiser un nombre de fonctions objectif regroupées dans $\vec{f}(\vec{x})$ telle que $\vec{g}(\vec{x}) \leq 0$ et $\vec{h}(\vec{x}) = 0$ avec :

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

Plusieurs solutions peuvent répondre à un problème d'optimisation multicritère car les objectifs sont souvent contradictoires, c'est-à-dire que la diminution d'un objectif entraîne une augmentation de l'autre. Ces solutions ne seront pas *optimales* car elles n'optimisent pas tous les objectifs du problème. Dans ce cas, il est intéressant d'adopter le concept du *compromis* ; Des solutions de compromis optimisent un certain nombre d'objectifs tout en dégradant les performances sur d'autres objectifs. De ce fait, une question se pose : comment savoir si une solution est intéressante entre une multitude de solutions obtenues?

Pour qu'une solution soit jugée intéressante, il faut qu'elle présente une relation de *dominance* avec les autres solutions obtenues.

II.5.4 La dominance

Une relation de dominance permet de filtrer les éléments d'un ensemble pour ne retenir que les éléments incomparables entre eux. Mathématiquement parlant, une *relation de dominance* est exprimée de la manière suivante :

Relation de dominance

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

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

Les solutions qui dominent les autres mais ne se dominent pas entre elles sont appelées **solutions optimales au sens de Pareto** ou **solutions non dominées**.

Un vecteur \bar{x} est **optimal localement au sens de Pareto**, s'il est optimal au sens de Pareto sur une restriction de l'ensemble \mathbb{R}^n . L'**optimalité locale au sens de Pareto**, est illustrée par la Figure II.11 et définie comme suit :

Optimalité locale au sens de Pareto

Un vecteur $\bar{x} \in \mathbb{R}^n$ est optimal localement au sens de Pareto si $\exists \delta > 0$ tel que \nexists un autre vecteur \bar{x}' qui domine le vecteur \bar{x} avec :

- $\bar{x}' \in \mathbb{R}^n \cap B(\bar{x}, \delta)$
- $B(\bar{x}, \delta)$ est une boule de centre \bar{x} et de rayon δ .

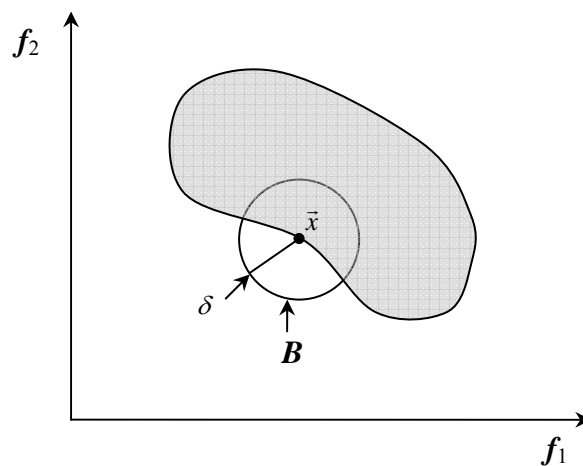


Figure II.11– L'optimalité locale au sens de Pareto

Par rapport à la définition précédente, un vecteur \bar{x} est **optimal globalement au sens de Pareto**, ou **optimal au sens de Pareto**, si la restriction de l'ensemble \mathbb{R}^n n'est plus considérée. L'**optimalité globale au sens de Pareto**, est définie comme suit :

Optimalité globale au sens de Pareto

Un vecteur \bar{x} est optimal globalement au sens de Pareto si \nexists un autre vecteur \bar{x}' qui domine \bar{x} .

L'**optimalité globale au sens de Pareto**, peut être définie graphiquement par le théorème du contact, illustré par la Figure II.12.

Théorème du contact

Un vecteur \bar{x} est optimal au sens de Pareto pour un problème d'optimisation multiobjectif donné si $(C^- + \bar{x}) \cap F = \{\bar{x}\}$ avec :

- C^- est un cône négatif défini dans \mathbb{R}^k comme suit : $C^- = \{\bar{x} / \vec{f}(\bar{x}) \in \mathbb{R}^k \wedge \vec{f}(\bar{x}) \leq 0\}$
- F représente l'espace des solutions réalisables.

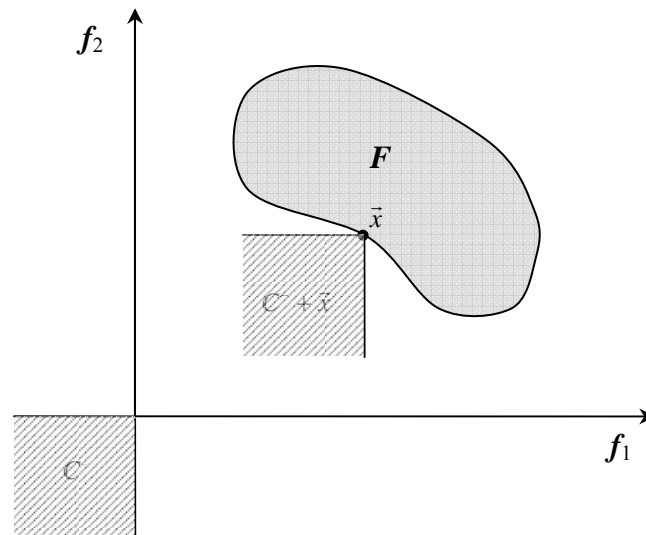


Figure II.12– Le théorème du contact

II.5.5 La ε – dominance

En se basant sur la notion de dominance présentée dans le paragraphe précédent, F représente l'ensemble des vecteurs $\bar{x} / \bar{x} \in \mathbb{R}^n$ et F^* l'ensemble des vecteurs \bar{g} de F qui ne sont dominés par aucun vecteur de F . F^* est donc appelé l'ensemble Pareto de F et les vecteurs de F^* sont appelés les vecteurs Pareto de F . L'ensemble de tous les ensembles Pareto de F est unique et noté par $P^*(F) = \{F^*\}$ [Laumanns et al., 02]. Le problème est que F^* représente un résultat d'optimisation qui peut être de taille importante et donc sa détermination numérique peut être prohibitive. Un résultat d'optimisation de grande taille représente une gêne pour une prise de décision rapide, d'où la notion de ε – **dominance** qui détermine une approximation de F^* pour dominer « approximativement » tous les éléments de F avec une taille polynomiale bornée. Grâce à cette approche, un décideur peut déterminer les parties intéressantes de l'espace objectif qui peut être exploré dans de nouvelles courses d'optimisation.

Pour le cas où tous les objectifs sont à minimiser alors l' ε -**dominance** est définie comme suit :

ε -dominance
 Un vecteur $\bar{x} \in \mathbb{R}^{+n}$ ε -**domine** un autre vecteur $\bar{g} \in \mathbb{R}^{+n}$ pour $\varepsilon > 0$ si et seulement si
 $\forall i \in \{1, \dots, n\} : (1 - \varepsilon) \cdot x_i \geq g_i$.

La Figure II.13 illustre la notion de l' ε -**dominance** dans le cas de deux objectifs à minimiser (n=2).

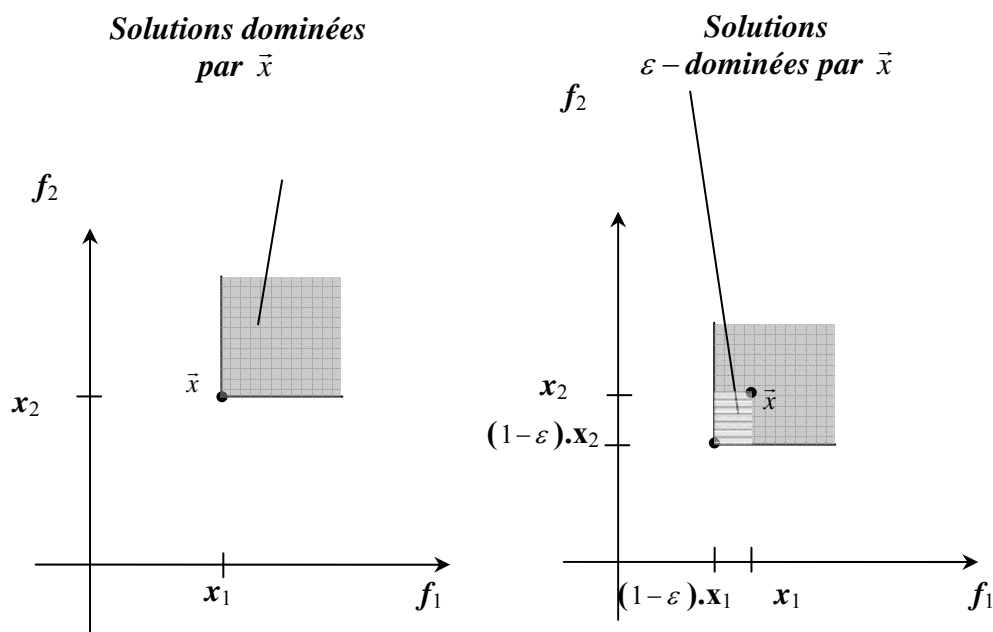


Figure II.13– La dominance et l' ε -dominance

II.5.6 Les méthodes de résolution des problèmes d'optimisation multiobjectif

Plusieurs méthodes de résolution existent pour les problèmes d'optimisation multiobjectif. Ces méthodes sont classées en cinq groupes [Collette et al.,02] :

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

II.5.6.1 Les méthodes scalaires

Appelée également l'approche naïve, cette méthode de résolution est la plus évidente car elle vise à convertir le problème multiobjectif à résoudre, à un problème d'optimisation monobjectif, dont il existe de nombreuses méthodes de résolution. La manière la plus simple à procéder est d'obtenir une nouvelle fonction objectif par la sommation de toutes les fonctions objectif, en amplifiant chaque fonction par un coefficient de pondération.

II.5.6.2 Les méthodes interactives

Ce sont des méthodes progressives qui ne permettent de chercher qu'une seule solution. Au cours de l'optimisation, ces méthodes 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.5.6.3 Les méthodes floues

Ce sont des méthodes qui font intervenir la logique floue pour la résolution des problèmes d'optimisation multiobjectif. Contrairement à la logique classique où tout est décrit en termes de VRAI ou FAUX, la logique floue permet de traiter l'incertitude et l'imprécision des connaissances humaines ainsi que les transitions progressives entre états. Cette logique se base sur les ensembles flous et a été élaborée par l'automaticien L.Zadeh.

II.5.6.4 Les méthodes d'aide à la décision

Ce sont les seules méthodes de résolution de problèmes d'optimisation multiobjectif 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.5.6.5 Les méthodes exploitant une métaheuristique

Les métaheuristiques forment une grande famille de méthodes d'optimisation qui visent à résoudre les problèmes d'optimisation difficile. Les problèmes d'optimisation concernés représentent aussi bien les problèmes d'optimisation combinatoire à variables discrètes que les problèmes d'optimisation globale à variables continues. Ces méthodes sont caractérisées par un haut niveau d'abstraction, ce qui permet de les adapter à une large gamme de problèmes différents, allant de la simple recherche locale aux algorithmes complexes de recherche globale.

II.5.7 Les métaheuristiques

II.5.7.1 Définition d'un problème d'optimisation combinatoire

L'optimisation combinatoire est le secteur des métaheuristiques discrètes qui consiste à trouver la meilleure solution dans l'ensemble des solutions réalisables, noté par X . En général, cet ensemble est fini mais compte un très grand nombre d'éléments qui doivent tous satisfaire un ensemble C de contraintes. Si f est la fonction objectif qui permet d'évaluer chaque solution réalisable. Alors un problème d'optimisation combinatoire vise à déterminer une solution $x^* \in X$ qui minimise f comme suit : $f(s^*) = \min_{s \in X} f(s)$

II.5.7.2 Contribution des métaheuristiques par rapport aux limites des méthodes classiques

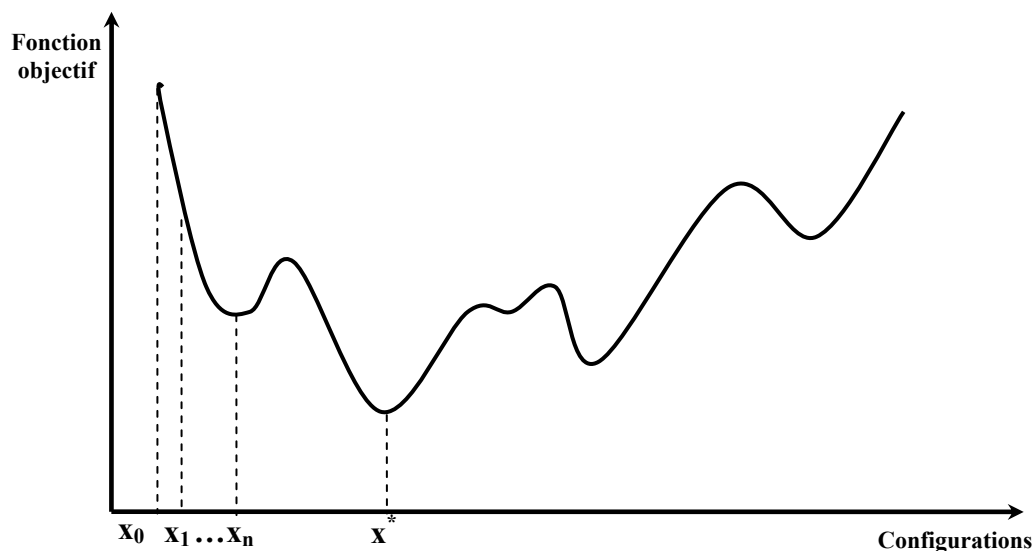


Figure II.14– Allure de la fonction objectif d'un problème d'optimisation difficile

Une *méthode classique d'amélioration itérative*, appelée aussi *méthode de descente*, risque de tomber dans le piège d'un optimum local. C'est une méthode de recherche locale qui part d'une configuration initiale, souvent choisie au hasard, pour générer ensuite une nouvelle solution par une modification élémentaire. Si le changement conduit à une amélioration de la fonction objectif, alors il est accepté et la nouvelle configuration obtenue, qui est « voisine » de la précédente, sert de point de départ pour un nouvel essai. Dans le cas contraire, la configuration précédente est reprise avant de faire une autre tentative. Le processus est itéré jusqu'à ce que toute modification rende le résultat moins bon. La Figure

II.14 montre qu'un algorithme classique ne conduit pas en général, à l'optimum absolu, mais seulement à un optimum local, qui constitue la meilleure des solutions accessibles compte tenu de l'hypothèse initiale.

Pour augmenter son efficacité, cette méthode peut être appliquée plusieurs fois, avec des conditions initiales différentes et retenir comme solution finale le meilleur des optimums locaux obtenus. Cette technique ne garantit pas de trouver la configuration optimale, en plus elle augmente le temps de calcul de l'algorithme. L'application répétée d'une méthode classique est donc sans utilité lorsque le nombre des optimums locaux croît exponentiellement avec la taille du problème.

Accepter une dégradation temporaire de la fonction objectif, lors du changement de la configuration courante fut la base de toutes les métaheuristiques pour surmonter les obstacles des optimums locaux. La divergence du procédé peut être évitée par un mécanisme de contrôle des dégradations, spécifique à chaque métaheuristique. Nous présentons dans le paragraphe qui suit, les quatre différentes approches existantes de métaheuristiques.

II.5.7.3 Les différents types de métaheuristiques

II.5.7.3.1 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 (Figure II.15). Ces méthodes fonctionnent avec une seule configuration courante à la fois, actualisée au cours des itérations successives.

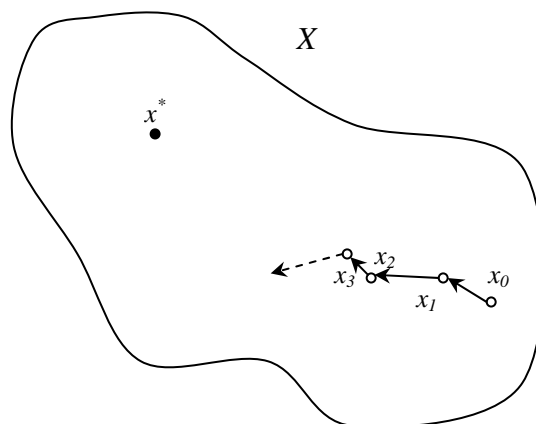


Figure II.15– Exploration de l'ensemble des solutions réalisables par une approche de recherche locale

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 : Le Recuit Simulé, les méthodes d'acceptation à seuil et la méthode Tabou.

II.5.7.3.1.1 Le Recuit Simulé (« Simulated Annealing »)

La méthode du Recuit Simulé tient ses sources aux expériences du domaine de la physique statique [Metropolis et al., 53] qui ont aboutit à un algorithme simple pour simuler l'évolution d'un système physique instable vers un état d'équilibre thermique à une température t fixée. Par analogie à l'énergie d'un matériau, la fonction objectif du problème d'optimisation à résoudre est donc à minimiser. A chaque itération, cette méthode génère une solution voisine x' qui est directement acceptée si elle est meilleure que la solution courante x . Dans le cas contraire, x' est acceptée avec une certaine probabilité $prob(\Delta f, t)$ qui dépend de l'importance de la détérioration $\Delta f = f(x') - f(x)$ et d'un paramètre t , qui correspond à la température dont les changements sont effectués selon un schéma de refroidissement précis. En général, la température est diminuée par palier à chaque nombre d'itérations effectuées et l'algorithme est interrompu si aucune solution voisine n'a été acceptée pendant un cycle complet d'itérations à température constante. La meilleure solution trouvée est mémorisée dans une variable x^* . La méthode du Recuit Simulé est souple vis-à-vis des évolutions du problème et facile à implémenter, elle est décrite génériquement par l'algorithme II.1. Les **méthodes d'acceptation à seuil** 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.

Algorithme II.1 - Le 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 $x \in X$

$x^* = x, k = 0, \textit{nouveau_cycle} = \text{vrai}, t = t_0$

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

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

 Sinon

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

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

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

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

 Si $f(x) < f(x^*)$ alors faire

 | $x^* = x$

$t = a.t$

Enfin, l'inconvénient majeur de la méthode du Recuit Simulé se présente lorsqu'un minimum local se trouve au fond d'une « vallée profonde », dans ce cas il est impossible de s'en sortir en une seule itération. En plus, les déplacements vers des solutions voisines peuvent provoquer des déplacements inverses qui risqueraient de cycler autour d'un minimum local. La méthode Tabou décrite dans ce qui suit, porte une amélioration à ces inconvénients majeurs.

II.5.7.3.1.2 La méthode Tabou (« Tabou Search »)

Appelée *méthode de recherche avec tabous*, ou *recherche tabou* ou encore *méthode tabou*, ce procédé permet de tirer les leçons du passé en gardant en mémoire les dernières solutions visitées pour interdire le retour vers ces solutions pendant un nombre fixé d'itérations. Cette liste de solutions « interdites » est appelée *solutions taboues* et les mouvements qui ramèneraient à ces solutions sont appelés *mouvements tabous*. Pour une meilleure gestion de l'espace mémoire, les *mouvements tabous* sont souvent enregistrés à la place des *solutions taboues*. Ce sont donc les mouvements inverses qui sont interdits pour éviter de se retrouver dans des solutions déjà visitées. L'interdiction est levée lorsque la solution courante est suffisamment modifiée pour que les mouvements tabous ne puissent la transformer en une solution déjà rencontrée et former ainsi un cycle.

Le statut Tabou d'un mouvement peut être levé prématurément lorsqu'il mène à une solution meilleure que toutes celles déjà aperçues par la recherche au cours des itérations passées. Cette action est appelée « aspiration » car dans ce cas, le mouvement en question est aspiré de l'ensemble tabou en satisfaisant une certaine fonction, appelée *fonction d'aspiration* qui mémorise généralement le mouvement qui a mené à la dernière meilleure solution rencontrée.

II.5.7.3.2 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 (Figure II.16). A chaque étape k , l'ensemble des solutions réalisables X est donc réduit en un ensemble $X^k \subseteq X$.

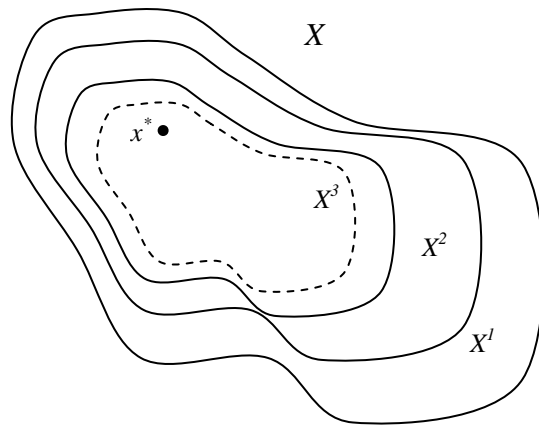


Figure II.16– Exploration de l'ensemble des solutions réalisables par une approche constructive

L'algorithme NEH [Nawaz et al., 83] 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., 01]. 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.

II.5.7.3.3 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 (Figure II.17). 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

inconvenient, 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 évolutionnaires et les algorithmes de colonies de fourmis.

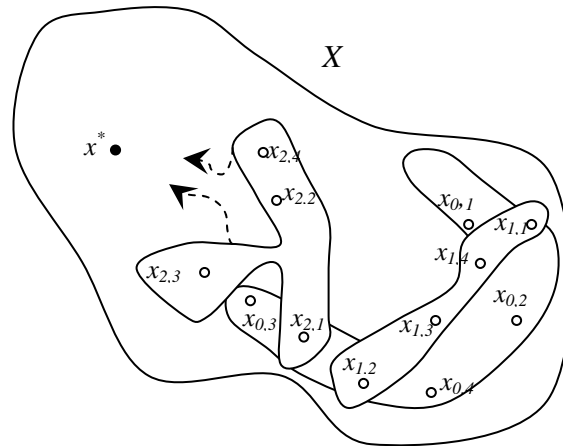


Figure II.17– Exploration de l'ensemble des solutions réalisables par une approche évolutive

II.5.7.3.3.1 Les algorithmes évolutionnaires

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 [Siarry et al., 03] 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.18):

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.

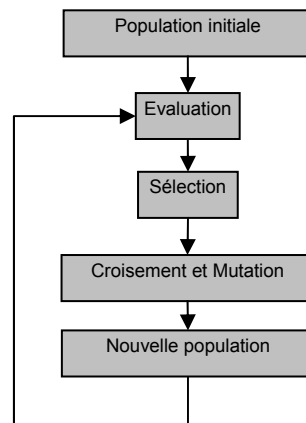


Figure II.18– Fonctionnement générique d'un algorithme évolutionnaire

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 et notamment dans le domaine de l'ordonnancement [Caux et al., 94] [Pierreval et al.,03].

II.5.7.3.3.1.1 Les algorithmes génétiques

C'est la variante la plus connue des algorithmes évolutionnaires, d'ailleurs, par abus de langage, beaucoup de spécialistes désignent et continuent à désigner les approches évolutionnaires par le terme « algorithmes génétiques ». Comme leur nom l'indique, les algorithmes génétiques s'inspirent du patrimoine héréditaire d'un individu (génotype) représenté par ses chromosomes. L'interaction du génotype d'un individu avec son environnement détermine son phénotype qui peut être modifié par mutation. Le phénotype est évalué par décodage du génotype, qui est une chaîne de symboles souvent binaires, dans le but de donner une valeur de performance exploitable par les opérateurs de sélection (couleur des yeux, des cheveux, traits du visage, etc.). Les opérateurs de variation (croisement et

mutation) présentés précédemment sont donc relatifs à la représentation binaire puisqu'ils agissent sur les chaînes binaires des génotypes. Dans un algorithme génétique simple, la recherche est réglée par l'application successive des opérateurs de variation. Le croisement correspond à la phase de coopération entre les individus alors que la mutation correspond à la phase d'adaptation individuelle.

II.5.7.3.3.1.2 La méthode de recherche distribuée (« Scatter Search »)

C'est une variante peu connue des algorithmes évolutionnaires vu l'échec qu'elle a subi. En fait, c'est une évolution qui n'est pas en relation directe avec le domaine de la génétique, ne représentant aucune restriction imposée sur la structure des solutions admissibles.

II.5.7.3.3.2 Les algorithmes de colonies de fourmis

Une fois encore, la nature fut d'une grande inspiration à la science humaine ; L'inspiration des algorithmes génériques tient sa source de la transcription génotype et l'inspiration des algorithmes de colonies de fourmis, quand à elle, tient sa source des performances collectives des insectes sociaux et des mécanismes qui permettent aux individus d'une même colonie, de régler leurs activités et de favoriser la survie de l'espèce. Chez certains insectes sociaux, et notamment les fourmis, les individus communiquent à travers une substance agissant en quantités infinitésimales, appelée *phéromone*, qu'ils secrètent, modifient, détectent et transportent même à plusieurs kilomètres. Cette substance extrêmement active joue un rôle très important dans la vie sociale des individus de la même espèce. Une colonie de fourmis communique donc indirectement avec les modifications dynamiques de son environnement, qui n'est d'autre que les pistes de phéromones. Ainsi, les individus d'une même colonie construisent ensemble une solution à un problème en s'appuyant sur leur expérience collective. Nous reprenons ici l'algorithme de colonies de fourmis de base (Algorithme II.2), appelé système de fourmis (« Ant System : AS »), cité dans [Siarry et al., 03]. Cet algorithme se base sur le problème du voyageur de commerce (« Travelling Salesman problem : TSP »).

Lorsqu'une fourmi k donnée se trouve dans une ville i , alors le trajet entre cette ville et une autre ville j dépend de la liste des villes déjà visitées, décrivant les mouvements possibles à chaque pas. Ces mouvements sont notés par J_i^k et le choix d'une ville j dans cette liste, à l'itération t , se fait par la formule suivante :

$$p_{ij}^k = \begin{cases} \frac{(\tau_{ij}(t))^\alpha \cdot (\eta_{ij})^\beta}{\sum_{l \in J_i^k} (\tau_{il}(t))^\alpha \cdot (\eta_{il})^\beta} & \text{si } j \in J_i^k \\ 0 & \text{si } j \notin J_i^k \end{cases} \quad (\text{II.1})$$

Avec :

- d_{ij} représente la **distance** entre la ville i et la ville j ;
- $\tau_{ij}(t)$ représente l'**intensité** de la piste en phéromone entre les deux villes i et j . Cette intensité change au cours des itérations, ce qui correspond au processus d'évaporation des pistes de phéromone. Le but est d'éviter les pièges des solutions sous-optimales, en permettant au système d'« oublier » en quelque sorte les mauvaises solutions. Sachant que la quantité initiale de phéromone sur les arêtes est une distribution uniforme d'une petite quantité $\tau_0 \geq 0$, alors la mise à jour de l'intensité des pistes suit la formule suivante :

$$\tau_{ij}(t+1) = (1 - \rho) \cdot \tau_{ij}(t) + \Delta \tau_{ij}(t) \quad (\text{II.2})$$

Avec : $\Delta \tau_{ij}(t) = \sum_{k=1}^m \Delta \tau_{ij}^k(t)$ et $\Delta \tau_{ij}^k(t)$ correspond à la quantité de phéromones laissée par la fourmi k , sur l'ensemble de son parcours, après un tour complet avec $1 \leq k \leq m$. Cette quantité de phéromone dépend de la qualité de la solution trouvée, suivant la formule suivante :

$$\Delta \tau_{ij}^k(t) = \begin{cases} \frac{Q}{L^k(t)} & \text{si } (i, j) \in T^k(t) \\ 0 & \text{si } (i, j) \notin T^k(t) \end{cases} \quad (\text{II.3})$$

Avec : $T^k(t)$ correspond au trajet effectué par la fourmi k à l'itération t , $L^k(t)$ correspond à la longueur du tour et Q correspond à un paramètre fixé.

- $\eta_{ij} = \frac{1}{d_{ij}}$ représente la **visibilité** entre les deux villes i et j . C'est une information statique, utilisée pour diriger le choix des fourmis vers des villes proches afin d'éviter les villes trop lointaines ;

- α et β : paramètres de contrôle de l'importance relative de l'intensité de la piste $\tau_{ij}(t)$ et de la visibilité η_{ij} . Donc si $\alpha = 0$ alors seule la visibilité de la ville est prise en compte et si $\beta = 0$ alors seules les pistes de phéromones jouent. Il est alors important de trouver un compromis entre ces deux paramètres (*intensification* et *diversification*) afin d'éviter une sélection trop rapide d'un trajet [Siarry et al., 03].

Algorithme II.2 - Colonies de fourmis de base

t : Itération avec $1 \leq t \leq t_{\max}$

k : Identifiant d'une fourmi avec $1 \leq k \leq m$

Pour t de 1 à t_{\max} faire:

Pour chaque fourmi k de 1 à m faire :

Choisir une ville au hasard

Pour chaque ville non visitée i faire :

Choisir une ville j , dans la liste J_i^k des villes restantes selon la **formule II.1**

Déposer une piste $\Delta\tau_{ij}^k(t)$ sur le trajet $T^k(t)$ selon la **formule II.3**

Evaporer les pistes selon la **formule II.2**

II.5.7.3.4 L'approche hybride

Des résultats intéressants ont été remarqués lors de la combinaison des algorithmes évolutifs avec des algorithmes de recherche locale (Figure II.19). Il s'agit d'insérer, lors de la phase d'adaptation individuelle d'une méthode évolutive, une méthode de recherche locale qui permet d'explorer en profondeur, une région de l'ensemble des solutions réalisables. De son côté, son alliée, la méthode de recherche évolutive, s'occupe d'introduire des règles de conduite générale, pour contrôler la recherche.

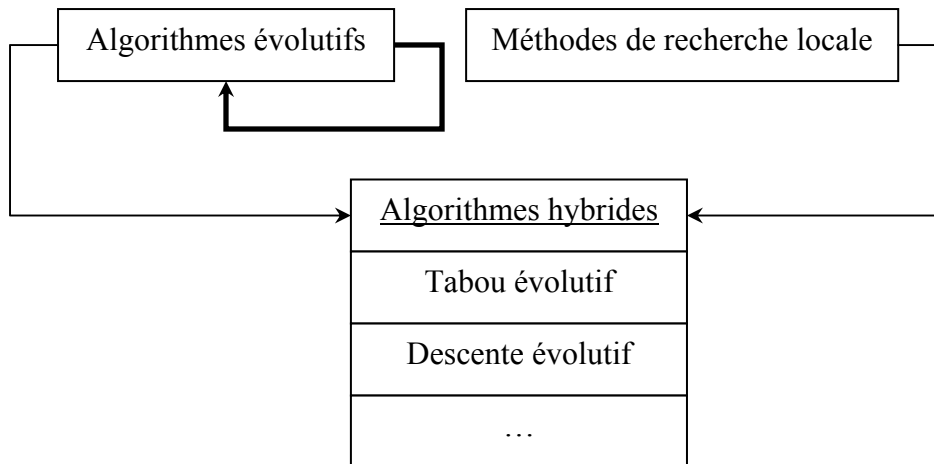


Figure II.19–Combinaisons de différentes approches

II.6 Les algorithmes évolutionnaires dans le domaine du transport

Dans le domaine du transport, les algorithmes évolutionnaires ont été utilisés pour la résolution des problèmes de régulation des réseaux de transport multimodal, pour le contrôle du trafic aérien, l'optimisation des itinéraires, etc.

Le problème de régulation des réseaux de transport multimodal nécessite non seulement le réordonnancement des courses des véhicules mais également la surveillance des réseaux et le diagnostic des incidents. Pour ce faire, *Fayech* [Fayech, 03] propose un Système d'Aide à la Décision (SAD) qui repose sur une approche évolutionnaire pour la régulation temporelle (AERSR) ou spatio-temporelle (AERAR), en cas de situations inhabituelles.

Le problème de contrôle du trafic aérien a été abordé par des approches évolutionnaires. Quatre thèmes sont particulièrement abordés : d'abord, la résolution des conflits entre les différentes trajectoires des avions, en prenant en considération une distance de séparation conforme à une norme donnée. Le deuxième thème aborde la division de l'espace bidimensionnel que représente le réseau aérien. Cette division doit représenter des secteurs équilibrés en terme de charge de travail, qui minimisent les tâches de coordination sachant que la charge de travail est liée au nombre d'avions et de conflits. Le troisième thème s'occupe de l'affectation du trafic entre les secteurs. Cette affectation doit réduire les augmentations des distances parcourues d'une part et la charge de travail des responsables des secteurs. Ces trois premières problématiques ont été traitées par [Delahaye et al, 96]. Enfin, le

dernier thème abordé concerne l'ordonnancement des temps d'atterrissage des avions sur un nombre limité de pistes [Ciesielski et al., 98] [Stevens, 95].

Pour le problème de l'optimisation des itinéraires, *Zidi* [Zidi, 06] propose une hybridation de l'algorithme de Dijkstra avec les algorithmes évolutionnaires pour optimiser des déplacements selon plusieurs critères.

II.7 Vers une architecture distribuée à base d'agents optimisateurs

Dans un SMA, deux modes de communication existent : un mode de communication explicite qui se base sur des protocoles véhiculant des concepts sémantiques de haut niveau et un mode de communication stigmergique qui se base sur l'environnement. Le premier mode est essentiel pour la communication directe entre agents mais pas toujours suffisant pour coordonner l'activité décentralisée d'un SMA. Pour ce faire, nous avons choisi de lier le mode de communication explicite à un mode de communication stigmergique afin de coordonner les tâches globales entre agents sans contrôle centralisé. Ce mode de communication mixte va permettre aux différentes entités d'un SMA de raisonner, de décider et d'interagir directement et/ou indirectement c'est-à-dire explicitement ou à travers l'environnement d'évolution.

Le mode de communication stigmergique permet une coordination émergente de l'activité décentralisée pour le problème complexe de partage des ressources critiques [Armetta, 06]. Les agents engagés dans la résolution de ce type de problème doivent trouver un bon compromis entre l'efficacité de résolution et la consommation de ressources et notamment le temps de résolution et les ressources de calcul.

Nous avons donc choisi de doter les agents d'algorithmes d'optimisation adaptés à leur connaissance et compétence (agents optimisateurs), de leur laisser des fois les moyens de communiquer directement via des protocoles de coordination et d'autoriser le mode de communication stigmergique par l'environnement.

II.8 Conclusion

Notre but est de concevoir et d'optimiser un système d'information d'aide à la mobilité urbaine qui fournit aux utilisateurs les services demandés en respectant les deux critères : le coût de l'information et le temps nécessaire pour sa procuration. C'est un problème multi-objectif à résoudre au sein d'un système largement distribué et dynamique, qui interconnecte des fournisseurs d'information de transport multimodal et d'information

connexe en concurrence. L'étude bibliographique que nous avons établie dans ce chapitre nous incite à concevoir un Système d'information multi-agent d'aide à la mobilité urbaine, en se basant sur une approche évolutionnaire pour la résolution de problème difficile multi-objectif. En effet, 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.

**CHAPITRE III. APPROCHE PROPOSEE D'OPTIMISATION
DISTRIBUEE DE RECHERCHE ET DE COMPOSITIONS DES
SERVICES MULTI-SOURCES AU PROFIT DES CLIENTS DE
TRANSPORT**

III.1 Introduction

Le problème de l'optimisation distribuée de recherche et de composition des services multi-sources dans le domaine du transport découle de l'accroissement continu des sources d'information distantes et distribuées sur les réseaux à grandes échelles. Les systèmes d'information distribués sont directement impliqués par ce problème d'optimisation qui nécessite le choix et la recherche de services aussi diversifiés que nombreux car un même service peut être proposé par plusieurs fournisseurs d'information, en concurrence, avec des coûts, des formats et des temps de réponses différents. Dans ce contexte, et bien que les requêtes utilisateurs peuvent être nombreuses et simultanées, le client du transport a besoin d'une information pertinente, de meilleur coût et en temps réel.

Pour ce problème, nous proposons une solution à trois niveaux d'optimisation, dissimulée dans un système d'information multi-agent ouvert et dynamique, basé sur le paradigme Agent Mobile (AM) pour optimiser la gestion du flux informatique. Le premier niveau d'optimisation permet de calculer un nombre effectif d'AMs pour explorer l'ensemble du réseau et de préparer des plans initiaux pour leur futur itinéraire. Ce niveau d'optimisation dépend de l'état du réseau, il n'est donc mis à jour que si cet état varie considérablement. Le deuxième niveau d'optimisation se base sur le premier et n'intervient qu'à la déclaration de requêtes utilisateurs. Il s'agit de déduire l'itinéraire définitif de chaque AM à partir des plans préalablement préparés, en optimisant les réponses des requêtes utilisateurs en termes de temps et coût de l'information. Enfin, le troisième niveau d'optimisation repose sur les deux premiers et vise à réduire le flux informatique de données en utilisant une technique d'archivage dynamique.

La suite de ce chapitre est organisée comme suit : le problème traité est décrit et formalisé dans le paragraphe III.2. L'architecture multi-agent proposée est développée dans le paragraphe III.3 et les approches d'optimisation adoptées sont déployées dans les paragraphes III.4 et III.5.

III.2 Formulation du problème

Un Système d'Information de Transport Multimodal doit satisfaire les requêtes utilisateurs du réseau de transport, en respectant les délais des réponses (dates de réponses au plus tard) et en réduisant le coût total de l'information. Ce problème devient difficile à résoudre si les requêtes utilisateurs sont nombreuses et simultanées, 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 une phase de décomposition de ces requêtes en sous-requêtes en identifiant les éventuelles similarités. Une sous-requête perçue correspond à une tâche indépendante dont le service correspondant peut être proposé par plusieurs fournisseurs d'information à la fois, avec différents coûts, temps de réponse et formats de l'information. Un fournisseur d'information peut aussi bien être un exploitant du réseau de transport ou un fournisseur d'information connexe. Ainsi, une réponse à une sous-requête donnée peut correspondre à un service de transport (une portion de chemin, une zone géographique bien reconnue, trafic routier, etc.) ou à un service connexe (événement culturel, prévisions météorologiques, services touristiques et promotionnels, etc..). Les fournisseurs de services, appelés nœuds, correspondent à des serveurs d'information distribués sur le réseau de transport, appelé Réseau Etendu de Transport Multimodal (RETM). Les nœuds qui proposent des services aux sous-requêtes (tâches) identifiées, sont reconnus lors de la phase de décomposition (Figure III.1). Les nœuds identifiés doivent par la suite être affectés aux tâches pour satisfaire les requêtes clients simultanées. L'affectation finale des nœuds aux tâches correspond à une solution globale qui doit respecter les contraintes du système.

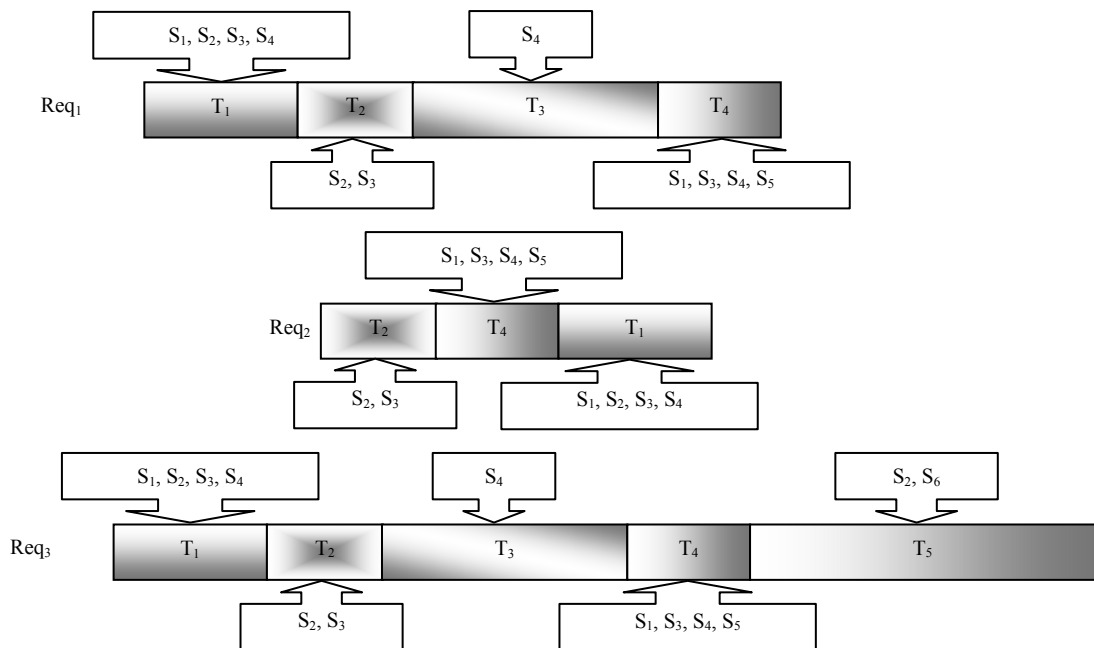


Figure III.1 - Identification des nœuds

Le problème décrit ci-dessus est défini par :

- N requêtes attendant des réponses au même instant t . L'ensemble de ces requêtes est noté par \mathfrak{R}_t ,
- Un ensemble de I tâches indépendantes qui correspondent à tous les services proposés par le RETM. L'ensemble de ces services est noté par $\Gamma = \{T_1, \dots, T_I\}$,
- Chaque requête $req_w \in \mathfrak{R}_t (1 \leq w \leq N)$ est décomposée en un ensemble de tâches indépendantes. Cet ensemble est noté par $\Gamma_{t,w} = \{T_{a1}, \dots, T_{an_w}\} (1 \leq n_w \leq I \text{ et } \Gamma_{t,w} \subseteq \Gamma)$,
- L'ensemble des I' tâches indépendantes, qui composent globalement un ensemble \mathfrak{R}_t , est noté par $\Gamma'_t = \{T_{b1}, \dots, T_{bI'}\} (\Gamma'_t \subseteq \Gamma \text{ et } \bigcup_{w=1}^N \Gamma_{t,w} = \Gamma'_t)$,
- Chaque requête $req_w \in \mathfrak{R}_t (1 \leq w \leq N)$ est caractérisée par une date de fin d'exécution au plus tard d_w initialement connue, une date de fin d'exécution D_w et un coût total C_w ,
- La réalisation de chaque tâche $T_i \in \Gamma$ nécessite une ressource, ou un nœud, à sélectionner à partir d'un ensemble de J fournisseurs d'informations préalablement inscrits dans le RETM. Cet ensemble est noté par $Y = \{S_1, \dots, S_J\}$,
- L'ensemble des J' nœuds ($J' \leq J$), sélectionnés à partir de Y pour répondre à Γ'_t est noté par $Y'_t = \{S_{c1}, \dots, S_{cJ'}\} (Y'_t \subseteq Y)$,
- Le temps d'exécution d'une tâche $T_i (1 \leq i \leq I)$ utilisant les ressources d'un nœud $S_j (1 \leq j \leq J)$ est prédéfini et noté par $P_{i,j}$; $P_{i,j}$ dépend donc du nœud sélectionné S_j ;
- Le coût de l'information à collecter d'un nœud $S_j (1 \leq j \leq J)$, correspondant au service référencé par $T_i (1 \leq i \leq I)$, est prédéfini et noté par $Co_{i,j}$;
- La taille des données à collecter d'un nœud $S_j (1 \leq j \leq J)$, pour assurer le service référencé par $T_i (1 \leq i \leq I)$, est prédéfinie et notée par $Q_{i,j}$.

Les trois caractéristiques décrites ci-dessus, à savoir $(P_{i,j}; Co_{i,j}; Q_{i,j})$, représentent successivement le premier, deuxième et troisième terme de chaque élément appartenant à la *table de services* (Table III-1).

Il est à noter que si un fournisseur ne propose pas de service pour une tâche donnée (flexibilité partielle), alors le terme correspondant dans la table de service est (0; 0; 0). Il est également possible d'avoir $P_{i,j} \neq 0$, $Q_{i,j} \neq 0$ et $Co_{i,j} = 0$ si l'information est gratuite comme dans le cas d'une offre promotionnelle.

	S ₁	S ₂	S ₃	...	S _j
T ₁	(0;0;0)	(0.2;5;3)	(0.4;3;3)		(0.2;5;3)
T ₂	(0.2;4;5)	(0.1;5;2)	(0.4;5;1)		(0.3;8;3)
T ₃	(0.1;0;3)	(0;0;0)	(0.2;0;3)		(0.4;2;2)
T ₄	(0.3;2;1)	(0.3;1;1)	(0;0;0)		(0,0,0)
...					
T ₁	(0.2;3;1)	(0.1;1;3)	(0.4;5;2)		(0.4;5;3)

Table III-1- Exemple de table de services

Afin de situer la complexité du problème décrit ci-dessus, que nous appelons Problème de Recherche et de Composition des Services (PRCS), nous l'avons comparé au Problème bien connu du Job Shop Flexible (PJSF) [Carlier, 89] [Kacem et al., 01] (Tableau III-1).

Dans le PRCS, nous prenons en considération les similitudes entre les différentes requêtes simultanées afin d'éviter une redondance de recherche de données. En plus, nous affectons aussi bien les serveurs aux différentes tâches identifiées que les AMs aux nœuds distants.

Par analogie avec le PJSF, le PRCS est un problème d'optimisation combinatoire NP-difficile qui traite conjointement le problème d'ordonnancement des tâches aux serveurs en optimisant leur coût et leur délai et le problème d'optimisation des plans de routes des AMs pour visiter les nœuds distants afin de récupérer, le plus rapidement possible, les réponses aux tâches.

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.

PJSF	PRCS
N jobs	N requêtes
M machines	J serveurs
n_w opérations non preemptables /job _w	n_w tâches non preemptable /req _w
L'exécution de chaque opération i d'un job w ($O_{i,w}$) nécessite une ressource (ou une machine) sélectionnée à partir d'un ensemble de machines disponibles	L'exécution de chaque tâche i (T_i) d'une requête w (req _w) nécessite une ressource (ou un serveur) sélectionnée à partir d'un ensemble de serveurs disponibles (similarités entre les requêtes)
L'affectation d'une machine M_j à l'opération $O_{i,w}$ exige l'occupation de cette machine pendant un temps d'exécution noté par $d_{i,w,j}$	L'affectation d'un serveur S_j à la tâche T_i exige l'occupation de ce serveur pendant un temps d'exécution noté par $P_{i,j}$
A un instant donné, une machine ne peut exécuter qu'une seule opération à la fois: cette même machine ne redevient disponible pour effectuer d'autres opérations que lorsque l'opération à laquelle elle est actuellement affectée est accomplie (contrainte des ressources).	A un instant donné, un serveur ne peut répondre qu'à une seule tâche à la fois: ce même serveur ne redevient disponible pour répondre à d'autres tâches que lorsque la tâche à laquelle il est actuellement affecté est accomplie (contrainte des ressources).
Problème : Organiser la réalisation des N jobs sur les M machines	Problème : Organiser la réalisation des N requêtes sur les J serveurs par m AMs

Tableau III-1- Analogie entre le PJSF et le PRCS

III.3 Architecture multi-agents proposée

Pour résoudre le problème décrit précédemment, nous proposons un système dynamique et ouvert (Figure III.2), basé sur l'interaction de cinq types d'agents logiciels: Les agents Interfaces Utilisateurs (AI), les agents identificateurs (AId), les agents Ordonnanceurs (AO) et les agents Mobiles Collecteurs Intelligents (ACI). Les agents AI jouent le rôle d'interface entre les utilisateurs et le système, les agents AId décomposent les requêtes et identifient les fournisseurs d'informations, les agents AO s'occupent des ordonnancements tâches/serveurs et serveurs/agents ACI, les agents ACI sont responsables de la recherche des services dans le RETM et enfin les agents AF s'occupent de la composition des réponses utilisateurs. 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, qu'on appelle *période d'inactivité* et qu'on note par Δ_∞ , alors aucun agent n'existe dans le système.

