



**HAL**  
open science

# Modélisation et Optimisation d'un Système de Transport à la Demande Multicritère et Dynamique

Issam Zidi

► **To cite this version:**

Issam Zidi. Modélisation et Optimisation d'un Système de Transport à la Demande Multicritère et Dynamique. Autre. Ecole Centrale de Lille; École Nationale des Sciences de l'Informatique (La Manouba, Tunisie), 2012. Français. NNT: 2012ECLI0012 . tel-00735879

**HAL Id: tel-00735879**

**<https://theses.hal.science/tel-00735879v1>**

Submitted on 27 Sep 2012

**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 : 192

ECOLE CENTRALE DE LILLE

## THESE

Présentée en vue  
d'obtenir le grade de

## DOCTEUR

En

**Spécialité : Automatique, Génie Informatique, Traitement du Signal et Images**

Par

**Issam ZIDI**

**DOCTORAT DELIVRE CONJOINTEMENT PAR L'ÉCOLE CENTRALE DE LILLE  
ET L'ÉCOLE NATIONALE DES SCIENCES DE L'INFORMATIQUE DE TUNISIE**

Titre de la thèse :  
Modélisation et Optimisation du Problème de Transport à la Demande  
Multicritère et Dynamique

Soutenue le 6 juillet 2012 devant le jury d'examen :

<b>Président</b>	Pierre BORNE, Professeur, Ecole Centrale de Lille, France
<b>Rapporteur</b>	Abdellah EL MOUDNI, Professeur, Université de Belfort-Montbéliard,
<b>Rapporteur</b>	Hanen BOUCHRIHA, Professeur, Ecole nationale d'ingénieurs de Tunis
<b>Membre</b>	Mekki KSOURI Professeur, Ecole nationale d'ingénieurs de Tunis
<b>Membre</b>	Mohammed CHADLI, Maître de Conférences, HDR, Université de Picardie Jules Vernes, Amiens
<b>Membre</b>	Kamel ZIDI, Docteur, Maître assistant, Faculté des sciences de Gafsa,
<b>Directeurs de thèse</b>	Khaled GHEDIRA Professeur, Institut Supérieur de gestion (ISG) Khaled MESGHOUNI, Maître de conférences, HDR, Ecole Centrale de Lille,

Thèse préparée dans les Laboratoires LAGIS et SOIE

Ecole Doctorale SPI 072 (Lille I, Lille III, Artois, ULCO, UVHC, EC Lille)  
**PRES Université Lille Nord-de-France**



*À la mémoire de ma mère,*

*À mon père*

*À mon frère et sœurs,*

*À toute ma grande famille*

## Remerciements

Les premières personnes qui me viennent à l'esprit et que je tiens à les remercier profondément sont les directeurs de cette thèse : le Dr. Khaled MESGHOUNI Maître de conférences, HDR et le Professeur Khaled GHEDIRA, pour l'énorme soutien scientifique et moral qu'ils m'ont apporté pendant ces années au sein du LAGIS et SOIE. Je les remercie aussi pour leurs qualités humaines et scientifiques et de m'avoir toujours encouragé à aller de l'avant.

Je tiens également à exprimer ma profonde gratitude au Dr. Kamel ZIDI, maître Assistant à la Faculté des Sciences de Gafsa et co-directeur de cette thèse, pour sa rigueur scientifique et son aide précieuse.

Il m'est très agréable aussi d'exprimer toute ma reconnaissance au Professeur Abdella EL MOUDNI, Professeur à l'université de Belfort-Montbéliard et au Professeur Hanen BOUCHRIHA, Professeur à Ecole nationale d'ingénieurs de Tunis de m'avoir fait l'honneur en s'intéressant à mon travail de thèse et en acceptant d'être les rapporteurs malgré leurs nombreuses occupations.

Je tiens également à remercier le

- Professeur Pierre BORNE, Professeur à Ecole Centrale de Lille, France
- Professeur Mekki KSOURI, Professeur à l'Ecole nationale d'ingénieurs de Tunis
- Dr. Mohammed CHADLI, Maître de Conférences, HDR à l'Université de Picardie Jules Verne.

D'avoir accepté d'examiner mon travail de thèse et de participer à ce jury.

Ces remerciements ne seraient pas complets si je n'y associais pas toutes les personnes qui ont contribué de près ou de loin à la réalisation de cette thèse, en particulier, tout le personnel des deux laboratoires dans lesquels j'ai réalisé mon travail de thèse ; le LAGIS et SOIE pour leur bonne humeur et leur disponibilité.



# Table des matières

Table des matières.....	5
Liste des figures .....	9
Liste des tableaux.....	10
Chapitre I : Introduction .....	12
I.1 Introduction : .....	12
I.2 Motivations :.....	12
I.3 Objectif et Problématique :.....	14
I.4 Positionnement Scientifique :.....	15
1. Recherche Opérationnelle : .....	15
2. Les Systèmes Multi-agent : .....	16
3. Personnalisation de l'information : .....	16
I.5 Plan de la thèse : .....	17
Chapitre II : Etat de L'art sur Le Problème de Transport à la Demande .....	20
II.1 Introduction :.....	20
II.2 Description du Problème de Transport à la Demande (PTD): .....	20
1. Contexte de l'étude : .....	20
2. Définition du PTD :.....	21
II.3. Travaux industrielles sur le Transport à la Demande (TAD):.....	23
1. Etat de l'art sur les TAD en France : .....	24
2. Etat de l'art sur le Transport à la Demande (TAD) à l'étranger :.....	25
II.4. Travaux de recherche sur le Transport à la Demande (TAD): .....	27
1. Les modèles mathématiques pour le Problème de Transport à la Demande (PTD) .....	28
2. Le Problème de Transport à la Demande (PTD) avec un seul véhicule .....	32
3. Le Problème de Transport à la Demande (PTD) avec plusieurs véhicules.....	34

II.5 Conclusion :	41
<b>Chapitre III :_Les Problèmes d’Optimisation Multi-Objectif et les Approches de Résolution</b>	<b>42</b>
III.1 Introduction :	42
III.2 Définitions et Vocabulaires :	42
III.3. Approches de résolution des problèmes d’optimisation multi-objectif :	44
1. Les approches de transformations des objectifs du POM à un seul objectif	45
2. Approches non-Pareto	48
3. Approches Pareto	49
III.3 Classification des méthodes d’optimisation multi-objectif :	53
III.4 Les métaheuristiques de l’optimisation multi-objectif :	53
1. Les méthodes de recherche locale	53
2. Le recuit simulé	54
3. La recherche Tabou	55
4. Les algorithmes évolutionnaires	56
III.5 Conclusion :	59
<b>Chapitre IV : Une Approche Centralisée de Résolution basée sur l’algorithme de Recuit Simulé pour le Problème de Transport à la Demande</b>	<b>60</b>
IV.1. Introduction :	60
IV.2. Formalisme mathématique du PTD :	60
IV.3. Une approche agrégative basée sur l’algorithme de recuit simulé pour la résolution du PTD statique :	64
1. L’algorithme de recuit simulé :	64
2. Résultats Numériques	69
IV.4. Une approche Pareto basée sur l’algorithme de Recuit Simulé Multi-Objectif (RSMO) pour la résolution du PTD statique :	72
1. L’algorithme de recuit simulé multi-objectif:	72
2. Résultats Numériques	75

IV.5. Une approche Pareto basée sur l’algorithme de recuit simulé multi-objectif pour la résolution du PTD dynamique : .....	81
1. Le PTD dynamique.....	81
IV.6. Conclusion : .....	86
<b>Chapitre V : Une approche distribuée basée sur l’algorithme de Recuit Simulé pour la Résolution du Problème de Transport à la Demande .....</b>	<b>88</b>
V. 1. Introduction : .....	88
V. 2. Les Systèmes Multi-agent (SMA) : .....	88
1. Caractéristiques des Agents.....	89
2. Caractéristiques des Systèmes multi-Agent.....	90
3. Les SMA et leur application dans le domaine de transport .....	92
V. 3. Approche distribuée basée sur le recuit simulé multi-objectif pour la résolution du PTD : .....	95
1. L’architecture Multi-agent .....	95
2. Le processus d’optimisation.....	97
V. 4. Applications et Résultats Numériques : .....	99
1. L’outil informatique utilisé.....	101
2. Le Système Multi-Agent.....	102
3. Application de l’approche sur un PTD statique.....	104
4. Application de l’approche sur un PTD dynamique .....	107
V.5 Conclusion : .....	116
<b>Chapitre VI :_Conclusion et Perspectives .....</b>	<b>118</b>
VI. 1. Conclusion : .....	118
VI. 2. Perspectives : .....	120
<b>Les Références Bibliographiques .....</b>	<b>122</b>

## Liste des figures

Figure I. 1 : Positionnement du TAD par rapport aux autres modes de transport .....	13
Figure II. 1 : Présentation schématique du problème.....	22
Figure II.2 : Affectation des véhicules à des demandes de transport dans la matrice .....	23
Figure II.3 : Une solution au PTD présentée par la matrice <i>sol</i> .....	23
Figure III.1 : Relation entre le type de véhicule et prix. ....	43
Figure III.2 : Sélection parallèle dans l’algorithme VEGA.....	49
Figure III.2 : Pseudo-code de la méthode de descente. ....	54
Figure IV.1 : Heuristique de distribution .....	65
Figure IV.2 : Exemple de composition de la solution initiale .....	66
Figure IV.3 : La structure de voisinage.....	68
Figure IV.4 : Comparaison de nos résultats avec ceux obtenus par (Mauri et al, 2006).....	71
Figure IV.5 : Pseudo-code de l’algorithme de Recuit Simulé Multi-Objectif (RSMO) .....	73
Figure IV.6 (a) : Itinéraire du véhicule numéro 1 .....	77
Figure IV.6 (b) : Itinéraire du véhicule numéro 2.....	78
Figure IV.6 (c) : Itinéraire du véhicule numéro 3 .....	78
Figure IV.6 (d) : Itinéraire du véhicule numéro 4.....	79
Figure IV.7 : Comparaison sur la durée de la route (RSMO, AG et RT).....	80
Figure IV.8 : Comparaison sur le temps mis pour le voyage (RSMO, AG et RT).....	81
Figure IV.9: Le problème de transport à la demande dynamique.....	83
Figure IV.10 : Fonctionnement de l’algorithme de RSMO pour la résolution du PTD dynamique. ....	84
Figure V.1 : L’architecture globale du modèle multi-agents.....	96
Figure V.2 : Le scénario de traitement d’une demande de transport .....	96
Figure V.3 : L’architecture comportementale de l’agent véhicule .....	98
Figure V.4 : Structure de voisinage .....	99
Figure V.9 : comportement de l’agent centre d’appels .....	100
Figure V.5 : comportement de l’agent centre d’appels .....	100