A un instant t , la formulation d'une requête utilisateur nécessite la création d'un agent AI et déclenche la création d'un agent AId, un agent AF, un agent AO et un certain nombre

d'agents ACI. On appelle les agents AI_d, AF, AO et ACI ainsi créés à l'instant t , la société d'agents P_t . Ensuite, et pendant une courte période de temps, qu'on appelle *période d'acquisition* et qu'on note par Δ_ε , chaque nouvelle requête formulée par un nouvel agent AI, est gérée par la même société d'agents P_t dont la création a été déclenchée par la première requête formulée à l'instant t . Si à l'instant $t + \Delta_\varepsilon$, toutes les sociétés d'agents préalablement créées sont indisponibles, alors la formulation de nouvelles requêtes déclenche la création d'une nouvelle société $P_{t+\Delta_\varepsilon}$ et ainsi de suite. Dès qu'une société P_{t_i} créée à l'instant t_i , est disponible, elle devient prête à gérer un ensemble de requêtes formulées pendant une nouvelle période Δ_ε à un nouvel instant t_j avec $t_i \leq t_j$, on appelle les requêtes formulées pendant la période Δ_ε , des requêtes Δ_ε – simultanées. Cependant, dès que la disponibilité d'une société ou d'un agent AI atteint la période Δ_∞ , alors cette société ou cet agent est automatiquement détruit. La Figure III.4 représente un diagramme de séquence qui illustre le comportement global du système pendant une période de formulation Δ_ε . Un exemple de scénario simplifié, exprimé par le diagramme de timing de la Figure III.3, montre les changements d'état d'une société P_{t_2} créée à l'instant t_2 et deux agents interfaces, l'un créé au même moment que P_{t_2} (AI₂₁) et l'autre a déjà été créé auparavant et n'est encore pas détruit (AI₁). D'après ce diagramme de timing, l'agent AI₂₁ interagit avec la société P_{t_2} entre les instants t_3 et $t_3 + \Delta_\varepsilon$. L'agent AI₁ prend la relève et interagit avec P_{t_2} entre les instants t_4 et $t_4 + \Delta_\varepsilon$ alors que l'agent AI₂₁ se met en état d'attente pendant cet intervalle de temps. Cet agent s'éteint à l'instant $t_{3_1} + \Delta_\infty$ parce que son état d'attente a atteint la période d'inactivité Δ_∞ . L'agent AI₁ est actif pendant les instants t_5 et $t_{3_1} + \Delta_\infty$, il interagit pendant cette période avec une société différente, autre que P_{t_2} (disponible en état d'attente) qui n'a pas été intégrée dans ce diagramme. Nous supposons l'existence d'un module réseau, représenté par un Agent Réseau (AR) qui détecte ses changements d'état pour informer le système concerné. Dès qu'un ensemble de nœuds est indisponible (panne, goulet d'étranglement, etc.), l'agent AR envoie la liste des ces nœuds à chaque agent ACI. Si un agent ACI est en train de patrouiller le RETM et devrait visiter un nœud déclaré indisponible, alors cet agent ACI doit négocier avec l'agent AO, de la même société, dans le but de réorganiser son itinéraire restant. Le nœud déclaré indisponible doit être déjà programmé dans l'itinéraire restant de l'agent ACI, par rapport à sa position courante à la réception du message. L'interaction d'une société avec un agent AI est illustrée par le même diagramme de séquences de la figure III.4.

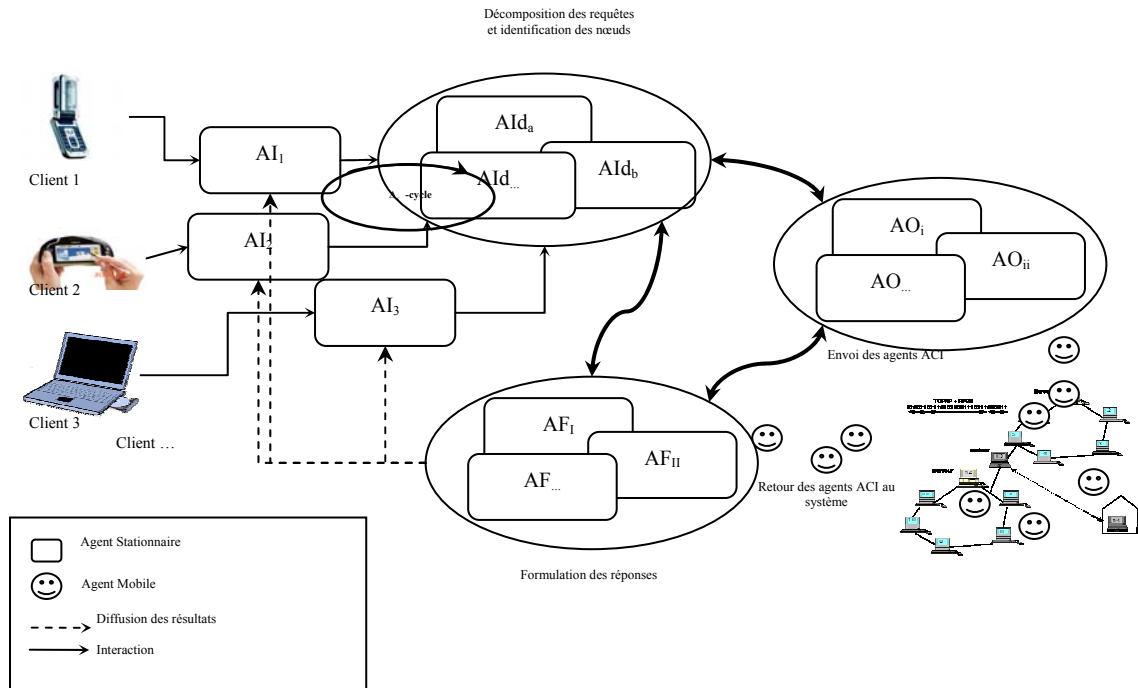


Figure III.2 - Architecture du système

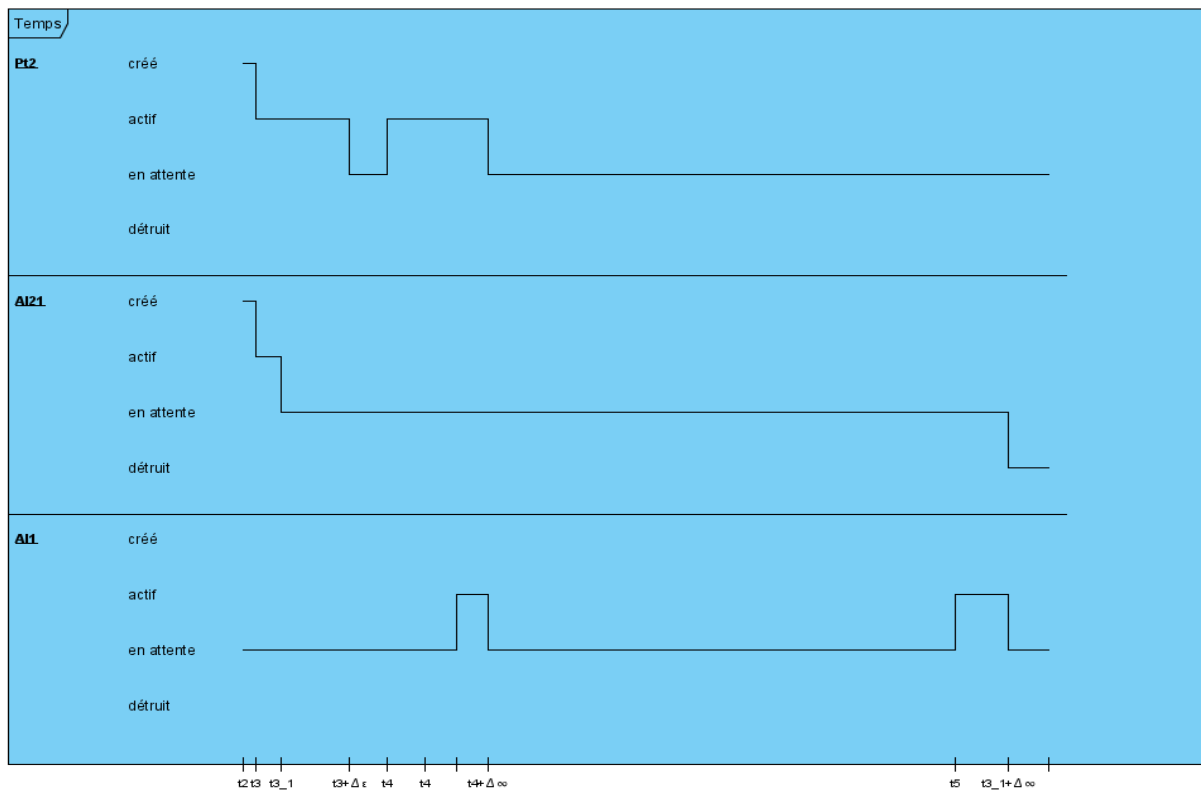


Figure III.3- Exemple de scénario de changement d'état d'une société

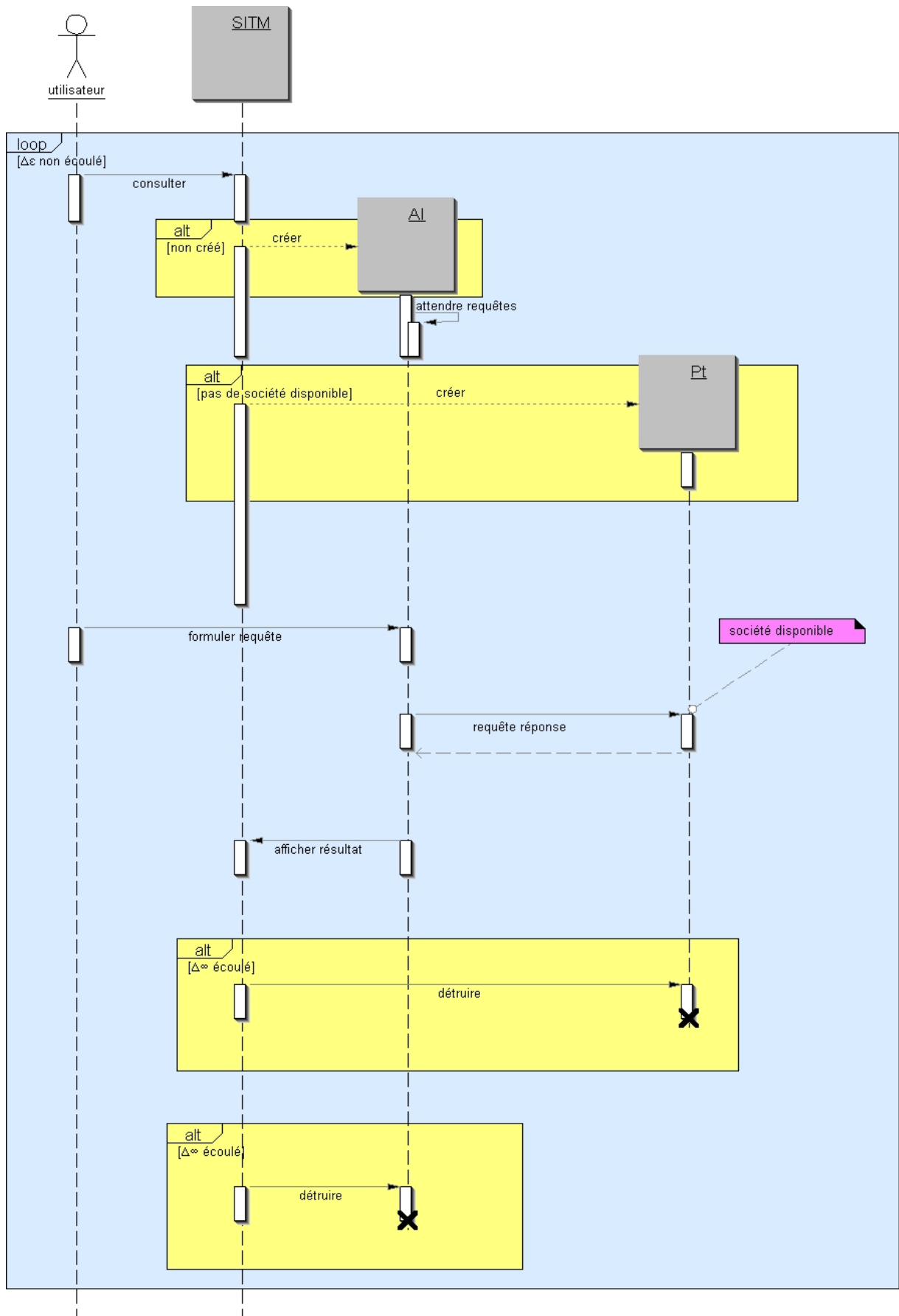


Figure III.4- Comportement global du SITM pendant une période de formulation

Le fonctionnement du système proposé nécessite l'intégration de bases de données locales que nous décrivons dans ce qui suit.

III.3.1 Les bases de données locales

Pour commercialiser ses données, un fournisseur d'information doit s'inscrire au système pour pouvoir enregistrer ou mettre à jour les services qu'il propose. Un service se caractérise par un coût, un temps de réponse et une taille de données et correspond à la réponse à une tâche par un fournisseur d'information. Un service est également caractérisé par un temps de pertinence qui permet de sauvegarder l'information localement pour un certain temps dans le but de réduire, si c'est possible, la transmission des données. Le fonctionnement du système nécessite donc l'existence de deux bases de données locales :

- La BDI (Base de Données d'Inscription): chaque entrepreneur, voulant proposer ses services à travers notre système, doit obligatoirement s'inscrire à cette base en enregistrant tous les services qu'il propose. Un fournisseur doit donc inscrire le label de chaque service proposé, sa référence, le temps de réponse estimé, le coût et la taille des données correspondantes. Il doit également mentionner l'adresse de son ou de ses serveurs. Un même service (même label) peut être donc proposé par plusieurs fournisseurs avec des coûts, des temps de réponse et des tailles différents ;
- La BDA (Base de Données d'Archivage) : Cette base joue le rôle de « zone tampon » pour contenir les données statiques à un certain degré. En d'autres termes, nous avons classé les données distribuées sur le RETM selon leur degré de stabilité grâce à une méthode utilisant les indicateurs de classification. Le but de cette méthode est d'éviter la recherche redondante des mêmes données non encore modifiées, à court et à long terme. La méthode de classification des données ainsi que la mise en œuvre et l'utilisation de la BDA sont expliqués en détail dans le paragraphe III.5.

L'organisation du SITM, qui intègre les bases de données locales décrites ci-dessus, est illustrée par le diagramme de classes de la Figure III.5.

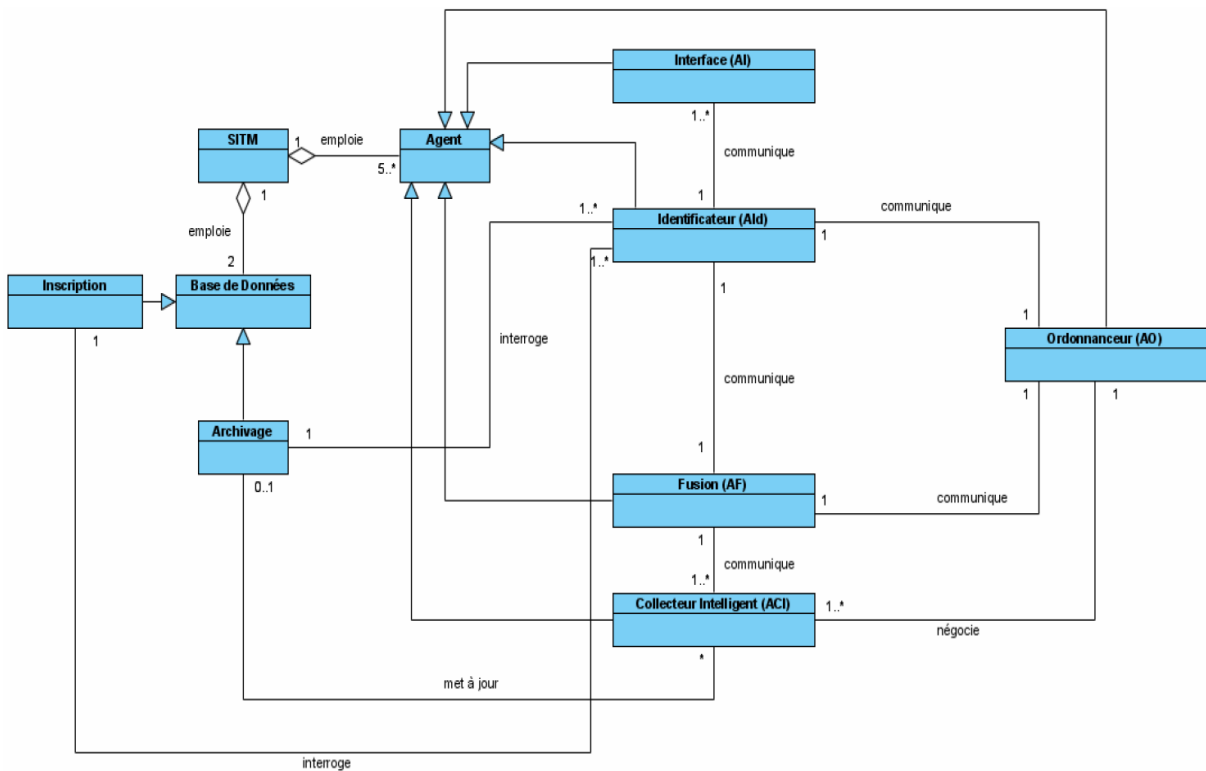


Figure III.5- Organisation du SITM

III.3.2 Comportements des agents dans le système

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

III.3.2.1 Agent Interface (AI)

Cet agent interagit avec l'utilisateur du système en lui permettant d'une part d'exprimer et d'émettre sa requête et d'autre part de choisir la forme de réponse appropriée à sa demande. Dès qu'un utilisateur accède au système, un agent AI est créé en temps réel pour l'aider et l'assister à formuler sa requête. Le même agent peut assurer cette tâche plusieurs fois de suite sur le même support tant que sa période d'inactivité n'est pas encore écoulée. Sinon, le système le détruit et un autre agent AI est créé au besoin et ainsi de suite. Une requête utilisateur formulée via un agent AI, est envoyée à un agent identificateur disponible, associé à une société disponible préalablement créée et tenue en état d'attente. S'il n'existe encore pas d'agent IdA disponible dans le système, un nouvel agent AId est automatiquement créé, au sein d'une nouvelle société d'agents. Dès qu'un agent AI 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 AI est illustré par le diagramme d'activité de la Figure III.6.

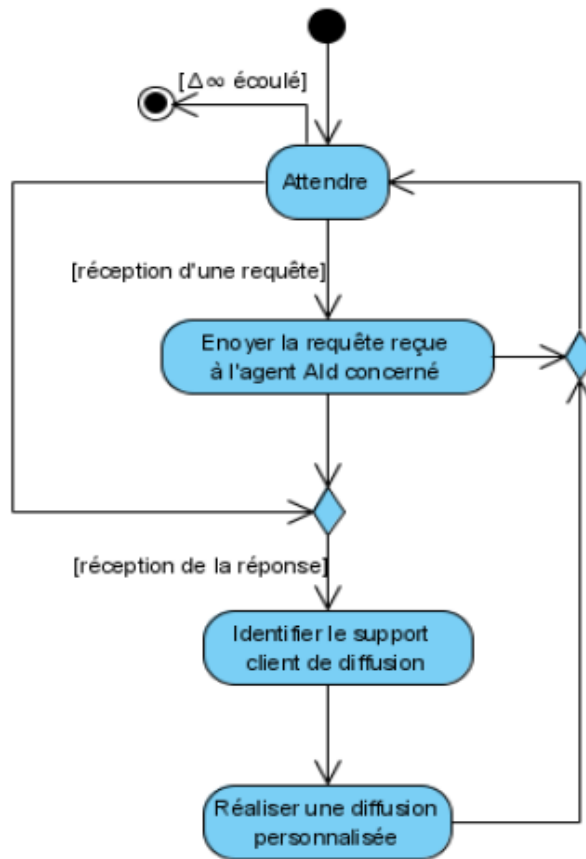


Figure III.6- Comportement de l'agent AI

III.3.2.2 Agent Identificateur (AId)

Un agent AId disponible peut recevoir un ensemble de requêtes Δ_e – simultanées et les décompose en un ensemble de tâches indépendantes en identifiant les éventuelles similarités afin d'éviter une recherche redondante des données. La procédure de décomposition engendre l'identification des nœuds qui proposent des services aux tâches identifiées. Ensuite, l'agent AId doit se conformer au protocole de l'archivage dynamique qui peut alléger le travail de l'agent AO en réduisant le transfert des données sur le RETM. Ce protocole sera détaillé au paragraphe III.5. Enfin, l'agent AId transmet les données générées à l'agent AO, issu de la même société, qui doit optimiser le choix des fournisseurs, selon les contraintes du système. Le comportement d'un agent AId est illustré par le diagramme d'activité de la Figure III.7.

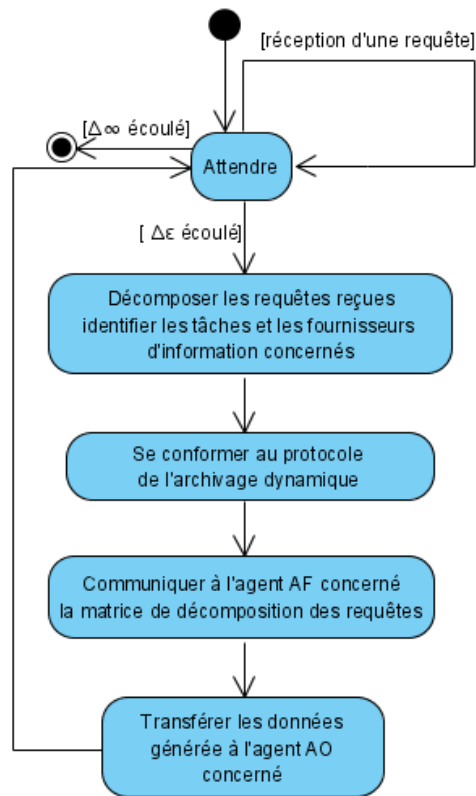


Figure III.7- Comportement de l'agent AId

III.3.2.3 Agent Ordonnanceur (AO)

Dès sa création, un agent AO calcule un nombre effectif d'agents ACI qu'il crée à son tour et attribue ensuite à chacun un Plan De Route Initial (PDRI) qu'il met à jour à chaque fois que l'état du réseau varie considérablement. Si l'agent AId, issu de la même société, lui procure un certain nombre de données alors l'agent AO procède par une étape d'optimisation. Les données reçues correspondent à un ensemble de services demandés ainsi qu'à l'ensemble des fournisseurs d'informations qui les proposent. Cependant, plusieurs nœuds peuvent proposer le même service avec différents coûts, temps de réponses et tailles des données. Un agent AO doit optimiser le choix des nœuds pour répondre aux services demandés, en minimisant le coût total et le temps de réponse pour respecter les temps de réponse au plus tard (contrainte de données). L'ensemble définitif sélectionné de nœuds construit pour chaque agent ACI, son Plan de Route Définitif (PDRD). Une perturbation réseau déclenche un processus de négociation entre l'agent AO et les agents ACIs correspondants pour reconstruire les PDRDs restants. Le comportement d'un agent AO est illustré par le diagramme d'activité de la Figure III.8 et le protocole de négociation adopté est détaillé dans le chapitre IV.

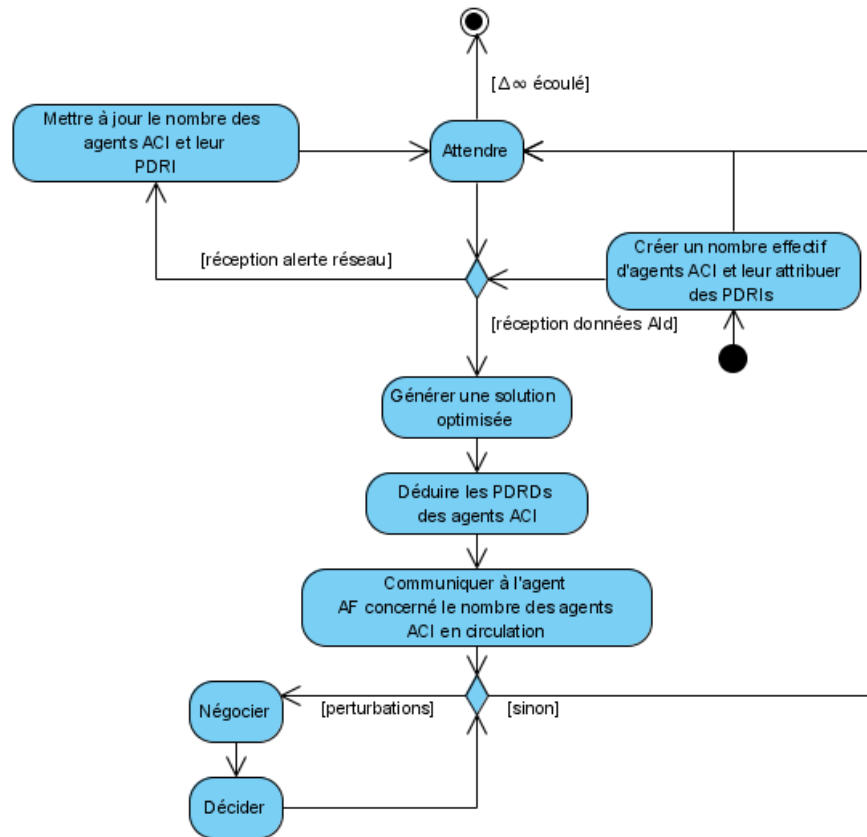


Figure III.8- Comportement de l'agent AO

III.3.2.4 Agent Collecteur Intelligent (ACI)

C'est un agent logiciel mobile capable de se déplacer intelligemment d'un nœud à l'autre à travers un réseau pour accomplir son travail et revient en fin de parcours à son nœud d'origine appelé nœud hôte et noté par H. La taille de l'information qu'un agent ACI doit collecter ne doit pas dépasser un seuil de capacité afin d'éviter la surcharge de données. L'agent AO doit donc prendre en considération cet aspect lors de la génération de la solution optimisée pour déduire les PDRDs. Ainsi, dès la réception de son PDRD, un agent ACI patrouille le réseau en respectant le plan qui lui a été attribué. Le comportement intelligent d'un agent ACI s'explique par sa capacité de changer son itinéraire en temps réel lors de son déplacement selon les disponibilités des nœuds du réseau suite à une perturbation. Pour cela il doit interagir avec l'agent AO correspondant suivant le protocole de négociation adopté (§IV.4). Un agent ACI doit proposer un plan de reconstruction de son itinéraire restant à partir de sa position courante, en se basant sur ses priorités, préférences et contraintes. L'agent AO, de son côté, doit négocier avec tous les agents ACI actifs correspondants pour réattribuer de nouveaux nœuds aux services dont les nœuds qui lui ont déjà été attribués, sont atteints par la

perturbation. Lorsque les agents ACI réintègrent le système, ils transmettent les données collectées à l'agent fusion correspondant et mettent à jour la BDA. Le comportement d'un agent ACI est illustré par le diagramme d'activité de la Figure III.9.

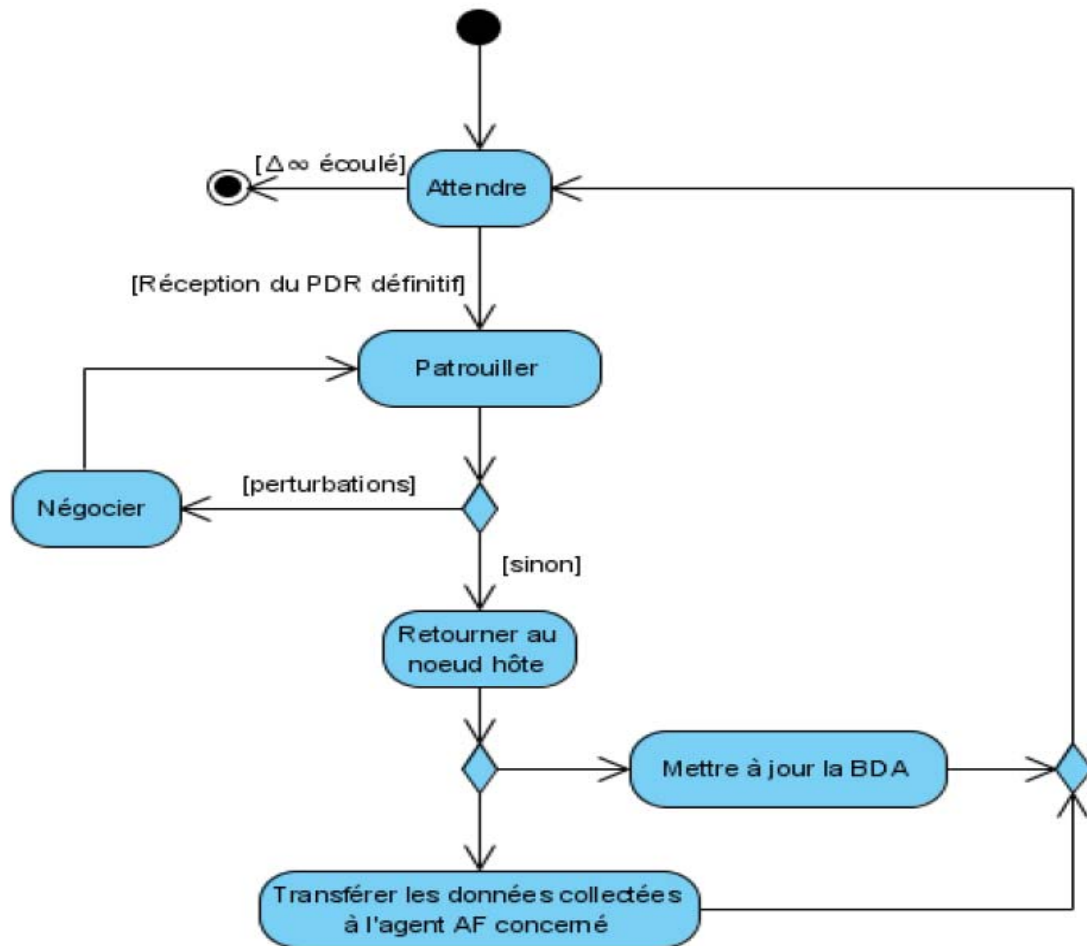


Figure III.9- Comportement de l'agent ACI

III.3.2.5 Agent Fusion (AF)

Cet agent doit fusionner correctement les données afin de composer les réponses aux requêtes simultanées. La procédure de fusion nécessite une communication entre les agents AF, AO et AId et progresse selon la disponibilité des données rassemblées. Chaque nouveau composant de réponse doit être complémentaire aux données qui ont déjà été rassemblées. Un agent AF doit d'abord recevoir le nombre des agents ACI et leur PDRD de la part de l'agent AO correspondant afin de déclencher la procédure de fusion des réponses dès la réception de toutes les données nécessaires pour cette fusion, qui correspondent à tous les bouts de réponses collectés par les agents ACI. Pour déclencher la procédure de fusion, l'agent AF doit également recevoir la matrice de décomposition des requêtes générée par l'agent AId

correspondant. Une fois les réponses formées, elles sont envoyées aux agents AI correspondants. Le comportement de l'agent AF est illustré par le diagramme d'activité de la Figure III.10.

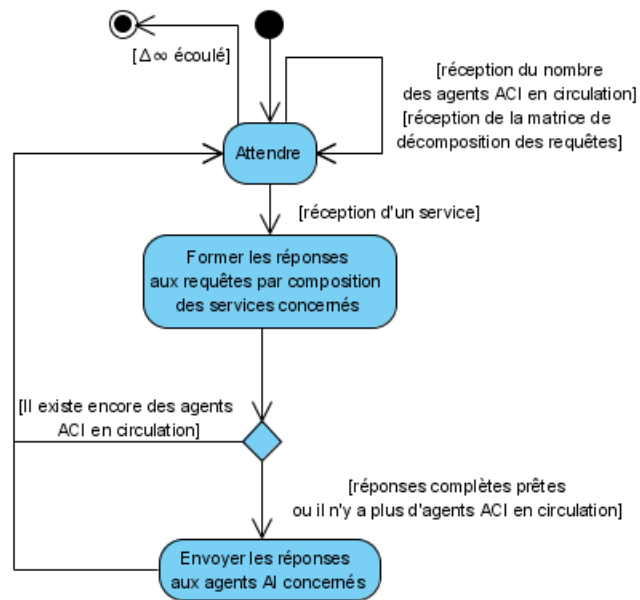


Figure III.10- Comportement de l'agent AF

Le comportement d'une société, en interaction avec les agents AI, composée d'un agent AI_d, un agent AO, un agent AF et un certain nombre d'agents ACI, est illustré par le diagramme de séquence de la Figure III.11.

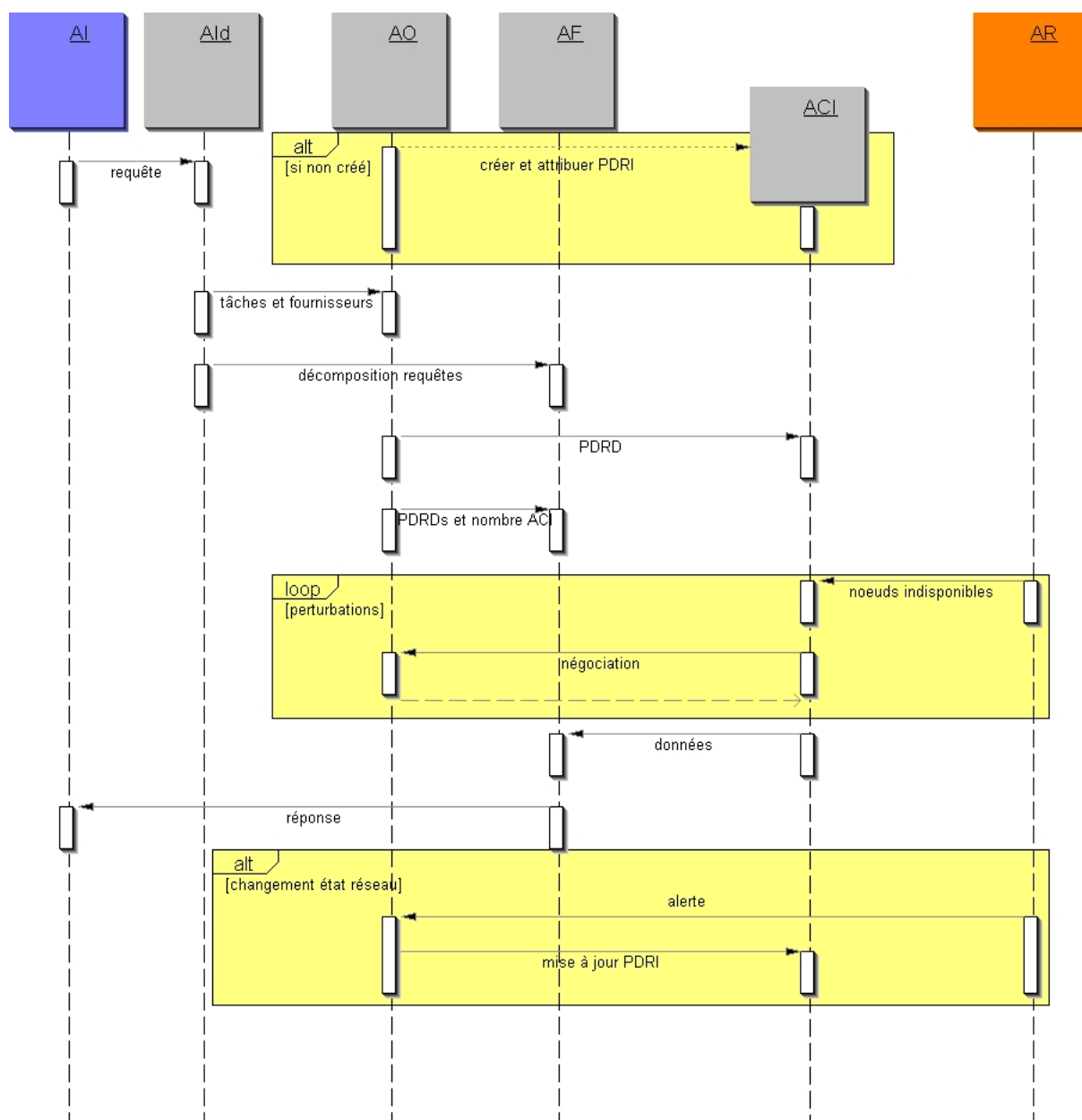


Figure III.11- Comportement d’une société d’agents

Les données nécessaires pour répondre aux services demandés sont réparties sur le RETM et leur collecte est à la charge des agents ACI dont les itinéraires initiaux (PDRI) et définitifs (PDRD) sont construits par un agent AO suivant une approche d’optimisation que nous présentons dans le paragraphe qui suit.

III.4 Approches et algorithmes d’optimisation proposés

Le comportement de l’agent AO, illustré par le diagramme d’activité de la Figure III.8, permet d’élaborer les PDRIs des agents ACI ainsi que la déduction des PDRDs grâce à deux approches d’optimisation. Ces approches se basent sur des algorithmes d’optimisation

efficaces que nous présentons dans ce paragraphe. L'élaboration des PDRIs représente une première étape d'optimisation qui permet de préparer une sorte de plan de route pour chaque agent ACI dès sa création. Chaque agent ACI sera donc désigné « patrouilleur » sur une partie du réseau qu'il visitera en totalité, partiellement ou pas du tout selon la solution optimisée générée par l'agent AO issue de la deuxième étape d'optimisation. Cette dernière permet de déduire les PDRDs qui représentent les itinéraires concrets des agents ACI.

III.4.1 Elaboration des PDRIs des agents ACI

Les algorithmes à coûts effectifs de planification des parcours d'agents mobiles « Cost Effective Mobile Agent Planning » (CE-MAP), suggérés par [Baek et al., 01], sont les plus appropriés au PRCS. En fait, un algorithme dynamique proposé, appelé BYKY2, optimise le nombre d'AMs et réduit le temps d'exécution selon l'état du réseau. Dans des travaux antérieurs [Zgaya et al., 05a] [Zgaya et al., 06c] [Zgaya et al., 07b], nous nous sommes inspirés de cette approche en considérant les variations d'état des entités mobiles par les données transportées. L'approche proposée se base sur les conditions suivantes :

- La collecte des données sur un nœud visité nécessite un certain temps d'exécution. Lors de cette étape d'optimisation, nous supposons que la taille des données à collecter sur un nœud de réseau est égale à la moyenne de la taille totale de toutes les données stockées sur ce nœud,
- Initialement, un agent ACI n'est pas totalement vide puisqu'il contient une quantité initiale de données Q_0 qui correspond à son code,
- Le débit minimal entre chaque paire de nœuds dans le réseau est disponible par un module d'administration existant, représenté par l'agent AR (§III.3),
- L'information peut avoir un aspect multimédia, ainsi nous supposons que la transmission d'une quantité de données d'un nœud à l'autre dépend de l'état courant du réseau.

III.4.1.1 Description

Le problème des PDRIs peut être décrit comme suit: les agents ACI sont créés et lancés dans un RETM à partir du nœud hôte H. Les autres nœuds du réseau représentent les fournisseurs d'information là où un agent ACI peut se déplacer pour collecter les données correspondantes aux services demandés (Figure III.12). Le but est de trouver un nombre minimum d'agents ACI pour explorer la totalité du RETM en prenant en considération son

état. Nous présentons d'abord quelques définitions des variables déclarées dans le Tableau III-2.

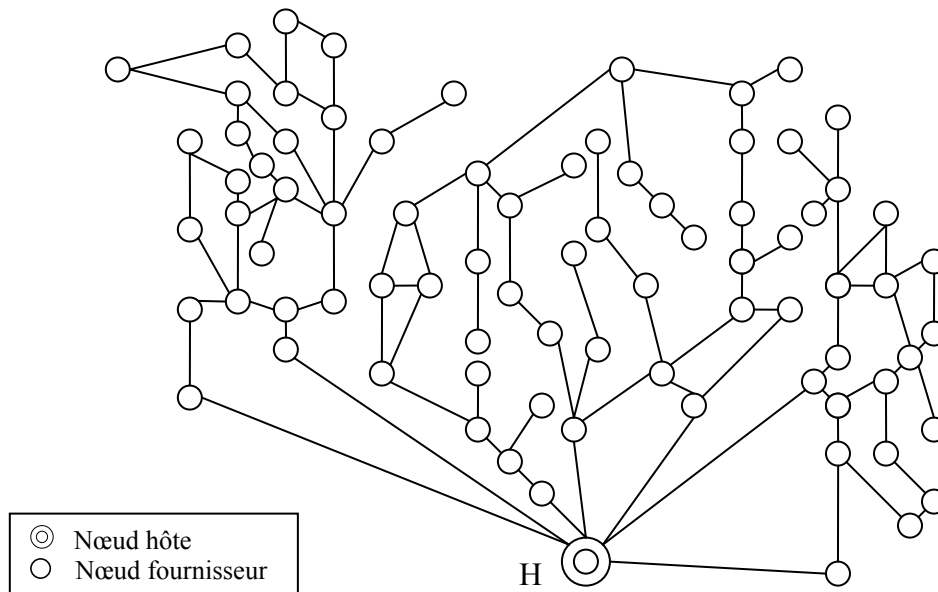


Figure III.12- Exemple de RETM

Variable	Description
m	Nombre d'agents ACI
ACI_1, \dots, ACI_m	Identificateurs des agents ACI
Wk_i	Séquence de nœuds représentant le PDRI d'un agent ACI_i avec $Wk_i = (S_{i_1}, \dots, S_{i_p})$ et $1 \leq p \leq J$
$T(Wk_i)$	Le temps total pour parcourir Wk_i
$Qte_{k,u}$	Taille des données transportées par un agent ACI_k jusqu'au nœud S_u inclus
$Tr(Qte_{k,u}, S_u, S_v)$	Temps de transmission de $Qte_{k,u}$ du nœud S_u au nœud S_v
CT_i	Temps d'exécution sur le nœud S_i
Qt_i	Quantité de données sur le nœud S_i
$d(S_i, S_j)$	Débit entre les nœuds S_i et S_j

Tableau III-2– Notations

Définition 1: Qt_j (Quantité de données sur le nœud S_j) est la quantité moyenne des données à extraire à partir de S_j .

T_i représente la tâche qui correspond au service proposé par S_j . Nous rappelons que $P_{i,j}$ représente le temps d'exécution de la tâche T_i utilisant les ressources du nœud S_j et $Q_{i,j}$

représente la taille des données à collecter du nœud S_j correspondant au service de la tâche T_i selon la table de service correspondante (Table III-1). Donc:

$$Qt_j = \frac{\sum_{i=1}^I a_{i,j} Q_{i,j}}{\sum_{i=1}^I a_{i,j}} \quad (1) \quad CT_j = \frac{\sum_{i=1}^I a_{i,j} P_{i,j}}{\sum_{i=1}^I a_{i,j}} \quad (2)$$

$a_{i,j}$ est un booléen tel que: $a_{i,j}=1$ si le nœud S_j propose un service pour la tâche T_i et $a_{i,j}=0$ sinon (suivant la table de service correspondante).

Définition 2: $Qte_{k,u}$ (Quantité de données transportée jusqu'à S_u inclus par ACI_k) est la quantité de données collectée et transportée par ACI_k au cours de sa route, jusqu'au nœud S_u .

$Qte_{k,u}$ est calculé comme suit:

$$Qte_{k,u} = Q_0 + \sum_{r=1}^u Qt_{k_r} \quad (3)$$

Définition 3: $Tr(Qte_{k,u}, S_u, S_v)$ (Temps de transmission) est le temps nécessaire pour la migration de l'agent ACI_k du nœud S_u au nœud S_v transportant la quantité de données $Qte_{k,u}$.

$Tr(Qte_{k,u}, S_u, S_v)$ est calculé comme suit:

$$Tr(Qte_{k,u}, S_u, S_v) = \frac{Qte_{k,u}}{d(S_u, S_v)} \quad (4)$$

Définition 4 : $T(Wk_k)$ (Le temps du parcours Wk_k) est le temps nécessaire pour que l'agent ACI_k visite la séquence des nœuds du réseau $(S_{k_1}, \dots, S_{k_p})$ avec $1 \leq p \leq J$ avant de retourner au nœud Hôte.

En se basant sur le Tableau III-3, $T(Wk_k)$ est calculé comme suit:

$$T(Wk_k) = T_{\text{départ}} + T_{\text{voyage}} + T_{\text{retour}} \quad (5)$$

p	Wk _k	T _{départ}	T _{retour}	T _{voyage}
1	(S_{k_1})	$Tr(Q_0, H, S_{k_1})$	$Tr(Qte_{k,k_p}, S_{k_p}, H)$	CT_{k_1}
[1..J]	(S_{k_1}, \dots, S_{k_p})			X_k

Tableau III-3- Temps de parcours

Avec: $X_k = \sum_{i=1}^p CT_{k_i} + \sum_{i=1}^{p-1} Tr(Qte_{k,k_i}, S_{k_i}, S_{k_{i+1}}) \quad (6)$

III.4.1.2 Algorithme de Construction des Schémas de Parcours (CoSPa)

La proposition d'une solution effective de construction des schémas de parcours des agents ACI nécessite l'existence d'un module administratif qui fournit des informations sur l'état du réseau et notamment les valeurs de débits entre toute paire de nœuds appartenant au RETM. Le but est de trouver un nombre minimal d'agents ACI pour explorer tout le réseau, en réduisant leur temps de navigation.

Il est clair que l'expédition d'un agent ACI vers chaque nœud du réseau ($m=J$) nous assure le meilleur temps de réponse car, dans ce cas, les agents sont lancés en même temps à partir du nœud hôte vers tous les autres nœuds du réseau. Ainsi, chaque agent opère de son côté et retourne au nœud hôte dès que son travail se termine. Dans ce cas, le meilleur temps de réponse est conservé comme un seuil noté par δ , il correspond au temps de parcours du dernier agent qui retourne au système. A partir de ce seuil, nous partitionnons le réseau en différentes parties par rassemblement des nœuds tel que le temps de parcours de chaque partie ne dépasse pas le seuil δ . Le but ici est de respecter δ en réduisant le nombre d'agents ACI ($m \leq J$) pour réduire l'utilisation des ressources du système.

Pour construire un PDRI, cet algorithme cherche à chaque fois le prochain nœud à visiter, à partir de la position courante de l'agent ACI, en recalculant le nouveau temps de parcours. Un nœud est sélectionné si le temps nécessaire pour effectuer l'itinéraire en cours de construction ne dépasse pas le seuil δ . Sinon, un PDRI est prêt à être affecté à un agent ACI et l'algorithme prend fin si chaque nœud fait partie d'un PDRI déjà attribué. L'algorithme distribue donc tous les nœuds du réseau à un ensemble m d'agents ACI, chaque agent devient ainsi le patrouilleur potentiel de la partie du réseau qui lui a été attribuée (Algorithme CoSPa).

CoSPa utilise les notations suivantes :

- $S_{a_i} \in R_a$ signifie que le nœud S_{a_i} fait partie de la séquence de nœuds R_a ;
- $PDR_k = PDR_k \oplus S_j$ signifie l'ajout du nœud S_j à la fin du plan de route PDR_k ;
- $PDR_k = PDR_k \ominus S_j$ signifie la suppression du nœud S_j de la fin du plan de route PDR_k ;

Algorithme CoSPa

1. Calculer le temps de parcours nécessaire pour visiter chaque nœud:

Pour $\forall S_j \in Y$ faire:

| $Wk_j = S_j$
| Calculer $T(Wk_j)$

2. Classer les nœuds par ordre décroissant selon le temps de parcours :

Soit $R_d = (S_{d_1}, \dots, S_{d_n})$ la suite des nœuds classés par ordre décroissant selon $T(Wk_j) = S_j$

3. Fixer le seuil $\delta = T(Wk_{d_1} = S_{d_1})$

4. Partitionner le réseau en différentes parties en rassemblant les nœuds tels que le temps de parcours de chaque partie ne dépasse pas le seuil δ . Le nombre de partitions correspond au nombre m effectif d'agents ACI, initialisé à 0:

Pour $\forall S_j \in Y$ faire:

| $Wk_j = \emptyset$

| Si $\exists S_{d_k} \langle R_d$ "non affecté" / k est minimum alors faire

| Sélectionner S_{d_k}

| Incrémenter m

| $Wk_j = Wk_j \oplus S_{d_k}$

| Marquer S_{d_k} comme "affecté"

| Sinon Fin du programme

| Tant que vrai faire

| Classer les nœuds par ordre décroissant selon $d(S_{d_k}, S_{d_l}) / 1 < l \leq J$ et S_{d_l} "non affecté"

| soit $R_e = (S_{e_1}, \dots, S_{e_g})$ la séquence résultat

| $preced_k = k$

| Pour $\forall S_{e_y} \langle R_e / y$ est minimum faire:

| $Wk_j = Wk_j \oplus S_{e_y}$

| Si $T(Wk_j) \leq \delta$ faire

| Marquer S_{e_y} comme "affecté"

| Trouver k / d_k est égal à e_y

| Interrompre la boucle pour

| Sinon $Wk_j = Wk_j \ominus S_{e_y}$

| Si $preced_k$ est égal à k alors interrompre la boucle tant que

L'approche d'optimisation adoptée dans le prochain paragraphe permet de sélectionner un sous-ensemble Y'_t de l'ensemble de tous les nœuds inscrits Y pour répondre aux services. Le but est d'optimiser la gestion du flux d'information en cas d'acquisition de requêtes, ce qui implique la recherche d'un nombre utile et optimal d'agents ACI à partir de l'ensemble des agents ACIs préparés par CoSPa. Cet ensemble peut être réduit lorsque tous les nœuds qui composent le PDRI d'un agent ACI, n'ont pas été sélectionnés lors de la deuxième phase d'optimisation. Les PDRDs seront donc déduits à partir des PDRIs afin de collecter les données nécessaires pour satisfaire, le plus rapidement possible et avec des coûts raisonnables, les requêtes clients à un instant t . Ce deuxième niveau d'optimisation se base sur une approche évolutionnaire que nous décrivons dans le prochain paragraphe. Soit m' le nouveau nombre d'agents ACI. Nous avons également $J'=|Y'_t|$ le nouveau nombre de nœuds ; ainsi : $m' \leq m$, $J' \leq J$ et $Y'_t \subseteq Y$.