Figure V.6 : le conteneur des agents dans la plate-forme JADE .....	103
Figure V.7 : La communication des agents via l'envoi de messages.....	104
Figure V.8 : le PTD dynamique.....	108

## Liste des tableaux

Figure III.3 : Architecture générale d'un algorithme évolutionnaire. ....	59
Tableau IV.1: Paramètres de l'algorithme du recuit simulé.....	70
Tableau IV.2: Les résultats obtenus par notre approche (Zidi et al, 2010 ) <sup>a</sup> .....	70
Tableau IV.3: Les résultats obtenus par (Mauri et al, 2006) .....	70
Tableau IV.4: les résultats obtenu par l'algorithme de RSMO .....	76
Tableau IV.5 : les résultats obtenus par l'algorithme génétique (Claudio et al, 2009).....	76
Tableau IV.6 : Les résultats obtenus par l'algorithme de recherche tabou (Cordeau & Laporte ,2003) .....	76
Tableau IV.7 : Simulation de 4 h avec 10 véhicules et 20 demandes par heure.....	85
Tableau V.1: Les résultats obtenus par l'approche multi-agents basé sur le RSMO (Zidi et al, 2011) [101].....	105
Tableau V.2: Les résultats obtenus par l'Algorithme Génétique (AG) (Claudio et al, 2009) .....	106
Tableau V.3: Les résultats obtenus par l'algorithme de Recherche Tabou (RT) (Cordeau & Laporte, 2003) .....	106
Tableau V.4: simulation de 4 heures avec 10 véhicules et 20 demandes par heure.....	111
Tableau V.5: simulation de 7 heures avec 20 véhicules et 20 demandes par heure.....	113
Tableau V.6 : Simulation de 4 heures avec 10 véhicules et 35 demandes par heure.....	115
Tableau V.7 : Temps de calcul en seconde.....	116



# Chapitre I :

## Introduction

### I.1 Introduction :

Lors de ces dernières années, l'offre de transport public n'a cessé de se développer et de s'améliorer, et avec elle les demandes et les exigences des usages, rendant ces moyens de transport moins compétitifs surtout dans les milieux ruraux et périurbains. L'utilisation de la voiture individuel reste assez répandue dans ces zones de part sa disponibilité et de son efficacité malgré les quelques inconvénients qu'elle possède en l'occurrence le coût d'utilisation (prix de l'essence) et les problèmes environnementaux qu'elle cause.

L'une des solutions qui peut résoudre ce problème et permet de profiter des avantages des deux modes de transport (en commun et individuel) et de réduire leurs inconvénients est le partage des voitures privées. Le service de partage des voitures privées est appelé service de **Transport À la Demande (TAD)** ou en anglais **Dial a Ride Problem (DRP)**. Dans ce mode de transport, les clients (voyageurs) envoient une demande de transport à un opérateur comprenant le lieu de ramassage (origine), la destination finale, le nombre de personnes à transporter et l'heure de passage désiré. Le problème de TAD consiste à élaborer des tournées de véhicules pour répondre aux demandes de transport en minimisant les coûts engendrés par ces tournées (distance parcourue, temps de service, nombre de véhicules utilisés) et en fournissant une bonne qualité de service aux voyageurs (temps de voyage, nombre de stations visitées, etc..).

### I.2 Motivations :

Il existe plusieurs motivations qui nous font proposer les systèmes de **Transport À la Demande (TAD)** comme une solution pour les problèmes de transport de nos jours. Dans ce qui suit nous présentons les motivations les plus intéressantes.

Aujourd'hui, nous constatons une évolution dans la structure des déplacements. Cette évolution est due à l'étalement urbain et à l'apparition des pôles attractifs aux périphéries des villes. Les flux de transports actuels sont caractérisés par des grandes distances couvrant les trajets banlieue-banlieue. C'est pour ces raisons que le mode de transport privilégié pour ce type de déplacements est la voiture privée. En effet, l'augmentation considérable du nombre de citoyens, en particulier dans les grandes villes, la croissance de l'usage des voitures pose des problèmes aigus, tant sociaux qu'économiques et environnementaux. Dans ce contexte, l'objectif des nouvelles politiques gouvernementales est de diminuer l'utilisation des voitures privées afin de lutter contre les embouteillages dans les villes et de diminuer les rejets polluants. C'est surtout au sein des villes que ces questions, vraisemblablement peu compatibles avec les objectifs d'un développement durable, sont les plus sensibles. En effet, le TAD présente une solution très intéressante pour les personnes âgées, les personnes à mobilité réduite et pour les aspects environnementaux.

Les individus cherchent des moyens de transport plus souples, flexibles et proches de leurs attentes. Le TAD permet de répondre à ces attentes. En effet, il est considéré comme un mode de transport collectif, individualisé et activé à la demande. Le TAD offre un meilleur rapport qualité/coût de service. Sachant que le coût de fonctionnement est lié à la qualité de service offerte, les concepteurs d'un tel système doivent trouver un compromis entre ces deux critères. Le coût d'exploitation et la qualité de service offerte aux utilisateurs augmentent avec le nombre de véhicules mis en service.

Dans la figure I.1, nous présentons la position du TAD par rapport aux autres modes de transport en termes de rapport qualité/coût de service.

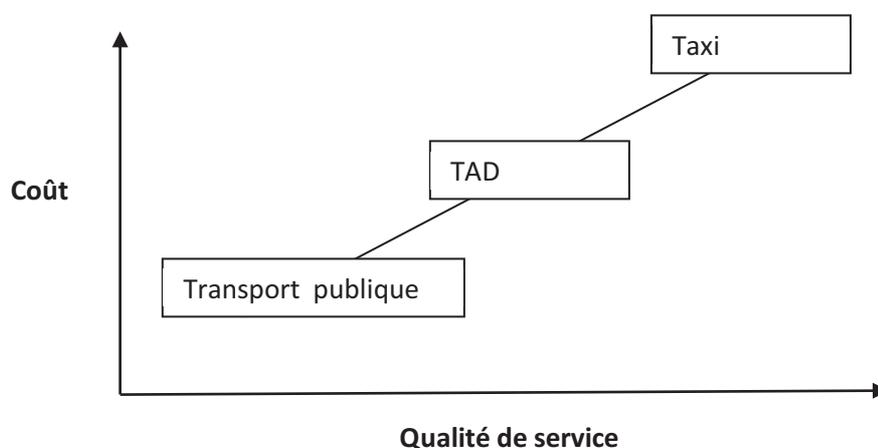


Figure I. 1 : Positionnement du TAD par rapport aux autres modes de transport [4].

### **I.3 Objectif et Problématique :**

Afin d'offrir un service de TAD remédiant aux problèmes écologiques et économiques mentionnés dans la section I.2, il va falloir trouver une solution formée par un ensemble de tournées de véhicules pour répondre aux demandes de transport. Dans le cadre de cette thèse, nous nous intéressons en particulier à trouver une solution informatique permettant le pilotage en temps réel ou différé des véhicules mis en jeu pour résoudre le problème de TAD. En plus de l'intérêt écologique et environnemental, le système ainsi mis au point doit optimiser les solutions à proposer aux utilisateurs. Cette optimisation nécessite la satisfaction de plusieurs critères qui peuvent être contradictoires. Par exemple, le confort du voyageur peut augmenter le coût pour l'opérateur offrant le service de TAD. Les critères à optimiser dans le système de TAD sont respectivement : La distance parcourue, le temps de service des véhicules, le nombre de véhicules utilisés dans le service et le confort du voyageur. En ce qui concerne les critères confort voyageur et la qualité de service rendue à ces derniers, ils sont composés de plusieurs sous-critères suivants : la durée de voyage de chaque client, le nombre de stations visitées dans un trajet pour un client et le niveau de satisfaction en termes de véhicule lorsqu'il s'agit des véhicules hétérogènes.

Nous pouvons distinguer deux différents problèmes à résoudre pour atteindre notre objectif. Le premier est un problème d'optimisation multicritère complexe. Il consiste à gérer l'affectation des voyageurs à des voitures et l'élaboration des tournées de véhicules pour servir les demandes de transport. Le deuxième est la gestion des profils utilisateurs pour personnaliser l'offre de service.

L'objectif principal visé dans cette thèse, est d'aider l'utilisateur dans sa démarche de recherche d'un voyage. Partant du fait que les utilisateurs ont besoin d'avoir des systèmes leur permettant d'accéder facilement et rapidement à des informations les plus pertinentes en fonction de leurs besoins et de leurs attentes. Une des solutions utilisées pour résoudre cette problématique consiste à utiliser la personnalisation de l'information afin d'optimiser la solution retournée à l'utilisateur. Le système utilisera ce que l'on connaît de l'utilisateur (voyageur), de ses préférences, de ses habitudes ou encore des choix effectués par des utilisateurs qui ont un profil similaire.

L'affectation des clients (voyageurs) à des véhicules instantanément, peut provoquer des modifications sur leurs itinéraires. Ce phénomène engendre des itinéraires dynamiques. Il devient donc nécessaire de développer une application de recherche et de gestion d'itinéraires permettant de proposer, en temps réel, des itinéraires en cas d'affectation dynamique.

Le but est de développer une nouvelle approche couplant des méthodes d'optimisation combinatoire et la personnalisation d'informations. Notre approche se base sur des systèmes multi-agents afin de mieux gérer les affectations et l'établissement des tournées de véhicules en temps réel ou différé.

## **I.4 Positionnement Scientifique :**

### **1. Recherche Opérationnelle :**

Le problème que nous traitons dans cette thèse concerne l'affectation et le routage d'un ensemble de véhicules homogènes ou hétérogènes pour servir des demandes de transport. Le cas le plus simple est l'affectation statique d'un véhicule à un parcours de stations. Ce problème d'élaboration de tournées de véhicules est connu sous le nom « **Dial a Ride Problem** » (DRP). Il représente un problème d'optimisation combinatoire multicritère. Il appartient à la classe des problèmes NP-difficile de l'optimisation combinatoire. Dans sa version de base, le DRP modélise un problème de transport qui consiste à livrer et collecter des personnes auprès d'un ensemble de lieux de départ vers des destinations à l'aide d'une flotte de véhicules. La résolution de ce problème consiste à déterminer un ensemble de tournées en minimisant plusieurs critères tel que la distance totale parcourue, le nombre de véhicules utilisés, etc... Le problème est donc un problème d'optimisation multicritère.

Le système envisagé doit répondre dynamiquement aux appels des voyageurs, d'où la nécessité d'utilisation des variantes du (DRP) classique. Ce problème est connu sous le nom de « **Dynamic Dial a Ride Problem** » (DDRP). La deuxième variante à utiliser pour la modélisation est le problème transport à la demande avec fenêtres de temps. Il est connu sous le nom de « **Dial a Ride Problem Time Window** » (DRPTW). Cette exploitation permettra la prise en compte de l'aspect temps réel de l'application envisagée.

## **2. Les Systèmes Multi-agents :**

Le but de l'outil développé dans cette thèse est de produire une solution de planification de tournées de véhicules pour transporter un ensemble de voyageurs d'une station de départ à une station d'arrivée. Le système doit affecter un ensemble de véhicules à un ensemble de tournées ayant la possibilité d'être changées selon la demande d'un voyageur. Avec l'aspect dynamique du système, la complexité du problème augmente en fonction du nombre de demandes de transport reçues. Cette complexité nous amène à décentraliser le traitement du problème en utilisant un système multi-agents. Les systèmes multi-agents se caractérisent par l'autonomie et l'intelligence des composants impliqués. Toutefois, un agent ne dispose que d'une vision locale de son environnement. Un agent peut être défini comme une entité (physique ou virtuelle) capable d'agir sur elle-même et sur son environnement. Un agent permet la représentation partielle de l'environnement et qui peut communiquer avec d'autres agents. Le comportement d'un agent est la conséquence de ses observations, de sa connaissance et ses interactions avec les autres agents. L'utilisation des systèmes multi-agents nous permet de décomposer le problème en plusieurs sous-problèmes. Le but de cette décomposition est de partager les tâches du système sur des agents autonomes. Ces derniers coopèrent ensemble pour atteindre le but envisagé.

Le système de transport à la demande, qui est l'objectif de cette recherche, est un système qui demande une optimisation d'un problème multicritère compliqué. Pour résoudre ce problème nous recommandons l'utilisation d'une plateforme de système multi-agents. Cette plateforme nous permet d'effectuer des simulations des algorithmes à proposer pour la résolution du problème.

## **3. Personnalisation de l'information :**

Le système envisagé doit fournir une solution d'affectation de véhicules à un ensemble de voyageurs en respectant un certain niveau de satisfaction de ces derniers. Cette satisfaction est tenue par le système comme un critère d'optimisation. Le fait d'offrir un service propre à un client (voyageur) selon ses préférences, revient à personnaliser ce service. Le but de la personnalisation de l'information qui représente tout un domaine de recherche, est de mieux répondre aux besoins de l'utilisateur. La personnalisation de l'information se définit généralement comme un ensemble de préférences représentées par des couples (*attribut,*

*valeur*). Ces préférences nous permettent de connaître les désirs de l'utilisateur (dans notre cas il s'agit d'un voyageur). L'ensemble des demandes de transport est représenté souvent par un *profil* qui regroupe les informations déjà connues sur un utilisateur du système. Le contenu d'un profil utilisateur varie selon l'application et le domaine d'études (**IHM** : Interface Homme Machine, **RI** : Recherche de l'Information, **BD** : Base de Données).

L'objectif du système envisagé est de tenir compte des besoins des utilisateurs (voyageurs) en utilisant la gestion des profils. Un profil utilisateur est relatif à un ou plusieurs utilisateurs du système. Une présentation des utilisateurs avec leurs profils nous permettra de regrouper ces derniers ou de les disperser. Le sens de regroupement ici est d'accepter des voyageurs dans le même véhicule.

## **I.5 Plan de la thèse :**

Le reste de ce document est organisé en cinq chapitres, qui sont décrits brièvement ici

- Dans le **chapitre 2**, nous présentons le service de **Transport À la Demande (TAD)** en détaillant la problématique du système de TAD. Ensuite nous rappelons les différents éléments qui composent ce problème, ainsi que les contraintes à satisfaire et les objectifs à optimiser. Nous présentons également une étude bibliographique concernant les problèmes de TAD dans les cas statique et dynamique.
- Dans le **chapitre 3**, nous présentons les Problèmes Multi-Objectifs (PMO) et les méthodes existantes dans la littérature pour les résoudre. Ces méthodes cherchent des solutions au PMO appelées Pareto-optimales. La qualité de ces solutions est jugée selon deux principes ; l'optimalité et la diversification sur l'espace de recherche. Dans ce chapitre nous présentons les critiques des méthodes d'optimisation multi-objectifs afin de choisir une méthode pour la résolution du PTD multicritères.
- Dans le **chapitre 4**, nous détaillons notre modélisation mathématique du PTD multicritères. Suite à cette modélisation, nous contribuons à la résolution du PTD par le développement d'une approche basée sur l'algorithme du recuit simulé multi-objectifs. Dans ce chapitre nous résoudrons le PTD statique et dynamique et nous

montrons que notre approche est efficace pour la résolution du problème dans le cas dynamique.

- Dans le **chapitre 5**, nous étendons notre modèle mathématique pour le PTD afin de traiter le critère de satisfaction en termes de véhicules. Cette satisfaction est déduite lorsqu'il s'agit de l'utilisation de véhicules hétérogènes pour servir les demandes de transport. Pour résoudre le PTD avec véhicules hétérogènes, nous développons un Système Multi-Agents (SMA) basé sur l'algorithme de recuit simulé multi-objectifs. Ce système prend en compte les préférences du voyageur d'où la gestion de profil utilisateur. Dans ce cas, les véhicules et les voyageurs sont des agents.
- Enfin, dans le **dernier chapitre** on présente les conclusions et les perspectives de ce travail de recherche.



# Chapitre II :

## Etat de L'art et Le Problème de Transport à la Demande

### II.1 Introduction :

Le Problème de Transport à la Demande (PTD) ou encore en anglais Dial a Ride Problem (DRP), consiste à prendre en charge des voyageurs ayant activé une demande de transport on les acheminant d'un point de départ vers un point d'arrivée. Le PTD est caractérisé par un ensemble de demandes de transport de taille «  $n$  » et d'un nombre de véhicules «  $m$  » pour servir ces derniers. La résolution d'un PTD est une solution de transport alternative permettant l'augmentation des taux d'occupation des véhicules. Cette augmentation a pour rôle de lutter contre la congestion routière et de réduire les émissions de gaz à effet de serre, et par conséquent la pollution de l'environnement. Il est également économiquement viable vu que le coût est partagé entre les usages sollicitant ce service.

Ce chapitre est organisé de la manière suivante : dans la deuxième partie, nous détaillons et nous décrivons le PTD. La troisième partie est dédiée à la présentation des travaux commercialisés et fonctionnels pour la résolution du problème de transport à la demande. Dans la quatrième partie nous ferons un survol de l'état de l'art académique concernant le PTD. Nous finirons ce chapitre par une conclusion.

### II.2 Description du Problème de Transport à la Demande (PTD):

#### 1. Contexte de l'étude :

Avec la hausse du prix des carburants, le véhicule personnel devient une charge de plus en plus lourde pour les revenus modestes. En effet, Le transport pèse très lourd dans le budget des ménages, il occupe la deuxième place avec plus de 15 % des dépenses [2]. En effet entre 1975 et 1990, en France, les distances domicile-travail sont passées d'une moyenne de 7 km pour un aller simple à environ 12 km. Seul un quart des actifs travaillent dans la commune où ils résident [3].

L'étude présentée dans cette thèse concerne l'extension du Problème de Tournées de véhicules PTV. En effet, Le PTV consiste à déterminer les trajets d'un ensemble de véhicules homogènes ou hétérogènes pour acheminer des marchandises tout en optimisant un ou plusieurs critères.

Notre objectif est de modéliser et de résoudre d'une manière optimale un PTD qui consiste à transporter des personnes. Ce problème est similaire au transport collectif avec parcours et horaires figés et le transport par véhicule individuel.

## **2. Définition du PTD :**

Le Problème de Transport à la Demande (PTD ou DRP) consiste à prendre en charge le transport d'un certain nombre de voyageurs d'un lieu de départ à un lieu d'arrivée. Ce dernier est caractérisé par un ensemble de demande de transport de taille «  $n$  » et d'un nombre de véhicules «  $m$  » pour servir ces dernières. Chaque demande de transport est modélisée par une requête contenant des informations sur la demande. L'exécution d'une requête consiste à prendre en charge une personne du point de ramassage «  $i$  » et de la déposer au point «  $i+n$  ». Le ramassage en «  $i$  » doit démarrer dans la fenêtre de temps  $[a_i, b_i]$  et l'arrivée à la destination doit se faire dans la fenêtre de temps  $[a_{i+n}, b_{i+n}]$ . En effet, le PTD est une extension du Problème de Tournées de Véhicules (PTV) avec pour le PTD une contrainte supplémentaire qui est la cohérence de l'ordre de passage d'un véhicule pour servir une demande. Il est évident que nous ne pouvons pas passer d'un point de livraison avant de récupérer la personne effectuant la demande.