III.4.2 Optimisation et Composition des flux de services par une approche évolutionnaire : calcul des PDRDs

III.4.2.1 Présentation

Dans cette partie, nous considérons les importantes caractéristiques des algorithmes évolutionnistes 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 tâche (sous-requête) doit être traitée par un seul serveur d'information sélectionné à partir de l'ensemble des serveurs qui proposent le service correspondant. 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 ;

- Les fonctions de fitness évaluent les solutions selon deux critères : le coût et le temps de réponse.

III.4.2.2 La méthode évolutionnaire proposée

Le choix d'une représentation appropriée à une solution possible du problème global est fondamental pour le succès des applications des algorithmes évolutionnaires. Dans des travaux antérieurs [Zgaya et al., 05b], nous avons conçu un codage efficace pour une solution possible, en respectant les contraintes du problème traité. Nous avons donc proposé une représentation flexible du chromosome, appelée Représentation Flexible d'Assignment des Tâches « Flexible Tasks Assignment Representation » (FeTAR).

Le chromosome est représenté par une matrice $CH(I' \times J')$ dont les lignes représentent l'ensemble des tâches indépendantes (les services demandés), qui composent globalement les requêtes simultanées Γ'_t et les colonnes représentent les différents nœuds identifiés Y'_t . Chaque élément de la matrice indique l'attribution d'un nœud $S_{c_j} \in Y'_t$ à une tâche $T_{c_i} \in \Gamma'_t$ avec $1 \leq i \leq I'$ et $1 \leq j \leq J'$ comme suit:

$$CH[c_i, c_j] = \begin{cases} 1 & \text{Si } S_{c_j} \text{ est affecté à } T_{c_i} \\ * & \text{Si } S_{c_j} \text{ peut être affecté à } T_{c_i} \\ X & \text{Si } S_{c_j} \text{ ne peut pas être affecté à } T_{c_i} \end{cases}$$

Un nœud ne peut pas être affecté à une tâche s'il ne propose pas de service correspondant. Sinon, la solution instance de FeTAR peut affecter le serveur à la tâche selon le résultat généré. La Table III-2 montre un exemple de solution instance de FeTAR. Par exemple, le service correspondant à la tâche T_{32} , est proposé par plusieurs fournisseurs d'information: S_5, S_{18}, S_1, S_{14} et S_{10} mais c'est le nœud S_{14} qui a été choisi pour fournir le service. Cependant, pour le service correspondant à la tâche T_6 , le serveur S_{50} est l'unique fournisseur d'information possible alors le système subit l'affectation parce qu'il n'a pas d'autre choix.

	S ₅	S ₁₈	S ₁	S ₁₄	S ₂₀₁	S ₅₀	S ₈	S ₉	S ₁₀	S ₃
T ₁₂	*	*	*	1	*	*	*	*	*	*
T ₃	*	*	*	*	X	*	*	*	*	1
T ₅	1	*	*	*	X	*	*	*	*	*
T ₆	X	X	X	X	X	1	X	X	X	X
T ₂₇	X	1	*	X	X	X	*	*	X	*
T ₁₀	X	X	*	*	X	X	X	*	1	X
T ₃₂	*	*	*	1	X	X	X	X	*	X
T ₁₃	*	*	*	*	X	X	*	*	*	1

Table III-2- Exemple d'une instance FeTAR

L'aspect flexible d'une solution FeTAR permet de modifier l'affectation des nœuds aux tâches grâce aux opérateurs génétiques présentés dans ce qui suit.

III.4.2.3 Les opérateurs génétiques adoptés

III.4.2.3.1 L'opérateur 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 **CrossFeTAR**) :

Algorithme CrossFeTAR
1. Sélectionner aléatoirement 2 chromosomes parents et un serveur :
P_1, P_2 et $S_{c_j} \in Y_t'$
2. Procéder par les affectations des nœuds des chromosomes enfants E_1 et E_2 :
Pour $\forall S_{c_k} \in Y_t'$ faire:
Si j est différent de k alors faire
Les affectations des tâches de S_{c_k} dans E_1 (resp. E_2) doivent correspondre aux mêmes affectations de S_{c_k} dans P_2 (resp. P_1)
Sinon faire
Les affectations des tâches de S_{c_j} dans E_1 (resp. E_2) doivent correspondre aux mêmes affectations de S_{c_j} dans P_1 (resp. P_2)

III.4.2.3.2 Exemple d'application pour l'opérateur de croisement

L'opérateur de croisement est illustré par l'exemple ci-dessous:

Etape 1:

P_1 , P_2 et S_2 représentent respectivement deux chromosomes et un nœud choisis aléatoirement. S_2 correspond à la deuxième colonne de la matrice qui représente le chromosome:

P_1	S_{16}	S_2	S_{33}	S_4
T_{11}	X	1	*	*
T_7	*	*	1	*
T_{12}	1	X	*	*
T_4	1	X	X	X
T_{55}	*	*	1	*

P_2	S_{16}	S_2	S_{33}	S_4
T_{11}	X	*	*	1
T_7	*	*	1	*
T_{12}	*	X	1	*
T_4	1	X	X	X
T_{55}	1	*	*	*

Etape 2:

Les affectations de S_2 dans E_1 (resp. E_2) doivent correspondre aux mêmes affectations de S_2 dans P_1 (resp. P_2)

E_1	S_{16}	S_2	S_{33}	S_4
T_{11}		1		
T_7		*		
T_{12}		X		
T_4		X		
T_{55}		*		

E_2	S_{16}	S_2	S_{33}	S_4
T_{11}		*		
T_7		*		
T_{12}		X		
T_4		X		
T_{55}		*		

Les affectations des autres nœuds, à savoir S_{16} , S_{33} et S_4 dans E_1 (resp. E_2) doivent correspondre aux mêmes affectations de S_{16} , S_{33} et S_4 dans P_2 (resp. P_1)

E_1	S_{16}	S_2	S_{33}	S_4
T_{11}	X	1	*	1
T_7	*	*	1	*
T_{12}	*	X	1	*
T_4	1	X	X	X
T_{55}	1	*	*	*

E_2	S_{16}	S_2	S_{33}	S_4
T_{11}	X	*	*	*
T_7	*	*	1	*
T_{12}	1	X	*	*
T_4	1	X	X	X
T_{55}	*	*	1	*

Une solution viable affecte un seul nœud à chaque tâche. Dans ce cas, E_1 et E_2 ne sont pas des solutions viables car l'Enfant1 (E_1) affecte la tâche T_{11} deux fois par les serveurs S_2 et S_4 et l'Enfant2 (E_2) n'affecte pas du tout la tâche T_{11} . Il faut donc les corriger. Nous proposons donc un opérateur de correction pour les solutions non viables.

III.4.2.3.3 Opérateur de correction adopté

L'opérateur de correction est nécessaire pour transformer une solution non viable en une solution viable. Une solution non viable n'affecte pas une tâche donnée ou l'affecte plusieurs fois par différents nœuds. Pour un chromosome Ch , nous proposons un opérateur de correction décrit par l'algorithme **CorrectFeTAR** suivant :

Algorithme CorrectFeTAR	
Pour $\forall T_{c_i} \in \Gamma_t'$ faire:	
	Initialiser à 0 le tableau <i>TabIndAffectations</i> de dimension maximale J'
	Initialiser l'indice k à 0 ;
	Pour $\forall S_{c_j} \in Y_t'$ faire:
	Si $Ch[c_i, c_j]$ est égal à 1 alors faire
	Incréments k
	$TabIndAffectations[k] = j$
	Si k est égal à 0 alors faire
	Tirer au hasard un indice p avec $1 \leq p \leq J' / Ch[c_i, c_p]$ est égal à *
	$Ch[c_i, c_p] = 1$
	Sinon Si $k > 1$ alors faire
	Tirer au hasard un indice p avec $1 \leq p \leq J' / Ch[c_i, c_p]$ est égal à 1
	Pour $\forall S_{c_x} \in Y_t'$ faire:
	Si $Ch[c_i, c_x]$ est égal à 1 et x différent de p alors faire
	$Ch[c_i, c_x] = *$

III.4.2.3.4 Exemple d'application pour l'opérateur de correction

Les solutions E_1 et E_2 de l'exemple du paragraphe III.4.2.3.2 nécessitent une étape de correction car E_1 affecte T_{11} deux fois par S_2 et S_4 et E_2 n'affecte pas T_{11} .

Enfant 1

E ₁	S ₁₆	S ₂	S ₃₃	S ₄
T ₁₁	X	1	*	1
T ₇	*	*	1	*
T ₁₂	*	X	1	*
T ₄	1	X	X	X
T ₅₅	1	*	*	*

Enfant 2

E ₂	S ₁₆	S ₂	S ₃₃	S ₄
T ₁₁	X	*	*	*
T ₇	*	*	1	*
T ₁₂	1	X	*	*
T ₄	1	X	X	X
T ₅₅	*	*	1	*

L'opérateur de correction, appliqué à E₁, donne le résultat suivant: E₁ sera au hasard E₁₁ ou E₁₂:

E ₁₁	S ₁₆	S ₂	S ₃₃	S ₄
T ₁₁	X	*	*	1
T ₇	*	*	1	*
T ₁₂	*	X	1	*
T ₄	1	X	X	X
T ₅₅	1	*	*	*

E ₁₂	S ₁₆	S ₂	S ₃₃	S ₄
T ₁₁	X	1	*	*
T ₇	*	*	1	*
T ₁₂	*	X	1	*
T ₄	1	X	X	X
T ₅₅	1	*	*	*

L'opérateur de correction, appliqué à E₂, donne le résultat suivant: E₂ sera au hasard E₂₁, E₂₂ ou E₂₃:

E ₂₁	S ₁₆	S ₂	S ₃₃	S ₄
T ₁₁	X	1	*	*
T ₇	*	*	1	*
T ₁₂	1	X	*	*
T ₄	1	X	X	X
T ₅₅	*	*	1	*

E ₂₂	S ₁₆	S ₂	S ₃₃	S ₄
T ₁₁	X	*	*	1
T ₇	*	*	1	*
T ₁₂	1	X	*	*
T ₄	1	X	X	X
T ₅₅	*	*	1	*

E ₂₃	S ₁₆	S ₂	S ₃₃	S ₄
T ₁₁	X	*	1	*
T ₇	*	*	1	*
T ₁₂	1	X	*	*
T ₄	1	X	X	X
T ₅₅	*	*	1	*

III.4.2.3.5 Opérateur de mutation contrôlée

La mutation représente un autre opérateur génétique important après le croisement. Cet opérateur introduit une certaine variation dans les gènes des individus de la population.

Typiquement, la mutation s'applique sur un chromosome pour en créer un nouveau modifié [Michalewicz, 92]. L'algorithme de mutation contrôlée que nous proposons est décrit par l'algorithme **MuteFeTAR**.

Algorithme MuteFeTAR	
1. Sélectionner aléatoirement un chromosome, une tâche et un serveur :	
$Ch, T_{c_i} \in \Gamma_t$ et $S_{c_j} \in Y_t$	
2. Si $Ch[c_i, c_j]$ est égal à * alors faire	
	trouver $j_1 / 1 \leq j_1 \leq J'$ et $Ch[c_i, c_{j_1}]$ égal à 1
	$Ch[c_i, c_{j_1}] = *$
	$Ch[c_i, c_j] = 1$
Sinon Si $Ch[c_i, c_j]$ est égal à 1 et $\exists j_1 / 1 \leq j_1 \leq J'$ et $Ch[c_i, c_{j_1}]$ égal à * alors faire	
	$Ch[c_i, c_{j_1}] = 1$
	$Ch[c_i, c_j] = *$

III.4.2.3.6 Exemple d'application pour l'opérateur de mutation

L'opérateur de mutation est illustré par l'exemple suivant:

Etape 1:

Le chromosome CH a été choisi aléatoirement, ainsi que la tâche T_{12} et le nœud S_{16} :

Chromosome

CH	S_{16}	S_2	S_{33}	S_4
T_{11}	X	1	*	*
T_7	*	*	1	*
T_{12}	1	X	*	*
T_4	1	X	X	X
T_{55}	*	*	1	*

Etape 2:

$CH[c_3, c_1] = 1$ et $\exists j_1$ (3 or 4) avec $CH[c_3, c_{j_1}] = *$ donc il faut affecter * à $CH[c_3, c_1]$, choisir au hasard j_1 (3 par exemple) et affecter 1 à $CH[c_3, c_3]$. L'ancien chromosome CH transformé par l'opération de mutation devient CH' :

Nouveau Chromosome

CH'	S ₁₆	S ₂	S ₃₃	S ₄
T ₁₁	X	1	*	*
T ₇	*	*	1	*
T ₁₂	*	X	1	*
T ₄	1	X	X	X
T ₅₅	*	*	1	*

III.4.2.4 Génération des parcours définitifs (PDRDs)

Suivant l'instance générée CH du FeTAR, les équations (1) et (2) (§III.4.1.1) se transforment en:

$$Q_{t_{c_j}} = \sum_{i=1}^{I'} (a_{c_i, c_j} \times Q_{c_i, c_j}) \quad (1')$$

$$CT_{c_j} = \sum_{i=1}^{I'} (a_{c_i, c_j} \times P_{c_i, c_j}) \quad (2')$$

Tel que a_{c_i, c_j} une valeur booléenne telle que si $CH[c_i, c_j] = 1$ ($1 \leq i \leq I'$ et $1 \leq j \leq J'$) alors $a_{c_i, c_j} = 1$ et $a_{c_i, c_j} = 0$ sinon. Les temps de parcours pour les PDRD sont donc déduits en utilisant les équations (1'), (2'), (3), (4), (5) et (6) (§III.4.1.1). Un PDRD d'un agent ACI correspond à la somme des temps d'exécution de toutes les tâches qu'il exécute sur tous les serveurs qu'il visite, à laquelle nous additionnons le temps maximum moyen estimé de parcours des agents ACI qu'on note par γ avec $\gamma = \delta - \sum_{j=1}^J CT_j / J$. Nous rappelons que δ est le seuil qui correspond au temps maximum de parcours des agents ACI pour explorer la totalité du réseau: $\delta = \max_{1 \leq i \leq J} (T(Wk_i = S_i))$

III.4.2.5 Fonctions d'évaluation du chromosome FeTAR

A chaque itération, un chromosome Ch est évalué par les deux fonctions ***Fitness_1*** et ***Fitness_2***. La première fonction calcule le temps de réponse de chaque requête qui correspond au temps maximum de parcours de tous les agents ACI, responsables de toutes ces tâches et la deuxième fonction ***Fitness_2*** calcule le coût total de chaque requête, qui correspond à la somme de toutes les tâches qui la composent.

Algorithme Fitness_1

FR_w : le temps final d'exécution de la $req_w \in \mathfrak{R}_t$

$\forall ACI_k / 1 \leq k \leq m$:

U_k : l'ensemble de tâches à accomplir par l'agent $ACI_k / 1 \leq k \leq m$

FU_k : le temps total nécessaire pour réaliser U_k

B_k : valeur booléenne de marquage

γ : le temps maximum moyen estimé de parcours des agents ACI

1. Initialisation :

$\forall ACI_k / 1 \leq k \leq m : U_k = \emptyset, FU_k = 0$ et $B_k = faux$

$\forall req_w \in \mathfrak{R}_t : FR_w = 0$

2. Construction de U_k et calcul de $FU_k \forall ACI_k / 1 \leq k \leq m$:

Pour $\forall T_{c_i} \in \Gamma_t$ faire:

Trouver k / ACI_k réalise T_{c_i}

$U_k = U_k \cup \{T_{c_i}\}$

Chercher $j / Ch[c_i, c_j]$ égal à 1

$FU_k = FU_k + P_{c_i, c_j}$

3. Calculer $FR_w \forall req_w \in \mathfrak{R}_t$:

Pour $\forall req_w \in \mathfrak{R}_t$ faire:

Pour $\forall T_{c_i} \in \Gamma_t$ faire:

Si $T_{c_i} \in req_w$ alors faire

Chercher $k / T_{c_i} \in U_k$

Si B_k est *faux* alors faire

$FR_w = \max(FR_w, FU_k)$

$B_k = vrai$

$FR_w = FR_w + \gamma$

Algorithme Fitness_2

C_w : le coût total de la $req_w \in \mathfrak{R}_t$

1. Initialisation:

$$\forall req_w \in \mathfrak{R}_t, C_w = 0$$

2. Calcul du coût total de chaque requête :

Pour $\forall \Gamma_{t,w}$ qui compose req_w faire :

Pour $\forall T_{c_i} \in \Gamma_{t,w}$ faire:	Chercher $j / Ch[c_i, c_j]$ égal à 1
	$C_w = C_w + Co_{c_i, c_j}$

L'évaluation d'un chromosome est illustrée par un vecteur d'évaluation (Table III-3), qui exprime, pour chaque req_w , son coût total (C_w) et le temps nécessaire total pour sa réponse (D_w). Le coût moyen de toutes les requêtes et le temps de réponse maximum peuvent donc être déduits à partir du vecteur généré.

	S ₁₂	S ₃₅	S ₁	S ₃	S ₂₂	S ₁₈	S ₂₁	S ₁₄	w	d _w	C _w	D _w
T ₁₇	X	1	*	X	X	*	*	*	1	10	5	6
T ₃	*	*	1	*	*	X	X	*	2	5	1	1
T ₁₂	1	X	*	*	*	*	*	*	3	10	4	2
T ₂	*	*	*	1	*	*	*	*	4	5	3	2
T ₁₁	*	1	*	*	*	*	*	*	4	3	2	1
T ₃₅	*	*	X	X	*	*	1	X	6	5	3	2
T ₁₄	X	*	*	*	1	*	X	*				
T ₇	*	X	X	*	*	1	*	X				
T ₅	*	*	*	*	*	*	*	1				

Table III-3 - Exemple de vecteur d'évaluation

III.4.2.6 Choix de la meilleure solution

Pour déterminer la meilleure solution, nous adoptons une approche élitiste [Zitzler et al., 98] qui se base sur un archivage externe pour y mémoriser les individus les plus adaptés pendant la recherche (Figure III.13). Les meilleures solutions sont sauvegardées et mises à jour dans deux archives externes, distingués par la satisfaction du critère temps:

- L'archive des solutions principales (M): représente les meilleures solutions qui respectent les temps de fin d'exécution au plus tard. En d'autres termes, un chromosome $CH \in M$ si

et seulement si $\forall 1 \leq w \leq N, D_w \leq d_{\max} = \max_{1 \leq w \leq N}(d_w)$. Cette archive est décomposée en deux sous-archives :

- o Le sous-archive des solutions dominantes (d)
 - o Le sous-archive des solutions ε – dominantes ($\varepsilon - d$)
- L'archive des solutions secondaires (M'): représente les meilleures solutions qui excèdent au moins un temps de fin d'exécution au plus tard. En d'autres termes, un chromosome $CH \in M'$ si et seulement si $\exists w, 1 \leq w \leq N$ et $d_{\max} < D_w$.

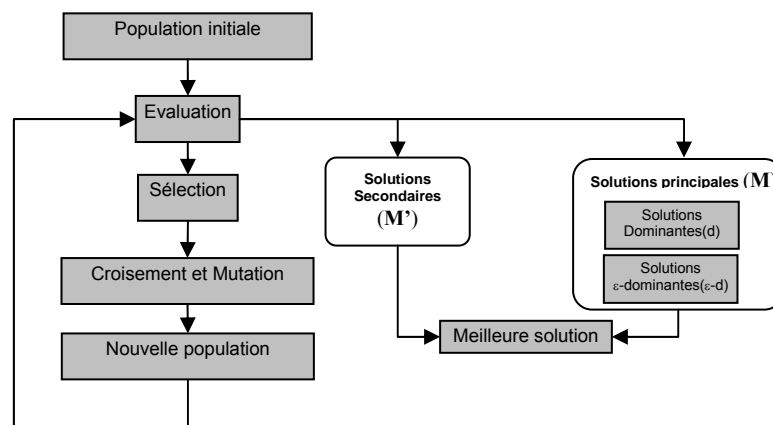


Figure III.13- Approche évolutionnaire

Le choix de séparer deux archives selon le critère temps se justifie par le fait qu'une réponse à une requête ne satisfait pas toujours ce critère. Dans ce cas, la réponse n'est pas très utile mais nous faisons le choix de la procurer à l'utilisateur qui peut rester en attente, bien que nous considérons que si la réponse à la demande de l'utilisateur lui a été procurée après le temps de fin d'exécution au plus tard correspondant alors l'utilisateur n'est pas satisfait. Nous considérons donc que le critère temps est plus prioritaire que le critère coût, ce qui signifie que les solutions retenues dans les différentes archives sont toutes d'abord triées par le critère temps puis par le critère coût.

f_1 (resp. f_2) représente la fonction délai évaluée par l'algorithme $fitness_1$ (resp. $fitness_2$) donc pour deux solutions différentes CH et CH' : si $CH \in M$ et $CH' \in M'$ alors CH domine CH' sinon Si $CH \in M$ et $CH' \in M$ ou $CH \in M'$ et $CH' \in M'$ alors CH domine CH' si et seulement si $f_1(CH) = f_1(CH')$ et $f_2(CH) < f_2(CH')$.

Chaque archive a une taille maximale de capacité, égale à la taille de la population. Donc si le nombre d'individus sélectionnés dans une archive dépasse cette taille fixe alors un

processus de « crowding » [Deb et al., 02] se déclenche pour sélectionner les individus qui doivent être gardés dans l'archive. Les solutions non sélectionnées sont supprimées; les autres contribuent à la prochaine procédure de sélection pour que les différents membres de l'archive puissent transmettre leurs caractéristiques aux populations descendantes. Les archives M et M' contiennent les solutions générées ayant les valeurs minimales de f_1 et de f_2 , donc si un nouveau chromosome CH généré domine n'importe quel élément de l'archive CH' , alors ce dernier est supprimé pour laisser sa place au nouvel élément.

Mise à part la classification des solutions FeTAR générées par les agents AO pour l'élaboration des PDRIs des agents ACI par l'approche évolutionniste adoptée, le système intègre une nouvelle catégorie d'archivage dans un autre niveau d'optimisation qui ne concerne pas cette fois les solutions évolutionnaires générées par les agents AO mais les données collectées par les agents ACI. Ce niveau d'optimisation est présenté dans ce qui suit.

III.5 Modèle d'archivage dynamique des données (MADD)

III.5.1 Classification des données et archivage

L'information, qui correspond à un service de transport multimodal et/ou à un service connexe, est représentée par des données localisées dans des nœuds distribués qui représentent les serveurs de données des fournisseurs d'information sur le RETM. Par conséquent, selon l'architecture choisie (§III.3) et l'approche d'optimisation adoptée (§III.4), les agents ACI doivent visiter ces nœuds pour collecter les données nécessaires à la satisfaction des utilisateurs du système. Cependant, lancer des agents ACI à travers le réseau peut-être limité pour les données périodiquement statiques. Il est donc possible d'optimiser le flux de données transférées sur le RETM, en réduisant le nombre effectif calculé des agents ACI grâce à une approche de classification des données selon des indicateurs. Dans des travaux antérieurs [Khaled et al., 05], l'information a été classée selon 4 types d'indicateurs: l'indicateur de localisation, l'indicateur de personnalisation, l'indicateur temporel et l'indicateur de mise à jour et suivant les valeurs attribuées à ces indicateurs, l'information peut appartenir aux catégories suivantes : statique, dynamique, événementielle, localisée, non-localisée, générale ou personnalisée. Nous nous intéressons seulement à la classification de l'information suivant les indicateurs temporels et de mise à jour dans le but de classifier l'information dans la catégorie statique, dynamique ou événementielle pour un archivage dynamique des données.

Lorsqu'un fournisseur d'information propose un service, il doit mentionner aussi bien la valeur de son indicateur temporel (IndTemp) que la valeur de son indicateur de mise à jour (IndMaj). L'indicateur temporel d'une information représente son degré de stabilité alors que l'indicateur de mise à jour représente sa période de validité. En effet, plus la valeur de l'indicateur temporel est faible, plus l'information est dynamique et dès que la période de validité s'achève, l'information n'est plus valide, dans ce cas elle doit être restaurée. L'information périodiquement statique est archivée dans la BDA (§III.3.1 et Figure III.14). Les données y sont archivées avec leur indicateur temporel et mise à jour respectif. La signification des valeurs attribuées à un indicateur temporel est donnée par le Tableau III-4.

Indicateurs		
IndTemp	IndMaj	Signification
0	N'existe pas	L'information est très dynamique
1	$\leq x$ heures	L'information peut changer dans moins de x heures
2	$> x$ heures	L'information peut changer dans plus de x heures
3	N'existe pas	L'information est statique

Tableau III-4– Indicateur temporel et de mise à jour de classification de l'information

Lorsque l'information est très dynamique, elle n'est pas archivée dans la BDA et les agents ACI doivent à chaque fois aller la chercher. Ceci explique pourquoi, pour ce type d'information, l'indicateur de mise à jour n'existe pas. D'un autre côté, lorsque l'information est complètement statique, il n'y a plus besoin non plus d'un indicateur de mise à jour car l'information est figée.

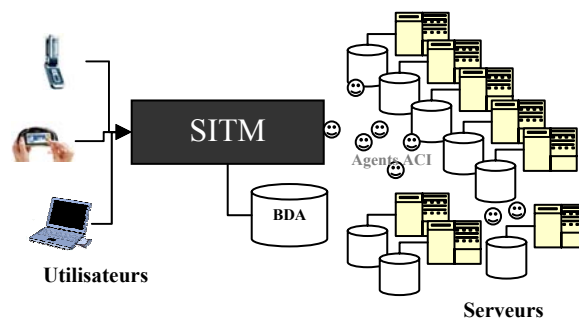


Figure III.14- Archivage dynamique de l'information

Dans ce cas si l'information n'existe pas déjà dans la BDA et un ACI est parti la chercher alors à son retour, les données correspondantes sont archivées une fois pour toute

dans la BDA. Néanmoins, dans la pratique, les données statiques ne le restent peut être pas éternellement et changent d'une manière exceptionnelle. Dans ce cas, le fournisseur d'information en question doit informer l'agent AId concerné du système, responsable de la BDA. Cette procédure est obligatoire pour assurer la pertinence de l'information qui n'est autre que la responsabilité de ses fournisseurs. Enfin, à leur retour au système, les agents ACI se chargent de la restauration des informations périmées.

III.5.2 Fonctionnement

Dès qu'un agent AId, disponible, reçoit un ensemble de requêtes Δ_e – simultanées, il génère l'ensemble des tâches Γ_i qui les composent en identifiant les fournisseurs d'information, susceptibles de répondre à ces tâches. Par la suite, cet agent doit évaluer l'indicateur *IndTemp* de chaque tâche identifiée ; si l'information correspondante n'est pas très dynamique (*IndTemp* $\neq 0$) alors l'agent AId doit vérifier si cette information a déjà été archivée localement dans la BDA. Si c'est le cas, l'agent AId doit vérifier la valeur de l'indicateur *IndMaj*. Si l'information est toujours valable, alors l'agent AId envoie les données directement à l'agent AF disponible. L'agent AO concerné charge des agents ACI de chercher l'information dans 3 cas de figures :

- a) Si l'information est très dynamique (*IndTemp* = 0),
- b) Si l'information n'est pas très dynamique (*IndTemp* $\neq 0$) et n'existe pas encore dans la BDA,
- c) Si l'information n'est pas très dynamique (*IndTemp* $\neq 0$), existe déjà dans la BDA mais elle n'est plus pertinente,

Dans les deux derniers cas, une mise à jour des données doit avoir lieu au niveau de la BDA. Le modèle d'archivage dynamique est assuré par les agents AId, AO et ACI, il est donc illustré par un diagramme d'activité à 3 partitions (Figure III.15), correspondante chacune à la participation de chaque agent à ce modèle.

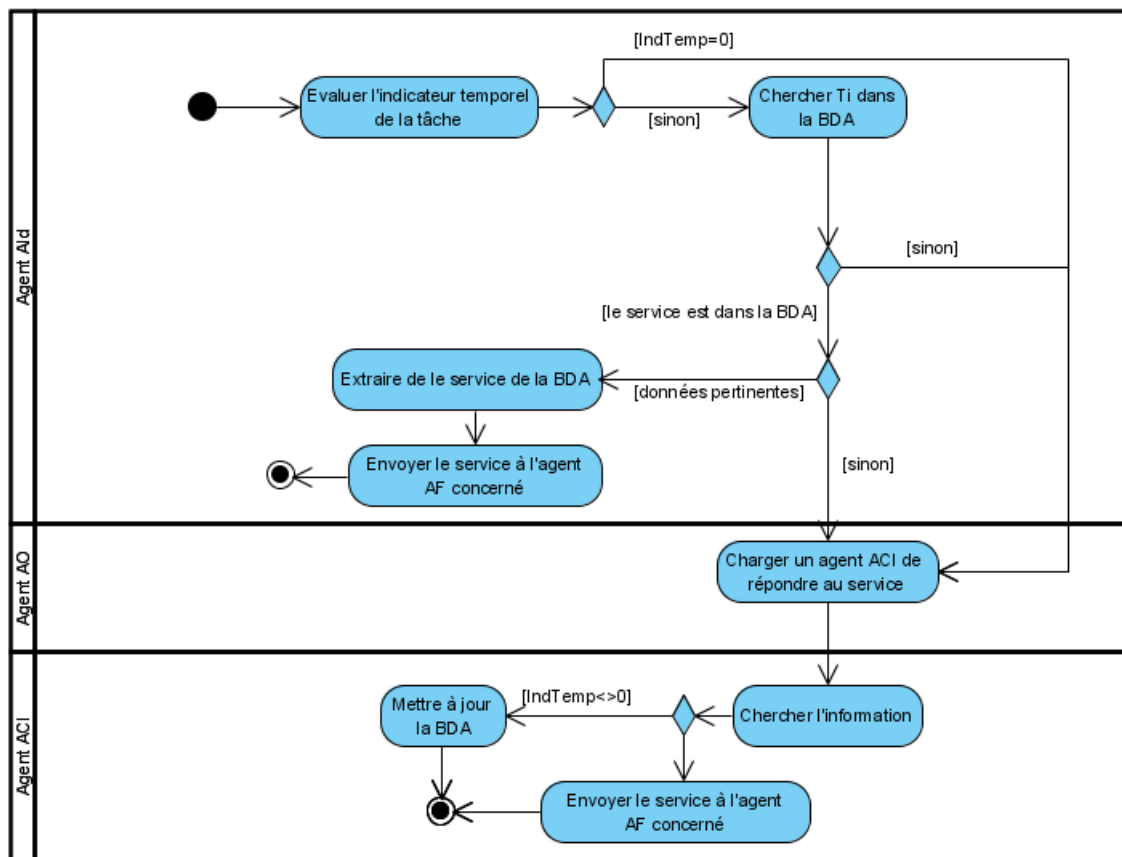


Figure III.15- Modèle de l'Archivage dynamique des données

III.6 Conclusion

Dans ce chapitre, nous avons présenté une approche d'optimisation à deux niveaux pour la gestion du flux d'information des requêtes utilisateurs. L'approche adoptée est intégrée dans un système d'information multi-agent dont le comportement vise à satisfaire les utilisateurs du système en termes de coût de l'information et du temps de réponse en réduisant le flux de données à travers le réseau. L'approche d'optimisation à deux niveaux se base sur l'interaction des agents AO et ACI et intègre un MADD pour une meilleure gestion des flux d'information.

Nous avons mentionné dans ce chapitre que les agents AO et ACI du système procèdent par une phase de négociation s'il y a des nœuds indisponibles sur le RETM à cause des aléas qui peuvent dégrader le fonctionnement du réseau. Dans le chapitre suivant, nous présentons le protocole de négociation adopté en utilisant une ontologie flexible de transport que nous avons conçu pour automatiser les interactions correspondantes au protocole.

CHAPITRE IV. PROTOCOLE DE NEGOCIATION ENTRE AGENTS ET CONCEPTION D'UNE ONTOLOGIE DE TRANSPORT

IV.1 Introduction

Dans le chapitre précédent, nous avons présenté une approche d'optimisation multiservices multi-sources dans un système d'information multi-agent. La solution proposée ne prend pas en considération les éventuelles perturbations du RETM (goulets d'étranglements, pannes, etc.) pour satisfaire les clients du transport dans tous les cas de figures. Dans ce chapitre, nous proposons un protocole de négociation entre les agents AO et les agents ACI du système. Le protocole proposé vise à réaffecter dynamiquement les services, dont les fournisseurs préalablement attribués ne sont plus disponibles, en se basant sur les états courants des agents ACI.

Peu de travaux ont évoqué la négociation dans le contexte des agents mobiles, la plupart parlent de négociation classique et asynchrone avec des hôtes distants [Kalyani et al., 06] [Huang et al., 03].

Dans ce chapitre, les agents ACI et AO négocient à distance la réattribution en temps réel des serveurs aux tâches en cas de perturbation. Cette réattribution dynamique reconstruit les itinéraires restants des agents ACI jusqu'à ce qu'il n'y ait plus de tâche à réattribuer. Dans le cas contraire, et si tous les agents ACI rejoignent le nœud hôte alors l'agent AO, initiateur de la négociation, lance à nouveau des agents ACI pour finir le travail. Le protocole proposé est inspiré du protocole réseau contractuel (« Contract Net Protocol ») [Smith, 80] et permet des accords partiels entre les différentes parties.

L'aspect combinatoire et complexe du protocole de négociation proposé, nous a incités à mettre en œuvre une ontologie flexible pour le transport qui permet de procurer un vocabulaire adapté pour exécuter les contrôles sémantiques appropriés pour une expression donnée.

L'évolution de ce chapitre est organisée comme suit :

- Dans le prochain paragraphe, nous présentons un exemple typique de situation de perturbation qui démontre l'impact de telles circonstances sur le système et le besoin de la mise en œuvre d'une forme d'interaction évoluée entre les agents concernés,
- Dans le troisième paragraphe, nous montrons la structure de base d'une négociation avant d'élaborer une brève description des principaux mécanismes et des modèles existants pour ce type d'interaction,

- Dans le quatrième paragraphe, nous présentons le protocole de négociation proposé ainsi que les critères d'évaluation des coûts des différents intervenants,
- La stratégie adoptée pour la réattribution des services est développée dans le cinquième paragraphe,
- Le terme « ontologie » est décrit dans le sixième paragraphe accompagné d'un bref état de l'art sur les supports ontologiques pour les processus de négociation,
- L'ontologie proposée est traitée dans le septième paragraphe.

IV.2 Situation de perturbation

Une situation de perturbation se présente lorsque des nœuds du RETM, qui devraient être visités par des agents ACI, ne sont plus disponibles. Ces agents doivent éviter les nœuds touchés par la perturbation, en modifiant leur itinéraire restant. Dans ce cas, un procédé de réattribution des services est nécessaire pour décerner de nouveaux fournisseurs disponibles aux différentes tâches en attente. Or, le caractère flexible que nous avons particulièrement élaboré pour un chromosome, permet une mise en œuvre complètement appropriée à ce problème de réattribution. Prenons l'exemple de la solution Ch (Table IV-1) touchée par une perturbation réseau qui concerne l'ensemble des nœuds $\{S_{18}, S_{10}, S_3\}$. Cette perturbation nécessite la réaffectation de l'ensemble des tâches $\{T_3, T_{27}, T_{10}, T_{13}\}$.

Ch	S ₅	S ₁₁	S ₁₈	S ₁	S ₁₄	S ₅₅	S ₂₀₁	S ₅₀	S ₁₀	S ₇₁	S ₃
T ₁₂	*	*	*	*	1	*	*	*	*	X	*
T ₃	*	X	*	*	*	*	X	*	*	*	1
T ₅	1	*	*	*	*	X	X	*	*	X	*
T ₆	X	X	X	X	X	X	X	1	X	*	X
T ₂₇	X	*	1	*	X	*	X	X	X	*	*
T ₁₀	X	X	X	*	*	*	X	X	1	X	X
T ₃₂	*	1	*	*	*	X	X	X	*	*	X
T ₁₃	*	*	*	*	*	*	X	X	*	*	1
T ₂₈	X	X	X	X	*	1	*	X	*	*	*
T ₁₉	*	*	*	1	*	*	*	*	*	*	X
T ₁₄	*	X	*	*	X	1	X	*	*	*	*
T ₄₄	*	*	X	*	*	X	1	X	*	*	*
T ₇₈	X	*	X	X	*	X	*	*	*	1	X

Table IV-1– Exemple de chromosome touché par une perturbation

Nous choisissons d'optimiser la procédure de réaffectation en abordant, de la manière la plus idéale possible, les agents ACI actifs en cours de route. Ainsi les agents ACI et AO

doivent interagir afin de trouver la meilleure solution appropriée à la situation courante de perturbation (Figure IV.1).

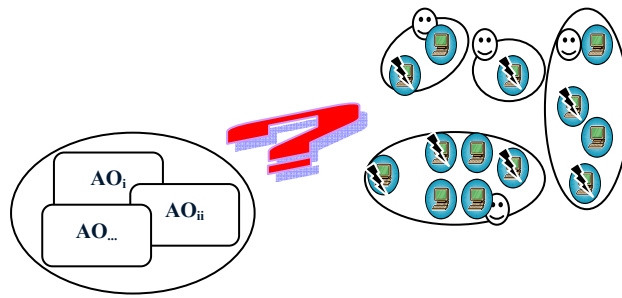


Figure IV.1- Situation de perturbation

Ch	S ₅	S ₁₁	S ₁₈	S ₁	S ₁₄	S ₅₅	S ₂₀₁	S ₅₀	S ₁₀	S ₇₁	S ₃
T ₁₂	*	*	*	*	1	*	*	*	*	X	*
T ₃	*	X	*	*	*	*	X	*	*	*	1
T ₅	1	*	*	*	*	X	X	*	*	X	*
T ₆	X	X	X	X	X	X	X	1	X	*	X
T ₂₇	X	*	1	*	X	*	X	X	X	*	*
T ₁₀	X	X	X	*	*	*	X	X	1	X	X
T ₃₂	*	1	*	*	*	X	X	X	*	*	X
T ₁₃	*	*	*	*	*	*	X	X	*	*	1
T ₂₈	X	X	X	X	*	1	*	X	*	*	*
T ₁₉	*	*	*	1	*	*	*	*	*	*	X
T ₁₄	*	X	*	*	X	1	X	*	*	*	*
T ₄₄	*	*	X	*	*	X	1	X	*	*	*
T ₇₈	X	*	X	X	*	X	*	*	*	1	X

Transformation

Ch	S ₅	S ₁₁	S ₁₈	S ₁	S ₁₄	S ₅₅	S ₂₀₁	S ₅₀	S ₁₀	S ₇₁	S ₃
T ₁₂	*	*	*	*	1	*	*	*	*	X	*
T ₃	*	X	*	*	*	'1'	X	*	*	*	'*'
T ₅	1	*	*	*	*	X	X	*	*	X	*
T ₆	X	X	X	X	X	X	X	1	X	*	X
T ₂₇	X	*	'*'	'1'	X	*	X	X	X	*	*
T ₁₀	X	X	X	*	*	'1'	X	X	'*'	X	X
T ₃₂	*	1	*	*	*	X	X	X	*	*	X
T ₁₃	'1'	*	*	*	*	*	X	X	*	*	'*'
T ₂₈	X	X	X	X	*	1	*	X	*	*	*
T ₁₉	*	*	*	1	*	*	*	*	*	*	X
T ₁₄	*	X	*	*	X	1	X	*	*	*	*
T ₄₄	*	*	X	*	*	X	1	X	*	*	*
T ₇₈	X	*	X	X	*	X	*	*	*	1	X

Figure IV.2– Exemple de chromosome modifié suite à une perturbation

De son côté, l'agent AO doit recalculer le vecteur d'évaluation du chromosome afin de respecter les délais autorisés d'obtention des réponses. La forme d'interaction adoptée, que nous présentons par la suite, doit émerger d'une forme d'accord entre l'agent AO concerné et les agents ACI correspondants, afin de réaffecter les tâches dont les fournisseurs préalablement choisis sont touchés par la perturbation (Figure IV.2).

Cette situation nous engage dans un contexte de compétition où l'objet de négociation étant l'ensemble des tâches à réaffecter. Avant de présenter le protocole de négociation proposé, nous élaborons, dans le paragraphe qui suit, une brève description des mécanismes et des modèles existants pour cette forme évoluée d'interaction,

IV.3 La négociation : la forme d'interaction adoptée

Dans un SMA, un objet est considéré comme un sujet de négociation entre un groupe d'agents, s'il se retrouve dans une situation de désaccord pour la réalisation d'un objectif commun. Ainsi, la négociation est la forme de contribution la plus appropriée pour la résolution de conflits car elle garantit un pouvoir équitable des agents, ce qui permet la décentralisation du contrôle. La négociation est également considérée comme une méthode de coordination d'actions qui conduit à une fixation d'un ordre d'exécution des opérations par relaxation de buts initiaux [Weiss, 00]. Dans tous les cas, nous parlons de négociation lorsque les différents intérêts sont en conflits pour la réalisation de buts communs.

IV.3.1 Structure de base

La structure de base d'une négociation entre deux agents A et B, est illustrée par le diagramme de séquences de la Figure IV.3 ; l'agent A commence par faire une proposition à l'agent B, ce dernier évalue la proposition reçue et détermine s'il en est satisfait ou pas. Si B n'est pas satisfait, alors il élabore une contre-proposition en fonction de ses propres buts et contraintes. Lorsque l'agent A évalue cette nouvelle proposition, il détermine s'il en est satisfait ou pas et ainsi de suite. Les tours de négociation s'enchaînent jusqu'à la satisfaction des buts communs ou l'arrêt accidentel ou volontaire de l'interaction. L'arrêt accidentel peut être la cause d'une dégradation des performances du système et l'arrêt volontaire peut être la cause d'une annulation de la part de l'initiateur de la négociation.

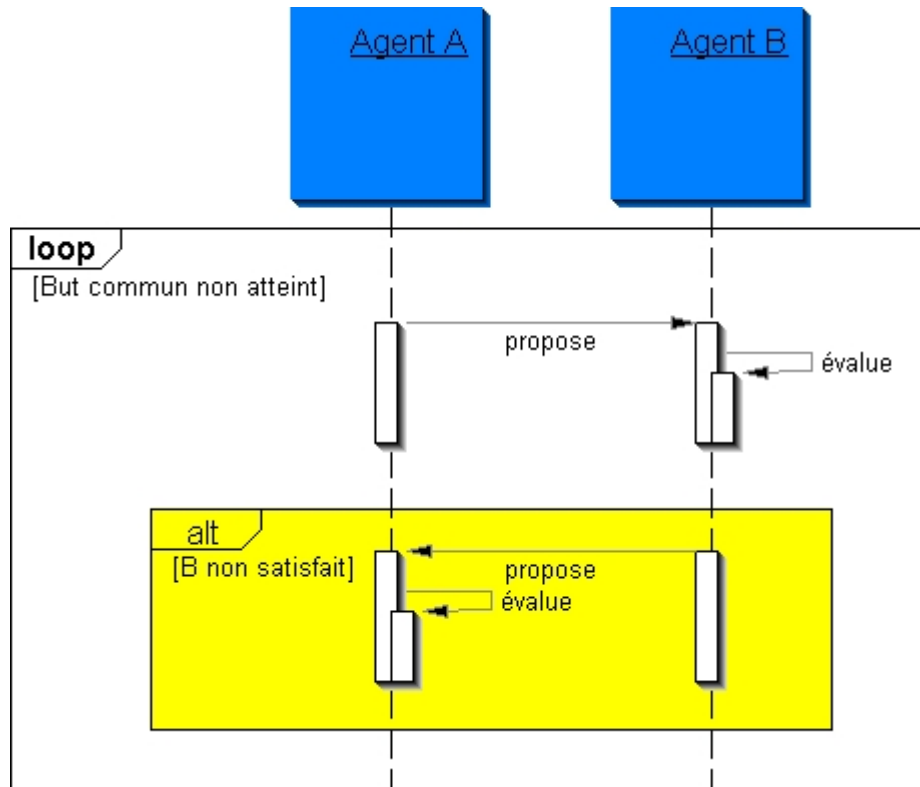


Figure IV.3– Structure de base d’une négociation

IV.3.2 Mécanismes de négociation

Les agents engagés dans un processus de négociation doivent prendre certaines initiatives qui peuvent conduire à un accord entre les différentes parties. De ce fait, trois grands mécanismes de négociation existent dans la littérature :

- La négociation par compromis : dans ce type de négociation, chacune des parties relâche les contraintes qu’elle juge moins importantes afin d’augmenter les chances d’atteindre le but global,
- La négociation intégrante : dans ce type de négociation, chacune des parties est amenée à changer ses objectifs selon le nouveau tour de négociation courant, en cherchant les buts profonds,
- La formation de coalitions: il s’agit de partitionner le groupe d’agents en conflit en sous-groupes appelés coalitions. Une coalition représente une organisation d’agents à court terme basée sur des engagements spécifiques et contextuels, ce qui permet aux agents de coexister tout en bénéficiant de leurs compétences

respectives [Vauvert et al., 00]. Le problème qui se pose ici se trouve dans la manière même de former les coalitions.

IV.3.3 Modèles de négociation

Dans ce paragraphe, nous présentons les modèles les plus connus en matière de négociation entre agents.

IV.3.3.1 Théorie de la Vente Aux Enchères (VAE)

Le principe de la VAE consiste de vendre un produit avec le prix le plus cher possible à des acheteurs potentiels qui essaient d'obtenir ce produit avec le prix le plus bas possible. Les protocoles qui résument cette théorie sont classés en deux catégories:

- Enchères ouvertes : Dans ce type d'enchère, tous les agents détiennent une vue globale sur le déroulement de la procédure ; chaque agent connaît la mise des autres agents et peut ainsi évaluer sa prochaine proposition en fonction de ses contraintes et des capacités observées des autres agents. Les deux modèles les plus courants dans ce type d'enchère sont :
 - Enchère anglaise : c'est le protocole le plus commun de la VAE qui représente une enchère ouverte à prix ascendant. Au commencement d'une session, si l'initiateur n'annonce pas le prix le plus bas acceptable (prix de réserve), alors le prix initial enregistré est égal à 0. Ensuite, chaque acheteur propose une somme supérieure à celle enregistrée devenant ainsi la nouvelle somme enregistrée et l'enchère se termine lorsqu'il n'y a plus d'enchérisseur. L'enchère est alors remportée par l'agent enchérisseur dont la proposition est à la hauteur du dernier prix enregistré,
 - Enchère hollandaise : c'est une enchère ouverte à prix descendant. Au commencement d'une session, l'initiateur décide du prix maximal pouvant être remporté par l'un des acheteurs. Ensuite, il décrémente ce prix jusqu'à atteindre un prix jugé acceptable par l'un des acheteurs qui remporte l'enchère.
- Enchères fermées : ce sont des enchères qui se déroulent en un seul tour de négociation où aucun agent ne connaît la mise des autres agents. Ainsi un agent décide de sa mise, en fonction de son estimation de la valeur de l'objet et des offres des autres agents. Les deux modèles les plus courants dans ce type d'enchère sont :

- Enchères sur enveloppe scellée : L'agent gagnant soumet la meilleure offre qu'il a proposée,
- Enchères Vickrey : L'agent gagnant soumet le deuxième meilleur prix offert.

IV.3.3.2 Le protocole réseau contractuel (« contract-net protocol »)

Le Protocole Réseau Contractuel (PRC) [Smith, 80], pour l'allocation des tâches décentralisées, a révolutionné le domaine de l'intelligence artificielle distribuée par le principe de la négociation entre des managers (initiateurs) et des contractants (participants). Dans sa version d'origine, le PRC ne présente aucun moyen d'évaluation de coût ; la simple convenance du contractant était suffisante pour sa sélection selon par exemple, son adjacence ou sa capacité. La version de base du PRC est donc aujourd'hui considérée comme un modèle de coopération qui se base sur des hypothèses de performances, de bénévolat et de modularités vérifiées.

Dans une version plus évoluée [Sandholm, 93], le PRC fut enrichi de notions de coût pour le choix de la tâche à effectuer par le contractant en ajoutant la possibilité de négocier sur un ensemble atomique de tâches (Figure IV.4). La notion de coût correspond à :

1. l'annonce d'une demande (« Announcing »), qui n'est obligatoirement pas déléguée à tous les agents [Parunak , 87],
2. la soumission d'une offre (« Bidding »),
3. l'attribution d'une tâche (« Awarding »).