### **a. Exemple d'un PTD:**

Dans la figure II.1 présentée dans cette section, nous montrons une présentation schématique et globale du PTD. Cette représentation illustre les composants principaux du problème. Ces composants sont essentiellement, l'ensemble des voyageurs, le nombre finis de stations (points de départ et destination) et la flotte des véhicules utilisée pour transporter les voyageurs (demandes de transport). Une vue globale du système nous permet de dégager une classification du problème parmi l'ensemble des problèmes issues du domaine de la logistique.

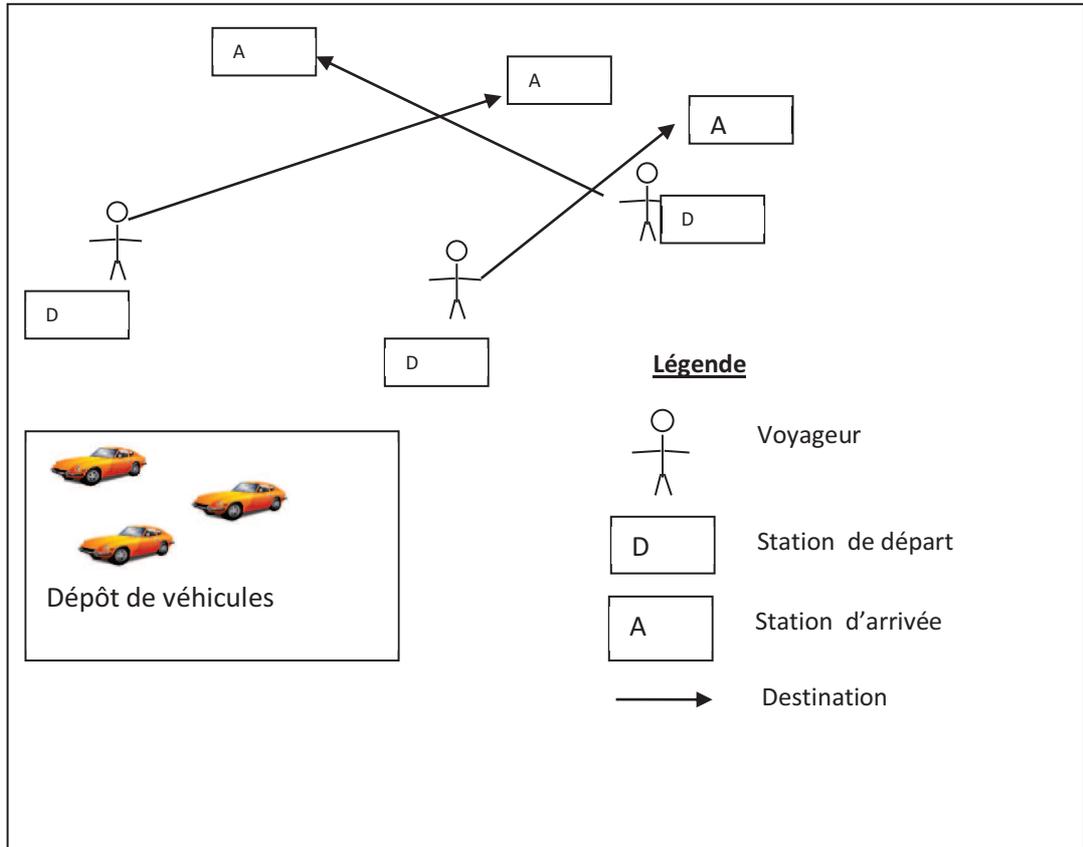


Figure II. 1 : Présentation schématique du problème

***i. Présentation matricielle d'une solution pour le PTD***

La solution du PTD peut être présentée par une matrice. Soit la matrice *sol* présentant une solution de ce dernier, les lignes modélisent les véhicules et les colonnes représentent les points de ramassage et de livraison concernant les demandes de transport. Si *i* est l'indice des lignes et *j* est l'indice des colonnes, *sol*[*i*][*j*] modélise l'ordre de passage du véhicule *i* par le point *j*. Comme nous avons mentionné dans la section (II.2.1), le PTD est une extension du PTV, donc pour résoudre ce problème il faut tout d'abord affecter les véhicules aux demandes puis optimiser les tournées de ces derniers. Dans la figure II.2 nous présentons la phase d'affectation des véhicules à des demandes de transport, et dans la figure II.3 nous présentons une solution complète du problème. Rappelons que pour une demande *j*, (*j* étant le point de départ et *j+n* le point de livraison).



Véhicules		Points de ramassage					Points de livraison				
		1	2	3	4	5	6	7	8	9	10
	1	1	1	1	0	1	1	1	1	0	1
2	0	0	0	1	0	0	0	0	1	0	

Figure II.2 : Affectation des véhicules à des demandes de transport dans la matrice *sol*

Si  $so[i][j]=1$ , donc le véhicule *i* sert le point de ramassage ou de livraison *j*

Véhicules		Points de ramassage					Points de livraison				
		1	2	3	4	5	6	7	8	9	10
	1	2	1	3	0	6	4	5	8	0	7
2	0	0	0	1	0	0	0	0	2	0	

Figure II.3 : Une solution au PTD présentée par la matrice *sol*

### II.3. Travaux industriels sur le Transport à la Demande (TAD):

Une étude exploratoire a été réalisée par le CERTU (Centre d'Etudes sur les Réseaux, les Transports, l'Urbanisme et les constructions) en 2007 sur les services de transport à la demande afin de mieux cerner cette alternative de transport, un état de l'art a été réalisé en France et à l'étranger, et en se focalisant sur certaines expérimentations particulièrement novatrices.

Nous survolons les travaux industriels existants sur les TAD en tenant compte, à la fois des expériences françaises et les expériences étrangères. Pour cette étude nous avons utilisé les différents rapports des études existantes.

## **1. Etat de l'art sur les TAD en France**

Dans cette section, nous présentons trois expériences françaises pour l'implantation des services de TAD en France. Les TAD représentés ici ne sont pas les seuls, ils ont été choisis parce qu'ils existent depuis plus de dix ans et nous disposons des toutes les informations les concernant.

Il est important de présenter ces services de TAD, puisque notre étude a pour objectif de réaliser une solution informatique permettant d'optimiser des tournées de véhicules pour offrir un service de TAD à la fois économique en terme de coût d'utilisation de véhicules et offrant une bonne qualité de service pour les voyageurs. Dans notre recherche, nous avons pu dégager trois grandes logiques de création de services de transport à la demande :

- a. Une logique politique, qui privilégie la couverture spatiale et la présence du service plus que l'efficacité commerciale ou même sa pertinence.
- b. Une logique financière visant à réduire les coûts, et qui voit dans les TAD une solution alternative à moindre frais.
- c. Une logique commerciale, pour laquelle le TAD est un produit commercialisable, apte à répondre au mieux à des exigences locales particulières, voire même à conquérir des nouvelles parts de marché [2].

### **d. La Rochelle (1985)**

**Taxis Autoplus** est un système de TAD mis en place en 1985 à la Rochelle afin d'offrir un service de transport en commun à la population 24h/24. En effet, ce service est basé sur le concept de borne d'appel permettant une réservation de transport instantanée aux voyageurs de la région. Le service fonctionne de la façon suivante : le client se rend à l'une des 82 bornes d'appels où il effectue sa réservation, puis l'un des 40 taxis vient alors le prendre en charge. La spécificité de ce service vient de la destination qui est imposée vers l'une des 3 bornes d'appel du centre-ville. Il s'agit donc d'une desserte du centre-ville. La caractéristique principale qui le raccroche à un TAD est sa spécificité liée à la destination centre-ville. La

gestion et la réalisation de ce service sont entièrement confiées au Groupement d'Intérêt Economique des taxiteurs [2].

#### **e. Saint-Nazaire (1991)**

T'y Bus à la demande a été créé en 1991. Ce TAD représente une nouvelle offre pour certaines communes périphériques en journée et offre une substitution du réseau urbain le soir. Le TAD est utilisé par l'utilisateur du réseau urbain avec des véhicules de petite taille aux couleurs identiques afin d'améliorer l'image du réseau. Ce service représente un complément du réseau urbain en proposant une offre 24 h /24h. En effet, les réservations sont gérées directement par le personnel du transport qui se donne le droit de modifier l'heure de la course prévue de quelques minutes en rappelant le voyageur afin d'optimiser au maximum les différents trajets demandés. La ville de Saint-Nazaire a été découpée en trois zones au sein desquelles chaque temps et distance inter arrêt ont été définis.

#### **f. Douai (1993)**

En 1993 **Taxitub** a été créé à Douai dans le but de favoriser les régions non desservies par le réseau urbain et d'améliorer la desserte de communes peu desservies (1 à 2 aller – retour quotidiens). **Taxitub** est basé sur un ensemble de lignes virtuelles à itinéraires fixes avec un mode automatique de réservation et de gestion des appels par serveur vocale (logiciel OPTITOD). En effet, le dimanche matin, la totalité des services urbains est remplacée par les Taxitub. La tarification par trajet correspond à la tarification la plus élevée du réseau urbain avec correspondance gratuite avec le réseau régulier.

### **2. Etat de l'art sur le Transport à la Demande (TAD) à l'étranger :**

#### **a. Allemagne**

Il existe En Allemagne, de nombreux sites internet proposent des trajets de longue distance avec au premier plan le site national « <http://www.mitfahrzentrale.de> » qui compte 700 000 inscrits. Ces sites nationaux concernent les trajets réguliers, dits Pendlernetz, mais de manière indirecte à travers les collectivités locales à qui ils vendent leur plateforme.