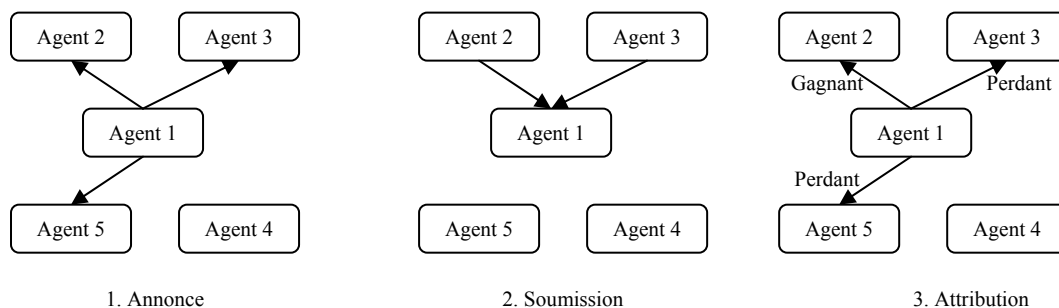


Figure IV.4 – Le Protocole du Réseau Contractuel

Un modèle générique pour la négociation de contrat, proposé par [Mathieu et al., 02], est caractérisé par la succession d'échange de messages entre un initiateur et des participants comme c'est le cas dans le PRC, qui représente aujourd'hui l'un des modèles les plus élaborés et les plus complets pour la coopération et la compétition

Par rapport à ce qui a été proposé dans la littérature en matière de négociation, nous avons adapté le PRC à notre problème de réattribution en brisant l'atomicité de l'ensemble des tâches sur lesquelles les agents négocient [Zgaya et al., 07a]. De ce fait, un agent participant peut accepter une partie de la proposition initiale de l'agent initiateur et non la refuser totalement et donc ce dernier peut confirmer une sous-partie de la partie préalablement acceptée. Cette flexibilité du protocole permet de diminuer le nombre de tours de négociation en permettant des accords partiels de contrat. Le protocole de négociation que nous proposons est détaillé dans ce qui suit.

IV.4 Le protocole de négociation proposé

Pour une situation de perturbation, nous proposons un processus de négociation inspiré du PRC entre les agents ACI, qui représentent les participants de la négociation, et l'agent AO correspondant, qui représente l'initiateur de la négociation. Dans la solution proposée, un accord partiel du contrat proposé est possible de la part de chaque agent ACI. Cette flexibilité représente l'originalité du protocole proposé. La partie acceptée est à confirmer partiellement ou totalement par l'initiateur de la négociation. Un procédé de renégociation est nécessaire tant qu'il y a encore des tâches à réaffecter après chaque tour de négociation. Le but de cette solution est de permettre aux agents ACI de coopérer et de coordonner leurs actions afin de trouver des ordonnancements robustes proches de l'optimum global selon leurs priorités, préférences et contraintes qui dépendent de leurs positions courantes dans leur itinéraire correspondant. D'autre part, les agents AO doivent assurer un temps minimal de réponse et des coûts raisonnables lors de chaque tour de négociation.

IV.4.1 L'initiateur

Un initiateur est un agent AO qui ignore les positions courantes des agents ACI en cours de route. Cependant, il connaît tous les schémas des PDRIs et les affectations des serveurs (PDRDs). L'agent AO n'attend pas toutes les réponses des participants pour prendre sa décision, il peut donc accepter une partie des réponses pour prendre des décisions à chaque courte période de temps et doit mettre à jour la liste des services à réaffecter après chaque

étape de confirmation. Ensuite, un agent AO doit proposer un nouveau contrat selon les différentes propositions des participants à la négociation.

IV.4.2 Les participants

Un participant est un agent ACI qui dispose de ses propres priorités, préférences et contraintes et qui ignore l'état des autres participants du même processus de négociation. Ces caractéristiques changent en temps réel et varient selon :

- sa position courante dans son PDRD et donc selon son PDRD restant,
- de l'état du réseau,
- des décisions prises au cours du processus de négociation engagé, suite à une confirmation avérée de l'agent initiateur.

A chaque fois qu'un agent ACI reçoit un nouveau contrat, il l'analyse et prend une décision (refus ou acceptation totale ou partielle).

IV.4.2.1 Le PDRD restant

Un PDRD (§III.3.2.3) Restant d'un agent ACI_k à un instant t , est une séquence de Δ_k nœuds qui est notée par $P2DR_{t,k} = (S_{x_1}, \dots, S_{x_{\Delta_k}})$, constituée de l'ensemble des nœuds qui appartiennent au PDRD de l'agent ACI_k , à partir du nœud qui suit sa position courante, noté par $Pos_{t,k}$. La Figure IV.5 montre un exemple d'itinéraire d'un agent ACI. Tous les nœuds de cet itinéraire forment le PDRD de cet agent. Le PDRD déduit est constitué de la série des nœuds $(S_{12}, S_{23}, S_4, S_1, S_{48}, S_7, S_{95})$. Si à l'instant t , l'agent ACI_k se trouve dans le nœud S_{23} , alors $P2DR_{t,k} = (S_4, S_1, S_{48}, S_7, S_{95})$.

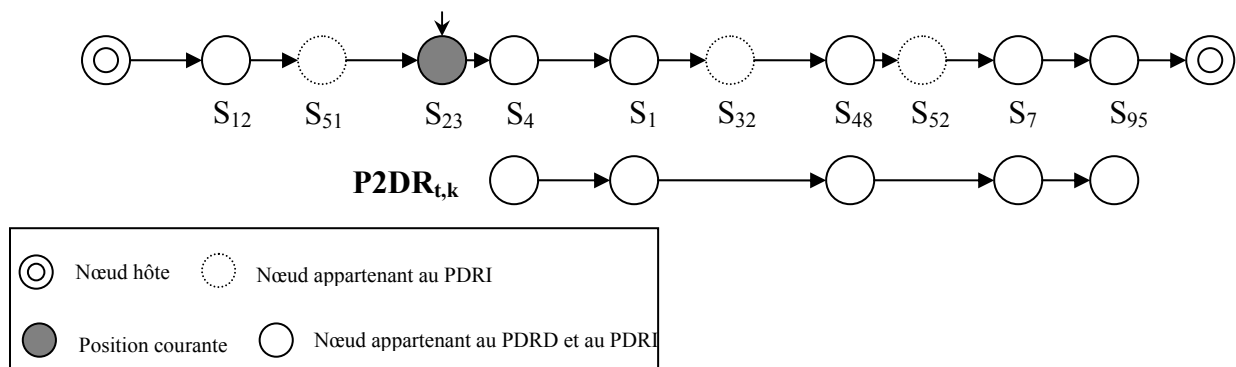


Figure IV.5– Exemple de PDRD restant d'un agent participant

IV.4.2.2 Les priorités

La séquence de priorités d'un agent ACI_k à un instant t , notée par $Priorités_{t,k}$, est constituée de la séquence des nœuds disponibles qui forment le PDRD de l'agent ACI_k , à partir de sa position courante $Pos_{t,k}$. Si nous reprenons le même exemple de la Figure IV.5, avec les nœuds S_1 et S_{52} qui ne sont pas disponibles (Figure IV.6), alors $Priorités_{t,k} = (S_{23}, S_4, S_{48}, S_7, S_{95})$. Le nœud S_{52} , ne faisant même pas partie du PDRD de ACI_k , il n'est pas pris en considération donc qu'il soit disponible ou pas ne change pas le fait qu'il n'appartienne pas à $Priorités_{t,k}$. Par contre, le nœud S_1 , faisant partie du PDRD, à partir de $Pos_{t,k}$, n'appartient pas à $Priorités_{t,k}$ car il n'est pas disponible.

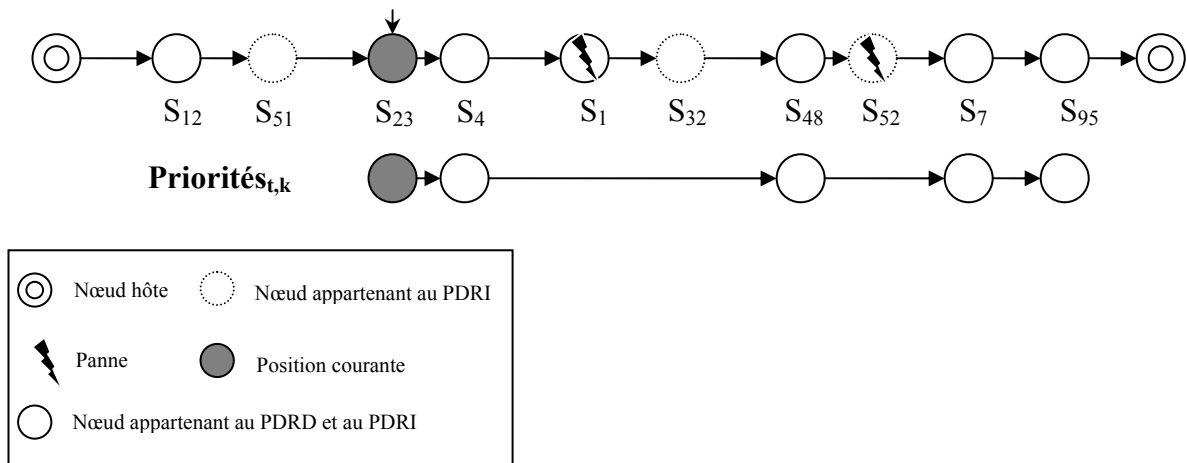


Figure IV.6– Exemple de priorités d'un agent participant

IV.4.2.3 Les préférences

La séquence de préférences d'un agent ACI_k à un instant t , notée par $Préférences_{t,k}$, est constituée de l'ensemble des nœuds disponibles qui appartiennent au PDRI et non au PDRD de l'agent ACI_k , à partir de sa position courante. Si nous reprenons le même exemple de la Figure IV.6, alors $Préférences_{t,k} = (S_{32})$. Par analogie aux priorités, le nœud S_1 , ne faisant même pas partie du PDRI de ACI_k , il n'est pas pris en considération donc qu'il soit disponible ou pas ne change pas le fait qu'il n'appartienne pas à $Préférences_{t,k}$. Par contre, le nœud S_{52} , faisant partie du PDRI, à partir de $Pos_{t,k}$, n'appartient pas à $Préférences_{t,k}$ car il n'est pas disponible. Ce résultat est illustré par la Figure IV.7.

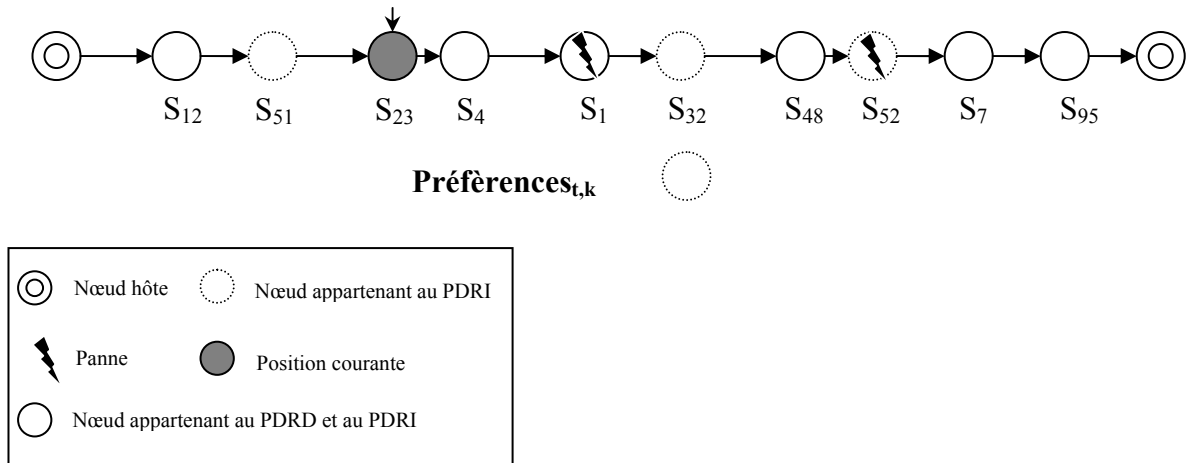


Figure IV.7– Exemple de préférences d'un agent participant

IV.4.2.4 Les contraintes

L'ensemble des contraintes d'un agent ACI_k à un instant t , noté par $Contraintes_{t,k}$ est illustré par l'exemple de la Figure IV.8. L'ensemble des nœuds sur le RETM que l'agent ACI_k ne peut pas visiter à partir de sa position courante, parce qu'ils causent des problèmes (surcharge, consommation excessive de ressources, etc.).

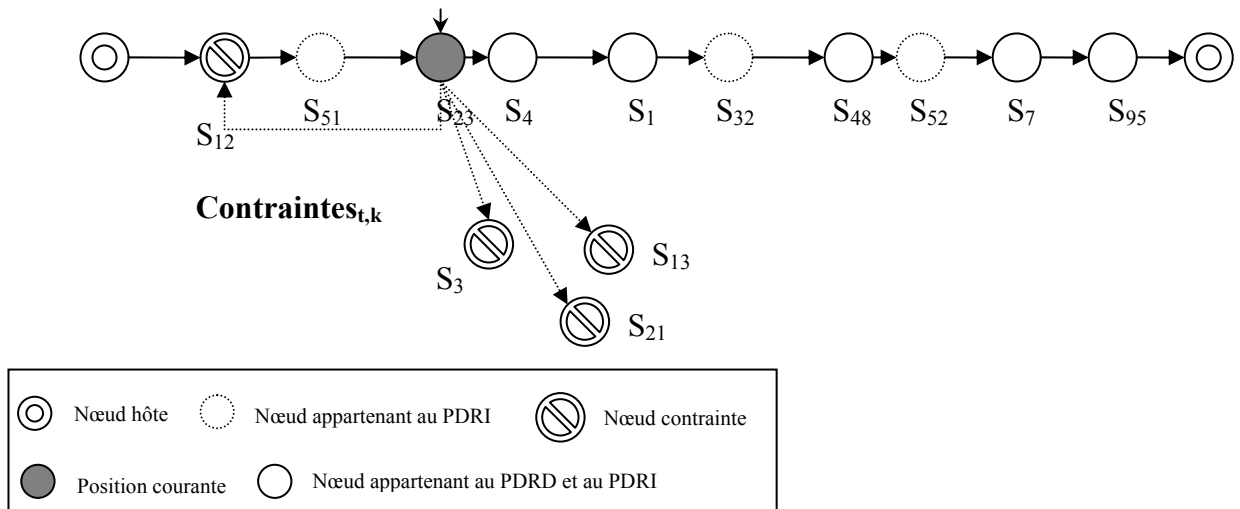


Figure IV.8– Exemple de contraintes d'un agent participant

IV.4.3 Le protocole de négociation proposé

Un protocole définit le langage utilisé par les agents pour échanger de l'information. Le protocole de négociation proposé (Figure IV.9), est caractérisé par une succession d'échanges de messages entre l'initiateur, représenté par l'agent AO, et les participants, représentés par les agents ACI. Une négociation est déclenchée par une perturbation qui nécessite la

réaffectation des nœuds disponibles à l'ensemble des tâches dont les nœuds préalablement attribués ne sont plus disponibles à cause de la perturbation. L'ensemble des tâches qui nécessite une réaffectation à l'instant t est noté par Φ_t .

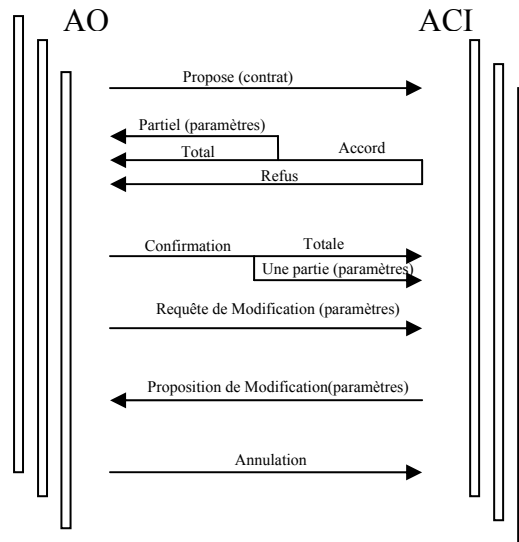


Figure IV.9- Le protocole de négociation proposé

Lorsqu'un agent AO génère une solution FeTAR, il envoie les PDRDs déduits des agents ACI sous forme de contrat. En cas de perturbation, tous les agents ACI ne vont probablement pas accepter totalement ce contrat initial parce que les nœuds appartenant à leur PDRD ont été touchés par la perturbation. L'agent AO va d'abord confirmer des parties acceptées du contrat avant de mettre à jour Φ_t . Ensuite, il va envoyer une requête de proposition de modification à chaque participant actif qui va proposer un plan de modification selon ses préférences, ses priorités et ses contraintes. A la réception des propositions, l'agent initiateur va trancher en choisissant parmi les différentes offres reçues pour un nouveau tour de négociation. La décision prise sera donc envoyée aux agents ACI sous forme d'un nouveau contrat et ainsi de suite jusqu'à ce qu'il n'y a plus de tâches à réaffecter. Le processus de négociation peut être interrompu si les temps de fin d'exécution au plus tard des requêtes ne sont plus respectés ou s'il y a un manque de ressources.

Le protocole proposé peut s'adapter à la négociation de plusieurs initiateurs avec plusieurs participants, si les différentes sociétés actives d'agents s'interagissent. Ce cas plus général dépasse le cadre de cette thèse mais fait partie de ses perspectives. Dans ce qui suit, nous détaillons les différents échanges de messages entre un unique initiateur et les participants correspondants.

IV.4.3.1 La forme du message

Nous adoptons la structure suivante pour un message échangé entre les agents :

$\langle \text{émetteur}, \text{receveur}, \text{service}, \text{performative}, \text{contenu}, \text{lang_contenu}, \text{ontologie}, f \rangle$ avec :

- *émetteur* : l'émetteur du message,
- *receveur* : la liste des receveurs (récipients) du message,
- *service* : le service "pages jaunes" proposé par le receveur du message,
- *performative* : exprime l'intention de la communication,
- *contenu* : l'information incluse dans le message,
- *lang_cont* : le langage utilisé pour exprimer le contenu,
- *ontologie* : le vocabulaire des symboles utilisé dans le contenu et leur signification,
- $f = \langle f_1, f_2, f_3, f_4, f_5 \rangle$ représente quelques champs pour le contrôle des conversations concurrentes et aussi pour la gestion des temps de réponse avec :

f_1 . *réponse-à A*: le récipient de la réponse du message est l'agent A,

f_2 . *conversation-id ide*: une identification de la conversation qui peut être fixée par l'émetteur du message dans le but d'identifier la séquence des actes communicatifs correspondants, qui forment ensemble une conversation,

f_3 . *réponse-avec exp*: identifie la réponse au message courant avec l'expression *exp*,

f_4 . *en-réponse-à exp*: indique que le message courant est une réponse à une action antérieure à laquelle la réponse a été dénotée par *exp* (f_3),

f_5 . *réponse-par d*: expression de temps et/ou date qui indique le temps d'attente maximal toléré par un agent émetteur d'un message pour recevoir une réponse à ce message.

Cette partie dépasse le cadre de cette thèse mais fait partie de ses perspectives, ces champs ne seront donc pas instanciés dans ce qui suit. Le langage utilisé pour exprimer le contenu ainsi que le vocabulaire utilisé ne seront également pas précisés pour l'instant. Dans ce qui suit, un message agent aura la forme suivante :

$\langle \text{émetteur}, \text{receveur}, \text{service}, \text{performative}, \text{contenu} \rangle$

IV.4.3.2 Proposition du contrat

La proposition du contrat correspond au message de la première organisation (le premier contrat) ou d'une réorganisation des PDRDs. Si $\Phi_t \neq \emptyset$ alors il s'agit bien du phénomène de

réorganisation. L'initiateur envoie un contrat individuel à chaque agent ACI actif de la même société P_t :

$$\langle P_t(\mathbf{AO}), P_t(\mathbf{ACI}_k), \text{réception-contrat}, \text{propose}, \partial \rangle$$

Avec: $\partial = \partial_1$ si $\Phi_t = \emptyset$ et $\partial = \partial_2$ sinon.

∂_1	∂_2
PDR (Propriétaire: \mathbf{ACI}_k RefI: v_1, \dots, v_{α_k} RefD: $f_1 \{ \psi_{f_1} \}, \dots, f_{\beta_k} \{ \psi_{f_{\beta_k}} \}$)	PDRD (Propriétaire: \mathbf{ACI}_k RefD: $f_1 \{ \psi_{f_1} \}, \dots, f_{\beta_k} \{ \psi_{f_{\beta_k}} \}$)

La séquence v_1, \dots, v_{α_k} représente les références des nœuds correspondants au PDRI de l'agent récepteur \mathbf{ACI}_k , la séquence f_1, \dots, f_{β_k} représente les références des nœuds de son PDRD et la séquence $\{ \psi_{f_1} \}, \dots, \{ \psi_{f_{\beta_k}} \}$ représente les ensembles des tâches à accomplir respectivement sur ces nœuds avec $\alpha_k \leq \beta_k$.

IV.4.3.3 Réponse au contrat

Lorsqu'un participant reçoit un contrat, il l'analyse et répond par:

- Un accord total s'il approuve l'affectation choisie des tâches par l'initiateur. Cette affectation concerne ce qui reste de son parcours (PDRD restant), selon sa position courante. Ce type de message ne porte pas de contenu, juste une performative classique d'acceptation:

$$\langle P_t(\mathbf{ACI}_k), P_t(\mathbf{AO}), \emptyset, \text{Accord-proposition}, \emptyset \rangle$$

- Un accord partiel s'il approuve une partie du contrat proposé. Le contenu du message exprime les références des tâches annulées:

$$\langle P_t(\mathbf{ACI}_k), P_t(\mathbf{AO}), \emptyset, \text{AccordPartiel}, \partial \rangle$$

Avec $\partial \equiv (c_1, \dots, c_n)$

- Un refus s’il n’est pas du tout d’accord. Ce type de message ne porte pas de contenu, juste une performative classique de refus:

$$\langle P_t(\mathbf{ACI}_k), P_t(\mathbf{AO}), \emptyset, \mathbf{Refus}, \emptyset \rangle$$

L’agent initiateur n’attend pas l’intégralité des réponses de tous les participants parce qu’il doit agir rapidement. En effet, il collecte les réponses reçues pendant une courte période de temps afin de prendre une décision; cette caractéristique serait exprimée par le dernier champ f d’un message agent, à travers la facette *reply-by*.

IV.4.3.4 Confirmation

Un agent initiateur doit confirmer indépendamment la partie acceptée de chaque contrat proposé. La confirmation peut-être:

- Totale si l’initiateur approuve totalement la réponse au contrat. Ce type de message ne porte pas de contenu, juste une performative classique de confirmation:

$$\langle P_t(\mathbf{AO}), P_t(\mathbf{ACI}_k), \emptyset, \mathbf{Confirmation}, \emptyset \rangle$$

- Partielle si l’initiateur approuve une partie de la réponse du contrat précédemment proposé, le contenu du message exprime les références des tâches accordées:

$$\langle P_t(\mathbf{AO}), P_t(\mathbf{ACI}_k), \emptyset, \mathbf{ConfirmationPartielle}, \hat{\partial} \rangle$$

Avec $\hat{\partial} \equiv (g_1, \dots, g_p)$

Il est important de noter, que la phase de confirmation nécessite la mise à jour de Φ_t . Si $\Phi_t = 0$ suite à une phase de confirmation, alors la négociation s’achève à ce niveau parce qu’un compromis a été construit au fur et à mesure des tours de négociation précédents. Sinon, l’agent initiateur demande aux participants de proposer une modification de leur PDRD restant.

IV.4.3.5 Demande de modification

Si Φ_t n'est pas encore vide, l'agent initiateur demande aux participants de proposer une nouvelle distribution des affectations. Le contenu *requête-modification* du message exprime Φ_t et la table des services pour que les agents ACI puissent proposer de nouvelles affectations:

$$\langle P_t(\text{AO}), P_t(\text{ACI}_k), \emptyset, \text{DemandeModification}, \partial \rangle$$

Avec $\partial \equiv (\Phi_t, \text{tabServices})$.

IV.4.3.6 Annulation

Le processus de négociation peut être interrompu dans les deux cas suivants :

- Si le plus petit temps d'exécution au plus tard d_{\min} de toutes les requêtes concernées est écoulé depuis l'acquisition des requêtes. Dans ce cas, l'agent AO peut lancer à nouveaux des agents ACI pour exécuter le contrat courant et tenter de répondre aux requêtes même avec du retard,
- S'il n'y a pas de solution possible (manque de ressources, pas de fournisseur disponible...). Dans ce cas, les réponses, même incomplètes, sont transférées aux utilisateurs.

Ce type de message ne porte pas de contenu, juste une performative classique d'annulation:

$$\langle P_t(\text{AO}), P_t(\text{ACI}_k), \emptyset, \text{Annulation}, \emptyset \rangle$$

IV.4.3.7 Proposition de modification

Dans un travail antérieur [Zgaya et al., 06a], nous avons conçu une stratégie de réattribution des serveurs aux tâches, suivant les contraintes, priorités et préférences des agents participants. Le contenu *proposition* du message exprime l'offre de chaque participant:

$$\langle P_t(\text{ACI}_k), P_t(\text{AO}), \emptyset, \text{PropositionModification}, \partial \rangle$$

Avec $\partial \equiv PDRD($

Propriétaire: ACI_k

D: $f_1 \{ \psi_{f_1} \}, \dots, f_{dk} \{ \psi_{f_{dk}} \}$

)

La séquence f_1, \dots, f_{d_k} représente les références des nœuds de la proposition du PDRD reconstruit restant et la séquence $f_1 \{\psi_{f_1}\}, \dots, f_{d_k} \{\psi_{f_{d_k}}\}$ représente les ensembles des tâches à accomplir respectivement sur ces nœuds.

La stratégie de réattribution des tâches, permettant d'établir une nouvelle proposition de modification pour un nouveau contrat, est présentée dans le paragraphe qui suit.

IV.5 Réattribution des services

La procédure de réattribution des services vise à reconstruire, de la manière la plus optimale possible, ce qui reste des parcours des agents ACI. Ce processus est déclenché à chaque fois qu'un agent ACI_k détecte une quelconque perturbation sur le RETM qui l'empêche de suivre normalement son $P2DR_{t,k}$ afin de respecter le contrat proposé par l'agent AO. Dans ce cas, les agents ACI et AO concernés entrent dans une phase de négociation comme il a été décrit dans le paragraphe précédent, dans le but de réaffecter les services qui ont encore besoin de fournisseur. Les agents AO négocient en priorité le temps de réponse et ensuite, si c'est possible, ils négocient le coût de l'information. D'autre part, les agents ACI négocient la reconstruction de leur itinéraire restant en temps réel, en se basant sur leurs contraintes, préférences et priorités.

IV.5.1 Algorithme de Reconstruction Dynamique des Chemins des Agents ACI (ARDyCA)

ARDyCA utilise les notations suivantes:

- Les deux ensembles de nœuds Y et Y'_t correspondant respectivement à l'ensemble des nœuds inscrits au SITM et à l'ensemble des nœuds qui offrent les services demandés à l'instant t , avec $|Y| = J$, $|Y'_t| = J'$, $J' \leq J$ et $Y'_t \subseteq Y$ (§III.2),
- L'ensemble des tâches à réaffecter Φ_t à l'instant t ,
- L'ensemble Ind_t des nœuds indisponibles à l'instant t , avec $Ind_t \subseteq Y$,
- Les PDRI des agents ACI. Un PDRI d'un agent ACI_k contient α_k nœuds et noté par $PDRI_k = (S_{v_1}, \dots, S_{v_{\alpha_k}})$ et $\forall k, S_{v_{\alpha_k+1}} = H$ (le nœud hôte),
- Les PDRD des agents ACI. Un PDRD d'un agent ACI_k à, un instant t , contient β_k nœuds et sur chaque nœud S_{f_j} , il exécute l'ensemble de tâches ψ_{f_j} . Il est noté par $PDRD_{t,k} = (S_{f_1} \{\psi_{f_1}\}, \dots, S_{f_{\beta_k}} \{\psi_{f_{\beta_k}}\})$ et $\forall k, S_{f_{\beta_k+1}} = H$ (le nœud hôte),

- $S_a \prec_{PDR_k} S_b$ signifie que le nœud S_a précède le nœud S_b dans le plan de route PDR_k ;
- $S_a \langle PDR_k$ signifie que le nœud S_a fait partie du plan de route PDR_k ;
- $S_a \notin PDR_k$ signifie que le nœud S_a ne fait pas partie du plan de route PDR_k ;
- $PDR_{t,k} = PDR_{t,k} \oplus S_j(PDRI_k)$ signifie l'ajout du nœud S_j à $PDR_{t,k}$, dans sa position préprogrammée dans le plan de route $PDRI_k$, sachant que $PDR_{t,k}$ est déduit à partir de $PDRI_k$;
- $PDR_k = PDR_k \oplus_{\prec_u, PDR_k} S_j$ signifie l'ajout du nœud S_j juste après le nœud S_u dans le plan de route PDR_k ;
- $PDR_{t,k} = PDR_{t,k} \oplus S_{x_j} \{\Psi_{x_j}\}$ signifie l'ajout de Ψ_{x_j} à l'ensemble de tâches à exécuter sur le nœud S_{x_j} déjà programmé à l'instant t dans le plan de route $PDR_{t,k}$;
- $PDR_k = PDR_k \ominus S_j$ signifie la suppression du nœud S_j de la fin du plan de route PDR_k ;
- $Pos_{t,k}$ représente la position courante de l'agent ACI_k à l'instant t ;
- Ω_t correspond à l'ensemble d'agents ACI qui n'ont encore pas fini leur route à l'instant t avec $|\Omega_t| \leq m'$. A titre de rappel, m' correspond au nombre d'agents ACI déduit suite à la deuxième étape d'optimisation par l'instance FeTAR générée du système (§III.4.2).

IV.5.2 L'algorithme ARDyCA

Les agents ACI doivent ignorer les nœuds indisponibles dans leur PDRDs restant respectif. Cependant, ces agents doivent trouver des nœuds de rechange pour exécuter les tâches en attente de fournisseur. Par conséquent, les PDRDs restants doivent, dans la mesure du possible, être mis à jour selon les positions courantes des agents ACI par une réaffectation dynamique de nouveaux nœuds à l'ensemble des tâches Φ_t . Cette nouvelle affectation dépend aussi bien des positions courantes des agents ACI dans leur PDRD que de leurs contraintes, priorités et préférences. La nouvelle affectation est représentée par un contrat entre les agents ACI et les agents AO , qui peut être accepté totalement ou partiellement ou encore il peut être refusé par chaque agent ACI . Sachant que Ω_t est mis à jour en temps réel dans un autre processus, l'algorithme ARDyCA de réattribution dynamique des tâches est décrit comme suit:

Algorithme ARDyCA

Procéder par la construction de la proposition de chaque agent participant $ACI_k \in \Omega_t$ selon le tour de négociation en cours, depuis le déclenchement du protocole (Etape = 1) ; le passage d'une étape à une autre se fait après un nombre défini de tours de négociation :

Si $\Omega_t \neq \emptyset$ et $\Phi_t \neq \emptyset$ alors faire

Selon la valeur de Etape faire:

Etape est égale à 1 : Pour $\forall ACI_k \in \Omega_t$ faire:

Construire le contrat selon les *priorités* de l'agent ACI_k

Etape est égale à 2 : Pour $\forall ACI_k \in \Omega_t$ faire:

Construire le contrat selon les *préférences* de l'agent ACI_k

Etape est égale à 3 : Pour $\forall ACI_k \in \Omega_t$ faire:

Construire le contrat selon les *contraintes* de l'agent ACI_k

Incrémenter Etape

Si Etape est égal à 4 alors faire

Etape =1

IV.5.2.1 Construction du contrat selon les priorités

Algorithme ARDyCA_priorités

Pour $\forall S_{x_j} \langle P2DR_{t,k}$ par décrémentation de j de Δ_k à 1 faire :

Si $S_{x_j} \notin Ind_t$ alors faire

Chercher l'ensemble des tâches $\psi_{x_j} \subseteq \Phi_t / S_{x_j}$ propose des services à ψ_{x_j}

Si $\psi_{x_j} \neq \emptyset$ alors faire

$$P2DR_{t,k} = P2DR_{t,k} \oplus S_{x_j} \{ \Psi_{x_j} \}$$

Le PDRD restant d'un agent ACI_k à l'instant t est noté par $P2DR_{t,k} = (S_{x_1} \{ \psi_{x_1} \}, \dots, S_{x_{\Delta_k}} \{ \Psi_{x_{\Delta_k}} \})$ avec Ψ_{x_j} correspond à l'ensemble des tâches à exécuter sur le nœud S_{x_j} avec $1 \leq j \leq \Delta_k$. L'idée de l'algorithme **ARDyCA_priorités** est d'explorer les nœuds appartenant au PDRD restant de chaque agent ACI , dans le but de chercher les fournisseurs disponibles qui proposent des services à l'ensemble des tâches Φ_t en attente de

réaffectation à l'instant t . Nous proposons d'explorer la séquence du PDRD restant à partir du dernier nœud qui le compose pour augmenter les chances d'exécuter l'ensemble des tâches en attente de réaffectation le plus rapidement possible.

IV.5.2.2 Construction du contrat selon les préférences

Nous rappelons que α_k correspond au nombre de nœuds du PDRI de l'agent ACI_k . Nous utilisons spécialement dans cette partie, l'algorithme CoSPA (§III.4.1.2) dans le but d'optimiser les itinéraires restants:

Algorithme ARDyCA_préférences
<p>Pour $\forall S_{v_j} \langle PDRI_k$ par incrémentation de j de 1 à α_k faire :</p> <div style="border-left: 1px solid black; padding-left: 10px; margin-left: 20px;"> <p>Si $S_{v_j} \langle P2DR_{t,k}, S_{v_j} \notin Ind_t$ et $Pos_{t,k} \prec_{PDRI_k} S_{v_j}$ alors faire</p> <div style="border-left: 1px solid black; padding-left: 10px; margin-left: 20px;"> <p>Chercher l'ensemble des tâches $\psi_v \subseteq \Phi_t / S_{v_j}$ propose des services à ψ_{v_j}</p> <p>Si $\psi_{v_j} \neq \emptyset$ alors faire</p> <div style="border-left: 1px solid black; padding-left: 10px; margin-left: 20px;"> <p>$P2DR_{t,k} = P2DR_{t,k} \oplus S_j(PDRI_k)$</p> <p>$P2DR_{t,k} = P2DR_{t,k} \oplus S_{v_j} \{ \Psi_{v_j} \}$</p> </div> </div> </div> <p>Réorganiser la séquence de nœuds du parcours $P2DR_{t,k}$ selon CoSPA (§III.4.1.2)</p>

L'idée est de chercher les nœuds encore disponibles, qui n'appartiennent pas au PDRD restant d'un agent ACI mais qui appartiennent à son PDRI. Si un nœud donné respecte toutes ces contraintes en succédant le nœud courant, alors il est ajouté au PDRD restant, dans sa position préprogrammée du premier niveau d'optimisation (Algorithme CoSPA, §III.4.1.2).

IV.5.2.3 Construction du contrat selon les contraintes

Algorithme ARDyCA_contraintes
<p>Pour $\forall S_j \in Y / S_j \notin Ind_t$ et $S_j \langle P2DR_{t,k}$ faire:</p> <div style="border-left: 1px solid black; padding-left: 10px; margin-left: 20px;"> <p>Si $\exists S_{x_j} \langle P2DR_{t,k} / S_{x_j} \notin Ind_t$ et $Contraintes(x_j, j)$ est égal à <i>faux</i> alors faire</p> <div style="border-left: 1px solid black; padding-left: 10px; margin-left: 20px;"> <p>Chercher l'ensemble des tâches $\Psi_j \subseteq \Phi_t / S_j$ propose des services à Ψ_j</p> <p>Si $\psi_{v_j} \neq \emptyset$ alors faire</p> <div style="border-left: 1px solid black; padding-left: 10px; margin-left: 20px;"> <p>$P2DR_{t,k} = P2DR_{t,k} \oplus_{\prec_{x_j, P2DR_{t,k}}} S_j \{ \Psi_j \}$</p> </div> </div> </div>

L'idée de l'algorithme **ARDyCA_contraintes** est de chercher n'importe quel nœud du RETM, encore disponible et qui n'appartient pas au PDRD restant d'un agent ACI. Si un

nœud donné respecte ces conditions, il est ajouté au PDRD restant dans une position appropriée à l'agent ACI concerné. Par exemple, nous pouvons citer la contrainte de la latence réseau; dans ce cas, un nœud S_j du RETM peut être ajouté au PDRD restant si et seulement si $d(S_{x_j}, S_j) \approx 0$ et $d(S_j, S_{x_{j+1}}) \approx 0$.

Le protocole de négociation proposé se base sur une sémantique de niveau élevé qui dépasse les limites d'une communication classique. Cette difficulté nous a incités à concevoir une bibliothèque extensible de vocabulaire et de sémantique spécialisés pour le protocole de négociation proposé. Cette bibliothèque est extensible en une ontologie dynamique pour le domaine du Transport que nous enrichirons et sur laquelle nous nous baserons pour nos travaux futurs.

IV.6 Conception d'une ontologie de Transport

IV.6.1 Définition

A l'origine, l'ontologie est une notion philosophique de la métaphysique générale qui représente l'étude de l'être en tant qu'être dans le contexte de la recherche des fondements premiers. Dans le domaine médical, l'ontologie s'intéresse à la genèse des maladies.

En informatique, l'objectif premier d'une ontologie est de modéliser un ensemble de connaissances dans un domaine donné. La connaissance est alors structurée en un ensemble organisé de concepts, reliés par des liens sémantiques ainsi que par des liens de composition et d'héritage au sens objet. Une ontologie représente donc une organisation hiérarchique de la connaissance sur un ensemble d'objets par leur regroupement en sous-catégories suivant leurs caractéristiques essentielles.

Dans un SMA, les agents interagissent via des langages de communication (FIPA-ACL, KQML, etc.) dans le but de se révéler des informations en s'envoyant des messages. Dans un message lié à un langage de communication donné, une information est représentée sous forme d'une expression de contenu, facile à transférer, codée avec le format approprié (chaîne de caractère, séquence d'octets...) et conforme à un langage de contenu, qui peut être compréhensible ou pas par l'être humain (SL, LEAP, etc.). Cependant, chaque agent peut avoir sa propre disposition de représenter l'information qui n'est pas toujours évidente à manipuler d'où la notion de l'ontologie qui a pour but de faciliter la manipulation des messages échangés entre agents.

Une ontologie définit un vocabulaire et un ensemble de relations qui relient les différents éléments de ce vocabulaire, ce qui représente une solution adéquate au protocole de négociation proposé, qui dépasse les limites d'une communication agent traditionnelle.

IV.6.2 L'apport d'un support ontologique pour les processus de négociation

L'apport d'un support ontologique pour les processus de négociation a été établi par [Bravo et al., 05] qui introduit un module sémantique de désambiguïté interagissant avec une ontologie pour transformer les messages non compréhensibles en des messages compréhensibles. Lorsqu'un agent reçoit une expression qu'il ne comprend pas, le module lui procure l'ensemble des mots qui lui sont sémantiquement liés. Les résultats expérimentaux permettent donc de prouver l'apport d'un support ontologique grâce à une exécution en deux étapes, en se basant sur des choix de stratégies entre les différents acteurs de la négociation ; la première étape engage les agents dans un processus de négociation sans activer le module de désambiguïté sémantique. Au bout de cette première étape, le nombre des messages de non compréhension entre les agents est assez important. Les mêmes stratégies sont employées dans la deuxième étape mais en activant cette fois le module de désambiguïté sémantique. Tous les messages de non compréhension sont alors supprimés.

Les autres travaux, proposant des systèmes de support pour la négociation, ne prennent pas en considération la complexité des phénomènes réels de négociation. Nous présentons quelques travaux réalisés dans le paragraphe qui suit.

IV.6.3 Travaux antérieurs

Des bibliothèques d'outils de structuration spécialisés ont été élaborés par [Kim et al., 03] pour l'implémentation des systèmes de négociation. Il existe aujourd'hui de différents systèmes de support pour la négociation utilisés comme c'est le cas pour SmartSettle³³ qui utilise un serveur central pour accomplir les accords entre agents sans exposer les données confidentielles. Nous pouvons également citer l'exemple de INSPIRE [Kersten et al., 97] qui est un système de négociation basé sur le web facilitant la spécification et l'évaluation des préférences.

³³ <http://www.oneaccordinc.com>

L'un des plus anciens sites Web où peuvent se rencontrer et négocier acheteurs et vendeurs est Kasbah [Maes et al., 96] qui permet aux utilisateurs de créer des agents autonomes pouvant vendre et acheter des services ou des biens pour leur propre compte. L'idée d'une place de marché web vise à faciliter l'interaction entre les agents qui communiquent via un langage spécifique se basant sur un ensemble prédéfini de méthodes. Dans Kasbah, les agents vendeurs essaient de se vendre en se déplaçant sur la place du marché web et en contactant les agents acheteurs intéressés. Les agents acheteurs et vendeurs entament alors une phase de négociation pour trouver le meilleur accord.

MAGMA [Tsvetovaty, 97] est un autre exemple d'architecture Web pour le e-Commerce qui inclut l'infrastructure nécessaire pour diriger et accomplir des actions commerciales sur internet, elle adopte le mécanisme Vickrey comme stratégie de négociation. MAGMA permet de supporter différentes formes de transactions humaines automatiques et contrôlées en intégrant différents agents commerçants, un serveur publicitaire et une banque. Les agents commerçants sont responsables pour l'achat et la vente de biens. Le serveur publicitaire fournit un ensemble ordonné d'affiches publicitaires qui inclut la recherche et le retrait de publicités par catégorie. Enfin, la banque fournit un ensemble basique de services bancaires comme la consultation de comptes ou les virements. Tous les agents communiquent via des sockets de connexion. Il s'agit d'une interface logicielle avec les services du système d'exploitation, grâce à laquelle un développeur exploitera facilement et de manière uniforme les services d'un protocole réseau.

OFFER [Bichler et al., 98] est une architecture d'intermédiaire commercial utilisant l'infrastructure de distribution CORBA qui représente une architecture commune des systèmes objet distribués proposée par l'OMG ; elle permet l'interopérabilité entre objets clients et objets serveurs distribués. OFFER se compose de trois parties principales : les fournisseurs, les clients et les intermédiaires. Un client peut chercher un service directement sur les catalogues proposés par les fournisseurs ou en utilisant l'intermédiaire pour chercher les catalogues de tous les fournisseurs, qui ont été préalablement inscrits à cet intermédiaire.

D'autres travaux similaires existent dans le domaine de l'e-commerce évoqués par [Bravo et al., 05]. Ces travaux ne supportent pas les phénomènes les plus complexes. Cependant, le protocole de négociation que nous proposons implique un dialogue de négociation entre des agents mobiles sur des sites distants et des agents stationnaires du système à travers l'architecture ouverte et dynamique proposée (§III.3). Ce mécanisme de

négociation adopté se base sur une ontologie flexible pour le Transport, présentée dans le paragraphe qui suit.

IV.7 Une ontologie flexible pour le Transport

Dans ce paragraphe, nous proposons une ontologie spécifique à une communication évoluée qui encadre le protocole de négociation proposé (§IV.4) par des niveaux d'ontologie liés.

IV.7.1 Niveaux d'ontologies

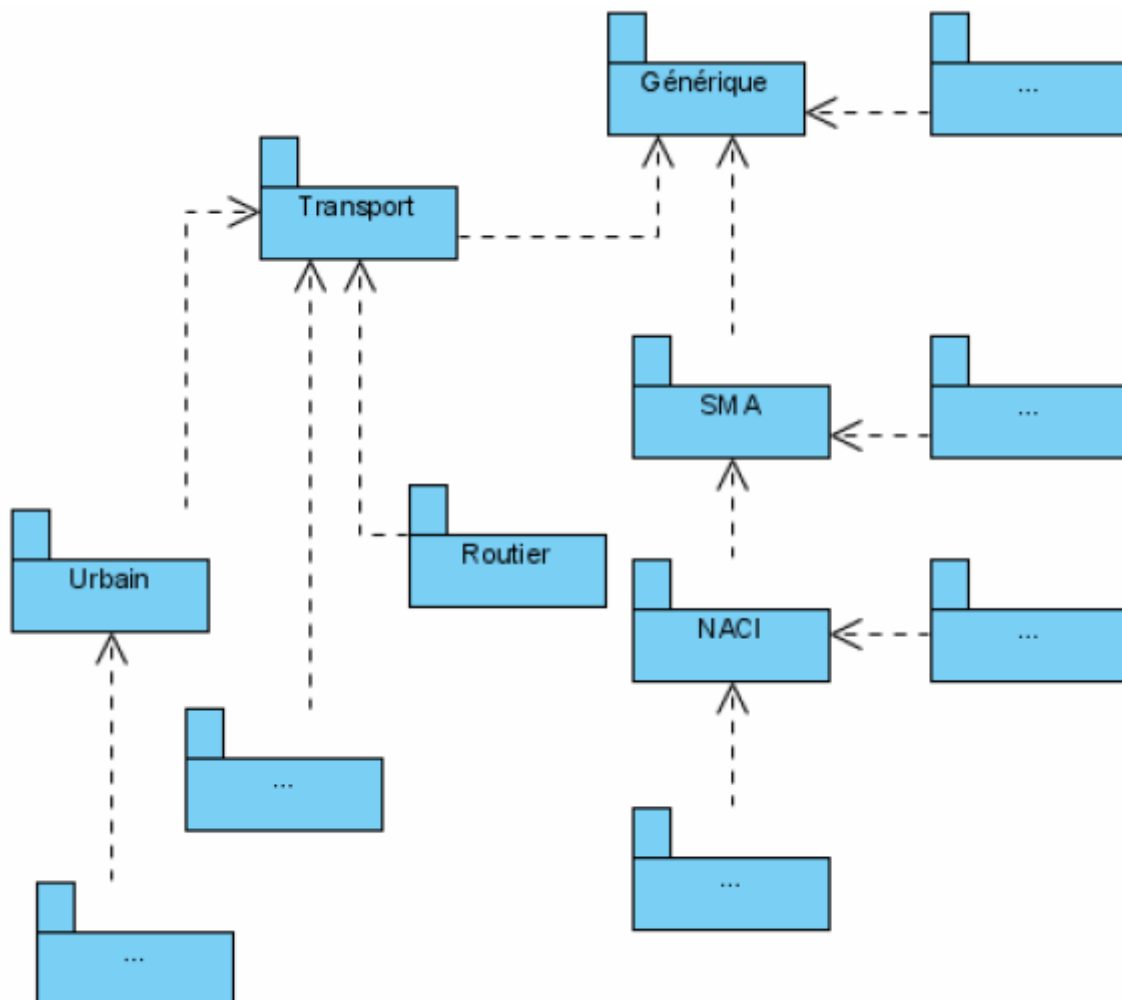


Figure IV.10- Système d'Ontologies Flexible pour de Transport (SOFT)

Le but est de définir un vocabulaire approprié pour l'ensemble du SMA proposé afin d'automatiser les différents types d'échanges entre les agents. Nous proposons donc une ontologie flexible, qui peut s'adapter à tout type possible d'interaction dans le SITM, représentée par un Système d'Ontologies Flexible pour de Transport (SOFT). Ce système est

composé d'un ensemble d'ontologies classées par catégories qui peuvent dépendre l'une de l'autre. SOFT est illustré par le diagramme de composants « packages » de la Figure IV.10 qui représente :

- Une ontologie générique qui se compose d'un ensemble d'éléments génériques pour le transport, pouvant être spécialisés dans différents domaines (routier, urbain...).
- Une ontologie pour le système d'information multi-agent proposé dans le chapitre III qui peut être spécialisé dans une ontologie de Négociation d'Agents ACI (NACI) pour supporter le protocole proposé dans le paragraphe IV.4.

Pour accomplir les contrôles sémantiques appropriés sur une expression donnée d'agent, il est nécessaire de classifier tous les éléments possibles dans le domaine du discours. Par conséquent, nous devons distinguer entre les prédicats et les termes que nous modélisons par des sous-catégories « sous-packages » dépendantes dans chacune des catégories sus-présentées. Cette classification est dérivée du langage de communication d'agent (ACL) défini dans la FIPA qui exige que le contenu de chaque message doit avoir une sémantique appropriée à sa performativité.

IV.7.2 Les prédicats

Les prédicats sont des expressions qui évaluent des faits du monde réel et peuvent donc avoir une valeur *vraie* ou *fausse* (Figure IV.11):

- **Disponible** : informe de la disponibilité d'une société d'agents ou d'un serveur :
 - **Société**: informe l'existence d'une société préalablement créée, disponible dans le système et dont la période d'inactivité correspondante n'est pas encore écoulée.
Exemple : (Disponible (Société : Pt₂))
 - **Serveur** : informe de la disponibilité d'un serveur donné
Exemple : (Disponible (Serveur : S₅₈))
- **Priorité**: informe l'appartenance d'un nœud du RETM aux priorités d'un agent ACI donné

Exemple : (Priorité (ACI : ACI₂) (RefServeur : S₁₀₉))

- **Préférence**: informe l'appartenance d'un nœud du RETM aux préférences d'un agent ACI donné

Exemple : (Préférence (ACI : ACI₁) (RefServeur : S₂₅))

- **Contrainte**: informe l'appartenance d'un nœud du RETM aux contraintes d'un agent ACI donné

Exemple : (Contrainte (ACI : ACI₉) (RefServeur : S₁₀))

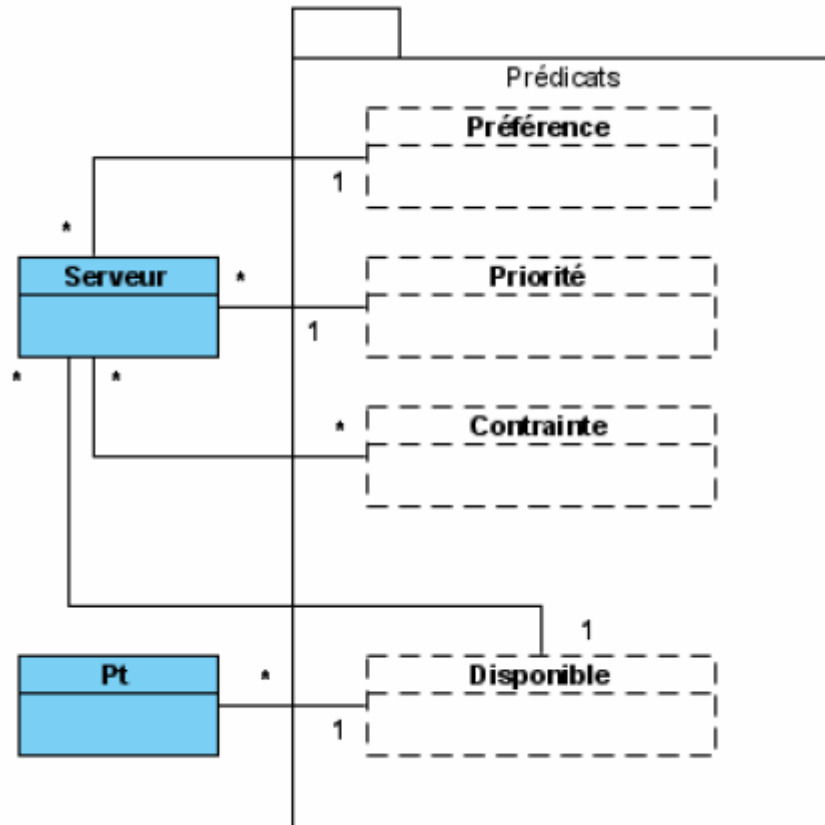


Figure IV.11- Prédicats

IV.7.3 Les termes

Les termes sont des expressions qui identifient les entités (abstraites ou concrètes) qui “existent” dans le monde et sur quoi les agents raisonnent. Nous distinguons les concepts et les actions des agents:

IV.7.3.1 Les concepts

Ce sont des expressions qui représentent des entités (Figure IV.12) avec des structures complexes pouvant être définies en termes de slots:

- **Élément** : représente un élément générique identifié par sa référence unique dans le RETM, il peut être spécialisé en une tâche, un fournisseur, etc.

Exemple : (Element :: Ref : 14 ; Label: Itinéraire_87)

- Tâche : représente une tâche dont le service correspondant peut être proposé par plusieurs fournisseurs

Exemple : (Tâche:: Ref : 2 ; Label: Météo ; Fournisseurs : 2 12 15 3),

- Fournisseur : représente un nœud du RETM qui peut proposer plusieurs services

Exemple : (Fournisseur :: Ref : 15 ; Label: Itinéraires_Zone_54 ; Tâches : 104 55 2 6 17),

- Requête : représente la demande d'un utilisateur formulée par un agent AI. Cette requête peut être décomposée en plusieurs tâches identifiables par le système,

Exemple : (Requête :: AI₅₄ ; RefTâches : 5 6 47 55 1)

- Le $PDRI_k$: représente le plan de route initial d'un agent ACI_k , il est constitué d'un ensemble de nœuds identifiés par leur référence unique. Peut être spécialisé en un PDRD.

Exemple : (PDRI :: ACI_9 ; RefServeurs : 12 5 98 102 35 7 1 29)

- Le $PDRD_{t,k}$: représente le plan de route définitif d'un agent ACI_k à un instant t . Constitué d'un ensemble de nœuds sélectionné du $PDRI_k$ correspondant. Le $PDRD_{t,k}$ spécifie l'ensemble des tâches à exécuter sur chaque nœud.

Exemple : (PDRD :: ACI_9 ; P : t_0 ; temps : t ; RefServeurs : 5 {6 9 1} 98{102 5} 35{6 81} 7{49})

- Φ_t : représente l'ensemble des tâches à réaffecter à l'instant t .

Exemple: (Φ_t :: t ; RefTâches : 59 4 2 19 6 142 3)

- Ind_t : représente l'ensemble des nœuds indisponibles à l'instant t , identifiés par leur référence unique sur le RETM.

Exemple : (Ind :: t ; RefServeurs : 100 13 68 5 14 26)

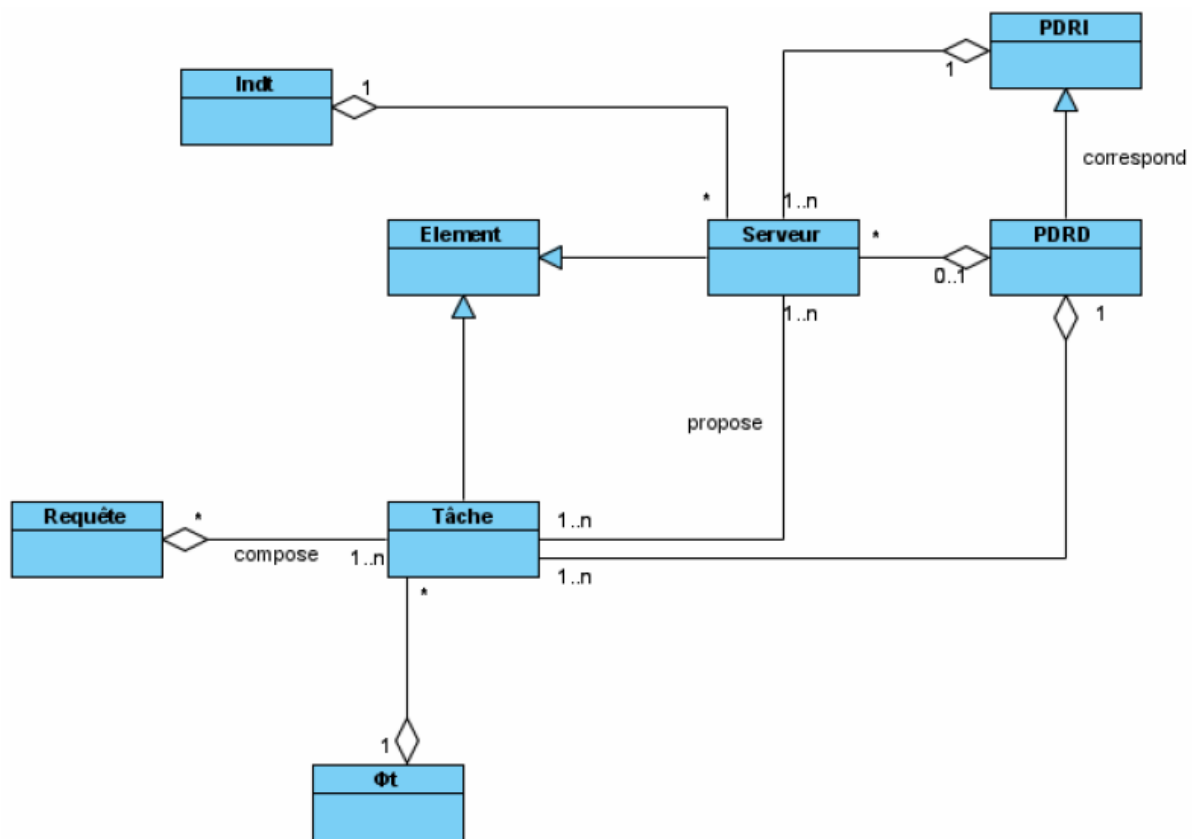


Figure IV.12- Concepts

IV.7.3.2 Les actions des agents

Ce sont des concepts spéciaux qui indiquent des actions qui peuvent être réalisées par des agents (Figure IV.13):

- **AccordPartiel**: une performative dans le processus de négociation qui exprime un accord partiel de la part d'un agent ACI_k envers l'agent AO associé à la société P_t . Cette performative est paramétrée, elle contient l'ensemble des tâches qui ont été annulées par la perturbation du réseau et que l'agent ACI en question n'a pas pu les achever.

Exemple : (AccordPartiel :: ACI_5 ; P : t ; RefTâches : 15 43 102 89)

- **ConfirmationPartielle** : une performative dans le processus de négociation qui exprime une confirmation partielle de la part d'un agent AO, associé à la société P_t , envers un agent ACI_k . Cette performative est paramétrée, elle contient l'ensemble des tâches qui ont été bien prises en compte par l'agent AO en question.

Exemple : (ConfirmationPartielle :: AO ; P : t ; RefTâches : 8 16 12)

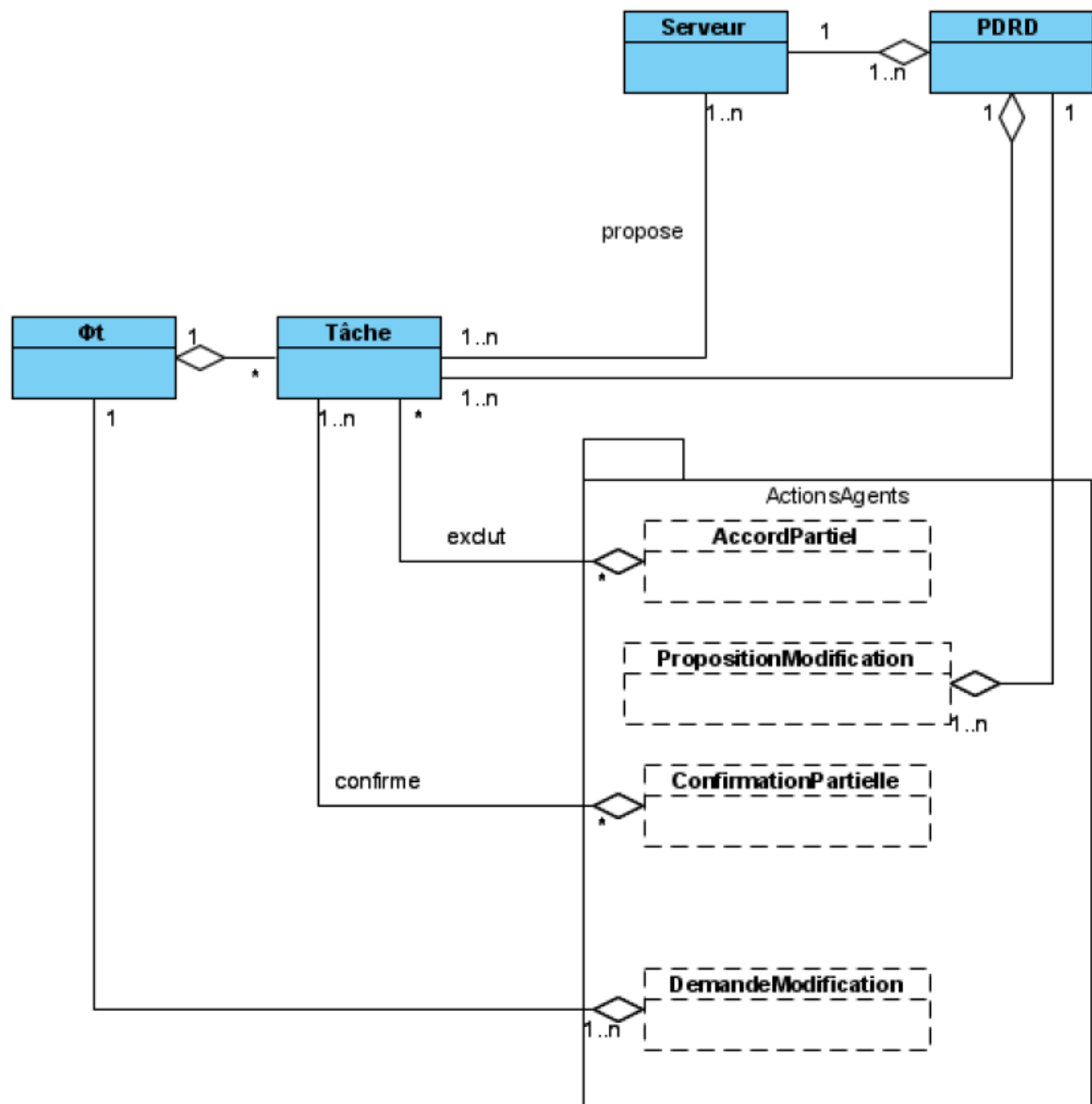


Figure IV.13- Actions d'agents

- DemandeModification: une performative dans le processus de négociation qui correspond à la demande de l'agent AO, issu de la société P_t , envers un agent ACI_k , pour que ce dernier propose une modification de son parcours pour exécuter les tâches en attente de réaffectation, qui représentent le paramètre de cette performative.

Exemple : (DemandeModification :: AO ; P : t_0 ; (Φ_t :: temps-sys ; RefTâches : 89 6 88))

- PropositionModification: une performative dans le processus de négociation qui correspond à la proposition d'un agent ACI_k , pour modifier son PDRD restant. Cette performative est paramétrée, elle contient le PDRD restant proposé.

Exemple: (PropositionModification : (P2DR : ACI₉ ; P : t₀ ; temps : t ; RefServeurs : 5 {6 9 1} 98 {102 5} 35 {6 81} 7 {49}))

IV.7.3.3 Identification des expressions référentielles

Ce sont des expressions qui identifient les entités pour lesquelles un certain prédicat est vrai :

- (tous ? x (Disponible ? (société) x) identifie toutes les sociétés disponibles dans le système
- (tous ? x (Contrainte ? x (RefServeur : S₉))) identifie tous les agents ACI dont le nœud S₉ représente une contrainte
- (tous ? Priorité ? (ACI : ACI₂) x) identifie tous les nœuds qui appartiennent aux priorités de l'agent ACI₂.

IV.7.4 Utilisation des ontologies

Un message agent doit désormais préciser l'ontologie utilisée. Dans le cadre de cette thèse, nous utilisons la sous-catégorie *NACI* dans les messages agents suivants (§IV.4.3):

- < P_t(ACI_k), P_t(AO), ∅, AccordPartiel, ∂, NACI >
- < P_t(AO), P_t(ACI_k), ∅, ConfirmationPartielle, ∂, NACI >
- < P_t(AO), P_t(ACI_k), ∅, DemandeModification, ∂, NACI >

IV.8 Conclusion

Dans ce chapitre, nous proposons une solution pour limiter la dégradation du SITM issue des aléas du RETM. L'approche proposée se base sur une forme de négociation entre les agents mobiles et les agents optimisateurs stationnaires du système. Le but est d'optimiser les réattributions des services aussi bien selon les positions, les contraintes, les préférences et les priorités des agents ACI que selon l'état variable du réseau. Cependant, les caractéristiques complexes d'une telle interaction dépassent les limites des systèmes de négociation traditionnels qui imposent diverses restrictions sur les types et formats des messages de négociation. Nous avons donc proposé une ontologie flexible pour le domaine du transport, essentielle pour une interaction complexe nécessitant un vocabulaire adapté pour l'exécution des contrôles sémantiques appropriés sur une expression donnée d'agent.

CHAPITRE V. SIMULATIONS ET RESULTATS

V.1 Introduction

Dans ce dernier chapitre, nous présentons les différents résultats issus de nos approches d'optimisation, intégrées dans l'architecture distribuée proposée (§III.3). En conséquence, nous présentons dans un premier temps la plate forme de simulation choisie ainsi que les outils utilisés (§V.2 et §V.3) puis nous justifions l'usage du paradigme Agent Mobile dans le SITM (§V.4) et l'apport du Modèle d'Archivage Dynamique des Données (§V.5). Dans le paragraphe V.6, nous présentons les résultats de simulation des fronts optimaux pour le choix de la meilleure solution, basé sur l'approche évolutionnaire adoptée. Enfin, un scénario de simulation complet est illustré dans le paragraphe V.7 qui met en œuvre le protocole de négociation proposé dans cette thèse entre les agents ACI et les agents AO.

V.2 Choix de la plateforme multi-agent

Le besoin de mettre en œuvre des systèmes à plusieurs composantes autonomes nécessite une infrastructure de logiciels utilisée comme environnement pour le déploiement et l'exécution d'un ensemble d'agents. Cette infrastructure est appelée *plate-forme de développement des systèmes multi-agents*. Cependant, l'implémentation de tels systèmes s'avère souvent difficile au niveau de la manipulation de structures de données complexes, de la distribution, de la communication ainsi qu'au niveau des contraintes matérielles imposées. En plus, l'intelligence artificielle est un domaine de recherche extrêmement riche et cette richesse induit une grande complexité et une grande multiplicité des approches proposées, ce qui conduit à de très nombreux modèles d'agents, d'environnement, d'interactions et d'organisations. Ces modèles sont souvent combinés au sein d'un même système multi-agent. Ainsi, le mieux est de choisir une plateforme multi-agent adaptée aux contraintes du système à mettre en œuvre. Plusieurs plates-formes multi-agents existent, telles que MadKit, JADE, ZEUS, AgentBuilder, Jack, etc.

Pour la sélection de la plateforme, nous avons négligé les critères insignifiants telles que la difficulté d'apprentissage ou la non disponibilité des sources. Néanmoins, nous avons souligné quelques critères importants :

- La possibilité d'implémenter des systèmes relativement complexes ;
- La flexibilité : éviter les plates-formes qui supportent une méthodologie particulière ;

- L'accélération de développement grâce à la présence suffisamment importante de briques logicielles pour pouvoir produire une application aboutie ;
- Le traitement distribué et notamment la présence d'un support pour le paradigme AM ;
- Possibilité d'intégration des services web.

Les deux plate-formes qui ne spécifient aucune méthodologie et peuvent être considérées comme des « frameworks »³⁴, sont JADE et Jack mais JADE l'emporte avec plusieurs autres caractéristiques intéressantes telles que la possibilité d'intégration des Web services et l'existence d'un bon support de langages de contenu et d'ontologies.

Ainsi, pour le développement du SITM et la simulation des résultats de nos approches d'optimisation distribuée, nous avons choisi la plateforme JADE³⁵ (Java Agent Development framework). JADE est un logiciel-médiateur³⁶ «middleware» qui permet une implémentation flexible des Systèmes Multi-Agents communiquant grâce à un transfert efficace des messages ACL (Agent Communication Language), conformes aux spécifications de la FIPA (§II.3.2.2.2). JADE est écrit en Java³⁷, supporte la mobilité, évolue rapidement et fait partie aujourd'hui des rares plateformes multi-agents qui offrent la possibilité d'intégration des services Web [Greenwood, 05]. D'un autre côté, JADE tente de faciliter le développement des applications agent en optimisant les performances d'un système d'agent distribué.

V.3 Outils et paramétrages

Pour apercevoir les communications et les comportements des agents, JADE offre des outils graphiques, qui sont eux même des agents, tels que Le « RMA » (Remote Management Agent) qui représente l'interface principale d'administration ; l'« introspector » qui permet de contrôler le cycle de vie d'un agent, ses messages ACL échangés et ses comportements ; le « sniffer » qui permet de suivre le cheminement des communications entre les différents agents du système (Figure V.1); etc.

³⁴ Un ensemble de bibliothèques permettant le développement rapide d'applications. Il fournit suffisamment de briques logicielles pour pouvoir produire une application

³⁵ <http://jade.tilab.com>

³⁶ Permet la communication entre des clients et des serveurs ayant des structures et une implémentation différentes

³⁷ www.java.sun.com

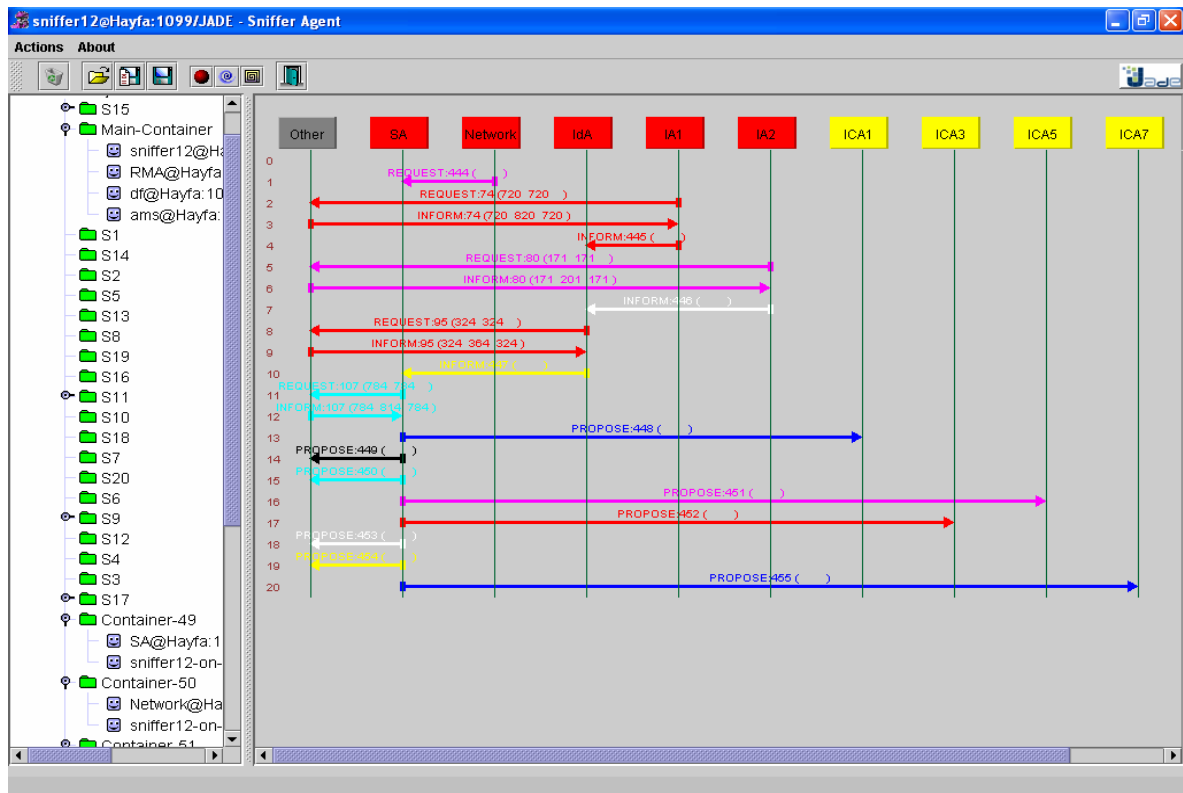


Figure V.1– Communication des agents dans le SITM

ICA	Service	Data	Source
MA3	T 8	S14T8	S14
MA2	T 14	S18T14	S18
MA5	T 1	S4T1	S4
MA5	T 10	S4T10	S4
MA5	T 7	S13T7	S13
MA5	T 12	S13T12	S13
MA4	T 2	S9T2	S9
MA4	T 6	S9T6	S9
MA4	T 13	S19T13	S19
MA1	T 15	S3T15	S3
MA1	T 5	S20T5	S20

Request	TotalCost	TotalTim	DueDate
IA1	16	2.0	2
IA7	9	2.0	2
IA4	34	2.0	2
IA3	8	0.6	2
IA5	5	1.5	2

Number of ICA : 5
Number of Servers : 8

New (nbRequests, nbTasks) = (5 , 11)

Figure V.2– Interface de résultats

Nous avons implémenté une interface graphique java de simulation générale, qui nous permet de récupérer les résultats issus d'un chromosome sélectionné. La Figure V.2 illustre un exemple de résultat de réponse à un ensemble de 5 requêtes simultanées, décomposées en 11 tâches différentes. Chaque requête req_i a été capturée par un agent AI_i pour $1 \leq i \leq 5$ et chaque

tâche identifiée a été servie par un agent ACI, choisi par l'agent AO concerné. Par exemple, l'agent ACI₃ (MA3 sur l'interface) est parti chercher le service T₈ sur le nœud S₁₄ pour récupérer l'information S14T8 et l'agent ACI₄ (MA4 sur l'interface) est parti chercher le service T₂ et T₆ sur le nœud S₉ et le service T₁₃ sur le nœud S₁₉ pour récupérer les informations S9T2, S9T6 et S19T13. Les résultats montrent ici que le chromosome sélectionné respecte le temps maximal de réponse de chaque requête « due date ».

Les données utilisées par le SITM, sont disponibles dans l'annexe A ; les résultats de simulation présentés dans ce chapitre se basent sur un RETM composé de 20 nœuds (J=20) proposant 100 services différents (I=100). Le RETM peut être exploré totalement par 5 agents ACI dotés par les PDRI_i suivants :

$$PDRI_1 = (S_{20}, S_{15}, S_1, S_3) ;$$

$$PDRI_2 = (S_{18}, S_7, S_{10}, S_{17}) ;$$

$$PDRI_3 = (S_2, S_{13}, S_{19}, S_6) ;$$

$$PDRI_4 = (S_{16}, S_{14}, S_5, S_{12}, S_4) ;$$

$$PDRI_5 = (S_{11}, S_8, S_9) ;$$

Le meilleur temps de réponse des agents ACI est $\delta = 16,28$ s et donc le temps maximum moyen estimé de parcours des agents ACI est $\gamma = 16,28 - \sum_{j=1}^{20} CT_j / 20 = 15,86$ s. Cette valeur est additionnée à la somme des temps d'exécution de toutes les tâches confiées à un agent ACI sur les serveurs correspondants, pour le calcul final des PDRDs.

V.4 Justification du choix du paradigme Agent Mobile et son efficacité dans le SITM

Dans le cadre des applications réparties dans les réseaux à grande envergure, l'apparition du paradigme Agent Mobile en 1994 fut une invention bouleversante qui promettait aux industriels des solutions alléchantes pour un meilleur accès à une information répartie et volumineuse. Au fil des années, les avantages de ce paradigme ont longtemps été discutés et il a été enfin prouvé qu'il peut être intéressant dans certains cas [Picco et al.,98] [Buse et al.,03]. En effet, l'efficacité du paradigme Agent Mobile dépend considérablement de l'architecture du système adopté [Lu et al.,03] [Buse et al.,03].

Dans cette partie, nous justifions l'usage de ce paradigme au sein du SITM en le comparant au paradigme classique Client/Serveur [Zgaya et al., 06b] [Picco et al.,98] [Ketel et

al.,05] sachant que la comparaison se base sur les mêmes solutions générées par l'approche d'optimisation adoptée par notre système. Dans ce qui suit, nous justifions la résignation des frais de transmission dans les résultats de simulation.

V.4.1 Frais de transmission de données : un modèle pseudo-formel

Pour le paradigme Client/Serveur (CS), η_{CS} (resp. $\tilde{\eta}_{CS}$) représente la fonction de frais de transmission des données correspondant à un envoi d'un message de requête (resp. réponse). T_{CS} et ρ_{CS} correspondent respectivement au temps de réponse maximum pour une requête donnée et au frais total de transmission, ils sont calculés comme suit:

$$T_{CS} = \sum_{j=1}^{J'} \left(\frac{\sum_{i=1}^{I'} a_{ci,cj} (\eta_{CS} I_{ci,cj} + \tilde{\eta}_{CS} Q_{ci,cj})}{d(H, S_{c_j})} \right) \quad \rho_{CS} = \sum_{j=1}^{J'} \left(\sum_{i=1}^{I'} a_{ci,cj} (\eta_{CS} + \tilde{\eta}_{CS}) \right)$$

Avec :

- $I_{ci,cj}$ et $Q_{ci,cj}$ correspondent respectivement à la taille du message de la requête et à la taille du message de la réponse à cette requête pour la tâche T_{ci} sur le serveur S_{c_j} ;
- T_{ci} et S_{c_j} représentent respectivement le service et le nœud qui correspondent à la $i^{\text{ème}}$ ligne et à la $j^{\text{ème}}$ colonne de la matrice qui symbolise le chromosome FeTAR CH généré par un agent AO avec $1 \leq i \leq I'$ et $1 \leq j \leq J'$ (§III.4.2.2) ;
- $a_{ci,cj}$ est un booléen tel que Si $CH[c_i, c_j] = 1$ ($1 \leq i \leq I'$ et $1 \leq j \leq J'$) alors $a_{ci,cj} = 1$ sinon $a_{ci,cj} = 0$.

Pour le paradigme Agent Mobile (AM), lorsqu'un agent ACI se déplace vers un nœud donné, il ramène avec lui toutes les réponses préalablement récupérées des autres nœuds déjà visités. Ainsi, lorsqu'un agent ACI finit de collecter les données du dernier nœud de son itinéraire, il retourne à son nœud hôte dans le système avec tous les résultats rassemblés.

Pour le paradigme AM, η_{AM} (resp. $\tilde{\eta}_{AM}$) représente la fonction de frais de transmission des données correspondant à un envoi d'un message de requête (resp. réponse). Par conséquent, le frais total de transmission ρ_{AM_k} pour un agent ACI_k est calculé comme suit:

$$\rho_{AM_k} = \sum_{j=1}^{J'} b_{c_j,ck} \eta_{AM} + \tilde{\eta}_{AM}$$

Avec $b_{c_j,ck}$ un booléen tel que: si S_{c_j} appartient au PDRD de l'agent ACI_{ck} ($1 \leq j \leq J'$ et $1 \leq k \leq m'$) alors $b_{c_j,ck} = 1$ sinon $b_{c_j,ck} = 0$. Le temps maximum d'une réponse à une requête donnée correspond au temps maximum de navigation de tous les agents ACI actifs concernés:

$$T_{AM} = \max_{1 \leq k \leq m'} (T(Wk_{ck}) + \rho_{AM_{ck}})$$

Cependant, nous nous intéressons aux systèmes ouverts, fortement sollicités et distribués, ces systèmes sont composés d'un nombre important de nœuds avec un flux intensif de requêtes utilisateurs. Dans ce cas, ρ_{CS} est susceptible d'être plus grand que ρ_{AM} , alors nous ne prenons pas en considération les frais de transmission des données dans les résultats de simulation.

V.4.2 Résultats de simulation

Pour un ensemble de 10 requêtes Δ_ε -simultanées avec $\Delta_\varepsilon = 2s$, formulées à un instant t (Figure V.3), l'agent AId élabore une décomposition en 54 tâches (Table V-1) et identifie 19 fournisseurs d'information, susceptibles d'offrir des services à ces tâches. L'agent AO génère donc la solution FeTAR CH₁ correspondante qui affecte 17 serveurs (parmi les 19 identifiés) aux différentes tâches demandées (Table V-2). Les PDRDs des agents ACI sont élaborés par la Table V-3.

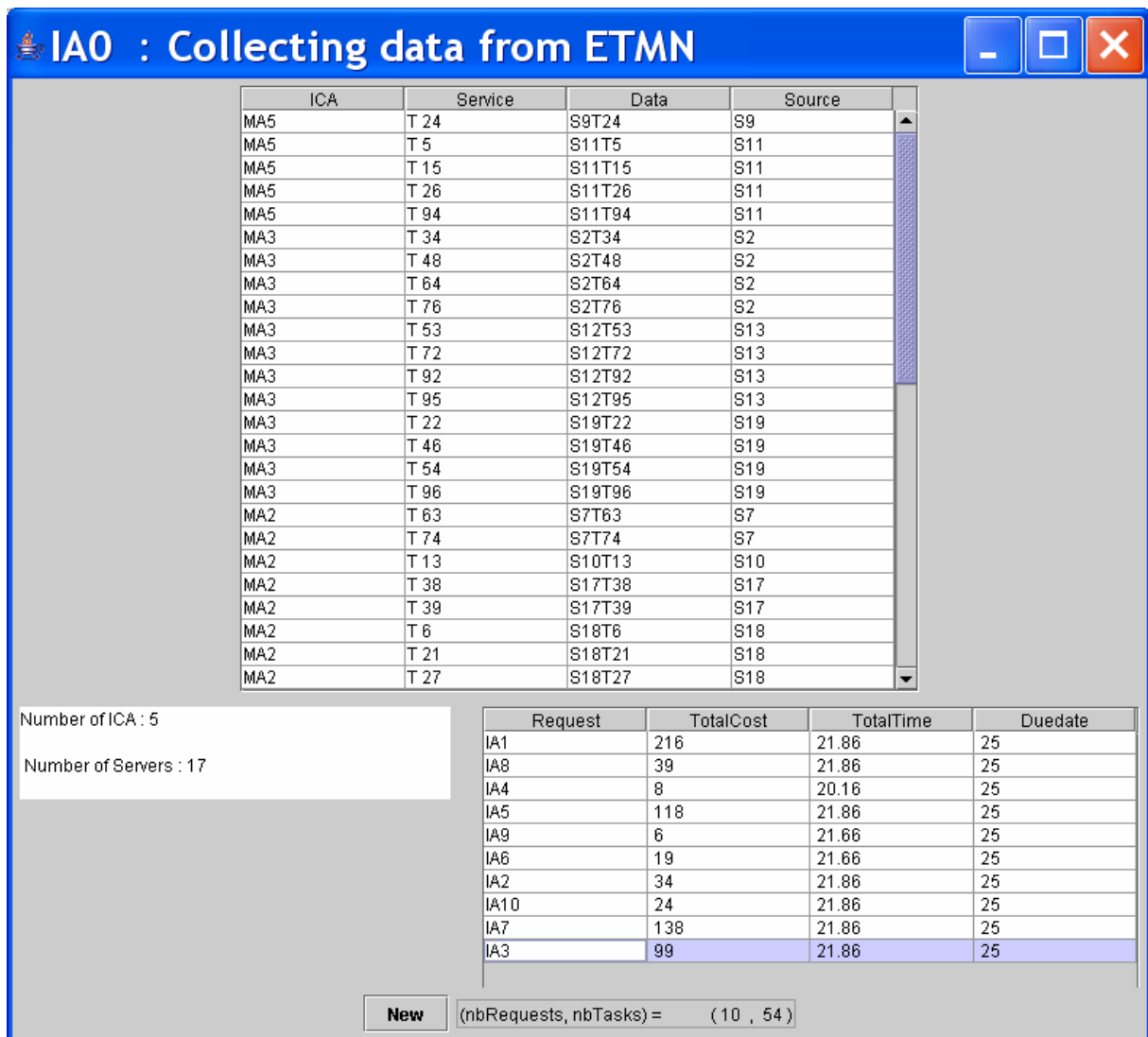


Figure V.3– Résultat de simulation

Quantité de donnée totale transférée : 15,95 unités

	req ₁	req ₂	req ₃	req ₄	req ₅	req ₆	req ₇	req ₈	req ₉	req ₁₀
T ₅	x									
T ₆										x
T ₈	x		x					x		
T ₉	x		x					x		
T ₁₃	x									
T ₁₅	x		x				x			
T ₁₈	x		x							x
T ₁₉	x		x							
T ₂₁	x				x		x			
T ₂₂	x				x		x			
T ₂₃	x				x		x			
T ₂₄	x		x		x		x			
T ₂₆	x		x				x			
T ₂₇	x		x				x			
T ₂₉			x		x					
T ₃₀	x				x					
T ₃₃	x		x		x					
T ₃₄	x		x		x					
T ₃₅	x				x					
T ₃₆	x				x					
T ₃₇	x		x		x		x			
T ₃₈	x		x		x		x			
T ₃₉	x				x		x			
T ₄₁	x						x			
T ₄₂	x						x			
T ₄₆				x	x	x				x
T ₄₈					x					
T ₅₁	x				x		x			
T ₅₃	x						x			
T ₅₄	x						x			
T ₅₉	x				x		x			
T ₆₁	x		x		x		x			
T ₆₃	x		x		x		x	x		
T ₆₄	x				x		x	x		
T ₆₅	x				x					
T ₆₆	x		x		x					
T ₇₂	x	x			x		x			
T ₇₄	x	x					x			
T ₇₆	x	x					x			
T ₇₈	x				x		x			
T ₇₉					x					
T ₈₃	x				x	x	x			
T ₈₄			x		x	x				
T ₈₅	x		x		x		x	x		
T ₈₆	x				x		x	x		
T ₈₇	x						x			
T ₈₈	x		x				x			
T ₉₂	x	x	x				x			
T ₉₃	x	x					x			
T ₉₄	x						x	x		
T ₉₅	x		x				x	x	x	
T ₉₆	x	x	x							
T ₉₉										x
T ₁₀₀	x	x	x						x	

Table V-1- Décomposition des requêtes

CH ₁	S ₁	S ₂	S ₃	S ₄	S ₅	S ₇	S ₈	S ₉	S ₁₀	S ₁₁	S ₁₂	S ₁₃	S ₁₄	S ₁₅	S ₁₆	S ₁₇	S ₁₈	S ₁₉	S ₂₀
T ₅	*	X	*	*	*	*	*	*	*	1	*	X	*	X	*	*	*	*	*
T ₆	*	X	*	*	*	*	*	*	*	*	*	X	X	*	*	*	1	*	*
T ₈	*	X	1	*	X	*	*	*	*	X	*	*	*	*	*	*	*	X	*
T ₉	X	X	X	*	X	X	*	*	*	*	*	*	X	1	X	*	X	X	*
T ₁₃	*	X	*	*	*	X	*	*	1	*	*	X	X	*	*	X	*	*	X
T ₁₅	*	X	*	*	*	X	*	*	*	1	*	*	X	*	*	X	*	*	*
T ₁₈	X	*	*	*	X	X	X	*	*	X	X	X	X	X	X	X	X	*	1
T ₁₉	*	X	X	X	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X
T ₂₁	X	X	*	*	X	X	X	X	X	X	X	X	X	X	X	X	1	X	X
T ₂₂	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	*	X	1	*
T ₂₃	*	X	X	X	X	X	X	*	X	X	X	*	X	X	*	X	X	X	1
T ₂₄	X	X	X	X	X	X	X	1	X	X	X	X	X	X	X	X	X	*	X
T ₂₆	X	*	*	X	X	X	X	X	X	1	*	X	X	X	X	X	*	X	X
T ₂₇	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	*	*
T ₂₉	1	*	X	X	X	X	X	X	X	*	X	*	X	X	X	X	X	X	X
T ₃₀	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	*	1	*	*
T ₃₃	*	X	*	1	X	*	X	X	X	X	*	X	X	X	*	X	X	X	X
T ₃₄	*	1	*	X	X	*	X	X	X	X	X	X	X	X	X	X	X	X	X
T ₃₅	1	X	*	X	X	X	*	X	X	X	X	X	X	X	X	X	*	X	X
T ₃₆	*	X	1	X	X	X	X	X	X	X	X	X	X	X	X	X	*	X	X
T ₃₇	X	X	*	X	X	X	X	X	X	X	X	X	X	X	X	X	*	X	1
T ₃₈	*	X	*	X	X	X	X	X	*	X	X	X	*	X	X	1	X	X	X
T ₃₉	X	X	X	X	X	X	X	X	*	X	X	X	X	X	X	1	X	X	*
T ₄₁	X	X	*	X	*	X	X	X	X	X	X	X	X	X	X	X	1	X	X
T ₄₂	*	X	*	X	X	X	X	X	X	*	X	X	X	X	X	X	1	X	X
T ₄₆	X	X	X	X	X	X	X	*	X	X	X	X	X	X	X	X	1	X	X
T ₄₈	X	1	*	*	X	X	X	X	X	X	X	X	X	X	X	X	X	*	X
T ₅₁	X	X	1	X	X	X	X	X	X	X	X	X	X	X	X	X	*	X	X
T ₅₃	*	X	X	X	X	X	X	X	X	X	X	1	X	X	*	X	X	X	X
T ₅₄	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	X
T ₅₉	*	*	X	X	X	X	X	X	X	1	X	*	X	X	X	X	X	X	X
T ₆₁	X	X	*	X	X	X	X	X	X	X	X	X	1	X	*	X	X	X	X
T ₆₃	*	X	*	*	*	1	*	*	X	X	X	X	X	X	*	X	*	X	X
T ₆₄	*	1	*	*	*	X	X	*	X	X	X	X	X	X	X	X	X	X	X
T ₆₅	*	X	X	X	X	X	X	X	X	X	X	X	1	X	X	X	*	X	X
T ₆₆	1	X	*	X	X	X	X	X	X	X	X	X	X	*	X	X	X	X	X
T ₇₂	*	X	*	X	X	X	X	X	*	X	X	1	X	X	X	X	*	X	X
T ₇₄	X	X	X	X	X	1	X	X	X	X	X	*	X	X	*	X	X	X	X
T ₇₆	X	1	*	X	X	X	X	X	X	X	X	X	X	X	X	X	*	X	X
T ₇₈	X	*	X	X	X	X	X	X	X	X	1	X	X	X	X	X	X	X	X
T ₇₉	1	*	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
T ₈₃	X	X	X	X	*	*	X	X	X	X	1	X	*	X	X	X	X	X	X
T ₈₄	1	*	*	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
T ₈₅	*	X	1	X	X	X	X	X	X	X	X	X	X	X	X	X	*	X	X
T ₈₆	*	X	*	X	1	X	X	X	X	X	X	X	X	X	X	X	*	X	X
T ₈₇	X	X	*	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1
T ₈₈	1	X	*	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
T ₉₂	X	X	*	X	X	X	X	X	X	X	1	X	X	X	*	*	X	X	X
T ₉₃	*	X	*	X	1	X	X	*	X	X	X	X	X	X	X	X	X	X	X
T ₉₄	X	X	X	X	X	X	X	X	X	1	*	*	*	X	X	X	*	X	X
T ₉₅	X	X	X	X	*	X	X	*	X	*	X	1	X	X	X	X	X	X	X
T ₉₆	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	X
T ₉₉	*	X	X	X	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X
T ₁₀₀	X	X	1	X	*	X	X	X	X	X	X	X	X	X	X	X	X	X	*

Table V-2- Chromosome CH₁

	S ₁	S ₂	S ₃	S ₄	S ₅	S ₇	S ₈	S ₉	S ₁₀	S ₁₁	S ₁₂	S ₁₃	S ₁₄	S ₁₅	S ₁₆	S ₁₇	S ₁₈	S ₁₉	S ₂₀
ACI ₁	x		x											x					x
ACI ₂						x			x							x	x		
ACI ₃		x										x							x
ACI ₄				x	x						x		x		x				
ACI ₅							x	x		x									

Table V-3– Table des PDRDs

Le chromosome généré (Table V-2) a été testé indépendamment par les deux paradigmes AM et CS. Le résultat obtenu montre que la réponse à l'ensemble des requêtes par le paradigme AM, selon la solution CH₁, nécessite au maximum 21,86 s de temps de réponse pour un ensemble de requêtes alors que le paradigme CS nécessite au maximum 108,95 s de temps pour le même ensemble de requêtes (Figure V.4).

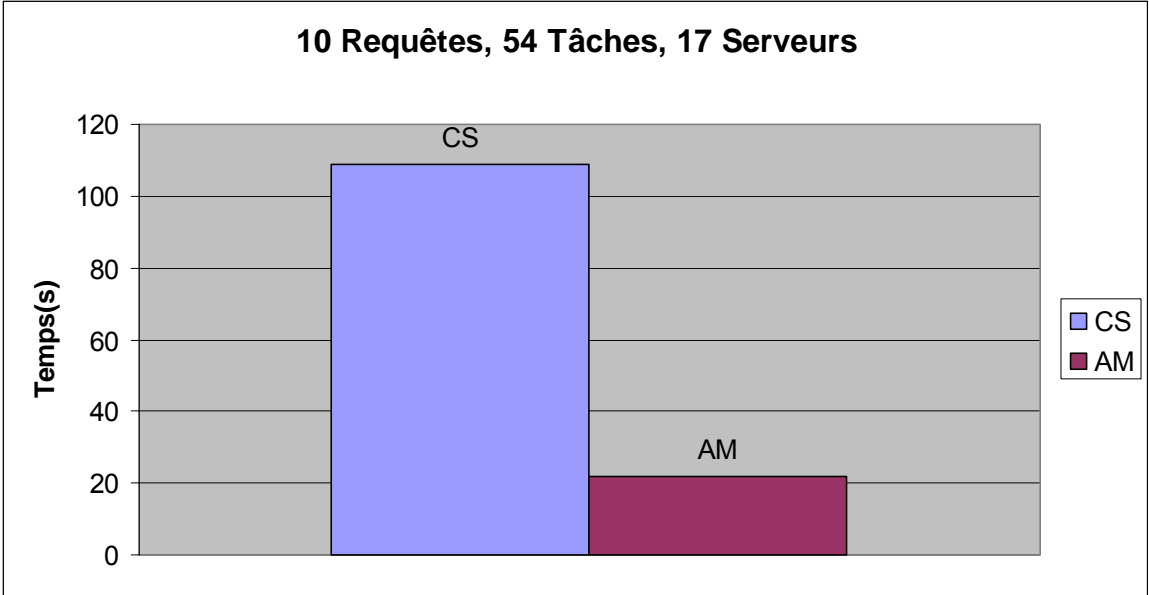


Figure V.4 – Comparaison des paradigmes AM et CS pour une seule instance de FeTAR

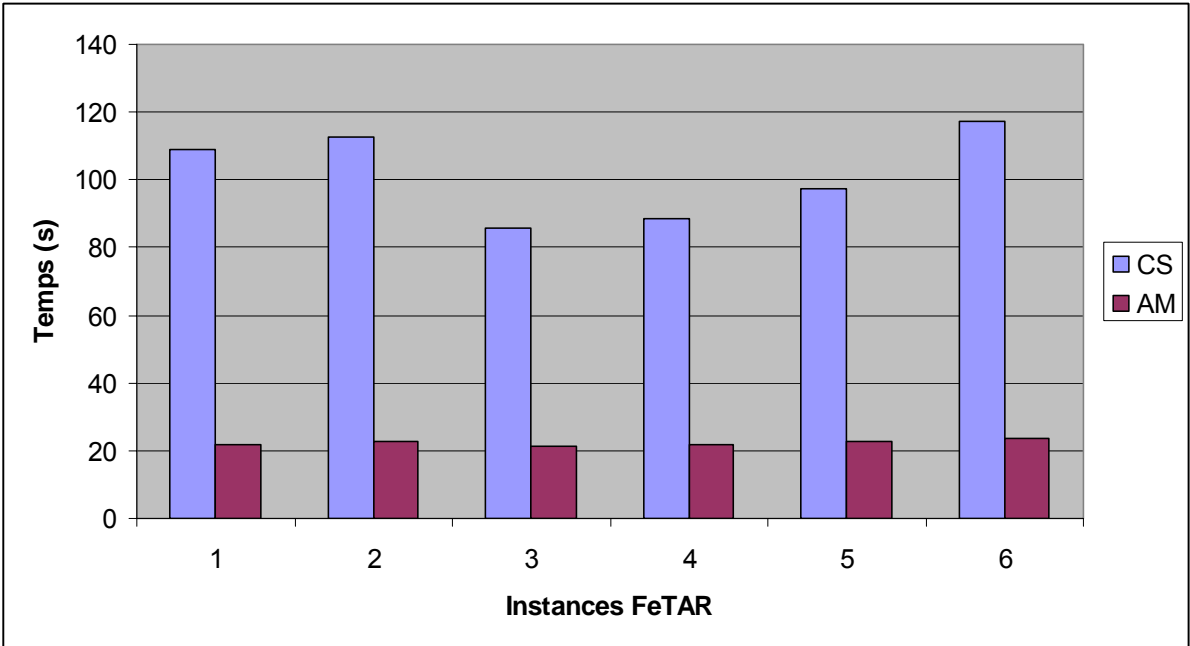


Figure V.5 - Variation des instances FeTAR

Pour le même exemple, nous avons généré un ensemble de solutions FeTAR (annexe B); les résultats montrent les avantages de l'utilisation du paradigme AM dans notre approche d'optimisation au sein du SITM par rapport au paradigme CS (Figure V.5). En outre, la variation de la taille des données des réponses (Annexes B et C) montre une meilleure performance du paradigme AM lorsque la taille des données devient importante (Figure V.6). Ceci prouve la pertinence du choix du paradigme AM dans les systèmes d'information à large transfert des flux de données, ce qui s'adapte bien au SITM. La courbe correspondante n'est pas régulière car la résolution du problème se situe dans le cas général des débits aléatoires dans le RETM.