Le service Pendlernetz de l'Etat fédéral de Westphalie Rhin Nord est un site de TAD disponible chez les 165 municipalités qui composent cette région de l'ouest de l'Allemagne. Ce service est accessible depuis un site central (<http://www.nrw.pendlernetz.de>) ou par les sites dispersés sur les différentes municipalités, l'ensemble des sites étant mis en réseau. Les personnes n'ayant pas accès à internet peuvent y avoir accès par l'intermédiaire de centrales de mobilité et des administrations locales. Le service Pendlernetz est utilisé pour trouver des solutions aux problèmes de trafic et à l'augmentation des coûts de déplacement. Ce dernier a été mis en place dans le cadre de la démarche Agenda 21 du district de Kleve.

Pour une organisation du service de TAD en Allemagne, un groupe de travail associant les différentes municipalités a été constitué en 2002. Ce groupe a été piloté par le centre public régional en charge du développement technologique. Cette organisation oblige chaque municipalité à participer financièrement en fonction de son nombre d'habitants. L'état fédéral de Westphalie Rhin Nord finance le marketing du service. La mise en place technique a été réalisée par l'entreprise privée Europe Alive Media. En effet, les coûts opérationnels du service représentent 20€/hab. Ce service s'adresse aux 2,2 millions de personnes effectuant des trajets domicile-travail. Entre 8 000 et 9 000 trajets sont proposés chaque jour sur le site.

## **b. Belgique**

En Belgique, le TAD se fait entre collègues de travail et entre membres d'une même famille. La société Taxistop a mis en place un site très connu en Belgique qui est <http://www.carpoolplaza.be>. Ce site couvre la Wallonie et les Flandres. Il offre la possibilité aux usagers de s'enregistrer en déclarant leurs horaires et leurs itinéraires.

La société Taxistop utilise un outil informatique permettant de mettre en place un service de TAD qui peut être développé au niveau régional et mis à la disposition via internet. Donc une banque de données de TAD est mise à la disposition des entreprises et des particuliers d'une manière directe.

Les demandes des entreprises et des particuliers d'une même région sont regroupées dans une base de données commune centralisée pour l'optimisation de la qualité des listes d'appariement produites.

L'outil informatique développé par Taxistop permet aussi d'observer et de suivre pour chaque nouvelle demande, les dossiers des personnes n'ayant pas pu être aidées dans un premier temps. Ce suivi permet de prendre en charge automatiquement ces derniers dans un second appel de transport.

### **c. Grande-Bretagne**

En Grande-Bretagne, le service Liftshare (<http://www.liftshare.com>) est un service de transport à la demande national qui propose également des services pour les entreprises et les collectivités. Ce service est accessible via le site national ou par des sites locaux. Les sites locaux, connectés en réseau, sont aussi accessibles par le site national.

Le service Liftshare identifie:

- Les itinéraires communs ;
- Les tranches horaires communes ;
- La situation de conducteur ou passager de l'utilisateur ;
- Les possibilités d'appariement au sein d'une même entreprise ou en dehors.

Ainsi, nous pouvons noter que :

- Le service effectue l'appariement selon l'itinéraire et non en fonction des adresses ;
- Il est possible de faire du covoiturage avec des salariés d'une autre entreprise ;

## **II.4. Travaux de recherche sur le Transport à la Demande (TAD):**

Le TAD est un service de transport qui doit être présent dans différents lieux avec une flotte de véhicules. Le problème dans les TAD est de déterminer un ensemble optimisé (selon certains critères) d'itinéraires de véhicules permettant de répondre aux différentes demandes de transport. La nature des services à accomplir conditionne fortement la structure des problèmes. On distingue ainsi des sous-familles de problèmes de calcul de tournées. En effet, le Problème de Transport à la Demande (PTD) généralise un certain nombre de problèmes de

tournées de véhicules tels que le ramassage et la livraison ou en anglais Pickup and Delivery Vehicle Routing Problem (PDVRP). Pour un état de l'art détaillé sur ces problèmes, voir l'article de Cordeau et al en 2007 [5]. Ce qui rend le PTD différent de la plupart des problèmes de routage est le point de vue humain. Lors du transport de passagers, le PTD consiste à donner un service qui respecte à la fois la qualité de service rendue aux voyageurs et le coût de transport engendré par les tournées de véhicules. Dans les problèmes de tournées de véhicules, la capacité et le temps sont des facteurs contraignants et en particulier ceux liés à la collecte et à la livraison des voyageurs dans le PTD.

## 1. Les modèles mathématiques du Problème de Transport à la Demande (PTD)

Plusieurs modèles mathématiques ont été proposés pour un certain nombre de variantes du PTD. Nous présentons dans la sous-section 1.a une formulation mathématique à trois indices de Cordeau (2006) [6]. Dans la section 1.b, nous présentons une modélisation à deux indices qui est une modélisation plus compacte que celle à trois indices en raison de l'utilisation d'un seul véhicule pour la résolution du PTD. Dans la section 1.c, nous détaillons le problème d'ordonnancement comme un sous problème pour le PTD. En effet le problème d'ordonnancement consiste à déterminer une séquence de routage, puis tenter de réduire la durée des parcours des véhicules afin de servir les demandes de transport dans un PTD.

### a. Formulation à trois indices

Cordeau formule le PTD avec un graphe orienté  $G = (V, A)$  [5]. L'ensemble des sommets  $V$  est divisé en  $\{0, 2n+1\}, P, D$ , où  $0$  et  $2n+1$  sont deux exemplaires de dépôts,  $P = \{1, \dots, N\}$  est l'ensemble des points de ramassage et  $D = \{n+1, \dots, 2n\}$  est l'ensemble des sommets de livraison. Une requête de transport est un couple  $(i, n+i)$ , où  $i \in P$  et  $n+i \in D$ . Pour chaque sommet  $v_i \in V$  est associé une charge  $q_i$ , avec  $q_0 = q_{2n+1} = 0$ ,  $q_i \geq 0$  pour  $i = 1, \dots, N$  et  $q_i = -q_{i+n}$  pour  $i = n+1, \dots, 2n$ ,  $d_i \geq 0$  est la durée de service avec  $d_0 = d_{2n+1} = 0$ . L'ensemble des arcs est défini par  $A = \{(i, j) : i = 0, j \in P \text{ ou } i, j \in P \cup D, i \neq j \text{ et } i \neq n+j, \text{ ou } i \in D, j = 2n+1\}$ . La capacité du véhicule  $k$  est  $Q_k$ . La durée maximale de la route  $k$  (durée du trajet)  $\in K$  est notée  $T_k$ . Le coût de transport sur l'arc traversant  $(i, j)$  avec le véhicule  $k$  est égal à  $C_{kij}$ , et

le temps de voyage sur l'arc (i, j) est désigné par  $t_{ij}$ . Le temps de trajet maximal est noté L et la fenêtre de temps du sommet i est noté par  $[e_i, l_i]$ .

Le modèle utilise trois index binaires variables,  $X_{kij}$  égale à 1 si et seulement si l'arc (i, j) est traversé par le véhicule  $k \in K$ . En outre,  $u_{ki}$  est le temps où le véhicule k démarre son service au sommet i,  $w_{ki}$  est la charge du véhicule k à la sortie du sommet i, et  $R_{ki}$  la durée du voyage du client i du service de TAD. Le modèle résultant est le suivant :

$$\text{Minimize } \sum_{i \in V} \sum_{j \in V} \sum_{k \in K} X_{ijk} C_{ijk} \quad (2.1)$$

Avec :

$$\sum_{k \in K} \sum_{j \in V} X_{ijk} = 1 \quad \forall i \in P \quad (2.2)$$

$$\sum_{v \in V} \sum_{j \in V} X_{0jk} = \sum_{v \in V} \sum_{j \in N} X_{j,2n+1,k} = 1 \quad \forall k \in K \quad (2.3)$$

$$\sum_{j \in V} X_{ijk} - \sum_{j \in V} X_{j,n+i,k} = 0 \quad \forall k \in K, \forall i \in P \quad (2.4)$$

$$\sum_{i \in V} X_{ijk} - \sum_{i \in V} X_{jik} = 0 \quad \forall j \in P \cup D, \forall k \in K \quad (2.5)$$

$$X_{ijk} (u_{ik} + t_{ijk} - u_{jk}) \leq 0 \quad \forall v \in K, (i, j) \in A \quad (2.6)$$

$$e_i \leq u_{jk} \leq l_i \quad \forall i \in V, k \in K \quad (2.7)$$

$$e_{i+n} \leq u_i \leq l_{i+n} \quad \forall i \in V, k \in K \quad (2.8)$$

$$X_{ijk} (w_{ik} + q_{kj} - Q_{jv}) = 0 \quad \forall k \in K, (i, j) \in A \quad (2.9)$$

$$X_{ijk} \in \{0,1\} \quad (2.10)$$

Dans cette formulation, l'équation (2.1) modélise la fonction objectif du problème. Les contraintes (2.2) et (2.4) garantissent que chaque requête est servie une fois par le même véhicule, tandis que les contraintes (2.3) et (2.5) garantissent que chaque véhicule démarre et termine sa route à partir de l'entrepôt. Les Contraintes (2.6), (2.7) et (2.8) définissent respectivement les temps de service, les fenêtres de temps, tandis que la contrainte (2.9) modélise les charges des véhicules, (2.10) représente la variable de décision du problème.