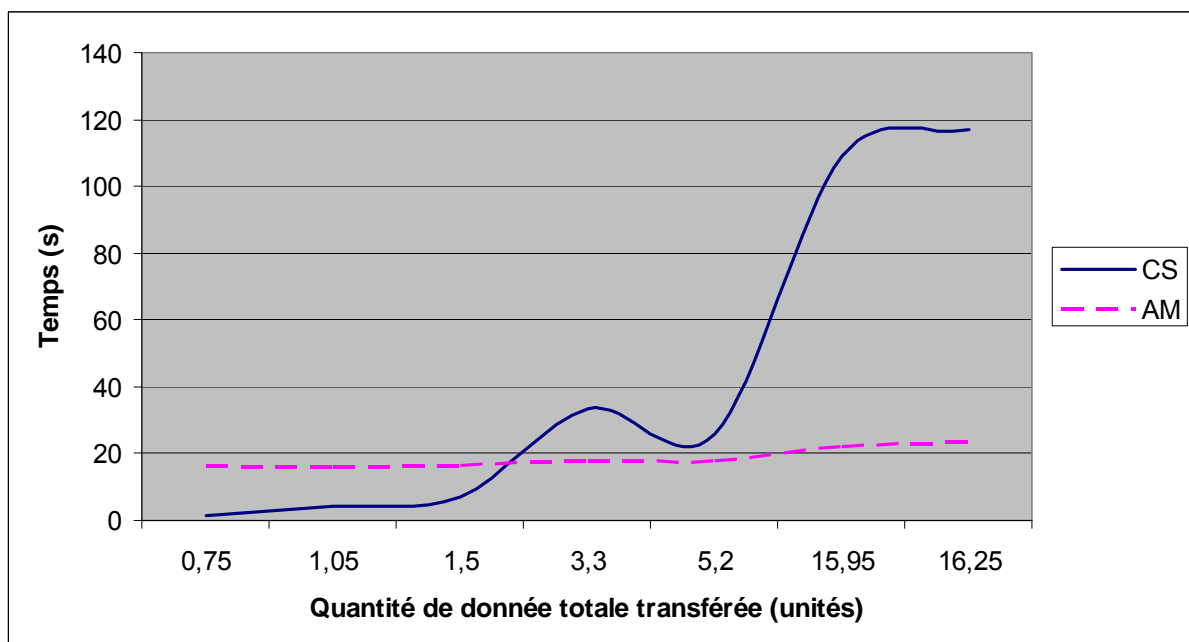


Figure V.6- Variation de la taille de données des réponses

V.5 L'apport du Modèle d'Archivage Dynamique des Données (MADD) dans le SITM

Nous rappelons que le premier niveau d'optimisation (§III.4.1) permet de préparer les PDRI des agents ACI pour explorer la totalité du RETM par 5 agents ACI (§V.3) comme suit :

$$PDRI_1 = (S_{20}, S_{15}, S_1, S_3) ;$$

$$PDRI_4 = (S_{16}, S_{14}, S_5, S_{12}, S_4) ;$$

$$PDRI_2 = (S_{18}, S_7, S_{10}, S_{17}) ;$$

$$PDRI_5 = (S_{11}, S_8, S_9) ;$$

$$PDRI_3 = (S_2, S_{13}, S_{19}, S_6) ;$$

Pour un ensemble de requêtes émises à l'instant t , une solution CH₂ est sélectionnée (Table V-4) afin de déduire les PDRDs des agents ACI à partir des PDRIs comme suit ($m'=4$ et $J'=7$):

$$\begin{aligned} \text{PDRD}_{t,1} &= (S_{15}\{T_7\}, S_3\{T_{13}\}); & \text{PDRD}_{t,4} &= (S_5\{T_5\}, S_{12}\{T_3\}); \\ \text{PDRD}_{t,2} &= (S_{18}\{T_1\}, S_{17}\{T_6, T_{10}\}); & \text{PDRD}_{t,5} &= \emptyset \\ \text{PDRD}_{t,3} &= (S_{13}\{T_{12}\}); \end{aligned}$$

CH ₂	S ₁	S ₂	S ₄	S ₅	S ₆	S ₇	S ₈	S ₉	S ₁₀	S ₁₁	S ₁₂	S ₁₃	S ₁₄	S ₁₅	S ₁₆	S ₁₇	S ₁₈	S ₁₉	S ₂₀
T ₁	*	*	*	*	*	*	*	*	*	*	*	X	*	*	*	X	*	*	X
T ₃	*	*	*	*	X	*	*	*	*	*	1	X	*	*	*	X	*	*	*
T ₅	*	*	*	1	X	*	*	*	*	*	*	X	*	X	*	*	*	*	*
T ₆	*	*	*	*	X	*	*	*	*	*	*	X	X	*	*	1	*	*	*
T ₇	X	*	*	X	X	X	*	*	X	X	*	*	*	1	X	X	*	*	*
T ₁₀	X	X	*	*	X	X	X	*	*	*	X	X	*	X	*	1	X	*	X
T ₁₂	*	*	*	*	X	X	X	X	*	*	X	1	X	*	*	*	*	X	*
T ₁₃	*	1	*	*	X	X	*	*	*	*	*	X	X	*	*	X	*	*	X

Table V-4 - Chromosome CH₂

Les indicateurs (Tableau V-1) des services identifiés de l'exemple en cours montre qu'au meilleur des cas, le nombre des agents ACI peut être réduit à un seul agent visitant 2 nœuds pour traiter 3 services (Figure V.7). Les PDRDs des agents ACI sont réduits comme suit :

$$\begin{aligned} \text{PDRD}_{t,1} &= \emptyset; & \text{PDRD}_{t,4} &= \emptyset; \\ \text{PDRD}_{t,2} &= (S_{18}\{T_1\}, S_{17}\{T_6, T_{10}\}); & \text{PDRD}_{t,5} &= \emptyset \\ \text{PDRD}_{t,3} &= \emptyset; \end{aligned}$$

Tâche	IndTemp	IndMaj
T ₁	0	-
T ₃	1	$12h \leq x = 12h$
T ₅	3	-
T ₆	0	-
T ₇	2	$60 \text{ jours} > x = 12h$
T ₁₀	0	-
T ₁₂	3	-
T ₁₃	3	-

Tableau V-1- Exemple d'indicateurs

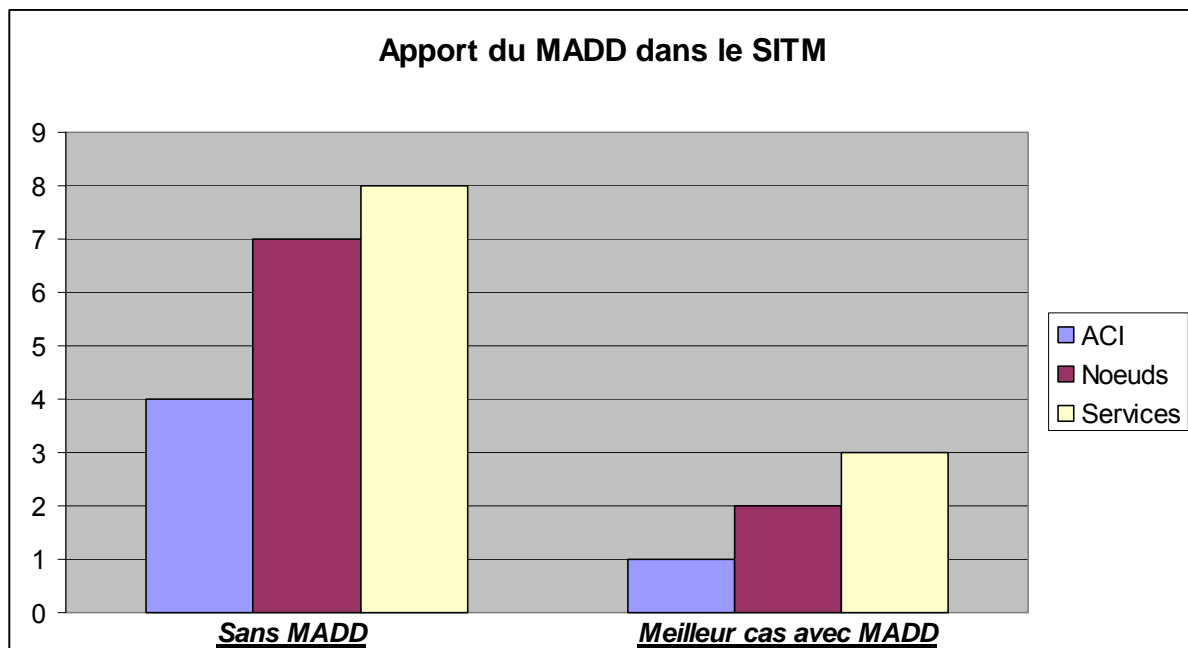


Figure V.7– Exemple d’amélioration du SITM par le MADD

Grâce au MADD, le flux de données peut être considérablement réduit, ce qui permet de limiter l’utilisation des ressources au niveau du SITM : navigation des ACI, accès aux serveurs et transfert de données.

V.6 L’approche évolutionnaire : choix de la meilleure solution

L’approche évolutionnaire adoptée (§III.4.2) pour la recherche et la composition des services, permet la sélection de la meilleure solution générée représentée par un chromosome instance de FeTAR.

La Figure V.8 représente le front Pareto-optimal obtenu avec $\varepsilon_1 = \varepsilon_2 = \varepsilon = 0.75$ pour l’exemple du paragraphe précédent (§V.5) sachant que les paramètres du programme ont été ajustés comme suit:

- Probabilité de croisement $p_c = 0.8$;
- Probabilité de mutation $p_m = 0.2$;
- Taille de la population = 100 ;
- Nombre de générations = 5000.

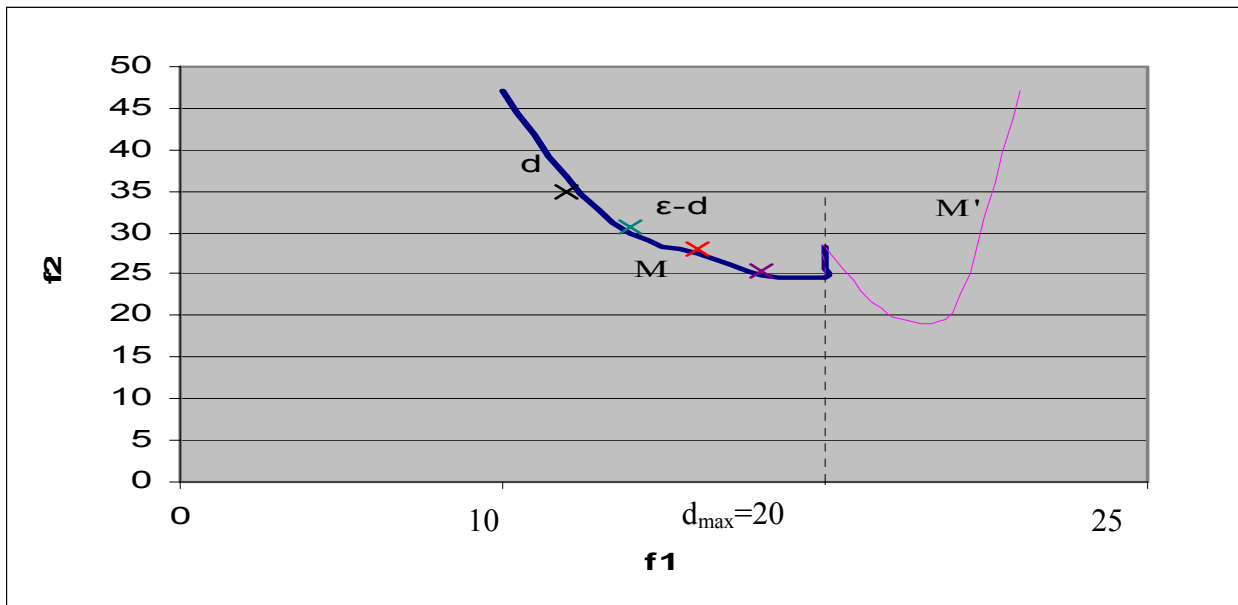


Figure V.8 - Fronts optimaux

V.7 Exemple détaillé de mise en œuvre du processus de négociation

Données de base

A l'instant $t=11h00$ et pendant 2 secondes ($\Delta_\epsilon = 2s$), nous supposons l'existence d'un nombre d'utilisateurs connectés à notre système formulant un nombre quelconque de requêtes. Soit l'ensemble suivant de requêtes :

- requête 1: « voyager à l'instant t de l'endroit B à l'endroit C » ;
- requête 2: « voyager le prochain week-end de l'endroit A à l'endroit B avec un coût minimum, demander les prévisions météorologiques et les événements culturels pour le prochain week-end à l'endroit B » ;
- requête 3: « voyager à l'instant t de l'endroit A à l'endroit C » ;
- requête 4: « demander les perturbations courantes du transport public de l'endroit B à l'endroit C » ;
- requête 5: « chercher le meilleur service de liaison avec le train X, prévu à l'endroit A à 12h00 aujourd'hui pour aller à l'endroit C » ;

- requête 6: « voyager à l'instant t de l'endroit A à l'endroit B » ;
- requête 7: « chercher un bon hôtel rapport qualité/prix à l'endroit D pendant le prochain week-end et faire la réservation, chercher le meilleur chemin et heure de départ pour aller de l'endroit B à l'endroit D avec la voiture, selon le trafic routier » ;
- etc.

Chaque agent AI envoie les requêtes Δ_ϵ – simultanées à l'agent AId de la même société d'agents (§III.3). Ce dernier décompose toutes les requêtes formulées en $l'=64$ tâches indépendantes. Le processus de décomposition des requêtes en services indépendants n'est pas détaillé dans le cadre de cette thèse car il représente en soi un axe de recherche.

La procédure de décomposition adoptée prend en considération le fait qu'il n'existe pas de connexion directe entre les points A et C. Ainsi, nous pouvons supposer que l'agent AId décompose \mathfrak{R}_t en un ensemble de tâches indépendantes comme suit:

- $T_1=$ « Perturbations du transport routier de l'endroit B à l'endroit C (à l'instant t) » ;
- $T_2=$ « Prévisions météorologiques à l'endroit B (le prochain week-end) » ;
- $T_3=$ « chercher un bon hôtel qualité/prix à l'endroit D durant le week-end prochain et faire la réservation » ;
- $T_6=$ « Chercher le plus court chemin pour aller avec la voiture de l'endroit B à l'endroit D » ;
- $T_9=$ « Chercher la meilleure heure de départ pour aller de l'endroit B à l'endroit D avec la voiture selon les prévisions du trafic routier pendant le prochain week-end » ;
- $T_{13}=$ « Evénements culturels à l'endroit B (le prochain week-end) " » ;
- $T_{16}=$ « Voyager de l'endroit B à l'endroit C (aujourd'hui, à l'instant t /aujourd'hui, à partir de 12:00) » ;
- $T_{19}=$ « Voyager de l'endroit A à l'endroit B (aujourd'hui, à l'instant t / le prochain week-end avec le meilleur prix/meilleur service de connexion avec le train X à 12h00 aujourd'hui) » ;

- etc.

Il est à noter qu'une tâche peut être représentée par plusieurs services avec différentes contraintes ; par exemple, T_{19} représente 3 services correspondant à la même tâche "Voyager de l'endroit A à l'endroit B" avec différentes contraintes: "maintenant", "le prochain week-end, avec un coût minimum" et "le meilleur service de connexion avec le train X à 12:00 aujourd'hui". En outre, un service est identifié grâce à un mot clé qui correspond, par exemple, à une « action » spécifiée selon des contraintes mentionnées entre parenthèses. Les réponses entières seront recomposées grâce à l'agent AF qui doit fusionner les services selon les contraintes de l'utilisateur, en prenant en considération la pertinence de l'information. La solution générée par le système est illustrée par le chromosome CH_3 correspondant à la Table V-5. Cette solution respecte tous les temps de fin d'exécution au plus tard des requêtes ($d_w = 25 \forall req_w \in \mathcal{R}_t$) avec un temps maximal $D_{\max}=23,46s$ et avec un coût total moyen égal à 30,5 unités de coût.

CH ₃	S ₁	S ₂	S ₃	S ₄	S ₅	S ₆	S ₇	S ₈	S ₉	S ₁₀	S ₁₁	S ₁₂	S ₁₃	S ₁₄	S ₁₅	S ₁₆	S ₁₇	S ₁₈	S ₁₉	S ₂₀
T ₁	*	X	*	*	*	*	*	*	*	*	*	*	X	*	*	X	*	1	X	X
T ₂	*	X	*	*	X	X	*	*	1	*	X	*	*	*	*	X	*	*	*	*
T ₃	*	X	1	*	*	X	*	*	*	*	*	*	X	*	*	*	X	*	*	*
T ₆	*	X	*	*	*	X	*	*	*	*	*	*	X	X	*	*	*	*	1	*
T ₉	X	X	X	*	X	X	X	*	*	*	*	*	*	X	*	X	*	X	X	1
T ₁₃	*	X	*	*	*	X	X	*	*	*	*	*	X	X	*	*	X	1	*	X
T ₁₆	X	X	X	X	*	X	*	X	X	X	X	X	X	X	X	X	X	X	1	X
T ₁₉	1	X	X	X	*	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
T ₂₀	X	X	*	X	1	X	X	X	X	*	X	X	X	X	*	X	X	X	X	*
T ₂₁	X	X	*	1	X	X	X	X	X	X	X	X	X	X	X	X	X	*	X	X
T ₂₂	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	*	X	1	*
T ₂₅	X	X	X	*	X	X	X	X	X	X	X	X	X	X	X	*	1	X	X	*
T ₂₆	X	*	1	X	X	X	X	X	X	*	*	X	X	X	X	X	X	*	X	X
T ₂₈	X	*	*	X	*	X	X	X	X	*	X	*	X	X	1	X	X	X	*	*
T ₂₉	1	*	X	X	X	X	X	X	X	*	X	*	X	X	X	X	X	X	X	X
T ₃₀	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	*	1	*	*
T ₃₁	*	X	*	*	X	*	X	1	X	X	X	*	X	*	X	X	X	*	X	X
T ₃₂	*	X	1	*	X	X	*	*	*	X	X	*	X	*	X	X	X	*	X	X
T ₃₃	*	X	1	*	X	X	*	X	X	X	X	*	X	X	X	*	X	X	X	X
T ₃₄	*	*	*	X	X	X	1	X	X	X	X	X	X	X	X	X	X	X	X	X
T ₃₅	*	X	*	X	X	X	X	1	X	X	X	X	X	X	X	X	X	*	X	X
T ₃₆	*	X	*	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	X	X
T ₃₇	X	X	*	X	X	X	X	X	X	X	X	X	X	X	X	X	X	*	X	1
T ₃₈	*	X	1	X	X	X	X	X	*	X	X	X	*	X	X	*	X	X	X	X
T ₃₉	X	X	X	X	X	X	X	X	*	X	X	X	X	X	X	X	*	X	X	1
T ₄₀	X	X	X	X	X	X	X	X	X	1	X	X	X	X	*	*	X	X	X	X
T ₄₁	X	X	*	X	*	X	X	X	X	X	X	X	X	X	X	X	X	1	X	X
T ₄₂	*	X	1	X	X	X	X	X	X	*	X	X	X	X	X	X	X	*	X	X
T ₄₄	X	X	X	X	X	X	X	X	X	*	X	X	X	X	X	X	1	X	X	X
T ₅₂	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	*	X	1	X
T ₅₃	*	X	X	X	X	X	X	X	X	X	X	1	X	X	*	X	X	X	X	X
T ₅₆	X	*	*	X	X	X	X	X	X	1	X	X	X	X	X	X	X	*	X	X
T ₅₇	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	*	1	X
T ₅₈	X	*	*	X	X	X	X	X	X	*	X	X	X	X	1	X	X	X	*	X
T ₅₉	*	1	X	X	X	X	X	X	X	*	X	*	X	X	X	X	X	X	X	X
T ₆₀	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	*	*	X
T ₆₁	X	X	1	X	X	X	X	X	X	X	X	X	X	X	*	X	*	X	X	X
T ₆₃	*	X	*	*	*	X	*	*	*	X	X	X	X	X	X	1	X	*	X	X
T ₆₄	*	*	*	1	*	X	X	X	*	X	X	X	X	X	X	X	X	X	X	X
T ₆₅	*	X	X	X	X	X	X	X	X	X	X	X	X	*	X	X	X	1	X	X
T ₆₆	1	X	*	X	X	X	X	X	X	X	X	X	X	*	X	X	X	X	X	X
T ₆₇	X	X	*	X	X	X	X	X	X	X	X	X	*	X	X	X	X	X	1	X
T ₆₈	*	X	*	X	X	1	X	*	X	X	X	X	X	X	X	X	*	X	X	X
T ₆₉	X	X	X	X	X	X	X	X	*	X	1	X	X	X	X	X	X	X	X	X
T ₇₁	X	X	*	X	X	X	*	X	X	X	X	X	X	1	X	X	X	X	X	X
T ₇₃	X	X	X	X	X	X	X	X	X	X	X	X	X	*	1	X	X	*	X	X
T ₇₄	X	X	X	X	X	X	*	X	X	X	X	X	X	*	X	1	X	X	X	X
T ₇₅	*	X	*	X	X	X	X	X	*	X	1	X	X	X	X	X	X	X	X	X
T ₇₆	X	*	*	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	X	X
T ₇₇	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	X	*
T ₇₈	X	1	X	X	X	X	X	X	X	X	*	X	X	X	X	X	X	X	X	X
T ₇₉	*	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
T ₈₀	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	X	X	*
T ₈₁	*	X	*	1	X	X	X	X	X	X	*	X	X	X	X	X	X	*	X	X
T ₈₂	*	X	*	X	X	X	*	*	1	X	X	X	X	*	X	X	X	X	X	X
T ₈₃	X	X	X	X	*	X	*	X	X	X	1	X	*	X	X	X	X	X	X	X
T ₈₄	*	1	*	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
T ₈₅	*	X	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	*	X	X
T ₈₆	*	X	*	X	1	X	X	X	X	X	X	X	X	X	X	X	X	*	X	X
T ₈₈	1	X	*	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
T ₉₀	X	X	X	X	X	X	X	*	X	X	X	X	*	X	1	*	X	X	X	X
T ₉₅	X	X	X	X	1	X	X	X	*	X	*	X	*	X	X	X	X	X	X	X
T ₉₆	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	X
T ₉₉	*	X	X	X	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Table V-5- Chromosome CH₃

Plans Des Routes Initiaux et Définitifs des agents ACIs

- $m=5$, les PDRIs sont:

- $PDRI_1 = (S_{20}, S_{15}, S_1, S_3) ;$
- $PDRI_2 = (S_{18}, S_7, S_{10}, S_{17}) ;$
- $PDRI_3 = (S_2, S_{13}, S_{19}, S_6) ;$
- $PDRI_4 = (S_{16}, S_{14}, S_5, S_{12}, S_4) ;$
- $PDRI_5 = (S_{11}, S_8, S_9) ;$

- $m'=5$, les PDRDs sont:

- $PDRD_{t,1} = \{S_{20}\{T_9, T_{37}, T_{39}\}, S_{15}\{T_{28}, T_{58}\}, S_1\{T_{19}, T_{29}, T_{66}, T_{88}\}, S_3\{T_3, T_{26}, T_{32}, T_{33}, T_{38}, T_{42}, T_{61}, T_{85}\}\} ;$
- $PDRD_{t,2} = \{S_{18}\{T_1, T_{13}, T_{30}, T_{36}, T_{41}, T_{65}, T_{76}, T_{77}\}, S_7\{T_{34}\}, S_{17}\{T_{25}, T_{44}, T_{60}, T_{80}\}\} ;$
- $PDRD_{t,3} = \{S_2\{T_{59}, T_{78}, T_{79}, T_{84}\}, S_{13}\{T_{53}\}, S_{19}\{T_6, T_{16}, T_{22}, T_{52}, T_{57}, T_{67}, T_{96}\}, S_6\{T_{68}\}\} ;$
- $PDRD_{t,4} = \{S_{16}\{T_{63}, T_{73}, T_{74}, T_{90}\}, S_{14}\{T_{71}\}, S_5\{T_{20}, T_{86}, T_{95}, T_{99}\}, S_{12}\{T_{75}, T_{83}\}, S_4\{T_{21}, T_{64}, T_{81}\}\} ;$
- $PDRD_{t,5} = \{S_{11}\{T_{40}, T_{56}, T_{69}\}, S_8\{T_{31}, T_{35}\}, S_9\{T_2, T_{82}\}\} .$

Application détaillée du Processus de négociation

Les agents ACI sont supposés visiter leur premier nœud respectif du PDRD sans problème avant l'annonce de l'ensemble des nœuds indisponibles.

Par exemple, nous supposons l'ensemble suivant de nœuds indisponibles:
 $Ind_t = \{S_1, S_3, S_7, S_{14}, S_5, S_{17}, S_{12}, S_9, S_{13}, S_{19}\}$ nous en déduisons les tâches à réaffecter :

$\Phi_i = \{T_{19}, T_{29}, T_{66}, T_{88}, T_3, T_{26}, T_{32}, T_{33}, T_{38}, T_{42}, T_{61}, T_{85}, T_{34}, T_{71}, T_{20}, T_{86}, T_{95}, T_{99}, T_{25}, T_{44}, T_{60},$

$T_{80}, T_{75}, T_{83}, T_2, T_{82}, T_{53}, T_6, T_{16}, T_{22}, T_{52}, T_{57}, T_{67}, T_{96}\}$, nous avons donc 34 tâches à réaffecter,

Le processus de négociation proposé nous permet de réaffecter l'ensemble Φ_i comme il est indiqué à la Figure V.9 qui montre que la plus grande partie de Φ_i est réaffectée grâce aux priorités des agents ACI, ensuite grâce à leurs préférences et enfin grâce aux contraintes. Pour ce scénario de simulation, chaque étape de l'algorithme ARDyCA correspond à deux tours de négociation par itération.

A la réception du contrat proposé, chaque agent ACI répond par un accord partiel parce que, dans ce cas, la perturbation a touché un sous-ensemble de chaque PDRD :

- L'agent ACI_1 ne visitera pas les nœuds S_1 et S_3 et donc il informe l'agent AO qu'il ne s'occupe plus des tâches $T_{19}, T_{29}, T_{66}, T_{88}, T_3, T_{26}, T_{32}, T_{33}, T_{38}, T_{42}, T_{61}, T_{85}$;
- L'agent ACI_2 ne visitera pas les nœuds S_7 et S_{17} et donc il informe l'agent AO qu'il ne s'occupe plus des tâches $T_{34}, T_{25}, T_{44}, T_{60}, T_{80}$;
- L'agent ACI_3 ne visitera pas les nœuds S_{13} et S_{19} et donc il informe l'agent AO qu'il ne s'occupe plus des tâches $T_{53}, T_6, T_{16}, T_{22}, T_{52}, T_{57}, T_{67}, T_{96}$;
- L'agent ACI_4 ne visitera pas les nœuds S_{14} , S_5 et S_{12} et donc il informe l'agent AO qu'il ne s'occupe plus des tâches $T_{71}, T_{20}, T_{86}, T_{95}, T_{99}, T_{75}, T_{83}$;
- L'agent ACI_5 ne visitera pas le nœud S_9 et donc il informe l'agent AO qu'il ne s'occupe plus des tâches T_2, T_{82} ;

A ce stade, l'agent AO construit Φ_i , confirme les parcours restants des agents ACI et demande à chacun de proposer une nouvelle série d'affectations à Φ_i selon ses priorités :

- Aucun agent ne propose une affectation à la tâche T_{19} puisque les fournisseurs possibles ne sont pas disponibles (S_1 et S_5) ;
- L'agent ACI_5 , qui n'a encore pas quitté le nœud S_{11} , propose d'y exécuter la tâche T_{29} ;

- L'agent ACI₁, qui s'apprête à quitter le nœud S₂₀, propose d'exécuter la tâche T₆₆ sur sa prochaine destination (le nœud S₁₅) ;
- Aucun agent ne propose une affectation à la tâche T₈₈ puisque les fournisseurs possibles ne sont pas disponibles (S₁ et S₃) ;
- La tâche T₃ est proposée par :
 - l'agent ACI₁ sur les nœuds S₂₀ et S₁₅ ;
 - l'agent ACI₂ sur le nœud S₁₈ et sur le nœud S₁₀ appartenant à ses préférences ;
 - l'agent ACI₄ sur les nœuds S₁₆ et S₄ ;
 - l'agent ACI₅ sur les nœuds S₁₁ et S₁₈ ;
- La tâche T₂₆ est proposée par :
 - l'agent ACI₂ sur le nœud S₁₈ ;
 - l'agent ACI₃ sur les nœuds S₂ ;
 - l'agent ACI₅ sur le nœud S₁₁ ;
- La tâche T₃₂ est proposée par :
 - l'agent ACI₂ sur le nœud S₁₈ ;
 - l'agent ACI₄ sur le nœud S₄ ;
 - l'agent ACI₅ sur le nœud S₈ ;
- La tâche T₃₃ est proposée par l'agent ACI₄ sur les nœuds S₁₆ et S₄ ;
- Aucun agent ne propose une affectation à la tâche T₃₈ puisque les fournisseurs possibles appartenant aux priorités des agents ACI ne sont pas disponibles (S₁, S₃, S₁₄, S₁₇), le nœud S₁₀ propose également un service à la tâche T₃₈ mais ce dernier appartient aux préférences de l'agent ACI₂ ;
- La tâche T₄₂ est proposée par :

- l'agent ACI₂ sur le nœud S₁₈ ;
- l'agent ACI₅ sur le nœud S₁₁ ;
- La tâche T₆₁ est proposée par l'agent ACI₁ sur le nœud S₁₅ ;
- La tâche T₈₅ est proposée par l'agent ACI₂ sur le nœud S₁₈ ;
- La tâche T₃₄ est proposée par l'agent ACI₃ sur le nœud S₂ ;
- Aucun agent ne propose une affectation à la tâche T₇₁ puisque les fournisseurs possibles ne sont pas disponibles (S₃ , S₇ et S₁₄) ;
- La tâche T₂₀ est proposée par :
 - l'agent ACI₁ sur les nœuds S₂₀ et S₁₅ ;
 - l'agent ACI₂ sur le nœud S₁₀ appartenant à ses préférences ;
- La tâche T₈₆ est proposée par l'agent ACI₂ sur le nœud S₁₈ ;
- La tâche T₉₅ est proposée par l'agent ACI₅ sur les nœuds S₁₁ ;
- Aucun agent ne propose une affectation à la tâche T₉₉ puisque les fournisseurs possibles ne sont pas disponibles (S₁ et S₅) ;
- La tâche T₂₅ est proposée par :
 - l'agent ACI₁ sur le nœud S₂₀ ;
 - l'agent ACI₄ sur le nœud S₁₆ et S₄ ;
- La tâche T₄₄ est proposée par l'agent ACI₅ sur le nœud S₁₁ ;
- La tâche T₆₀ est proposée par l'agent ACI₂ sur le nœud S₁₈ ;
- La tâche T₈₀ est proposée par l'agent ACI₁ sur le nœud S₂₀ ;
- La tâche T₇₅ est proposée par l'agent ACI₂ sur le nœud S₁₀ appartenant à ses préférences ;

- Aucun agent ne propose une affectation à la tâche T_{83} puisque les fournisseurs possibles ne sont pas disponibles (S_5 , S_7 , S_{12} et S_{14}) ;
- La tâche T_2 est proposée par :
 - l'agent ACI_1 sur les nœuds S_{20} et S_{15} ;
 - l'agent ACI_2 sur le nœud S_{18} et sur le nœud S_{10} appartenant à ses préférences ;
 - l'agent ACI_4 sur le nœud S_4 ;
 - l'agent ACI_5 sur le nœud S_8 ;
- La tâche T_{82} est proposée par l'agent ACI_5 sur le nœud S_8 ;
- La tâche T_{53} est proposée par l'agent ACI_4 sur le nœud S_{16} ;
- La tâche T_6 est proposée par :
 - l'agent ACI_1 sur les nœuds S_{20} et S_{15} ;
 - l'agent ACI_2 sur le nœud S_{18} et sur le nœud S_{10} appartenant à ses préférences ;
 - l'agent ACI_4 sur les nœuds S_{16} et S_4 ;
 - l'agent ACI_5 sur le nœud S_{11} et S_8 ;
- Aucun agent ne propose une affectation à la tâche T_{16} puisque les fournisseurs possibles ne sont pas disponibles (S_5 , S_7 et S_{19}) ;
- La tâche T_{22} est proposée par l'agent ACI_1 sur le nœud S_{20} ;
- Aucun agent ne propose une affectation à la tâche T_{52} puisque les fournisseurs possibles ne sont pas disponibles (S_{17} et S_{19}) ;
- La tâche T_{57} est proposée par l'agent ACI_2 sur le nœud S_{18} ;
- Aucun agent ne propose une affectation à la tâche T_{67} puisque les fournisseurs possibles ne sont pas disponibles (S_3 , S_{13} et S_{19}) ;

- Aucun agent ne propose une affectation à la tâche T_{96} puisque le seul fournisseur possible n'est pas disponible (S_{19}).

Chaque agent ACI propose toutes les affectations possibles selon leurs priorités. A la réception de toutes ces propositions, l'agent AO décide (nouveau contrat) :

1) Affectation directe des tâches (choix unique):

$T_{29}, T_{66}, T_{61}, T_{85}, T_{34}, T_{86}, T_{95}, T_{44}, T_{60}, T_{80}, T_{82}, T_{53}, T_{22}, T_{57}$;

2) Affectation des tâches suite à une optimisation (choix multiple):

$T_3(S_{20}), T_{26}(S_{11}), T_{32}(S_8), T_{33}(S_{16}), T_{42}(S_{18}), T_{20}(S_{20}), T_{25}(S_{16}), T_2(S_8), T_6(S_{18})$;

Soient les réponses suivantes des agents ACI :

- L'agent ACI_1 refuse le choix de l'agent AO pour les tâches T_3 et T_{20} sur le nœud S_{20} parce que ce dernier l'a déjà quitté et accepte le reste de la proposition (accord partiel);
- L'agent ACI_4 refuse le choix de l'agent AO pour les tâches T_{25} et T_{33} sur le nœud S_{16} parce que ce dernier l'a déjà quitté et accepte le reste de la proposition (accord partiel);
- Les agents ACI_2, ACI_3 et ACI_5 acceptent totalement le contrat en cours (accords totaux).

A ce stade, l'agent AO met à jour $\Phi_t = \{ T_3, T_{20}, T_{25}, T_{33}, T_{75}, T_{19}, T_{88}, T_{38}, T_{71}, T_{99}, T_{83}, T_{16}, T_{52}, T_{67}, T_{96} \}$, confirme les parcours restants des agents ACI et demande à chacun de proposer une nouvelle série d'affectations à Φ_t encore selon ses priorités :

- L'agent ACI_1 propose d'exécuter T_3 et T_{20} sur S_{15} ;
- L'agent ACI_4 propose d'exécuter T_{33} et T_{25} sur S_4 ;

A la réception de ces propositions, l'agent AO envoie un nouveau contrat contenant toutes les affectations proposées puisqu'il les approuve. Les agents ACI concernés (ACI_1 et ACI_4) envoient ainsi des accords totaux qui vont être confirmés par l'agent AO. L'ensemble Φ_t devient donc : $\Phi_t = \{ T_{75}, T_{19}, T_{88}, T_{38}, T_{71}, T_{99}, T_{83}, T_{16}, T_{52}, T_{67}, T_{96} \}$. A ce stade, l'agent AO demande à chaque agent ACI, participant à la négociation, de proposer une nouvelle série d'affectations à Φ_t encore selon ses préférences : l'agent ACI_2 propose donc d'exécuter la tâche T_{75} sur le nœud S_{10} . Dans ce cas, l'agent AO envoie à l'agent ACI_2 un contrat contenant

cette affectation puisqu'il l'approuve, l'agent ACI_1 accepte puis l'agent AO confirme et met à jour $\Phi_t = \{ T_{19}, T_{88}, T_{38}, T_{71}, T_{99}, T_{83}, T_{16}, T_{52}, T_{67}, T_{96} \}$.

Pour cet exemple, le processus de négociation s'arrête à ce niveau puisque tous les fournisseurs possibles de l'ensemble restant des tâches à réaffecter ne sont pas disponibles. L'évolution de la réduction de l'ensemble Φ_t en fonction du nombre de tours de négociation est exprimée par la Figure V.9.

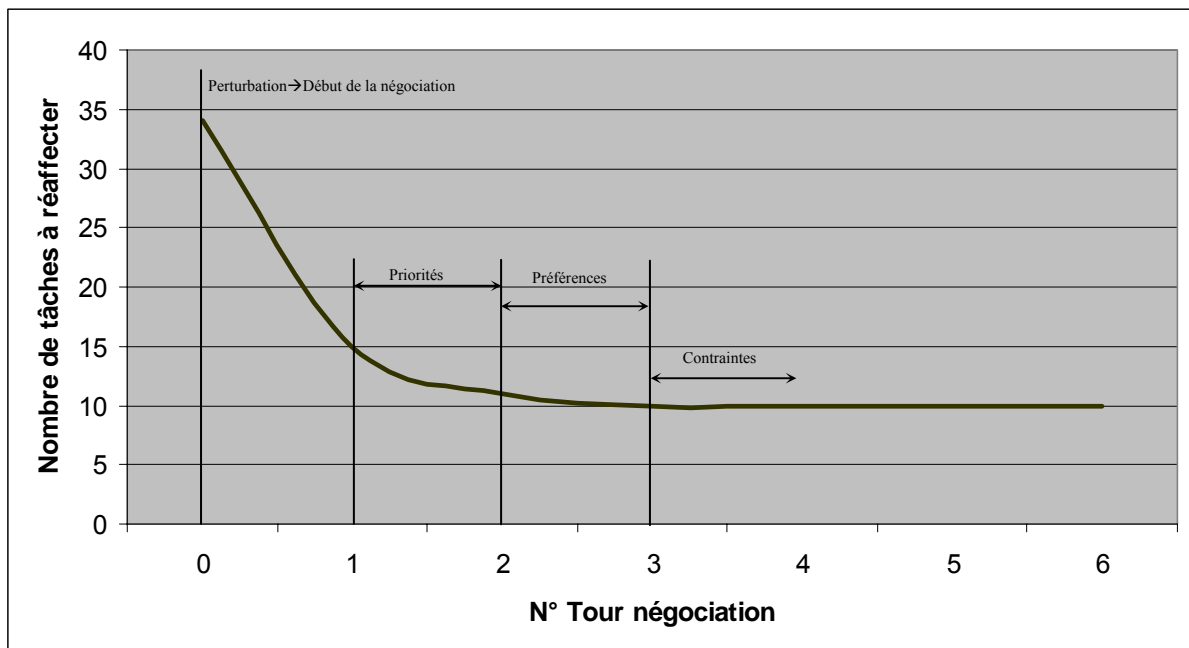


Figure V.9– Application du protocole de négociation

V.8 Conclusion

Dans ce chapitre, nous avons montré le fonctionnement des approches d'optimisation adoptées dans le SITM. Les résultats et scénarios de simulation montrent la robustesse du système face à un flux de données, éventuellement important, et face aux aléas du RETM. Nous avons donc démontré que le paradigme AM est parfaitement adapté à nos approches d'optimisation en réduisant le temps de réponses aux requêtes. En outre, les aléas du RETM déclenchent une sorte de négociation entre les agents ACI patrouilleurs et les agents optimisateurs stationnaires du système (les agents AO). Cette négociation permet de tenter de répondre aux utilisateurs du système en dépit des perturbations. Il faut dire que le caractère flexible du chromosome FeTAR (§III.4.2.2) a été d'une grande utilité à ce type complexe

d'interaction grâce à une possibilité d'appliquer une modification directe sur le chromosome telle qu'une mutation contrôlée.

CONCLUSION GENERALE

Dans ce travail de recherche, nous avons proposé un système d'information (SITM) d'aide à la mobilité urbaine qui se base sur une approche multi-agent en intégrant des algorithmes d'optimisation. L'architecture du SITM 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.

Ce système permet d'optimiser la gestion des flux de données pour faire face à un nombre important de requêtes simultanées. L'optimisation intègre le temps de réponse des requêtes, le coût de l'information, la taille des données transférées et l'état du réseau informatique, ce qui fait que le problème résolu est multi-contrainte multi-objectif. Le système se base sur le paradigme Agent Mobile (AM) dont l'utilisation a été prouvée et testée. La solution proposée se déroule en deux étapes principales avec une approche d'archivage dynamique de données :

- Pour la première étape d'optimisation, nous avons conçu un algorithme à coûts effectifs de planification des parcours d'AMs (CoSPa) pour préparer des agents mobiles dotés de plans de patrouille, en prenant en considération l'état du réseau. Cette étape d'optimisation se fait à priori et se déclenche à l'initiation du système ou à chaque fois que l'état du réseau varie considérablement ;
- Pour la deuxième étape d'optimisation, nous avons adopté une approche évolutionnaire en proposant un ensemble de solutions à usage flexible. Cette étape d'optimisation se déroule en temps réel à la réception des requêtes utilisateurs et permet l'optimisation des flux de services ;
- L'approche d'archivage dynamique proposée utilise la technique des indicateurs pour réduire la navigation des AMs.

Nous avons également proposé un protocole de négociation entre les agents mobiles et les agents stationnaires optimisateurs du système pour faire face aux aléas du réseau. Le protocole proposé utilise une technique particulière pour arriver le plus vite possible à trouver

un compromis entre les différents intervenants du processus, cette technique originale se base sur l'acceptation des contrats par parties en tolérant des confirmations et des accords partiels entre les parties. La complexité d'un tel protocole nous a incités à mettre en œuvre une ontologie flexible à enrichir ultérieurement pour couvrir l'ensemble du système. La mise en œuvre d'autres protocoles d'interaction entre les différentes sociétés actives du système est également envisageable mais ne doit pas tomber dans le piège de la consommation excessive des ressources.

Dans mes travaux futurs, la réalisation d'un système d'information d'aide à la mobilité urbaine, basé sur l'optimisation distribuée, les systèmes multi-agents et agents mobiles, nécessitera d'établir et dessiner **les futurs services** d'aide à la mobilité en mettant au point l'ensemble des éléments techniques pour une expérimentation réelle, par exemple dans une rame TER ou dans les pôles d'échanges. L'intégration de notre système dans le démonstrateur du projet **VIATIC-Mobilité** du pôle de compétitivité **I-TRANS** demande des développements supplémentaires et notamment les « web services ».

Une analyse fonctionnelle, technique et économique plus approfondie permettra de définir les services d'aide à la mobilité de demain, en lien étroit avec des acteurs régionaux. Mes travaux de recherche pourraient s'intégrer dans le démonstrateur **SIM** (Système d'Information Multimodale) en cours de réalisation dans la Région Nord Pas de Calais et qui traite comme service uniquement les itinéraires en lignes. Nous envisagerons d'interfacer SIM avec notre système pour intégrer nos approches dans la recherche et la composition d'autres services. Ces services ont pour objectif de proposer de l'information d'accompagnement du voyageur pendant son déplacement (information multimodale) et de l'information d'agrément au cours de sa mobilité (actualités, culture, divertissement, tourisme, jeux...). Ces informations sont accessibles à proximité des systèmes de transport et en embarqué.

Les approches utilisées dans mes travaux de recherche notamment les méthodes d'optimisation distribuées, les systèmes multi-agents et agents mobiles sont souples et évolutifs. Nous proposons donc d'adapter et d'étendre ces approches à des problèmes autres que les problèmes de transport, tels que les problèmes d'optimisation de la logistique de flux dans le domaine de la défense dans le cadre du projet EADS/EC-Lille.

GLOSSAIRE

- ACI : Agent Collecteur Intelligent
- AF : Agent Fusion
- AI : Agent Interface
- AId : Agent Identificateur
- AM : Agent Mobile
- AO : Agent Ordonnanceur
- AR : Agent Réseau
- BDA : Base de données d'Archivage
- BDI : Base de Données d'Inscription
- COSPA : Construction des Schémas de Parcours
- H : Nœud hôte
- NACI : Négociation d'Agents ACI
- PDR : Plan De Route
- PDRD : PDR Définitif
- PDRI : PDR Initial
- $P2DR_{t,k}$: PDRD Restant d'un agent ACI_k à l'instant t
- PRCS : Problème de Recherche et de Composition de Services
- RETM : Réseau Etendu de Transport Multimodal
- SI : Système d'Information
- SITM : Système d'Information de Transport Multimodal
- SMA : Système Multi-Agent
- Ψ_j : Ensemble des tâches à exécuter sur le serveur S_j
- Φ_t : Ensemble des tâches nécessitant une réaffectation à l'instant t
- Ω_t : Ensemble des agents ACI qui n'ont encore pas achevé leur route à l'instant t
- Ind_t : Ensemble des nœuds indisponibles du réseau à l'instant t
- Δ_ε : Période d'acquisition des requêtes
- Δ_∞ : Période d'inactivité d'un agent ou d'une société d'agent

BIBLIOGRAPHIE

- [Adam, 00] E.Adam. *Modèle d'organisation multi-agent pour l'aide au travail coopératif dans les processus d'entreprise : application aux systèmes administratifs complexes*. Thèse de doctorat, Université de Valenciennes et du Hainaut-Cambrésis, septembre 2000.
- [Armetta, 06] F.Armetta, S.Hassas et S.Pimont. *Coordination émergente de l'activité décentralisée d'agents pour le partage de ressources critiques*. Systèmes multi-agents : articulation entre l'individuel et le collectif : JFSMA 2006. Hermes Science/Lavoisier, ISBN 2-7462-1586-1.
- [Baek et al., 01] J.W.Baek, J.H.Yeo, G.T.Kim, H.Y.Yeom. Cost Effective Mobile Agent Planning for Distributed Information Retrieval. *In proceedings of the 21st IEEE International Conference on Distributed Computing Systems (ICDCS'01)* p. 0065, 2001.
- [Barron et al., 99] T.M.Barrons, R.H.L Chaing et V.C.Storey. A semiotic framework for information systems classification and development. *Decision support Systems*, Elsevier, 25, p.1-17, 1999.
- [Bernard, 99] G. Bernard. Technologie du code mobile : état de l'art et perspectives. Institut National des Télécommunications, 1999.
- [Bichler et al., 98] M.Bichler, C.Beam et A.Segev. OFFER : A broker-centered object framework for electronic requisitioning. *In Proceedings of IFIP Conference trends in Electronic Commerce*, 1998.
- [Bravo et al., 05] Maricela C. Bravo, Joaquin Pérez, Victor J. Sosa, Azucena Montes. *Ontology Support for Communicating Agents in Negotiation Processes*. *In Proceedings of the Fifth International Conference on Hybrid Intelligent Systems (HIS'05)* 0-7695-2457-5/05.
- [Buse et al.,03] D.P. Buse, J.Q. Feng and Q.H. Wu. *Mobile Agents for Data Analysis In Industrial Automation Systems*. In Proceedings of the IEEE/WIC IAT'03. August 2003.
- [Carrier, 89] J.Carrier. An algorithm for solving the job shop problem. *Management science*, Vol. 35, pp164-176, 1989.

- [Carzaniga et al.,97] A. Carzaniga, G. P. Picco et G. Vigna. Designing distributed applications with mobile code paradigms. *In Proceedings of the 19th International Conference on Software Engineering*. July 1997, pp22-32.
- [Caux et al., 94] C.Caux, H.Pierreval, M.C.Portmann. Les algorithmes génétiques et leurs applications aux problèmes d'ordonnement. *Actes des journées d'étude : ordonnancement et entreprise, applications concrètes et outils pour le futur*. 16-17 juin 1994, Toulouse, France.
- [Ciesielski et al.,98] V.Ciesielski et P.Scerri. Real time genetic scheduling of aircraft landing times. IEEE International Conference on Evolutionary Computation, ICEC98. In D.Fogel (Editor), pp.360-364, Anchorage, May 1998.
- [Chaib-draa,95] B.Chaib-draa. Industrial Applications of Distributed AI. *Communications of the ACM*, 38(11), pp.49-53. Novembre 1995.
- [Chaib-draa,96] B.Chaib-draa. Interaction between Agents in Routine, Familiar and Unfamiliar Situations. *International Journal of Intelligent & Cooperative Information Systems*, 5(1), pp.1-25,1996.
- [Collette et al.,02] Yann Collette et Patrick Siarry. *Optimisation multiobjectif*. Editions EYROLLES, septembre 2002.
- [Deb et al., 02] K. Deb, S. Agrawal, A. Pratap and T. Meyarivan, A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2): 182-197, 2002.
- [Delahaye et al., 96] D.Delahaye, N.Durand, J.M.Alliot et M.Schoenauer. Genetic algorithms for air traffic control systems. ENAC, INFAUTOM'96, Toulouse, France, 14-15 mars 1996.
- [Doniec, 06] A. Doniec. *Prise en compte des comportements anticipatifs dans la coordination multi-agent : application à la simulation de trafic en carrefour*. Thèse de doctorat. France, Université de Valenciennes et du Hainaut Cambrésis. Octobre 2006.
- [Durfee et al., 91] E.H.Durfee et V.R.Lesser. Partial global planning: A coordination framework for distributed hypothesis formation. *IEEE transactions on Systems, Man and Cybernetics*, 21(5), pp.1167-1183. Septembre - octobre 1991.
- [Fayech et al., 02] B.Fayech, S.Maouche, S.Hammadi et P.Borne. Multi-Agent Decision-Support System for an Urban Transportation Network. *World Automation Congress, CDROM Proceedings ISIAAC02. 9-13 June, Florida, USA, 2002*.