## b. Formulation à deux index

Plus récemment, en 2006, Ropke, Cordeau et Laporte [31] [34] ont proposé deux modèles et un algorithme de branch-and-cut pour le PDP avec fenêtres de temps (PDPTW) et le PTD. Dans cette modélisation, les véhicules sont identiques. Nous faisons remarquer que le PDPTW est un PTD sans la contrainte de temps de trajet maximum. Dans cette partie, nous décrivons ce qui paraît le meilleur des deux modèles selon Cordeau. Ce modèle présente une flotte homogène de véhicules de capacité  $Q$ . Une variable de décision à deux index  $X_{ij}$  est utilisée,  $r$  désigne un chemin,  $R$  est l'ensemble des chemins qui respectent la contrainte de fenêtres de temps et le temps de trajet maximal,  $A(r)$  est l'ensemble des arcs de  $R$ ,  $P$  étant l'ensemble des points de départ et  $D$  l'ensemble des points d'arrivée,  $S$  est l'ensemble des nœuds  $S \subset P \cup D$ , notons également que, à chaque nœud est associée une charge  $q_i$  avec  $q(S) = \sum_{i \in S} q_i$  la charge totale des sommets de  $S$ . Le modèle mathématique résultant est le suivant :

$$\text{Minimize } \sum_{i \in V} \sum_{j \in V} X_{ij} C_{ij} \quad (2.11)$$

Avec :

$$\sum_{i \in V} X_{ij} = 1 \quad \forall i \in P \cup D \quad (2.12)$$

$$\sum_{j \in V} X_{ij} = 1 \quad \forall j \in P \cup D \quad (2.13)$$

$$\sum_{i, j \in S} X_{ij} \leq |S| - 2 \quad \forall S \in \mathcal{S} \quad (2.14)$$

$$\sum_{i, j \in S} X_{ij} \leq |S| - \max \{1, \lceil q(S)/Q \rceil\} \quad (S \subseteq P \cup D, |S| \geq 2) \quad (2.15)$$

$$\sum_{(i, j) \in A(r)} X_{ij} \leq |A(r)| - 1 \quad \forall R \in \mathcal{R} \quad (2.16)$$

$$X_{ij} \in \{0, 1\} \quad \forall i, j \in P \cup D \quad (2.17)$$

Dans ce modèle, la contrainte (2.14) permet de respecter l'ordre de passage entre un point de ramassage et le point de livraison qui lui est associé. La contrainte (2.15) granite le respect de la capacité du véhicule. Alors que la contrainte (2.16) est celle de l'élimination des sous tours et les chemins non valides dans la solution du PTD.

### c. Le problème d'ordonnancement dans le PTD

Étant donné une route  $k = (V_0 = 0, \dots, V_i, \dots, V_q = 2n + 1)$  constituée d'une séquence de sommets, le problème d'ordonnancement est de déterminer l'heure de départ du dépôt et la date à laquelle le service devrait commencer à chaque sommet  $V_1, \dots, V_{q-1}$ , de telle sorte que les fenêtres de temps sont satisfaites et la durée de trajet est réduite au minimum. Ce problème est d'une importance critique à chaque fois qu'une limite supérieure est imposée sur la durée du trajet.

Nous utilisons la notation suivante:

- $[e_i, \ell_i]$ : une fenêtre de temps concernant le sommet  $V_i$  (Tous les véhicules doivent quitter le dépôt au plus tôt à  $e_0$  et retourner au plus tard à  $\ell_{2n+1}$ );
- $t_{ij}$ : le temps de Voyage de  $V_i$  vers  $V_j$ ;
- $d_i$ : la durée du service au sommet  $V_i$ ;
- $A_i$ : l'heure d'arrivée du véhicule à  $V_i$ ;
- $u_i$ : le moment où le service commence à  $V_i$ ;
- $D_i$ : l'heure de départ du sommet  $V_i$ ;
- $W_i$ : le temps d'attente au point  $V_i$ .

Nous Notons que  $u_i \geq \max \{e_i, A_i\}$  et  $D_i = u_i + d_i$ . La fenêtre de temps à  $V_i$  est violée si  $u_i > \ell_i$ . Arrivée à  $v_i$  avant  $e_i$  est autorisée et donc le temps d'attente à ce sommet est  $W_i = u_i - A_i$ .

S'il est possible de résoudre le problème d'ordonnancement, une solution peut être identifiée de manière séquentielle par l'attribution des valeurs suivantes :  $u_0 = E_0$  et  $u_i = \max \{e_i, A_i\}$  pour  $i = 1, \dots, Q$ . Pour réduire la durée du trajet et le temps d'attente au sommet, il est avantageux de retarder le départ du véhicule du dépôt et le début de service au niveau des sommets de ramassage. Pour cela, nous devons calculer pour chaque sommet  $V_i$  le délai maximum d'attente au dépôt  $F_i$  avant le démarrage du service de sorte qu'aucune fenêtre de temps dans la route  $k$  ne sera violée [7].  $F_i$  est calculée de la façon suivante :

$$F_i = \min_{j \leq i \leq q} \left\{ l_j - \left( u_i + \sum_{i \leq p \leq j} t_{p,p+1} + d_p \right) \right\} \quad (2.18)$$

qui peut être réécrite sous la forme :

$$F_i = \min_{i \leq j \leq q} \left\{ \sum_{i \leq p \leq j} W_p - (l_j - u_j) \right\} \quad (2.19)$$

Avec

$$u_j = u_i + \sum_{i \leq p \leq j} t_{p,p+1} + d_p + \sum_{i \leq p \leq j} W_p \quad (2.20)$$

Au lieu de calculer le délai d'attente au dépôt  $F_i$ , certains auteurs ont minimisé une fonction  $f_k$  calculée en fonction de la variable  $u_i$ . Sexton et Bodin (1985) [8] considère le cas où  $f_k =$

$$\sum_{i=0}^{q-1} \alpha_i u_i \text{ tels que } \alpha_i \text{ sont des paramètres prédéfinis. Si } V_{i+1} \text{ désigne le successeur de } V_i \text{ dans la}$$

route  $k$ , le problème peut être formulé de façon plus générale comme un problème d'optimisation de la forme suivante :

$$\text{Minimiser } \sum_{i=0}^q g_i(u_i) \quad (2.21)$$

$$u_i - u_{i+1} \leq -t_{i,i+1} - d_i \quad (i = 0, \dots, q-1) \quad (2.22)$$

$$-u_i \leq -e_i \quad (i = 0, \dots, q) \quad (2.23)$$

$$u_i \leq l_i \quad (i = 0, \dots, q) \quad (2.24)$$

Avec  $g_i(u_i)$  une fonction convexe définie par rapport à la fenêtre de temps  $[e_i, l_i]$ . Dumas, Soumis et Desrosiers (1989) [9] ont proposé une approche pour résoudre ce problème en effectuant  $q$  minimisations unidimensionnelles. Dans le cas particulier où les fonctions objectives sont quadratiques ou linéaires, la complexité de l'algorithme est  $O(q)$ .

Les auteurs Hunsaker et Savelsbergh (2002) [10] ont développé une procédure de test de la faisabilité d'une insertion pour les heuristiques de construction ou d'amélioration des solutions pour le PTD. Ils considèrent une variante du PTD avec fenêtres de temps avec une limite supérieure pour le temps d'attente d'une demande  $W_i$ , et une borne supérieure sur la durée du trajet proportionnelle à la durée de conduite pour chaque véhicule. Ils ont mentionné comment vérifier en temps  $O(q)$  si l'insertion d'une requête donnée dans une route est faisable.

## 2. Le Problème de Transport à la Demande (PTD) avec un seul véhicule

Un des cas les plus simples du problème de transport à la demande est celui où tous les utilisateurs sont desservis par un seul véhicule. Nous allons présenter dans cette section des

algorithmes pour le cas statique, où toutes les demandes sont connues à l'avance, et pour le cas dynamique, où les demandes de transport arrivent peu à peu en temps réel.

### **a. Le PTD statique**

En 1980, Psaraftis a formulé et résolu le PTD avec un programme dynamique dans lequel les fonctions objectif est la minimisation de la somme pondérée des temps de service pour un véhicule et le nombre de clients insatisfaits [11]. L'insatisfaction du client est exprimée comme une combinaison pondérée des temps d'attente avant l'instant de ramassage et le temps de parcours. Dans cette solution pour le PTD, le transporteur impose une contrainte limitant la différence entre la position d'un utilisateur dans la liste d'appel et sa position dans la route d'un véhicule. Cet algorithme a ensuite été mis à jour par le même auteur (Psaraftis, 1983) [12] pour traiter des fenêtres de temps spécifiées par les voyageurs. Comme c'est souvent le cas des approches basées sur la programmation dynamique, l'algorithme ne peut résoudre avec optimalité que des problèmes à petite taille, car la procédure a une complexité de  $O(n^2)$ . Le plus grand PTD résolu en utilisant cette approche contient neuf utilisateurs. Alors que la plupart des PTDs, dans la pratique sont formulé par un nombre de demandes de transport plus important. En effet l'approche proposée pourrait encore s'avérer utile en tant que sous-routine dans un algorithme multi-véhicules, à condition que le nombre d'utilisateurs desservis par le même véhicule reste relativement faible.

Sexton (1979) [13] et Sexton et Bodin (1985a, b) [14] [15] ont également résolue le PTD avec un seul véhicule. Cette résolution est une étape dans le problème avec plusieurs véhicules. Leur approche se base sur une heuristique dans laquelle les voyageurs ont déjà été regroupés. Leur algorithme alterne entre la résolution d'un problème de routage par le biais d'une heuristique d'insertion et la résolution du problème d'ordonnancement associé. Dans ce travail les auteurs minimisent une fonction objectif présentant la somme pondérée de deux termes. Le premier terme est la différence entre le temps réel de voyage et le temps de voyage direct d'un client. Le second terme est la différence entre le temps de livraison souhaitée et le délai de livraison effective, sous l'hypothèse que le premier est au moins aussi grand que ce dernier. Comme expliqué dans la section *II.1.c*, cet objectif peut être exprimé comme une fonction linéaire des variables  $u_i$ . Les résultats de cette approche sont présentés sur plusieurs

ensembles de données de Baltimore et Gaithersburgh, où le nombre de demandes de transport varie entre 7 et 20.