- [Fayech, 03] B.Fayech. *Régulation des réseaux de transport multimodal : systèmes multi-agents et algorithmes évolutionnistes*. Thèse de doctorat, l'Université des Sciences et Technologies de Lille (I3D) et l'Ecole Centrale de Lille (LAIL) et. France, octobre 2003.
- [Ferber, 95] J.Ferber. *Les Systèmes Multi-Agents vers une intelligence collective*. InterEditions, 1995.
- [Fischer et al.95] K.Fischer, J.P.Müller, M.Pishel et D.Schier; A Model for Cooperative Transportation Scheduling. *In proceedings of the 1st International Conference on MAS*. AAAI Press/MIT Press. pp.109-116, Menlo Park, California, 1999.
- [Florez, 99] R.A.Florez-Mendez. Towards a Standardization of Multi-Agent System Frameworks. *ACM Crossroads Student Magazine*. Canada, 1999.
- [Greenwood, 05] D.Greenwood. JADE Web Service Integration Gateway (WSIG). Whitestein Technologies. *Jade Tutorial*. AAMAS 2005.
- [Hanon, 06] D.Hanon. *Modèle décisionnel orienté comportement fondé sur le vote : Application à la navigation d'agents autonomes en environnement simulé*. Thèse de doctorat, Université de Valenciennes et du Hainaut-Cambrésis, décembre 2006.
- [Huang et al., 03] P.H.Huang et J.H.Yen. Integrate the structure of negotiation certification of set and the intelligent mobile agent. *Proceedings of the IEEE 37th Annual 2003 International Carnahan Conference on Security Technology*. 14(16), pp. 237 – 244, octobre 2003.
- [Huissman et al.,01] D.Huisman, R.Freling et A.P.M.Wagelmans. A Dynamic Approach to Vehicle Scheduling. *Report EI2001-17 accepted for Transportation Science (focussed issue on Real-time Fleet Management) on August 22, 2002 with a new title: A Robust Solution Approach to the Dynamic Vehicle Scheduling Problem*. Econometric Institute, Erasmus University Rotterdam, Netherlands.
- [Jennings et al., 96] N.Jennings et M.Wooldridge. Software Agents. *IEEE Review*. pp.17-20, janvier 1996.
- [Kacem et al., 01] I.Kacem, S.Hammadi, P.Borne. Approach by Localization and Genetic Manipulations Algorithm for Flexible Job-shop Problems. *Proceedings of International IEEE Conference on Systems, Man, and Cybernetics*, October 7-10, 2001, pp 2599-2604, Tucson, Arizona, USA.

- [Kalyani et al., 06] Y.Kalyani et C.Adams. Privacy negotiation using mobile agent. *In proceedings of Canadian Conference on Electrical and Computer Engineering, IEEE CCECE/CCGEI*, Ottawa, May 2006. pp.628-633.
- [Kamoun, 07] M.A.Kamoun. *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 (LAGIS) / ESTAS de l'INRETS. France, avril 2007.
- [Kersten et al., 97] Kersten G., Norohana S. Supporting International Negotiation with a WWW-based System. *Internet Research Report*, 1997.
- [Ketel et al.,05] M. Ketel, N.S. Dogan and A. Homaifar. Distributed Sensor Networks based on Mobile Agents Paradigms. Department. of Computer Science, North Carolina A&T State University, NC 27411. *System Theory, SSST'05. Greensboro*, USA 2005, pp411-414.
- [Khaled et al.,05] I. Ben Khaled, M.A. Kamoun, K.Zidi, S. Hammadi. Vers un système d'Information Voyageur Multimodal (SIM) à base de Système Multi Agent (SMA). *REE* 2005.
- [Kim et al., 03] J.B.Kim et A.Segev. A framework for dynamic eBusiness negotiation processes. In *Proceedings of IEEE International Conference on eCommerce, CEC 2003*. 24(27) pp.84-91. Juin 2003.
- [Laumanns et al., 02] M.Laumanns, L.Thiele, K.Deb et E.Zitzler. Combining Convergence and Diversity in Evolutionary Multi-objective Optimization. *Evolutionary Computation*, 0(3):263–282, Fall 2002.
- [Ljungberg et al., 92] M.Ljungberg et A.Lucas. The oasis air-traffic management system. *In Proceedings of the 2nd Pacific Rim International Conference on Artificial Intelligence, PRICAI'92*. Seoul, Korea,1992 .
- [Lopez et al., 01] P.Lopez et François Roubellat. *Ordonnancement de la production*. ISBN : 2-7462-0184-4. Paris, Hermès Science Publications, 2001.
- [Lu et al.,03] X.Lu et K.Mori. Autonomous Information Services Integration and Allocation in Agent-Based Information Service System. *In Proceedings of the IEEE/WIC Intelligent Agent Technology (IAT'03)*. 13(16) pp.290-296. Août 2003.

- [Maes et al., 96] P.Maes et A.Chavez. Kasbah : An agent marketplace for buying and selling goods. In Proceedings of the Practical Applications for Intelligent Agents and Multi-Agent technology Conference. Avril 1996.
- [Mandiau et al., 99] R.Mandiau, E.G.L.Strugeon et G.Agimont. Study of the influence of organizational structure of the efficiency of a multi-agent system. *Networking and Information Systems Journal*.2, 1999, pp.153-179.
- [Mandiau et al., 02a] R.Mandiau et E.G. L. Strugeon. Systèmes multi-agents. *Techniques de l'ingénieur , S7216 traité informatique Industrielle*. 2002.
- [Mandiau et al., 02b] R.Mandiau, E.G.L.Strugeon et A. Péninou. *Organisation et applications des SMA*. Lavoisier, 2002.
- [Mathieu et al., 02] P.Mathieu et M.H.Verrons. A generic Model for contract negotiation. In *Proceedings of the AISB'02 Convention*, pp. 1-8. London, UK, avril 2002.
- [Metropolis et al., 53] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, et E. Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, vol. 21, 1953.
- [Michalewicz, 92] Z. Michalewicz. Genetic Algorithms + Data Structures = Evolution programs. Springer Verlag, 1992.
- [Nawaz et al., 83] M.Nawaz, Jr.E.E. Ensocre et I.Ham. *A heuristic algorithm for the machine, n-job flow shop sequencing problem*. Omega, vol.11, 1983.
- [OuldSidi,06] M.M.OuldSidi. *Contribution à l'amélioration des systèmes d'aide à la décision pour la régulation du trafic des réseaux de transport collectif*. Thèse de doctorat, LAGIS de l'Ecole Centrale de Lille et ESTAS de l'Institut National de Recherche sur les Transports et leur Sécurité (INRETS). France, décembre 2006.
- [Parunak , 87] Parunak, H.V.D. 1987. *Manufacturing Experience with the Contract Net*. In: *Huhns, M. ed. Distributed Artificial Intelligence*, pp.285-310. Los Altos, Calif.: Morgan Kaufmann.
- [Pharm et al., 98] V.A. Pharm et A. Karmouch. Mobile Software Agents: an overview. *IEEE Communication Magazine*. University of Ottawa, Ontario, July 1998, pp. 26-37.
- [Picco et al.,98] G.P.Picco and M. Baldi. Evaluating the Tradeoffs of Mobile Code Design Paradigms in Network Management Applications. In *Proceedings of the 20th IEEE International Conference on Software Engineering (ICSE'97)*.

Edited by In R. Kemmerer and K. Futatsugi, pp. 146-155, Kyoto, Japan
April 1998.

- [Pierreval et al.,03] H. Pierreval, C. Caux, J.L.Paris et F.Viguier. Evolutionary approaches to the design and organization of manufacturing systems. *Computers and Industrial Engineering*, Vol. 44, pp.339-364, 2003.
- [Pruitt, 81] D.Pruitt. *Negotiation Behavior*. Academic Press ,1981.
- [Rosenschein et al.,86] J. S.Rosenschein, M.GINSBERG, et M. R.GENESERETH. Cooperation without communication. *Proceedings of the Fifth National Conference on Artificial Intelligence, AAAI'86*. 1986.
- [Rousseau et al., 85] J. M.Rousseau et J.Y.Blais. HASTUS: An interactive systems for buses and crew scheduling. *Computer Scheduling of Public Transport-2. J.M.Rousseau ed. pp.45-60, Amsterdam, North Holland, 1985*.
- [Rouvrais, 02] S. Rouvrais. *Utilisation d'agents mobiles pour la construction de services distribués*. Thèse de doctorat. Université de Rennes I IRISA, 5 juillet 2002.
- [Sandholm, 93] T.W. Sandholm. An Implementation of the Contract-Net protocol based on marginal cost calculations. *Proceedings 11th national conference on artificial intelligence*, 1993.
- [Siarry et al., 03] P.Siarry, J.Dréo, A.Pétrowski et E.Taillard. *Métaheuristiques pour l'optimisation difficile*. Eyrolles, 2003.
- [Smith, 80] Reid G. Smith. The Contract Net Protocol: highlevel communication and control in a distributed problem solver. *IEEE Transactions on computers*, C-29(12):1104–1113. Décembre. 1980.
- [Stevens, 95] G.Stevens. *An approach to scheduling aircraft landing times using genetic algorithms*. Honours thesis, RMIT, Department of Computer Science. Novembre 1995.
- [Strugeon, 98] E.G.L.Strugeon, E.Adam et C.Koski. Agents Intelligents en interaction homme-machine dans les systèmes d'information. In Kolski C.(dir.), *Environnements évolués et évaluation de l'IHM, Interaction homme-machine pour les SI 2*, hermès, p.209-248, 2001.
- [Theilmann et al., 99] W.Theilmann et K.Rothermel. Efficient Dissemination of Mobile Agents. In *Proceedings of 19th IEEE International Conference on Distributed Computing Systems Workshop*. IEEE Computer Society, May 1999, pp9-14.

- [Tsvetovatyy, 1997] M. Tsvetovatyy, M.Gini, B.Mobasher et Z.Wieckwski. MAGMA: An agent virtual market for electronic commerce. *Applied Artificial Intelligence*, 1997.
- [Tubbs, 84] S.L.Tubbs. *A system Approach to Small Group Interaction*. 2nd edition, Addison Wesley, Reading MA, 1984.
- [Vauvert et al., 00] G. Vauvert et A. E. Seghrouchni. Formation de coalitions pour agents rationnels. *Proceedings des JLIPN'2000*. Villetaneuse, France, 11-12 Septembre 2000.
- [Verrons, 04] M.H.Verrons. *Genca: un modèle général de négociation de contrats entre agents*. Thèse de doctorat, Université des sciences et technologies de Lille, laboratoire d'informatique fondamentale de Lille – UMR 8022. France, novembre 2004.
- [Weiss, 00] G.Weiss. *Multiagent Systems, A Modern Approach to distributed Artificial Intelligence*. The MIT Press, 2000.
- [Wooldridge et al., 95] M.J.Wooldridge et N.R. Jennings. Intelligent Agents: Theory and Practice. *The knowledge Engineering Review*, 10 (2), pp.115-152, 1995.
- [Zgaya et al., 05a] H.Zgaya, S.Hammadi et K.Ghédira. Workplan Mobile Agent for the Transport Network Application. *In Proceedings of the 17th IMACS World Congress Scientific Computation, Applied Mathematics and Simulation (IMACS'2005)*. Paris, Juillet 11-15, 2005.
- [Zgaya et al., 05b] H.Zgaya, S.Hammadi, K.Ghédira. Evolutionary method to optimize Workplan mobile agent for the transport network application. *In Proceedings of the International Conference on Systems, Man and Cybernetics (IEEE SMC'2005)*, Hawaii, USA, October 10-12, 2005.
- [Zgaya et al., 06a] H. Zgaya, S.Hammadi. Dynamic Approach to Reassign Tasks when Servers Breakdown in a Multi-Modal Information System. *In Proceedings of the 2006 IMACS Multiconference on Computational Engineering in Systems Applications (CESA'2006)*, Beijing, China, October 4-6, 2006. Tsinghua University Press, Beijing, China, Volume 1 pp. 985-990.
- [Zgaya et al., 06b] H. Zgaya, Slim Hammadi. Assignment and Integration of Distributed Transport Services in Agent-Based Architecture. *In Proceedings of the IEEE/WIC international conference on Intelligent Agent Technology (WI/IAT'2006)*, Hong Kong, China, December 18-22, 2006.

- [Zgaya et al., 06c] H. Zgaya, S. Hammadi. Stratégie de migration d'agents mobiles pour les applications du réseau de transport. *Revue de l'Electricité et de l'électronique (REE)*. N°5/2006 Méthodologies et heuristiques pour l'optimisation des systèmes industriels.
- [Zgaya et al., 07a] H. Zgaya et S.Hammadi. Multi-Agent Information System Using Mobile Agent Negotiation Based on a Flexible Transport Ontology. *In Proceedings of the 2007 International Conference on Autonomous Agents and Multi-agent Systems (AAMAS'2007)*. May14-182007, Honolulu, Hawai'i.
- [Zgaya et al., 07b] H. Zgaya, S. Hammadi, K.Ghédira. A migration strategy of mobile agents for the transport network applications. *IMACS International Journal, Mathematics and Computers in Simulation (MATCOM2835)*, PII : S0378-4754(07)00188-7.
- [Zgaya et al., 07c] H. Zgaya, S. Hammadi. Combination of mobile agent and evolutionary algorithm to optimize the client transport services. *RAIRO-Operations Research. Special Issue on Cooperative methods for Multiobjective Optimization (à paraître en 2007)*.
- [Zidi, 06] K.Zidi. *Système Interactif d'Aide au Déplacement Multimodal (SIADM)*. Thèse de doctorat, LAGIS de l'Ecole Centrale de Lille et l'Université des Sciences et Technologies de Lille. France, décembre 2006.
- [Zidi et al., 06] S.Zidi, S.Maouche et S.Hammadi. Ant colony optimization for spatial reconfiguration of multimodal transport network. *CTS2006 11th IFAC-Symposium on Control in Transportation Systems* Delft, Netherlands., August 2006.
- [Zitzler et al., 98] E. Zitzler and L. Thiele, Multiobjective Optimization Using Evolutionary Algorithms: A Comparative Case Study, *In Lecture Notes in Computer Science*, 1498: 292-301, UK (1998).

ANNEXES

Annexe A

Serveurs	
RéfServeur	NomServeur
1	S1
2	S2
3	S3
4	S4
5	S5
6	S6
7	S7
8	S8
9	S9
10	S10
11	S11
12	S12
13	S13
14	S14
15	S15
16	S16
17	S17
18	S18
19	S19
20	S20

Tâches				
RéfTâche	RéfServeur	temps	coût	taille
1	1	0,5	4	0,2
1	3	0,3	2	0,2
1	4	0,1	10	0,45
1	5	0,8	4	0,15
1	6	0,1	4	0,15
1	7	0,3	10	0,2
1	8	0,1	3	0,3
1	9	0,8	3	0,15
1	10	0,1	10	0,3
1	11	0,3	4	0,45
1	12	0,4	10	0,35
1	14	0,1	10	0,3
1	15	0,1	1	0,15
1	17	0,3	10	0,25
1	18	0,1	3	0,45
2	1	0,8	2	0,35
2	3	0,2	3	0,25

Tâches				
RéfTâche	RéfServeur	temps	coût	taille
2	4	0,5	4	0,3
2	7	0,1	10	0,45
2	8	0,1	2	0,3
2	9	0,4	6	0,15
2	10	0,2	5	0,1
2	12	0,3	6	0,15
2	13	0,4	3	0,05
2	14	0,8	10	0,3
2	15	0,4	5	0,2
2	17	0,4	6	0,3
2	18	0,5	2	0,35
2	19	0,4	6	0,05
2	20	0,1	5	0,15
3	1	0,8	5	0,3
3	3	0,3	4	0,4
3	4	0,8	3	0,45
3	5	0,8	7	0,05
3	7	0,3	11	0,2
3	8	0,8	5	0,15
3	9	0,8	1	0,45
3	10	0,5	5	0,15
3	11	0,8	2	0,1
3	12	0,3	1	0,35
3	14	0,3	5	0,25
3	15	0,8	5	0,2
3	16	0,8	6	0,15
3	18	0,5	5	0,35
3	19	0,8	9	0,2
3	20	0,8	5	0,3
4	1	0,1	10	0,05
4	2	0,3	1	0,1
4	3	0,1	4	0,05
4	4	0,2	2	0,3
4	5	0,1	4	0,15
4	7	0,1	1	0,2
4	8	0,5	6	0,45
4	9	0,1	3	0,25
4	10	0,1	6	0,3
4	13	0,2	1	0,1
4	14	0,1	4	0,05
4	15	0,1	3	0,35
4	16	0,5	8	0,05
4	20	0,1	1	0,3
5	1	0,5	6	0,1
5	3	0,5	4	0,45
5	4	0,5	3	0,2

Tâches				
RéfTâche	RéfServeur	temps	coût	taille
5	5	0,5	2	0,15
5	7	0,5	1	0,05
5	8	0,8	6	0,1
5	9	0,5	5	0,2
5	10	0,3	2	0,15
5	11	0,5	7	0,05
5	12	0,4	6	0,25
5	14	0,5	3	0,15
5	16	0,5	7	0,1
5	17	0,5	8	0,05
5	18	0,8	10	0,35
5	19	0,5	6	0,1
5	20	0,5	3	0,25
6	1	0,2	7	0,05
6	3	0,3	3	0,1
6	4	0,2	8	0,2
6	5	0,2	7	0,35
6	7	0,5	5	0,15
6	8	0,2	7	0,05
6	9	0,2	5	0,1
6	10	0,2	3	0,05
6	11	0,7	10	0,25
6	12	0,2	9	0,2
6	15	0,3	7	0,3
6	16	0,2	5	0,15
6	17	0,2	2	0,05
6	18	0,7	7	0,4
6	19	0,2	7	0,3
6	20	0,2	5	0,15
7	3	0,6	7	0,45
7	4	0,6	8	0,05
7	8	0,6	9	0,15
7	9	0,3	7	0,2
7	12	0,5	5	0,35
7	13	0,6	6	0,3
7	14	0,7	4	0,2
7	15	0,6	2	0,15
7	18	0,6	10	0,05
7	19	0,6	7	0,45
7	20	0,6	3	0,1
8	1	0,6	2	0,2
8	3	0,3	5	0,15
8	4	0,6	8	0,05
8	7	0,3	10	0,35
8	8	0,6	7	0,05
8	9	0,6	3	0,2

Tâches				
RéfTâche	RéfServeur	temps	coût	taille
8	10	0,1	6	0,45
8	12	0,2	6	0,15
8	13	0,5	8	0,15
8	14	0,5	9	0,05
8	15	0,9	3	0,25
8	16	0,3	7	0,35
8	17	0,6	8	0,35
8	18	0,1	2	0,45
8	20	0,2	6	0,25
9	4	0,1	7	0,15
9	8	0,3	10	0,15
9	9	0,2	5	0,4
9	10	0,1	1	0,15
9	11	0,2	4	0,1
9	12	0,9	1	0,4
9	13	0,5	6	0,1
9	15	0,8	8	0,2
9	17	0,1	4	0,4
9	20	0,9	1	0,25
10	4	0,9	4	0,25
10	5	0,1	6	0,35
10	9	0,9	8	0,05
10	10	0,8	10	0,05
10	11	0,8	1	0,05
10	14	0,1	3	0,1
10	16	0,4	1	0,4
10	17	0,9	10	0,05
10	19	0,6	5	0,1
11	3	0,1	6	0,35
11	4	0,3	10	0,15
11	5	0,1	9	0,4
11	7	0,6	7	0,05
11	8	0,7	2	0,15
11	9	0,8	3	0,2
11	10	0,3	2	0,45
11	12	0,1	1	0,05
11	14	0,7	3	0,35
11	15	0,3	9	0,01
11	16	0,1	2	0,45
11	17	0,9	1	0,05
11	18	0,5	10	0,05
11	19	0,7	3	0,3
11	20	0,6	8	0,05
12	1	0,7	6	0,2
12	3	0,1	3	0,1
12	4	0,9	8	0,1

Tâches				
RéfTâche	RéfServeur	temps	coût	taille
12	5	0,2	1	0,05
12	10	0,9	8	0,4
12	11	0,9	5	0,2
12	13	0,4	3	0,45
12	15	0,8	5	0,05
12	16	0,5	10	0,15
12	17	0,1	9	0,2
12	18	0,9	8	0,4
12	20	0,6	10	0,45
13	1	0,1	7	0,2
13	3	0,9	6	0,25
13	4	0,1	6	0,15
13	5	0,3	4	0,1
13	8	0,8	7	0,1
13	9	0,4	5	0,05
13	10	0,9	5	0,4
13	11	0,4	8	0,15
13	12	0,8	4	0,3
13	15	0,7	2	0,3
13	16	0,4	6	0,1
13	18	0,7	10	0,2
13	19	0,9	7	0,25
14	4	0,2	10	0,4
14	7	0,6	7	0,05
14	10	0,2	10	0,25
14	11	0,6	10	0,1
14	12	0,3	3	0,1
14	13	0,6	1	0,45
14	14	0,5	2	0,15
14	16	0,1	5	0,05
14	17	0,3	2	0,35
14	18	0,6	1	0,05
14	19	0,1	4	0,2
14	20	0,3	8	0,35
15	1	0,3	7	0,4
15	3	0,1	7	0,2
15	4	0,4	10	0,45
15	5	0,8	8	0,35
15	8	0,2	8	0,25
15	9	0,2	9	0,2
15	10	0,8	1	0,05
15	11	0,3	4	0,3
15	12	0,9	7	0,15
15	13	0,2	2	0,3
15	15	0,7	2	0,45
15	16	0,1	3	0,2

Tâches				
RéfTâche	RéfServeur	temps	coût	taille
15	18	0,9	2	0,35
15	19	0,7	5	0,45
15	20	0,4	5	0,2
16	5	0,3	5	0,3
16	7	0,5	7	0,15
16	19	0,2	8	0,45
17	1	0,2	1	0,5
17	6	0,6	2	0,25
17	19	0,5	5	0,3
17	20	0,9	6	0,9
18	2	0,8	1	0,2
18	3	0,3	2	0,7
18	4	0,6	9	0,9
18	9	0,6	10	0,4
18	10	0,9	5	0,4
18	19	0,2	6	0,5
18	20	0,3	3	0,25
19	1	0,8	4	0,1
19	5	0,2	6	0,3
20	3	0,3	5	0,2
20	5	0,9	1	0,5
20	10	0,1	9	0,8
20	15	0,8	3	0,1
20	20	0,1	1	0,9
21	3	0,2	2	0,9
21	4	0,2	3	0,1
21	18	0,8	8	0,7
22	17	0,2	3	0,5
22	19	0,8	1	0,4
22	20	0,4	8	0,4
23	1	0,5	1	0,9
23	9	0,8	5	0,2
23	13	0,2	9	0,5
23	16	0,1	5	0,8
23	20	0,3	2	0,25
24	9	0,3	2	0,6
24	19	0,2	3	0,55
25	4	0,5	5	0,5
25	16	0,3	6	0,25
25	17	0,2	7	0,3
25	20	0,1	9	0,15
26	2	0,1	5	0,4
26	3	0,2	9	0,8
26	11	0,3	7	0,7
26	12	0,8	5	0,25
26	18	0,2	9	0,2

Tâches				
RéfTâche	RéfServeur	temps	coût	taille
27	18	0,3	2	0,5
27	19	0,6	1	0,4
27	20	0,3	2	0,4
28	2	0,3	9	0,9
28	3	0,9	5	0,25
28	5	0,3	4	0,8
28	10	0,8	2	0,35
28	12	0,2	7	0,25
28	15	0,8	4	0,9
28	19	0,2	5	0,85
28	20	0,1	6	0,8
29	1	0,3	4	0,5
29	2	0,3	2	0,4
29	11	0,5	3	0,15
29	13	0,4	4	0,2
30	17	0,2	10	0,05
30	18	0,3	9	0,1
30	19	0,3	8	0,25
30	20	0,4	7	0,3
31	1	0,5	4	0,2
31	3	0,3	2	0,2
31	4	0,1	10	0,45
31	6	0,1	4	0,15
31	8	0,1	3	0,3
31	12	0,4	10	0,35
31	14	0,1	10	0,3
31	18	0,1	3	0,45
32	1	0,8	2	0,35
32	3	0,2	3	0,25
32	4	0,5	4	0,3
32	7	0,1	10	0,45
32	8	0,1	2	0,3
32	9	0,4	6	0,15
32	12	0,3	6	0,15
32	14	0,8	10	0,3
32	18	0,5	2	0,35
33	1	0,8	5	0,3
33	3	0,3	4	0,4
33	4	0,8	3	0,45
33	7	0,3	11	0,2
33	12	0,3	1	0,35
33	16	0,8	6	0,15
34	1	0,1	10	0,05
34	2	0,3	1	0,1
34	3	0,1	4	0,05
34	7	0,1	1	0,2

Tâches				
RéfTâche	RéfServeur	temps	coût	taille
35	1	0,5	6	0,1
35	3	0,5	4	0,45
35	8	0,8	6	0,1
35	18	0,8	10	0,35
36	1	0,2	7	0,05
36	3	0,3	3	0,1
36	18	0,7	7	0,4
37	3	0,6	7	0,45
37	18	0,6	10	0,05
37	20	0,6	3	0,1
38	1	0,6	2	0,2
38	3	0,3	5	0,15
38	10	0,1	6	0,45
38	14	0,5	9	0,05
38	17	0,6	8	0,35
39	10	0,1	1	0,15
39	17	0,1	4	0,4
39	20	0,9	1	0,25
40	11	0,8	1	0,05
40	16	0,4	1	0,4
40	17	0,9	10	0,05
41	3	0,1	6	0,35
41	5	0,1	9	0,4
41	18	0,5	10	0,05
42	1	0,7	6	0,2
42	3	0,1	3	0,1
42	11	0,9	5	0,2
42	18	0,9	8	0,4
43	1	0,1	7	0,2
43	3	0,9	6	0,25
43	9	0,4	5	0,05
43	10	0,9	5	0,4
43	11	0,4	8	0,15
43	19	0,9	7	0,25
44	11	0,6	10	0,1
44	17	0,3	2	0,35
45	1	0,3	7	0,4
45	3	0,1	7	0,2
45	8	0,2	8	0,25
45	18	0,9	2	0,35
46	9	0,4	5	0,2
46	19	0,2	8	0,45
47	1	0,2	1	0,5
47	6	0,6	2	0,25
47	19	0,5	5	0,3
48	2	0,8	1	0,2

Tâches				
RéfTâche	RéfServeur	temps	coût	taille
48	3	0,3	2	0,7
48	4	0,6	9	0,9
48	19	0,2	6	0,5
49	1	0,8	4	0,1
50	3	0,3	5	0,2
50	15	0,8	3	0,1
51	3	0,2	2	0,9
51	18	0,8	8	0,7
52	17	0,2	3	0,5
52	19	0,8	1	0,4
53	1	0,5	1	0,9
53	13	0,2	9	0,5
53	16	0,1	5	0,8
54	19	0,2	3	0,55
55	16	0,3	6	0,25
55	17	0,2	7	0,3
56	2	0,1	5	0,4
56	3	0,2	9	0,8
56	11	0,3	7	0,7
56	18	0,2	9	0,2
57	18	0,3	2	0,5
57	19	0,6	1	0,4
58	2	0,3	9	0,9
58	3	0,9	5	0,25
58	10	0,8	2	0,35
58	15	0,8	4	0,9
58	19	0,2	5	0,85
59	1	0,3	4	0,5
59	2	0,3	2	0,4
59	11	0,5	3	0,15
59	13	0,4	4	0,2
60	17	0,2	10	0,05
60	18	0,3	9	0,1
60	19	0,3	8	0,25
61	3	0,1	2	0,1
61	15	0,1	1	0,25
61	17	0,2	10	0,25
62	8	0,1	2	0,3
62	14	0,8	10	0,3
62	18	0,5	2	0,15
63	1	0,8	5	0,3
63	3	0,3	4	0,4
63	4	0,8	3	0,25
63	5	0,5	7	0,05
63	7	0,3	2	0,2
63	8	0,8	7	0,15

Tâches				
RéfTâche	RéfServeur	temps	coût	taille
63	9	0,8	1	0,15
63	16	0,8	6	0,15
63	18	0,5	5	0,35
64	1	0,2	10	0,15
64	2	0,3	1	0,1
64	3	0,1	4	0,05
64	4	0,2	2	0,3
64	5	0,1	4	0,15
64	9	0,1	3	0,25
65	1	0,5	6	0,1
65	14	0,4	3	0,15
65	18	0,8	10	0,35
66	1	0,2	7	0,05
66	3	0,1	3	0,1
66	15	0,3	8	0,2
67	3	0,6	7	0,45
67	13	0,6	6	0,3
67	19	0,6	7	0,25
68	1	0,6	2	0,2
68	3	0,3	5	0,1
68	6	0,3	5	0,15
68	8	0,6	7	0,05
68	17	0,6	8	0,25
69	9	0,2	5	0,4
69	11	0,2	4	0,1
70	9	0,9	8	0,15
70	11	0,8	1	0,05
70	14	0,1	3	0,1
71	3	0,7	6	0,35
71	7	0,4	7	0,05
71	14	0,3	3	0,25
72	1	0,8	6	0,2
72	3	0,2	3	0,1
72	10	0,5	8	0,4
72	13	0,5	3	0,15
72	18	0,5	8	0,4
73	15	0,7	2	0,3
73	16	0,4	2	0,1
73	19	0,9	7	0,25
74	7	0,6	7	0,05
74	14	0,5	2	0,25
74	16	0,1	5	0,15
75	1	0,3	7	0,4
75	3	0,1	7	0,2
75	10	0,8	1	0,05
75	12	0,9	2	0,25

Tâches				
RéfTâche	RéfServeur	temps	coût	taille
76	2	0,1	5	0,4
76	3	0,2	9	0,8
76	18	0,2	9	0,25
77	18	0,3	2	0,5
77	20	0,3	2	0,2
78	2	0,3	9	0,8
78	12	0,3	7	0,35
79	1	0,5	4	0,5
79	2	0,2	2	0,4
80	17	0,2	10	0,05
80	20	0,1	7	0,3
81	1	0,5	4	0,2
81	3	0,4	2	0,2
81	4	0,2	10	0,45
81	12	0,3	10	0,15
81	18	0,1	3	0,45
82	1	0,8	2	0,35
82	3	0,2	3	0,25
82	7	0,1	5	0,45
82	8	0,1	2	0,3
82	9	0,4	6	0,15
82	14	0,5	2	0,3
83	5	0,2	7	0,15
83	7	0,9	3	0,2
83	12	0,3	1	0,35
83	14	0,3	5	0,25
84	1	0,1	10	0,05
84	2	0,3	1	0,1
84	3	0,1	4	0,05
85	1	0,5	6	0,1
85	3	0,5	4	0,45
85	18	0,8	10	0,35
86	1	0,2	7	0,05
86	3	0,1	3	0,1
86	5	0,2	7	0,15
86	18	0,7	7	0,45
87	3	0,6	7	0,45
87	20	0,2	3	0,15
88	1	0,3	2	0,25
88	3	0,5	5	0,15
89	12	0,9	1	0,4
89	15	0,7	8	0,3
89	17	0,3	4	0,4
90	9	0,5	8	0,05
90	14	0,1	3	0,1
90	16	0,4	1	0,4

Tâches				
RéfTâche	RéfServeur	temps	coût	taille
90	17	0,3	10	0,05
91	6	0,1	9	0,4
91	18	0,5	5	0,05
91	19	0,7	3	0,3
92	3	0,1	3	0,1
92	13	0,4	3	0,45
92	17	0,1	9	0,2
92	18	0,9	8	0,4
93	1	0,1	7	0,2
93	3	0,9	6	0,15
93	5	0,3	4	0,1
93	9	0,4	5	0,05
94	11	0,6	10	0,15
94	12	0,3	3	0,1
94	13	0,6	1	0,35
94	14	0,5	2	0,15
94	18	0,6	1	0,05
95	5	0,6	8	0,35
95	9	0,5	9	0,2
95	11	0,3	4	0,35
95	13	0,2	2	0,3
96	19	0,3	8	0,35
97	1	0,1	1	0,5
97	19	0,5	5	0,3
98	2	0,8	2	0,2
98	3	0,3	2	0,7
98	9	0,6	10	0,4
98	20	0,3	2	0,25
99	1	0,8	4	0,15
99	5	0,2	6	0,3
100	3	0,3	4	0,1
100	5	0,3	3	0,5
100	20	0,2	2	0,5

Latencies		
RéfServeur1	RéfServeur2	SL
0	1	0,25
0	2	0,13
0	3	0,71
0	4	0,49
0	5	0,39
0	6	0,4
0	7	0,19

Latencies		
RéfServeur1	RéfServeur2	SL
0	8	0,79
0	9	5,29
0	10	0,52
0	11	0,67
0	12	0,88
0	13	0,89
0	14	0,25
0	15	0,85
0	16	0,2
0	17	0,89
0	18	0,11
0	19	0,38
0	20	0,02
1	2	0,25
1	3	0,36
1	4	0,46
1	5	0,46
1	6	0,68
1	7	0,28
1	8	0,13
1	9	0,14
1	10	0,82
1	11	0,49
1	12	0,87
1	13	0,41
1	14	0,02
1	15	0,1
1	16	0,79
1	17	0,95
1	18	0,94
1	19	0,11
1	20	0,34
2	3	0,98
2	4	0,61
2	5	0,76
2	6	0,74
2	7	0,82
2	8	0,44
2	9	0,28
2	10	0,98
2	11	0,26
2	12	0,81
2	13	0,17
2	14	0,71
2	15	0,42
2	16	0,44

Latencies		
RéfServeur1	RéfServeur2	SL
2	17	0,52
2	18	0,21
2	19	0,96
2	20	0,97
3	4	0,16
3	5	0,28
3	6	0,16
3	7	0,93
3	8	0,49
3	9	0,04
3	10	0,9
3	11	0,82
3	12	0,74
3	13	0,53
3	14	0,96
3	15	0,13
3	16	0,16
3	17	0,1
3	18	0,04
3	19	0,16
3	20	0,25
4	5	0,95
4	6	0,11
4	7	0,46
4	8	0,48
4	9	0,5
4	10	0,76
4	11	0,17
4	12	0,23
4	13	0,89
4	14	0,27
4	15	0,14
4	16	0,29
4	17	0,33
4	18	0,74
4	19	0,31
4	20	0,24
5	6	0,16
5	7	0,9
5	8	0,34
5	9	0,33
5	10	0,89
5	11	0,36
5	12	0,31
5	13	0,58
5	14	0,17

Latencies		
RéfServeur1	RéfServeur2	SL
5	15	0,15
5	16	0,38
5	17	0,1
5	18	0,45
5	19	0,75
5	20	0,13
6	7	0,35
6	8	0,5
6	9	0,37
6	10	0,34
6	11	0,51
6	12	0,68
6	13	0,6
6	14	0,39
6	15	0,87
6	16	0,3
6	17	0,62
6	18	0,61
6	19	0,19
6	20	0,51
7	8	0,71
7	9	0,28
7	10	0,09
7	11	0,69
7	12	0,96
7	13	0,01
7	14	0,56
7	15	0,52
7	16	0,51
7	17	0,21
7	18	0,1
7	19	0,84
7	20	0,64
8	9	0,04
8	10	0,67
8	11	0,72
8	12	0,42
8	13	0,4
8	14	0,2
8	15	0,33
8	16	0,46
8	17	0,15
8	18	0,17
8	19	0,73
8	20	0,41
9	10	0,7

Latencies		
RéfServeur1	RéfServeur2	SL
9	11	0,78
9	12	0,85
9	13	0,73
9	14	0,54
9	15	0,12
9	16	0,31
9	17	0,5
9	18	0,63
9	19	0,37
9	20	0,49
10	11	0,27
10	12	0,34
10	13	0,88
10	14	0,15
10	15	0,58
10	16	0,08
10	17	0,2
10	18	0,23
10	19	0,14
10	20	0,86
11	12	0,16
11	13	0,89
11	14	0,49
11	15	0,74
11	16	0,58
11	17	0,08
11	18	0,7
11	19	0,47
11	20	0,4
12	13	0,37
12	14	0,36
12	15	0,47
12	16	0,41
12	17	0,12
12	18	0,59
12	19	0,06
12	20	0,69
13	14	0,98
13	15	0,49
13	16	0,88
13	17	0,2
13	18	0,16
13	19	0,19
13	20	0,47
14	15	0,2
14	16	0,16

Latencies		
RéfServeur1	RéfServeur2	SL
14	17	0,66
14	18	0,2
14	19	0,9
14	20	0,32
15	16	0,16
15	17	0,96
15	18	0,18
15	19	0,94
15	20	0,1
16	17	0,84
16	18	0,81
16	19	0,96
16	20	0,57
17	18	0,14
17	19	0,81
17	20	0,4
18	19	0,72
18	20	0,6
19	20	0,11

Annexe B

Scénario N°1 :

IAO : Collecting data from ETMN

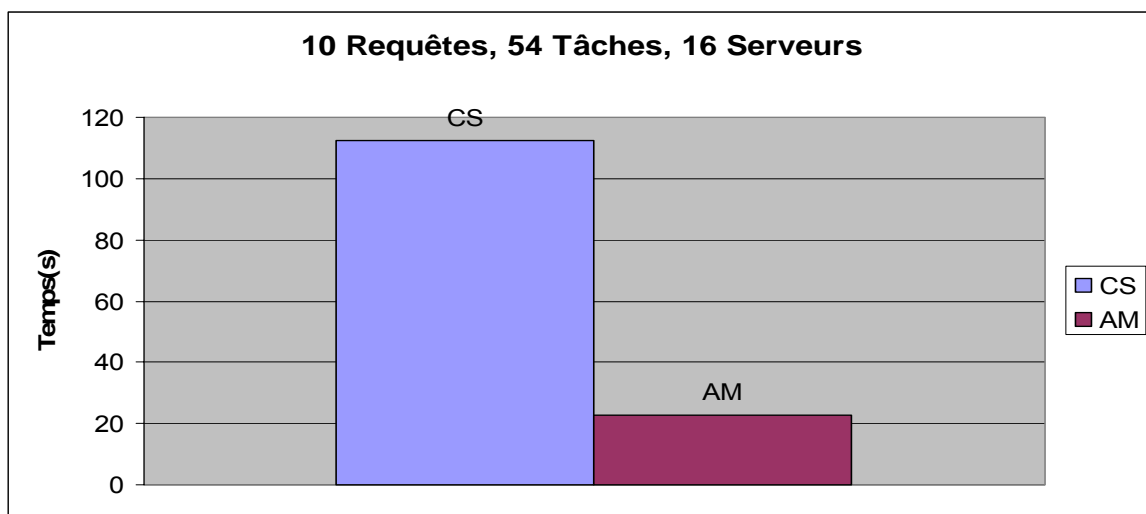
ICA	Service	Data	Source
MA5	T 93	S9T93	S9
MA2	T 34	S7T34	S7
MA2	T 83	S7T83	S7
MA2	T 39	S17T39	S17
MA2	T 21	S18T21	S18
MA2	T 35	S18T35	S18
MA2	T 36	S18T36	S18
MA2	T 37	S18T37	S18
MA2	T 51	S18T51	S18
MA3	T 18	S2T18	S2
MA3	T 29	S2T29	S2
MA3	T 64	S2T64	S2
MA3	T 76	S2T76	S2
MA3	T 84	S2T84	S2
MA3	T 15	S13T15	S13
MA3	T 23	S13T23	S13
MA3	T 22	S19T22	S19
MA3	T 24	S19T24	S19
MA3	T 46	S19T46	S19
MA3	T 48	S19T48	S19
MA3	T 54	S19T54	S19
MA3	T 96	S19T96	S19
MA1	T 8	S1T8	S1
MA1	T 65	S1T65	S1
MA1	T 79	S1T79	S1

Number of ICA : 5
Number of Servers : 16

Request	TotalCost	TotalTime	DueDate
IA1	229	22.56	25
IA5	138	22.56	25
IA4	8	19.96	25
IA8	32	22.56	25
IA2	33	22.56	25
IA9	12	22.56	25
IA7	151	22.56	25
IA6	12	20.66	25
IA3	97	22.56	25
IA10	22	22.56	25

New (nbRequests, nbTasks) = (10 , 54)

Quantité de donnée totale transférée : 16,55 unités



Instance FeTAR générée																			
CH	S ₁	S ₂	S ₃	S ₄	S ₅	S ₇	S ₈	S ₉	S ₁₀	S ₁₁	S ₁₂	S ₁₃	S ₁₄	S ₁₅	S ₁₆	S ₁₇	S ₁₈	S ₁₉	S ₂₀
T ₅	*	X	*	*	*	*	*	*	*	*	1	X	*	X	*	*	*	*	*
T ₆	*	X	*	*	*	*	*	*	*	*	1	X	X	*	*	*	*	*	*
T ₈	1	X	*	*	X	*	*	*	*	X	*	*	*	*	*	*	*	*	X
T ₉	X	X	X	1	X	X	*	*	*	*	*	*	X	*	X	*	X	X	*
T ₁₃	*	X	*	*	*	X	*	*	*	*	1	X	X	*	*	X	*	*	X
T ₁₅	*	X	*	*	*	X	*	*	*	*	*	1	X	*	*	X	*	*	*
T ₁₈	X	1	*	*	X	X	X	*	*	X	X	X	X	X	X	X	X	*	*
T ₁₉	*	X	X	X	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X
T ₂₁	X	X	*	*	X	X	X	X	X	X	X	X	X	X	X	X	1	X	X
T ₂₂	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	*	X	1	*
T ₂₃	*	X	X	X	X	X	X	*	X	X	X	1	X	X	*	X	X	X	*
T ₂₄	X	X	X	X	X	X	X	*	X	X	X	X	X	X	X	X	X	1	X
T ₂₆	X	*	1	X	X	X	X	X	X	*	*	X	X	X	X	X	*	X	X
T ₂₇	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	*	*	1
T ₂₉	*	1	X	X	X	X	X	X	X	*	X	*	X	X	X	X	X	X	X
T ₃₀	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	*	*	*	1
T ₃₃	*	X	*	*	X	*	X	X	X	X	1	X	X	X	*	X	X	X	X
T ₃₄	*	*	*	X	X	1	X	X	X	X	X	X	X	X	X	X	X	X	X
T ₃₅	*	X	*	X	X	X	*	X	X	X	X	X	X	X	X	X	1	X	X
T ₃₆	*	X	*	X	X	X	X	X	X	X	X	X	X	X	X	X	1	X	X
T ₃₇	X	X	*	X	X	X	X	X	X	X	X	X	X	X	X	X	1	X	*
T ₃₈	*	X	1	X	X	X	X	X	*	X	X	X	*	X	X	*	X	X	X
T ₃₉	X	X	X	X	X	X	X	X	*	X	X	X	X	X	X	1	X	X	*
T ₄₁	X	X	*	X	1	X	X	X	X	X	X	X	X	X	X	X	*	X	X
T ₄₂	*	X	1	X	X	X	X	X	X	*	X	X	X	X	X	X	*	X	X
T ₄₆	X	X	X	X	X	X	X	*	X	X	X	X	X	X	X	X	X	1	X
T ₄₈	X	*	*	*	X	X	X	X	X	X	X	X	X	X	X	X	X	1	X
T ₅₁	X	X	*	X	X	X	X	X	X	X	X	X	X	X	X	X	1	X	X
T ₅₃	*	X	X	X	X	X	X	X	X	X	X	*	X	X	1	X	X	X	X
T ₅₄	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	X
T ₅₉	*	*	X	X	X	X	X	X	X	1	X	*	X	X	X	X	X	X	X
T ₆₁	X	X	*	X	X	X	X	X	X	X	X	X	X	1	X	*	X	X	X
T ₆₃	*	X	1	*	*	*	*	*	X	X	X	X	X	X	*	X	*	X	X
T ₆₄	*	1	*	*	*	X	X	*	X	X	X	X	X	X	X	X	X	X	X
T ₆₅	1	X	X	X	X	X	X	X	X	X	X	X	*	X	X	X	*	X	X
T ₆₆	*	X	*	X	X	X	X	X	X	X	X	X	X	1	X	X	X	X	X
T ₇₂	*	X	1	X	X	X	X	X	*	X	X	*	X	X	X	X	*	X	X
T ₇₄	X	X	X	X	X	*	X	X	X	X	X	X	*	X	1	X	X	X	X
T ₇₆	X	1	*	X	X	X	X	X	X	X	X	X	X	X	X	X	*	X	X
T ₇₈	X	*	X	X	X	X	X	X	X	X	1	X	X	X	X	X	X	X	X
T ₇₉	1	*	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
T ₈₃	X	X	X	X	*	1	X	X	X	X	*	X	*	X	X	X	X	X	X
T ₈₄	*	1	*	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
T ₈₅	*	X	1	X	X	X	X	X	X	X	X	X	X	X	X	X	*	X	X
T ₈₆	*	X	1	X	*	X	X	X	X	X	X	X	X	X	X	X	*	X	X
T ₈₇	X	X	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	*
T ₈₈	*	X	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
T ₉₂	X	X	1	X	X	X	X	X	X	X	X	*	X	X	X	*	*	X	X
T ₉₃	*	X	*	X	*	X	X	1	X	X	X	X	X	X	X	X	X	X	X
T ₉₄	X	X	X	X	X	X	X	X	X	*	1	*	*	X	X	X	*	X	X
T ₉₅	X	X	X	X	1	X	X	*	X	*	X	*	X	X	X	X	X	X	X
T ₉₆	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	X
T ₉₉	1	X	X	X	*	X	X	X	X	X	X	X	X	X	X	X	X	X	X
T ₁₀₀	X	X	1	X	*	X	X	X	X	X	X	X	X	X	X	X	X	X	*

Scénario N°2 :

IAO : Collecting data from ETMN

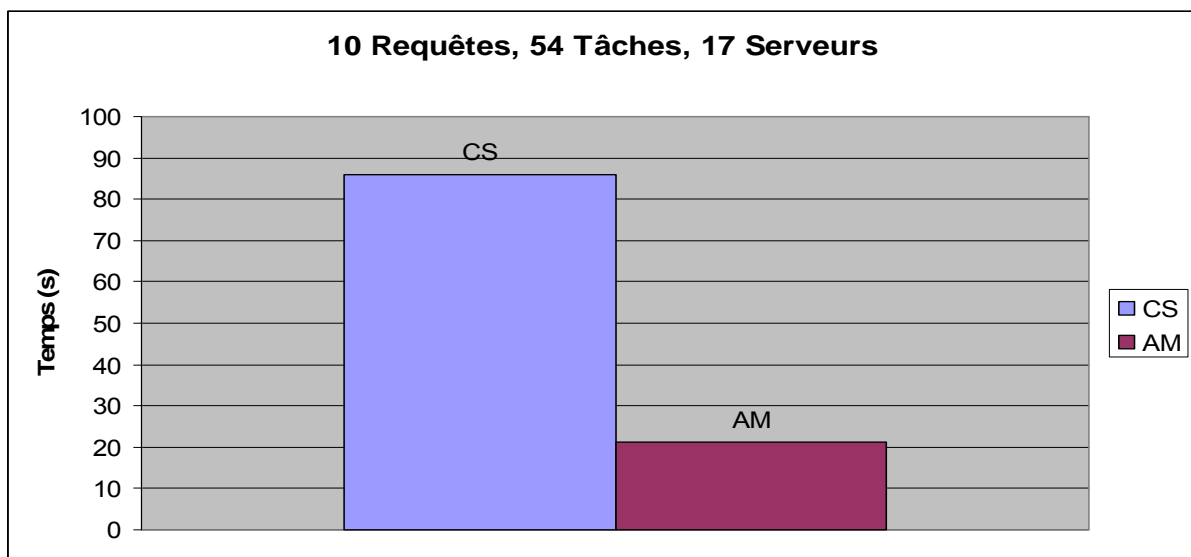
ICA	Service	Data	Source
MA5	T 13	S9T13	S9
MA5	T 24	S9T24	S9
MA5	T 64	S9T64	S9
MA5	T 93	S9T93	S9
MA5	T 9	S11T9	S11
MA5	T 29	S11T29	S11
MA3	T 26	S2T26	S2
MA3	T 78	S2T78	S2
MA3	T 79	S2T79	S2
MA3	T 92	S12T92	S13
MA3	T 94	S12T94	S13
MA3	T 95	S12T95	S13
MA3	T 46	S19T46	S19
MA3	T 54	S19T54	S19
MA3	T 96	S19T96	S19
MA1	T 34	S1T34	S1
MA1	T 35	S1T35	S1
MA1	T 66	S1T66	S1
MA1	T 99	S1T99	S1
MA1	T 18	S3T18	S3
MA1	T 37	S3T37	S3
MA1	T 38	S3T38	S3
MA1	T 51	S3T51	S3
MA1	T 61	S3T61	S3
MA1	T 76	S3T76	S3

Number of ICA : 5
Number of Servers : 17

Request	TotalCost	TotalTime	DueDate
IA9	5	19.259998	25
IA1	236	21.36	25
IA8	25	21.36	25
IA4	8	18.36	25
IA5	147	21.36	25
IA6	13	21.26	25
IA2	38	21.36	25
IA7	144	21.36	25
IA10	21	21.26	25
IA3	100	21.36	25

New (nbRequests, nbTasks) = (10 , 54)