Le PTD avec un seul véhicule a été formulé avec un programme entier par Desrosiers, Dumas et Soumis (1986). Dans cette formulation il existe des contraintes de fenêtres de temps ainsi que la capacité des véhicules et des contraintes de priorité. Les auteurs ont résolu le problème par la programmation dynamique. Des solutions optimales ont été obtenues pour un nombre de demandes de transport égal à 40.

### **b. Le PTD dynamique**

Le PTD dynamique a été également étudié par Psaraftis (1980) [11]. Dans le PTD dynamique des nouvelles demandes de transport sont produites de façon dynamique au cours du temps, mais aucune information sur le futur des demandes n'est disponible (contrairement à ce qui se passe dans la programmation stochastique). Lorsqu'une nouvelle demande est produite, une solution prévisionnelle sera disponible. Toutes les demandes qui ont été déjà traitées avant l'instant  $t$  ne seront pas retirées de la solution. Le problème est alors de ré-optimiser la partie de la solution après l'instant  $t$ , y compris la nouvelle demande. Ceci est réalisé par l'application de l'algorithme de programmation dynamique développé pour le cas statique. Une difficulté pratique qui paralyse cette approche est celle de la possibilité de résoudre le problème à l'instant  $t$  avant l'arrivée de la prochaine requête, ce qui ne peut être possible que si l'algorithme est lent et que les demandes arrivent rapidement. Une solution a été proposée dans le cadre de la localisation dynamique d'ambulance par Gendreau, Laporte et Semet (2001) pour contourner cette difficulté. La solution consiste à pré-calculer plusieurs scénarios, en utilisant le calcul parallèle par la prévision de demandes de transport futures. Malgré ses limites, le travail de Psaraftis sur le PTD dynamique a aidé à définir les concepts utilisés dans les recherches faites sur les problèmes de routage dynamique (voir par exemple, Psaraftis, 1988, 1995; Mitrović Minić, Krishnamurti et Laporte, 2004)[17] [18].

## **3. Le Problème de Transport à la Demande (PTD) avec plusieurs véhicules**

Beaucoup de recherches ont été effectués sur le PTD à plusieurs véhicules. Notre état de l'art concernant ce problème est subdivisé encore en cas statique et dynamique.

## a. Le cas statique

L'une des premières heuristiques pour la résolution du PTD statique avec plusieurs véhicules a été proposée par Jaw et al (1986) [19]. Dans leur travail, le modèle impose des fenêtres de temps strictes sur les points de ramassage des demandes et des fenêtres de temps soft (qui ne peuvent pas être respectées) sur les points de livraison. Dans ce travail, un temps maximum de voyage est formulé avec une fonction linéaire. En outre, les véhicules ne sont pas autorisés à être inactifs lorsqu'ils transportent des passagers. Dans cette recherche, une fonction objectif non-linéaire est utilisée pour évaluer la qualité des solutions obtenue pour le PTD. L'heuristique sélectionne les utilisateurs progressivement dans l'ordre du temps au plus tôt de ramassage progressif et les insère dans les itinéraires des véhicules. L'algorithme a été testé sur des cas artificiels de 250 requêtes de transport et sur un ensemble de données réelles avec 2617 demandes et 28 véhicules.

Une technique utilisée dans la résolution des PTD<sub>s</sub> consiste à définir des groupes de voyageurs pouvant être servi par le même véhicule puis exécuter la phase de routage. Cette idée est exploitée par Bodin et Sexton (1986) [20] qui construisent des clusters de voyageurs avant d'appliquer à chaque groupe l'algorithme de résolution par un seul véhicule de Sexton et Bodin (1985a, b) [14] [15] et de faire des swaps entre les clusters. Cette approche est testée sur deux instances extraites d'une base de données de Baltimore contenant environ 85 voyageurs. Plus tard, Dumas, Desrosiers et Soumis (1989) [21] ont amélioré cette approche en deux phases en créant des « mini-groupes » d'utilisateurs, c'est à dire, des groupes d'utilisateurs pouvant être servis dans la même zone à peu près au même moment. Ces mini-groupes sont ensuite combinés pour former les itinéraires de véhicules possibles, en utilisant une technique de génération de colonnes. Enfin, chaque véhicule ré-optimise sa route au moyen de l'algorithme avec un seul véhicule de Desrosiers, Dumas et Soumis (1986) [16] en plus, une étape de planification est exécutée. Les auteurs ont réussi à résoudre des cas réels présentant trois villes canadiennes : Montréal, Sherbrooke et Toronto. Les instances sont d'une taille de 200 voyageurs et donc sont facilement résolus. Cette facilité de résolution est imputable au fait que dans les plus grands cas il ne faut qu'une technique de décomposition spatiale et temporelle pour résoudre le problème. La phase de mini-regroupement a ensuite été améliorée par Desrosiers et al. (1991) qui ont présenté les résultats pour un ensemble de données comprenant près de 3000 utilisateurs [21]. Enfin, Ioachim et al. (1995) a montré qu'il

y a un avantage, en termes de qualité de la solution par l'utilisation d'une technique d'optimisation pour construire des clusters [22].

Toth et Vigo (1996) ont résolu un PTD formé par des demandes de transport réelles. Dans ce problème, les voyageurs spécifient les fenêtres de temps sur leur origine et leur destination [23]. Une limite proportionnelle à la distance directe est imposée sur la durée du trajet. Le transport des voyageurs est garanti par une flotte de minibus et de voitures particulières. Dans certains cas, les taxis peuvent également être utilisés, mais tant que ce mode n'est pas le meilleur mode de transport pour les personnes handicapées, une pénalité est imposée à leur utilisation. L'objectif dans le travail Toth et Vigo (1996) est de minimiser le coût total du service. Toth et Vigo ont développé une heuristique responsable en premier lieu d'affecter les demandes de transport à des véhicules au moyen d'une procédure d'insertion en parallèle, et en second lieu d'effectuer des échanges inter-route et intra-route. Les tests de ce travail sont effectués sur des cas impliquant entre 276 et 312 demandes de transport. Ces tests ont montré des améliorations significatives par rapport aux solutions précédentes faites à la main.

Une autre étude faite par Borndorfer et al. (1997) se base sur une approche en deux phases dans lesquelles les groupes des voyageurs sont d'abord construits, puis regroupés pour former des tournées de véhicules possibles [24]. Dans cette approche, un cluster est défini comme un « sous-tour maximal » tel que le véhicule n'est jamais vide. Dans la première phase de cette approche, un grand nombre de clusters est construit et puis un problème de partitionnement de l'ensemble de voyageurs est alors résolu. Cette résolution consiste à sélectionner un sous-ensemble de voyageurs pour qu'ils soient servis par le même véhicule. Dans la deuxième phase de cette approche, les itinéraires possibles de véhicules sont programmés. Les deux problèmes de partitionnement de l'ensemble de voyageurs et de routage de véhicules sont résolus par un algorithme de branch-and-cut. Cette approche est appliquée pour un PTD réel à Berlin qui comporte des demandes de transport dont la taille est comprise entre 859 et 1771 demandes par jour.

En 2003, Cordeau et Laporte ont proposé une heuristique basée sur l'algorithme de recherche tabou pour résoudre le PTD avec plusieurs véhicules [25]. Dans leur travail, les utilisateurs spécifient une fenêtre de temps sur l'heure d'arrivée et de départ de leur voyage. En effet, dans cette approche un temps de trajet maximal est associé à chaque demande de transport. Ce temps peut être soit le même pour toutes les demandes, ou calculé en utilisant un facteur de

déviations maximales de la durée du trajet direct de chaque demande en particulier. Dans cette approche, des contraintes de capacité maximale et de longueur de route sont imposées sur les véhicules. L'algorithme de recherche tabou itérative se base sur la suppression d'une demande de transport et sa réinsertion dans un autre itinéraire. L'algorithme proposé par Cordeau et Laporte a été testé sur des instances générées aléatoirement dont la taille varie entre 24 et 144 et un ensemble de demandes compris entre 200 et 295 fournies par un transporteur danois. Contrairement à d'autres algorithmes tels que la génération de colonnes et de branch-and-cut, l'algorithme de recherche tabou peut facilement répondre à une grande variété de contraintes et d'objectifs, même si elles sont non-linéaires.

Dans l'article de Aldaihani et Dessouky (2003), les auteurs traitent un PTD hybride dans lequel deux types de services sont offerts : le transport de porte à porte, et le transport vers et depuis un arrêt de bus [26]. Le choix entre les deux modes est fait par l'algorithme proposé. Les auteurs ont développé une heuristique basée sur une phase d'insertion-réinsertion, suivies par la recherche tabou. Deux objectifs sont pris en compte dans cette recherche : la minimisation de la distance parcourue par les véhicules et la minimisation du temps de voyage des passagers. Les auteurs ont trouvé des résultats après des tests effectués sur les données réelles du comité de Lancaster en Californie aux États-Unis. Diana et Dessouky (2004) ont appliqué une procédure pour résoudre une version du PTD dont l'objectif est une somme pondérée de la distance, du temps de trajet, et du temps d'inactivité des véhicules. Ils ont fourni des résultats à partir des tests effectués sur des PTDs de tailles qui varient entre 500 et 1000 demandes.