Quantité de donnée totale transférée : 19,6 unités



Instance FeTAR générée																			
CH	S1	S2	S3	S4	S5	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16	S17	S18	S19	S20
T5	*	X	*	*	*	*	*	*	*	*	*	X	*	X	*	1	*	*	*
T6	*	X	*	*	*	*	*	*	*	*	*	X	X	1	*	*	*	*	*
T8	*	X	*	*	X	*	*	*	1	X	*	*	*	*	*	*	*	X	*
T9	X	X	X	*	X	X	*	*	*	1	*	*	X	*	X	*	X	X	*
T13	*	X	*	*	*	X	*	1	*	*	*	X	X	*	*	X	*	*	X
T15	*	X	*	*	*	X	*	*	*	*	1	*	X	*	*	X	*	*	*
T18	X	*	1	*	X	X	X	*	*	X	X	X	X	X	X	X	X	*	*
T19	*	X	X	X	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X
T21	X	X	*	*	X	X	X	X	X	X	X	X	X	X	X	X	1	X	X
T22	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	X	*	*
T23	*	X	X	X	X	X	X	*	X	X	X	*	X	X	1	X	X	X	*
T24	X	X	X	X	X	X	X	1	X	X	X	X	X	X	X	X	X	*	X
T26	X	1	*	X	X	X	X	X	X	*	*	X	X	X	X	X	*	X	X
T27	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	*	*
T29	*	*	X	X	X	X	X	X	X	1	X	*	X	X	X	X	X	X	X
T30	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	*	1	*	*
T33	*	X	*	*	*	X	*	X	X	X	1	X	X	X	*	X	X	X	X
T34	1	*	*	X	X	*	X	X	X	X	X	X	X	X	X	X	X	X	X
T35	1	X	*	X	X	X	*	X	X	X	X	X	X	X	X	X	*	X	X
T36	*	X	*	X	X	X	X	X	X	X	X	X	X	X	X	X	1	X	X
T37	X	X	1	X	X	X	X	X	X	X	X	X	X	X	X	X	*	X	*
T38	*	X	1	X	X	X	X	X	*	X	X	X	*	X	X	*	X	X	X
T39	X	X	X	X	X	X	X	X	*	X	X	X	X	X	X	1	X	X	*
T41	X	X	*	X	1	X	X	X	X	X	X	X	X	X	X	X	*	X	X
T42	*	X	*	X	X	X	X	X	X	*	X	X	X	X	X	X	1	X	X
T46	X	X	X	X	X	X	X	*	X	X	X	X	X	X	X	X	X	1	X
T48	X	*	*	1	X	X	X	X	X	X	X	X	X	X	X	X	X	*	X
T51	X	X	1	X	X	X	X	X	X	X	X	X	X	X	X	X	*	X	X
T53	*	X	X	X	X	X	X	X	X	X	X	*	X	X	1	X	X	X	X
T54	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	X
T59	*	*	X	X	X	X	X	X	X	1	X	*	X	X	X	X	X	X	X
T61	X	X	1	X	X	X	X	X	X	X	X	X	*	X	*	X	*	X	X
T63	*	X	*	*	*	1	*	*	X	X	X	X	X	X	*	X	*	X	X
T64	*	*	*	*	*	X	X	1	X	X	X	X	X	X	X	X	X	X	X
T65	*	X	X	X	X	X	X	X	X	X	X	X	*	X	X	X	1	X	X
T66	1	X	*	X	X	X	X	X	X	X	X	X	X	*	X	X	X	X	X
T72	*	X	*	X	X	X	X	X	1	X	X	*	X	X	X	X	*	X	X
T74	X	X	X	X	X	*	X	X	X	X	X	X	1	X	*	X	X	X	X
T76	X	*	1	X	X	X	X	X	X	X	X	X	X	X	X	X	*	X	X
T78	X	1	X	X	X	X	X	X	X	X	*	X	X	X	X	X	X	X	X
T79	*	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
T83	X	X	X	X	*	*	X	X	X	X	1	X	*	X	X	X	X	X	X
T84	*	*	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
T85	*	X	1	X	X	X	X	X	X	X	X	X	X	X	X	X	*	X	X
T86	*	X	1	X	*	X	X	X	X	X	X	X	X	X	X	X	*	X	X
T87	X	X	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	*
T88	*	X	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
T92	X	X	*	X	X	X	X	X	X	X	1	X	X	X	*	*	*	X	X
T93	*	X	*	X	*	X	X	1	X	X	X	X	X	X	X	X	X	X	X
T94	X	X	X	X	X	X	X	X	X	*	*	1	*	X	X	X	*	X	X
T95	X	X	X	X	*	X	X	*	X	*	X	1	X	X	X	X	X	X	X
T96	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	X
T99	1	X	X	X	*	X	X	X	X	X	X	X	X	X	X	X	X	X	X
T100	X	X	*	X	1	X	X	X	X	X	X	X	X	X	X	X	X	X	*

Scénario N°3 :

IA0 : Collecting data from ETMN

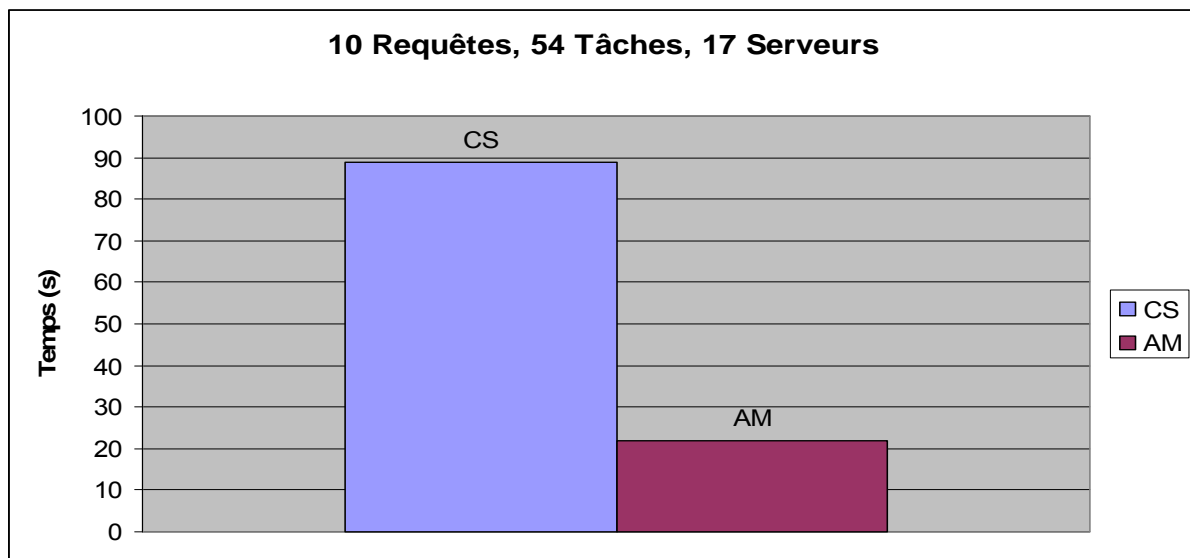
ICA	Service	Data	Source
MA3	T 76	S2T76	S2
MA3	T 78	S2T78	S2
MA3	T 84	S2T84	S2
MA3	T 72	S12T72	S13
MA3	T 15	S19T15	S19
MA3	T 27	S19T27	S19
MA3	T 30	S19T30	S19
MA3	T 46	S19T46	S19
MA3	T 54	S19T54	S19
MA3	T 96	S19T96	S19
MA1	T 53	S1T53	S1
MA1	T 65	S1T65	S1
MA1	T 79	S1T79	S1
MA1	T 85	S1T85	S1
MA1	T 88	S1T88	S1
MA1	T 99	S1T99	S1
MA1	T 21	S3T21	S3
MA1	T 35	S3T35	S3
MA1	T 36	S3T36	S3
MA1	T 41	S3T41	S3
MA1	T 48	S3T48	S3
MA1	T 66	S3T66	S3
MA1	T 86	S3T86	S3
MA1	T 87	S3T87	S3
MA1	T 100	S3T100	S3

Number of ICA : 5
Number of Servers : 17

Request	TotalCost	TotalTime	DueDate
IA4	8	19.36	25
IA5	130	21.76	25
IA1	214	21.76	25
IA8	35	21.76	25
IA9	13	21.76	25
IA6	12	21.56	25
IA10	19	21.76	25
IA2	35	21.76	25
IA3	110	21.76	25
IA7	151	21.76	25

New (nbRequests, nbTasks) = (10 , 54)

Quantité de donnée totale transférée : 16,55 unités



Instance FeTAR générée																			
CH	S ₁	S ₂	S ₃	S ₄	S ₅	S ₇	S ₈	S ₉	S ₁₀	S ₁₁	S ₁₂	S ₁₃	S ₁₄	S ₁₅	S ₁₆	S ₁₇	S ₁₈	S ₁₉	S ₂₀
T ₅	*	X	*	*	*	*	*	*	*	*	*	X	1	X	*	*	*	*	*
T ₆	*	X	*	*	*	*	*	*	*	*	*	X	X	*	*	1	*	*	*
T ₈	*	X	*	*	X	*	*	*	*	X	*	*	*	*	*	*	1	X	*
T ₉	X	X	X	*	X	X	*	*	1	*	*	*	X	*	X	*	X	X	*
T ₁₃	*	X	*	1	*	X	*	*	*	*	*	X	X	*	*	X	*	*	X
T ₁₅	*	X	*	*	*	X	*	*	*	*	*	*	X	*	*	X	*	1	*
T ₁₈	X	*	*	*	X	X	X	*	1	X	X	X	X	X	X	X	X	*	*
T ₁₉	*	X	X	X	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X
T ₂₁	X	X	1	*	X	X	X	X	X	X	X	X	X	X	X	X	*	X	X
T ₂₂	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	X	*	*
T ₂₃	*	X	X	X	X	X	X	*	X	X	X	*	X	X	*	X	X	X	1
T ₂₄	X	X	X	X	X	X	X	1	X	X	X	X	X	X	X	X	X	*	X
T ₂₆	X	*	*	X	X	X	X	X	X	*	1	X	X	X	X	X	*	X	X
T ₂₇	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	*	1	*
T ₂₉	*	*	X	X	X	X	X	X	X	1	X	*	X	X	X	X	X	X	X
T ₃₀	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	*	*	1	*
T ₃₃	*	X	*	1	X	*	X	X	X	X	*	X	X	X	*	X	X	X	X
T ₃₄	*	*	*	X	X	1	X	X	X	X	X	X	X	X	X	X	X	X	X
T ₃₅	*	X	1	X	X	X	*	X	X	X	X	X	X	X	X	X	*	X	X
T ₃₆	*	X	1	X	X	X	X	X	X	X	X	X	X	X	X	X	*	X	X
T ₃₇	X	X	*	X	X	X	X	X	X	X	X	X	X	X	X	X	1	X	*
T ₃₈	*	X	*	X	X	X	X	X	*	X	X	X	*	X	X	1	X	X	X
T ₃₉	X	X	X	X	X	X	X	X	1	X	X	X	X	X	X	*	X	X	*
T ₄₁	X	X	1	X	*	X	X	X	X	X	X	X	X	X	X	X	*	X	X
T ₄₂	*	X	*	X	X	X	X	X	X	1	X	X	X	X	X	X	*	X	X
T ₄₆	X	X	X	X	X	X	X	*	X	X	X	X	X	X	X	X	X	1	X
T ₄₈	X	*	1	*	X	X	X	X	X	X	X	X	X	X	X	X	X	*	X
T ₅₁	X	X	*	X	X	X	X	X	X	X	X	X	X	X	X	X	1	X	X
T ₅₃	1	X	X	X	X	X	X	X	X	X	X	*	X	X	*	X	X	X	X
T ₅₄	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	X
T ₅₉	*	*	X	X	X	X	X	X	X	1	X	*	X	X	X	X	X	X	X
T ₆₁	X	X	*	X	X	X	X	X	X	X	X	X	X	*	X	1	X	X	X
T ₆₃	*	X	*	*	*	1	*	1	*	X	X	X	X	X	X	*	X	*	X
T ₆₄	*	*	*	*	1	X	X	*	X	X	X	X	X	X	X	X	X	X	X
T ₆₅	1	X	X	X	X	X	X	X	X	X	X	X	*	X	X	X	*	X	X
T ₆₆	*	X	1	X	X	X	X	X	X	X	X	X	X	*	X	X	X	X	X
T ₇₂	*	X	*	X	X	X	X	X	*	X	X	1	X	X	X	X	*	X	X
T ₇₄	X	X	X	X	X	*	X	X	X	X	X	X	1	X	*	X	X	X	X
T ₇₆	X	1	*	X	X	X	X	X	X	X	X	X	X	X	X	X	*	X	X
T ₇₈	X	1	X	X	X	X	X	X	X	X	*	X	X	X	X	X	X	X	X
T ₇₉	1	*	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
T ₈₃	X	X	X	X	*	1	X	X	X	X	*	X	*	X	X	X	X	X	X
T ₈₄	*	1	*	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
T ₈₅	1	X	*	X	X	X	X	X	X	X	X	X	X	X	X	X	*	X	X
T ₈₆	*	X	1	X	*	X	X	X	X	X	X	X	X	X	X	X	*	X	X
T ₈₇	X	X	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	*
T ₈₈	1	X	*	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
T ₉₂	X	X	*	X	X	X	X	X	X	X	*	X	X	X	X	*	1	X	X
T ₉₃	*	X	*	X	*	X	X	1	X	X	X	X	X	X	X	X	X	X	X
T ₉₄	X	X	X	X	X	X	X	X	X	*	1	*	*	X	X	X	*	X	X
T ₉₅	X	X	X	X	*	X	X	1	X	*	X	*	X	X	X	X	X	X	X
T ₉₆	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	X
T ₉₉	1	X	X	X	*	X	X	X	X	X	X	X	X	X	X	X	X	X	X
T ₁₀₀	X	X	1	X	*	X	X	X	X	X	X	X	X	X	X	X	X	X	*

Scénario N°4 :

IA0 : Collecting data from ETMN

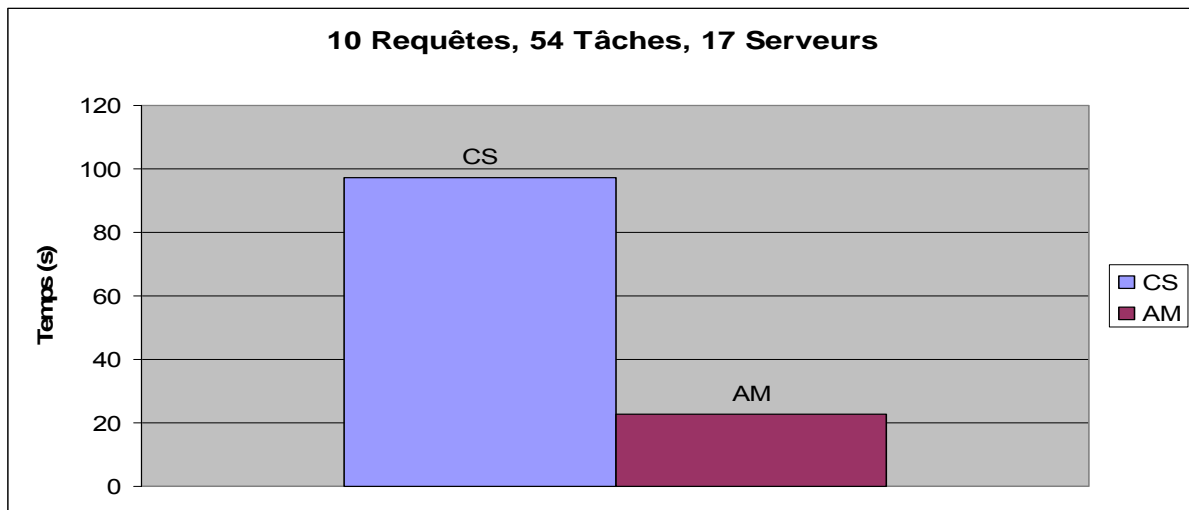
ICA	Service	Data	Source
MA1	T 66	S1T66	S1
MA1	T 6	S3T6	S3
MA1	T 33	S3T33	S3
MA1	T 37	S3T37	S3
MA1	T 38	S3T38	S3
MA1	T 41	S3T41	S3
MA1	T 51	S3T51	S3
MA1	T 72	S3T72	S3
MA1	T 84	S3T84	S3
MA1	T 85	S3T85	S3
MA1	T 86	S3T86	S3
MA1	T 88	S3T88	S3
MA1	T 100	S3T100	S3
MA1	T 61	S15T61	S15
MA1	T 39	S20T39	S20
MA1	T 87	S20T87	S20
MA4	T 48	S4T48	S4
MA4	T 93	S5T93	S5
MA4	T 95	S5T95	S5
MA4	T 99	S5T99	S5
MA4	T 83	S12T83	S12
MA4	T 94	S12T94	S12
MA4	T 65	S14T65	S14
MA4	T 15	S16T15	S16
MA4	T 74	S16T74	S16

Number of ICA : 5
Number of Servers : 17

Request	TotalCost	TotalTime	DueDate
IA9	12	22.56	25
IA8	28	22.56	25
IA6	13	22.56	25
IA4	8	19.46	25
IA1	215	22.56	25
IA10	27	22.56	25
IA2	41	22.56	25
IA5	121	22.56	25
IA7	131	22.56	25
IA3	103	22.56	25

New (nbRequests, nbTasks) = (10 , 54)

Quantité de donnée totale transférée : 16,55 unités



Instance FeTAR générée																			
CH	S ₁	S ₂	S ₃	S ₄	S ₅	S ₇	S ₈	S ₉	S ₁₀	S ₁₁	S ₁₂	S ₁₃	S ₁₄	S ₁₅	S ₁₆	S ₁₇	S ₁₈	S ₁₉	S ₂₀
T ₅	*	X	*	*	*	*	*	*	*	1	*	X	*	X	*	*	*	*	*
T ₆	*	X	1	*	*	*	*	*	*	*	*	X	X	*	*	*	*	*	*
T ₈	*	X	*	*	X	*	*	1	*	X	*	*	*	*	*	*	*	X	*
T ₉	X	X	X	*	X	X	*	*	*	*	*	X	*	X	1	X	X	*	*
T ₁₃	*	X	*	*	*	X	*	*	*	1	*	X	X	*	*	X	*	*	X
T ₁₅	*	X	*	*	*	X	*	*	*	*	*	*	X	*	1	X	*	*	*
T ₁₈	X	*	*	*	X	X	X	1	*	X	X	X	X	X	X	X	X	*	*
T ₁₉	1	X	X	X	*	X	X	X	X	X	X	X	X	X	X	X	X	X	X
T ₂₁	X	X	*	*	X	X	X	X	X	X	X	X	X	X	X	X	1	X	X
T ₂₂	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	*	X	1	*
T ₂₃	*	X	X	X	X	X	X	1	X	X	X	*	X	X	*	X	X	X	*
T ₂₄	X	X	X	X	X	X	X	1	X	X	X	X	X	X	X	X	X	*	X
T ₂₆	X	1	*	X	X	X	X	X	X	*	*	X	X	X	X	X	*	X	X
T ₂₇	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	*	1	*
T ₂₉	*	*	X	X	X	X	X	X	X	1	X	*	X	X	X	X	X	X	X
T ₃₀	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	*	*	1	*
T ₃₃	*	X	1	*	X	*	X	X	X	X	*	X	X	X	*	X	X	X	X
T ₃₄	*	1	*	X	X	*	X	X	X	X	X	X	X	X	X	X	X	X	X
T ₃₅	*	X	*	X	X	X	1	X	X	X	X	X	X	X	X	X	*	X	X
T ₃₆	1	X	*	X	X	X	X	X	X	X	X	X	X	X	X	X	*	X	X
T ₃₇	X	X	1	X	X	X	X	X	X	X	X	X	X	X	X	X	*	X	*
T ₃₈	*	X	1	X	X	X	X	X	*	X	X	X	*	X	X	*	X	X	X
T ₃₉	X	X	X	X	X	X	X	X	*	X	X	X	X	X	X	*	X	X	1
T ₄₁	X	X	1	X	*	X	X	X	X	X	X	X	X	X	X	X	*	X	X
T ₄₂	*	X	*	X	X	X	X	X	X	*	X	X	X	X	X	X	1	X	X
T ₄₆	X	X	X	X	X	X	X	*	X	X	X	X	X	X	X	X	X	1	X
T ₄₈	X	*	*	1	X	X	X	X	X	X	X	X	X	X	X	X	X	*	X
T ₅₁	X	X	1	X	X	X	X	X	X	X	X	X	X	X	X	X	*	X	X
T ₅₃	1	X	X	X	X	X	X	X	X	X	X	*	X	X	*	X	X	X	X
T ₅₄	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	X
T ₅₉	1	*	X	X	X	X	X	X	X	*	X	*	X	X	X	X	X	X	X
T ₆₁	X	X	*	X	X	X	X	X	X	X	X	X	X	1	X	*	X	X	X
T ₆₃	*	X	*	*	*	1	*	*	X	X	X	X	X	X	*	X	*	X	X
T ₆₄	*	1	*	*	*	X	X	*	X	X	X	X	X	X	X	X	X	X	X
T ₆₅	*	X	X	X	X	X	X	X	X	X	X	X	1	X	X	X	*	X	X
T ₆₆	1	X	*	X	X	X	X	X	X	X	X	X	X	*	X	X	X	X	X
T ₇₂	*	X	1	X	X	X	X	X	*	X	X	*	X	X	X	X	*	X	X
T ₇₄	X	X	X	X	X	*	X	X	X	X	X	X	*	X	1	X	X	X	X
T ₇₆	X	*	*	X	X	X	X	X	X	X	X	X	X	X	X	X	1	X	X
T ₇₈	X	1	X	X	X	X	X	X	X	X	*	X	X	X	X	X	X	X	X
T ₇₉	*	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
T ₈₃	X	X	X	X	*	*	X	X	X	X	1	X	*	X	X	X	X	X	X
T ₈₄	*	*	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
T ₈₅	*	X	1	X	X	X	X	X	X	X	X	X	X	X	X	X	*	X	X
T ₈₆	*	X	1	X	*	X	X	X	X	X	X	X	X	X	X	X	*	X	X
T ₈₇	X	X	*	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1
T ₈₈	*	X	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
T ₉₂	X	X	*	X	X	X	X	X	X	X	X	*	X	X	X	*	1	X	X
T ₉₃	*	X	*	X	1	X	X	*	X	X	X	X	X	X	X	X	X	X	X
T ₉₄	X	X	X	X	X	X	X	X	X	*	1	*	*	X	X	X	*	X	X
T ₉₅	X	X	X	X	1	X	X	*	X	*	X	*	X	X	X	X	X	X	X
T ₉₆	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	X
T ₉₉	*	X	X	X	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X
T ₁₀₀	X	X	1	X	*	X	X	X	X	X	X	X	X	X	X	X	X	X	*

Scénario N°5 :

IA0 : Collecting data from ETMN

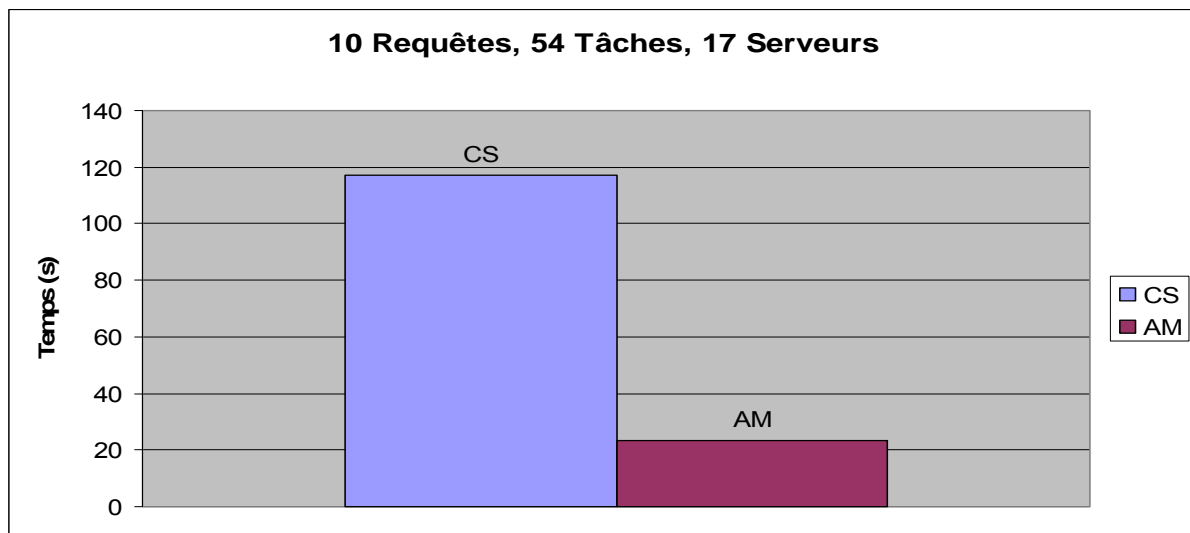
ICA	Service	Data	Source
MA3	T 26	S2T26	S2
MA3	T 34	S2T34	S2
MA3	T 76	S2T76	S2
MA3	T 78	S2T78	S2
MA3	T 79	S2T79	S2
MA3	T 84	S2T84	S2
MA3	T 46	S19T46	S19
MA3	T 54	S19T54	S19
MA3	T 96	S19T96	S19
MA5	T 35	S8T35	S8
MA5	T 6	S9T6	S9
MA5	T 18	S9T18	S9
MA5	T 24	S9T24	S9
MA5	T 9	S11T9	S11
MA5	T 29	S11T29	S11
MA5	T 42	S11T42	S11
MA5	T 94	S11T94	S11
MA5	T 95	S11T95	S11
MA2	T 15	S10T15	S10
MA2	T 22	S17T22	S17
MA2	T 27	S18T27	S18
MA2	T 41	S18T41	S18
MA2	T 51	S18T51	S18
MA2	T 72	S18T72	S18
MA2	T 92	S18T92	S18

Number of ICA : 5
Number of Servers : 17

Request	TotalCost	TotalTime	DueDate
IA8	46	23.56	25
IA9	7	20.76	25
IA1	222	23.56	25
IA4	8	17.86	25
IA5	114	23.56	25
IA2	41	23.56	25
IA6	10	19.96	25
IA10	27	23.56	25
IA7	146	23.56	25
IA3	93	23.56	25

New (nbRequests, nbTasks) = (10 , 54)

Quantité de donnée totale transférée : 16,25 unités



Instance FeTAR générée																			
CH	S ₁	S ₂	S ₃	S ₄	S ₅	S ₇	S ₈	S ₉	S ₁₀	S ₁₁	S ₁₂	S ₁₃	S ₁₄	S ₁₅	S ₁₆	S ₁₇	S ₁₈	S ₁₉	S ₂₀
T ₅	*	X	*	1	*	*	*	*	*	*	*	X	*	X	*	*	*	*	*
T ₆	*	X	*	*	*	*	*	1	*	*	*	X	X	*	*	*	*	*	*
T ₈	*	X	*	*	X	*	*	*	*	X	1	*	*	*	*	*	*	X	*
T ₉	X	X	X	*	X	X	*	*	*	1	*	*	X	*	X	*	X	X	*
T ₁₃	1	X	*	*	*	X	*	*	*	*	*	X	X	*	*	X	*	*	X
T ₁₅	*	X	*	*	*	X	*	*	1	*	*	*	X	*	*	X	*	*	*
T ₁₈	X	*	*	*	X	X	X	1	*	X	X	X	X	X	X	X	X	*	*
T ₁₉	1	X	X	X	*	X	X	X	X	X	X	X	X	X	X	X	X	X	X
T ₂₁	X	X	*	1	X	X	X	X	X	X	X	X	X	X	X	X	*	X	X
T ₂₂	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	X	*	*
T ₂₃	1	X	X	X	X	X	X	*	X	X	X	*	X	X	*	X	X	X	*
T ₂₄	X	X	X	X	X	X	X	1	X	X	X	X	X	X	X	X	X	*	X
T ₂₆	X	1	*	X	X	X	X	X	X	*	*	X	X	X	X	X	*	X	X
T ₂₇	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	*	*
T ₂₉	*	*	X	X	X	X	X	X	X	1	X	*	X	X	X	X	X	X	X
T ₃₀	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	*	*	*	1
T ₃₃	*	X	*	1	X	*	X	X	X	X	*	X	X	X	*	X	X	X	X
T ₃₄	*	1	*	X	X	*	X	X	X	X	X	X	X	X	X	X	X	X	X
T ₃₅	*	X	*	X	X	X	1	X	X	X	X	X	X	X	X	X	*	X	X
T ₃₆	*	X	1	X	X	X	X	X	X	X	X	X	X	X	X	X	*	X	X
T ₃₇	X	X	*	X	X	X	X	X	X	X	X	X	X	X	X	X	*	X	1
T ₃₈	1	X	*	X	X	X	X	X	*	X	X	X	*	X	X	*	X	X	X
T ₃₉	X	X	X	X	X	X	X	X	*	X	X	X	X	X	X	*	X	X	1
T ₄₁	X	X	*	X	*	X	X	X	X	X	X	X	X	X	X	X	1	X	X
T ₄₂	*	X	*	X	X	X	X	X	X	1	X	X	X	X	X	X	*	X	X
T ₄₆	X	X	X	X	X	X	X	*	X	X	X	X	X	X	X	X	X	1	X
T ₄₈	X	*	1	*	X	X	X	X	X	X	X	X	X	X	X	X	X	*	X
T ₅₁	X	X	*	X	X	X	X	X	X	X	X	X	X	X	X	X	1	X	X
T ₅₃	*	X	X	X	X	X	X	X	X	X	X	*	X	X	1	X	X	X	X
T ₅₄	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	X
T ₅₉	*	*	X	X	X	X	X	X	X	1	X	*	X	X	X	X	X	X	X
T ₆₁	X	X	*	X	X	X	X	X	X	X	X	X	X	1	X	*	X	X	X
T ₆₃	*	X	*	1	*	*	*	*	X	X	X	X	X	X	*	X	*	X	X
T ₆₄	1	*	*	*	*	X	X	*	X	X	X	X	X	X	X	X	X	X	X
T ₆₅	*	X	X	X	X	X	X	X	X	X	X	X	1	X	X	X	*	X	X
T ₆₆	*	X	*	X	X	X	X	X	X	X	X	X	X	1	X	X	X	X	X
T ₇₂	*	X	*	X	X	X	X	X	*	X	X	*	X	X	X	X	1	X	X
T ₇₄	X	X	X	X	X	*	X	X	X	X	X	X	1	X	*	X	X	X	X
T ₇₆	X	1	*	X	X	X	X	X	X	X	X	X	X	X	X	X	*	X	X
T ₇₈	X	1	X	X	X	X	X	X	X	X	*	X	X	X	X	X	X	X	X
T ₇₉	*	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
T ₈₃	X	X	X	X	*	*	X	X	X	X	1	X	*	X	X	X	X	X	X
T ₈₄	*	1	*	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
T ₈₅	1	X	*	X	X	X	X	X	X	X	X	X	X	X	X	X	*	X	X
T ₈₆	*	X	1	X	*	X	X	X	X	X	X	X	X	X	X	X	*	X	X
T ₈₇	X	X	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	*
T ₈₈	*	X	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
T ₉₂	X	X	*	X	X	X	X	X	X	X	X	*	X	X	X	*	1	X	X
T ₉₃	1	X	*	X	*	X	X	*	X	X	X	X	X	X	X	X	X	X	X
T ₉₄	X	X	X	X	X	X	X	X	X	1	*	*	*	X	X	X	*	X	X
T ₉₅	X	X	X	X	*	X	X	*	X	1	X	*	X	X	X	X	X	X	X
T ₉₆	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	X
T ₉₉	1	X	X	X	*	X	X	X	X	X	X	X	X	X	X	X	X	X	X
T ₁₀₀	X	X	*	X	1	X	X	X	X	X	X	X	X	X	X	X	X	X	*

Annexe C

Scénario N°1 :

IA0 : Collecting data from ETMN

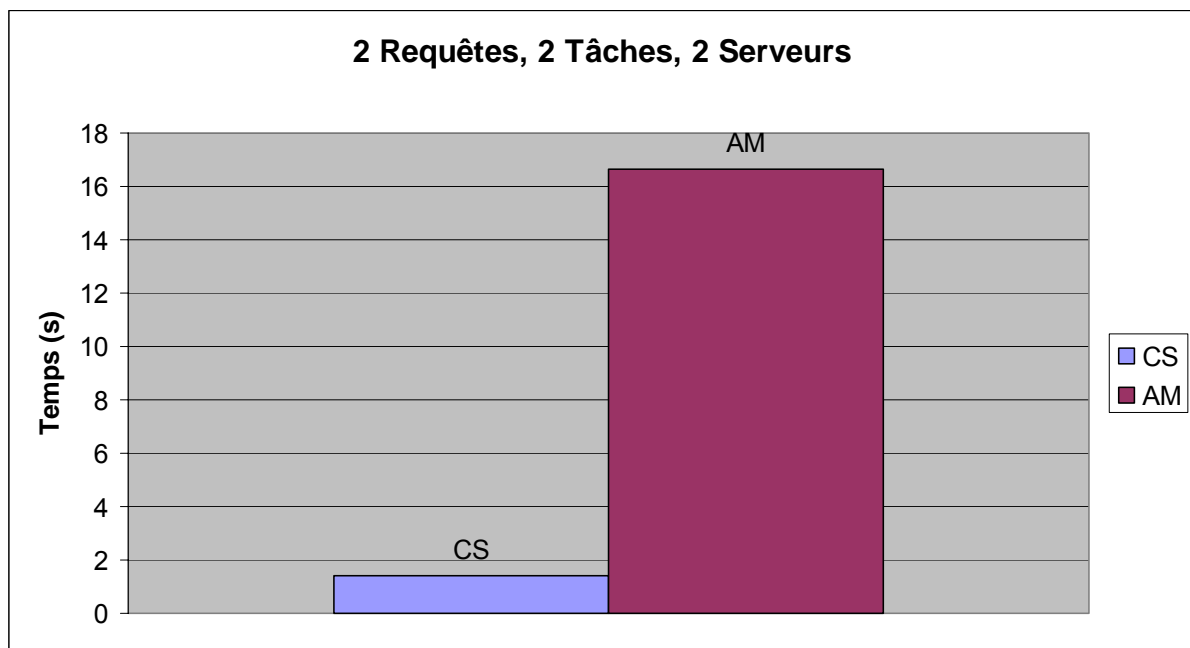
ICA	Service	Data	Source
MA3	T 15	S13T15	S13
MA5	T 3	S9T3	S9

Number of ICA : 2
Number of Servers : 2

Request	TotalCost	TotalTime	DueDate
IA8	1	16.66	25
IA3	3	16.66	25

New (nbRequests, nbTasks) = (2 , 2)

Quantité de donnée totale transférée : 0,75 unités



Instance FeTAR générée																			
CH	S ₁	S ₃	S ₄	S ₇	S ₈	S ₉	S ₁₀	S ₁₂	S ₁₃	S ₁₄	S ₁₅	S ₁₆	S ₁₇	S ₁₈	S ₁₉	S ₂₀	S ₁		
T ₃	*	*	*	*	*	*	1	*	*	*	X	*	*	*	*	*	*		
T ₁₅	*	*	*	*	X	*	*	*	*	*	1	X	*	*	*	*	*		

Scénario N°2 :

IA0 : Collecting data from ETMN
-
□
✕

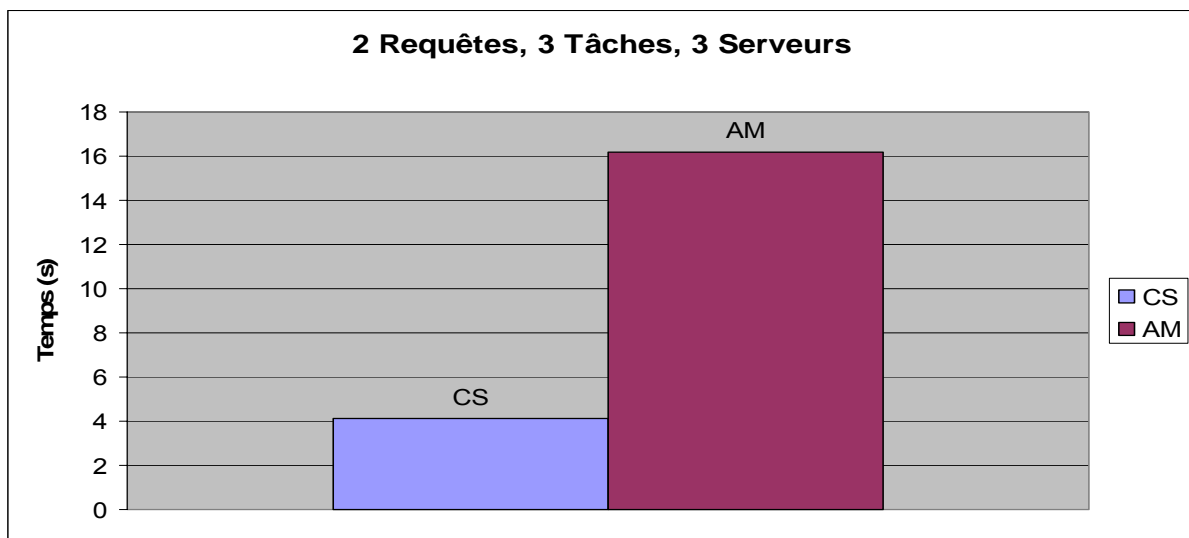
ICA	Service	Data	Source
MA5	T 2	S8T2	S8
MA1	T 97	S1T97	S1
MA4	T 25	S16T25	S16

Number of ICA : 3
 Number of Servers : 3

Request	TotalCost	TotalTime	DueDate
IA2	1	15.96	25
IA9	9	16.16	25

(nbRequests, nbTasks) = (2 , 3)

Quantité de donnée totale transférée : 1,05 unités



Instance FeTAR générée																
CH	S ₁	S ₃	S ₄	S ₇	S ₈	S ₉	S ₁₀	S ₁₂	S ₁₃	S ₁₄	S ₁₅	S ₁₆	S ₁₇	S ₁₈	S ₁₉	S ₂₀
T ₂	*	*	*	*	1	*	*	*	*	*	*	X	*	*	*	*
T ₂₅	X	X	*	X	X	X	X	X	X	X	X	1	*	X	X	*
T ₉₇	1	X	X	X	X	X	X	X	X	X	X	X	X	X	*	X

Scénario N°3 :

IA0 : Collecting data from ETMN
- □ ×

ICA	Service	Data	Source
MA3	T 76	S2T76	S2
MA5	T 2	S9T2	S9
MA1	T 97	S1T97	S1
MA2	T 25	S17T25	S17
MA4	T 6	S16T6	S16

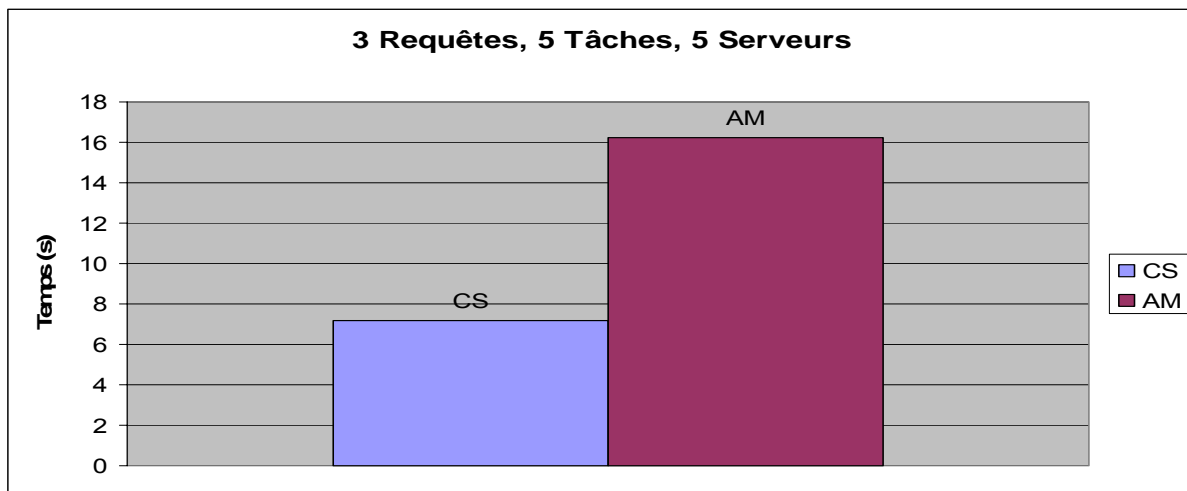
Number of ICA : 5

Number of Servers : 5

Request	TotalCost	TotalTime	DueDate
IA2	1	15.96	25
IA1	17	16.06	25
IA9	14	16.26	25

(nbRequests, nbTasks) = (3 , 5)

Quantité de donnée totale transférée : 1,5 unités



Instance FeTAR générée																				
CH	S ₁	S ₂	S ₃	S ₄	S ₅	S ₇	S ₈	S ₉	S ₁₀	S ₁₁	S ₁₂	S ₁₃	S ₁₄	S ₁₅	S ₁₆	S ₁₇	S ₁₈	S ₁₉	S ₂₀	
T ₂	*	X	*	*	X	*	*	1	*	X	*	*	*	*	X	*	*	*	*	*
T ₆	*	X	*	*	*	*	*	*	*	*	X	X	*	1	*	*	*	*	*	
T ₂₅	X	X	X	*	X	X	X	X	X	X	X	X	X	*	1	X	X	X	*	
T ₇₆	X	1	*	X	X	X	X	X	X	X	X	X	X	X	X	X	*	X	X	
T ₉₇	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	*	X	

Scénario N°4 :

IAO : Collecting data from ETMN
_ □ ✕

ICA	Service	Data	Source
MA2	T 11	S17T11	S17
MA5	T 24	S9T24	S9
MA4	T 3	S4T3	S4
MA4	T 100	S5T100	S5
MA4	T 26	S12T26	S12
MA1	T 23	S1T23	S1
MA1	T 50	S15T15	S15
MA1	T 80	S20T80	S20
MA1	T 87	S20T87	S20

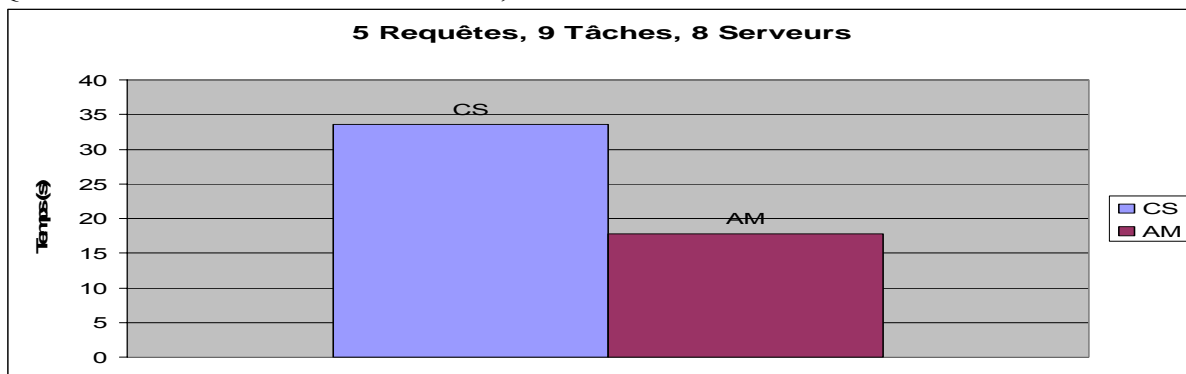
Number of ICA : 4

Number of Servers : 8

Request	TotalCost	TotalTime	DueDate
IA10	19	17.76	25
IA6	2	16.16	25
IA3	3	16.76	25
IA7	13	17.76	25
IA8	5	17.76	25

(nbRequests, nbTasks) = (5 , 9)

Quantité de donnée totale transférée : 3,3 unités



Instance FeTAR générée																				
CH	S ₁	S ₂	S ₃	S ₄	S ₅	S ₇	S ₈	S ₉	S ₁₀	S ₁₁	S ₁₂	S ₁₃	S ₁₄	S ₁₅	S ₁₆	S ₁₇	S ₁₈	S ₁₉	S ₂₀	
T ₃	*	X	*	1	*	*	*	*	*	*	*	X	*	*	*	X	*	*	*	
T ₁₁	X	X	*	*	*	*	*	*	*	X	*	*	*	*	*	1	*	*	*	
T ₂₃	1	X	X	X	X	X	X	*	X	X	X	*	X	X	*	X	X	X	*	
T ₂₄	X	X	X	X	X	X	X	1	X	X	X	X	X	X	X	X	X	*	X	
T ₂₆	X	*	*	X	X	X	X	X	X	*	1	X	X	X	X	X	*	X	X	
T ₅₀	X	X	*	X	X	X	X	X	X	X	X	X	X	1	X	X	X	X	X	
T ₈₀	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	*	X	X	1	
T ₈₇	X	X	*	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	
T ₁₀₀	X	X	*	X	1	X	X	X	X	X	X	X	X	X	X	X	X	X	*	

Scénario N°5

IAO : Collecting data from ETMN

-
□
✕

ICA	Service	Data	Source
MA2	T 51	S18T51	S18
MA2	T 65	S18T65	S18
MA5	T 45	S8T45	S8
MA5	T 62	S8T62	S8
MA4	T 99	S5T99	S5
MA4	T 83	S12T83	S12
MA4	T 94	S12T94	S12
MA1	T 75	S1T75	S1
MA1	T 34	S3T34	S3
MA1	T 85	S3T85	S3
MA1	T 86	S3T86	S3
MA1	T 92	S3T92	S3
MA1	T 50	S15T15	S15
MA3	T 79	S2T79	S2
MA3	T 91	S6T91	S6
MA3	T 29	S13T29	S13
MA3	T 67	S12T67	S13
MA3	T 96	S19T96	S19

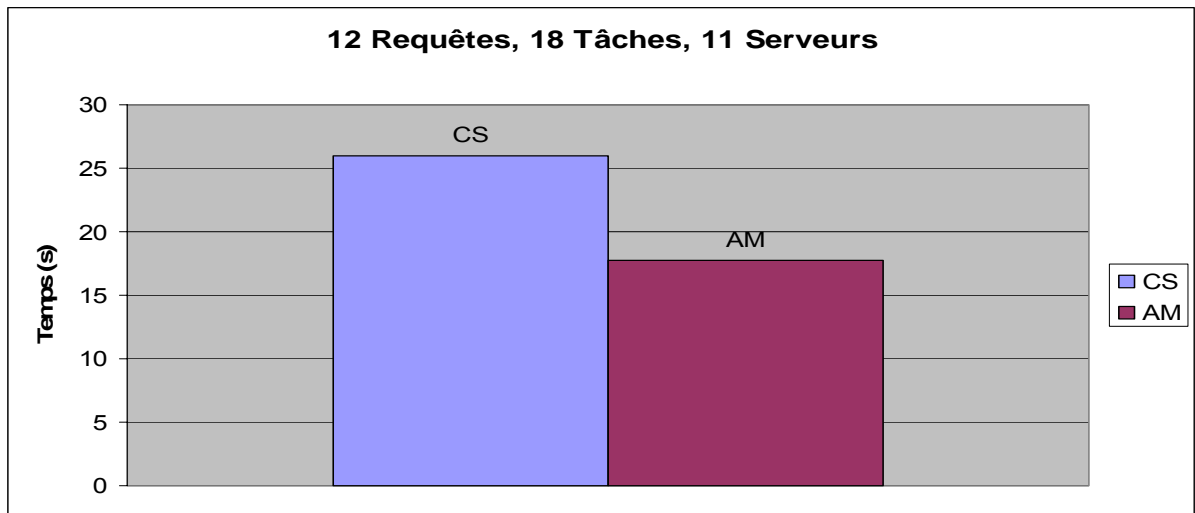
Number of ICA : 5

Number of Servers : 11

Request	TotalCost	TotalTime	DueDate
IA8	27	17.46	25
IA4	6	16.66	25
IA12	10	17.76	25
IA1	10	17.76	25
IA10	13	17.46	25
IA7	2	16.16	25
IA2	5	17.76	25
IA5	5	17.76	25
IA3	27	17.76	25
IA9	14	17.46	25
IA6	18	17.76	25
IA11	19	17.76	25

(nbRequests, nbTasks) = (12 , 18)

Quantité de donnée totale transférée : 5,2 unités



Instance FeTAR générée

CH	S ₁	S ₂	S ₃	S ₅	S ₆	S ₇	S ₈	S ₁₀	S ₁₁	S ₁₂	S ₁₃	S ₁₄	S ₁₅	S ₁₇	S ₁₈	S ₁₉
T ₂₉	*	*	X	X	X	X	X	X	*	X	1	X	X	X	X	X
T ₃₄	*	*	1	X	X	*	X	X	X	X	X	X	X	X	X	X
T ₄₅	*	X	*	X	X	X	1	X	X	X	X	X	X	X	*	X
T ₅₀	X	X	*	X	X	X	X	X	X	X	X	X	X	1	X	X
T ₅₁	X	X	*	X	X	X	X	X	X	X	X	X	X	X	1	X
T ₆₂	X	X	X	X	X	X	1	X	X	X	X	*	X	X	*	X
T ₆₅	*	X	X	X	X	X	X	X	X	X	X	*	X	X	1	X
T ₆₇	X	X	*	X	X	X	X	X	X	X	1	X	X	X	X	*
T ₇₅	1	X	*	X	X	X	X	*	X	*	X	X	X	X	X	X
T ₇₉	*	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X
T ₈₃	X	X	X	*	X	*	X	X	X	1	X	*	X	X	X	X
T ₈₅	*	X	1	X	X	X	X	X	X	X	X	X	X	X	*	X
T ₈₆	*	X	1	*	X	X	X	X	X	X	X	X	X	X	*	X
T ₉₁	X	X	X	X	1	X	X	X	X	X	X	X	X	X	*	*
T ₉₂	X	X	1	X	X	X	X	X	X	X	*	X	X	*	*	X
T ₉₄	X	X	X	X	X	X	X	X	*	1	*	*	X	X	*	X
T ₉₆	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1
T ₉₉	*	X	X	1	X	X	X	X	X	X	X	X	X	X	X	X