En 2006, quatre heuristiques ont été publiées pour résoudre PTD à plusieurs véhicules. Dans l'article rédigé par Rekiek, Delchambre et Saleh (2006) [27], le principal objectif est la minimisation du nombre de véhicules utilisés pour servir les voyageurs. Les auteurs proposent un algorithme génétique pour la phase de regroupement de voyageurs et un mécanisme d'insertion pour la phase de routage. Ils ont testé leur approche sur des données réelles de la Ville de Bruxelles où ( $100 \leq n \leq 164$ ). Xiang, Chu et Chen (2006) ont résolu une version de PTD dont l'objectif est la minimisation d'une combinaison de coûts fixes du véhicule, les coûts variables du véhicule, les frais de conducteur, la période d'attente et le temps de service [28]. Dans cette approche, les insertions et les échanges inter-route sont utilisés pour construire les itinéraires des véhicules. Les auteurs introduisent un élément de diversification

dans leur mécanisme de recherche en utilisant une fonction objectif secondaire responsable sur les temps d'inactivité des véhicules. Dans ce travail, des instances contenant entre 50 et 2000 demandes de transport ont été traitées. L'objectif considéré par Wong et Bell (2006) est la minimisation d'une combinaison linéaire des temps de fonctionnement total, du temps de parcours des passagers et des frais de taxi pour les demandes non affectées [29]. Les auteurs travaillent avec plusieurs types de véhicules. Certains véhicules sont équipés pour recevoir les usagers en fauteuil roulant, mais les capacités réservées pour les utilisateurs de cette catégorie sont réduites. Les auteurs proposent une procédure d'insertion parallèle. Les voyageurs sont d'abord classés selon un indice qui mesure la difficulté engendrée sur les autres requêtes déjà insérées dans l'itinéraire. Les insertions sont faites par l'examen des demandes les plus difficiles. Cette phase est suivie par une phase de post-optimisation composée de réinsertions des demandes dans d'autres itinéraires. L'algorithme a été testé sur des cas virtuels contenant 150 demandes de transport. Wolfler Calvo et Colorni (2006) ont développé une heuristique pour une version du PTD dans laquelle le nombre de véhicules disponibles est fixé et les fenêtres de temps sont imposées sur les points de ramassage et de livraison [30]. Le principe de leur heuristique consiste à construire un ensemble de routes  $m$  et un certain nombre de sous-tours en résolvant un problème d'affectation. Une phase de routage est ensuite effectuée pour insérer les sous-tours dans les routes  $m$  et de replanifier les demandes dans les routes. Des essais ont été effectués sur des instances contenant entre 10 et 180 demandes de transport.

Deux algorithmes exacts de branch-and-cut ont été proposés pour la résolution du PTD. En effet, Cordeau(2006) et Ropke [31], ont formulé le problème comme un programme linéaire avec des nombres entiers. Cette formulation est présentée respectivement dans les *sections II.1.a et II.1.b*. Dans leur travail, initialement, le modèle est résolu par la relaxation de certaines contraintes. Pendant le processus de branchement, des algorithmes de séparation sont appliqués pour identifier les contraintes violées parmi celles qui ont d'abord été relaxées. Ces contraintes sont ensuite introduites et le processus se termine lorsque l'arbre de recherche a été exploré en fonction des règles habituelles de branch-and-cut. Le plus grand exemple résolu avec le premier l'algorithme contenait 36 demandes, alors que le plus important problème est résolu par le second algorithme qui est composé de 96 demandes.

Melachinoudis, Ilhan et Min (2007) ont proposé une heuristique pour la résolution d'un PTD avec fenêtres de temps [32]. Dans ce problème, l'objectif est de minimiser une combinaison linéaire des coûts de transport et la non satisfaction des voyageurs. Ils ont développé une approche basée sur la recherche tabou qui utilise les réinsertions des demandes de transport. Des instances de PTD contenant jusqu'à 50 demandes ont été résolues. Jorgensen, Larsen et Bergvinsdottir (2007) ont travaillé sur le PTD avec un objectif multicritère contenant sept critères [33]. Leur heuristique alterne entre un mécanisme de recherche génétique pour construire des clusters de voyageurs et une méthode de recherche de voisinage pour construire les routes de véhicules. Ils ont résolu des PTD<sub>s</sub> contenant entre 24 et 144 demandes de transport.

Plus récemment, Claudio et al (2009) ont proposé un algorithme génétique pour la résolution du PTD avec plusieurs véhicules [34]. Les auteurs ont testé leur approche sur le benchmark de Cordeau et Lapote (2003) [35].

## **b. Le cas dynamique**

Borndorfer et al (1997) notent que la distinction entre PTD statique et dynamique est souvent floue dans la pratique, puisque les demandes sont souvent annulées [24]. Donc en conséquence, les transporteurs peuvent introduire de nouvelles demandes dans une solution conçue pour un problème statique. En outre, comme c'est déjà mentionné par ces auteurs, les PTDs dynamiques existent rarement dans la réalité, puisque un certain nombre de demandes de transport sont souvent connues avant le début de la planification. La difficulté ici est de concevoir des itinéraires pour les véhicules en tenant compte des demandes dynamique et avec une incertitude d'accueillir des futures demandes dynamiques.

Une résolution intéressante du PTD dynamique avec plusieurs véhicules est faite par Madsen, Ravn et Rygaard (1995), qui ont résolu un problème réel concernant les services de transport des personnes âgées et handicapées à Copenhague [37]. Les voyageurs peuvent spécifier une fenêtre de temps sur leur ramassage ou sur le délai de livraison, mais pas les deux à la fois. Des véhicules de plusieurs types sont utilisés pour fournir ce service de transport, mais tous ne sont pas disponibles à tout moment. Les demandes de transport arrivent en

dynamique tout au long de la journée. Les vitesses des véhicules sont variables. En plus les véhicules peuvent devenir indisponibles en raison de pannes. Les auteurs ont développé un algorithme d'insertion appelé « REBUS », sur la base de la procédure précédemment développée par Jaw et al (1986) [38]. Les nouvelles demandes sont insérées de manière dynamique dans les routes des véhicules en tenant compte de leurs difficultés d'insertion dans un itinéraire existant. L'algorithme a été testé sur 300 voyageurs servis par 24 véhicules. Les auteurs mentionnent que l'algorithme a été capable de générer de bonnes solutions en termes de qualité dans des temps de calcul très courts.

Le PTD résolu par Teodorovic et Radivojevic (2000) ne prend pas en compte une fonction objectif bien définie [39]. En effet le PTD résolu traite un ensemble de trois critères à minimiser: la distance parcourue, le temps d'attente des véhicules, et les temps de trajet des passagers. Chaque fois qu'une nouvelle demande arrive, elle est insérée dans l'un des itinéraires des véhicules en fonction de l'une des neuf règles qui varient selon que l'insertion de la demande dans un itinéraire donnerait une "petite", "moyenne" ou "grande" augmentation des critères déjà mentionnés. Tous les critères utilisés dans l'évaluation sont flous et toutes les mesures préférentielles sont calculées au moyen des fonctions floues. Des règles similaires sont utilisées pour déterminer la position de la demande dans l'itinéraire choisi. Le système a été testé sur des cas virtuels contenant 900 demandes.

Colorni et Righini (2001) ont testé trois objectifs différents du PTD dynamique, à savoir la maximisation des demandes de transport servies, la maximisation de la perception du niveau de service par des voyageurs, et la minimisation de la distance parcourue [40]. Dans leur recherche, un mécanisme d'insertion en alternance est mis en place. En effet, ce mécanisme consiste en une phase de regroupement de voyageur et d'une phase de routage de véhicules. L'algorithme de routage est un algorithme branch-and-cut appliqué à un ensemble de demandes dont les fenêtres de temps ne sont pas trop éloignées dans le futur. Les auteurs mentionnent qu'ils ont réalisé des expériences avec leur approche à Crema et Verbania, situés dans le nord de l'Italie.

Un algorithme développé par Coslovich, Pesenti et Ukovich (2006) suit une stratégie en deux phases pour l'insertion d'une nouvelle demande de transport dans un itinéraire existant [41]. Une phase off-line est d'abord utilisée pour créer un ensemble possible d'itinéraires

concernant les demandes connues en se basant sur le mécanisme 2-opt. Après Une phase on-line est utilisée pour insérer la nouvelle demande en tenant compte de l'objectif de minimisation de l'insatisfaction des utilisateurs.

## **II.5 Conclusion :**

Le PTD est un problème de routage important et difficile à résoudre. En effet, ce problème est rencontré dans plusieurs contextes et est susceptible de devenir très important dans les prochaines années. Il partage plusieurs caractéristiques avec les problèmes de ramassage et de livraison, mais comme il est un problème de transport de personnes, le niveau des critères de service devient plus important. Ainsi, la ponctualité, la réduction des temps d'inactivité et le coût de transport sont plus critiques dans le PTD. Après trois ans de recherche, nous pouvons dire que des heuristiques excellentes existent pour le cas statique. Il est maintenant possible de résoudre des problèmes avec plusieurs centaines d'utilisateurs dans des délais raisonnables et il devrait être possible d'appliquer des techniques de décomposition pour les grandes instances impliquant, par exemple, deux ou trois mille utilisateurs. Nous croyons que l'accent doit désormais être mis sur la version dynamique du problème. Cela implique la construction d'une solution initiale pour un ensemble limité de demandes connues à l'avance et la conception de fonctionnalités capables de déterminer si une nouvelle demande doit être insérée ou non. Si oui, comment les routes existantes devraient être modifiées pour en tenir compte. Dans le même esprit, il devrait être possible de mettre à jour une solution partiellement construite pour faire face à des annulations et autres événements imprévus tels que les retards de la circulation, des véhicules en panne. Enfin, les systèmes avancés devraient faire pleinement usage des nouvelles technologies comme les systèmes de positionnement du véhicule utilisés dans le domaine des services médicaux d'urgence (Brotcorne, Laporte et Semet, 2003) [42].

Le PTD est un problème d'optimisation de tournées de véhicules pour le transport des voyageurs. Ce dernier est un problème d'optimisation multi-objectif, il faut tenir compte d'un ensemble de critères qui sont généralement contradictoires. Dans le chapitre suivant, nous mettons l'accent sur les problèmes d'optimisation multi-objectif et les méthodes existantes pour la résolution de ces derniers.

# Chapitre III :

## Les Problèmes d'Optimisation Multi-Objectif et les Approches de Résolution

### III.1 Introduction :

Dans un Problème d'Optimisation Multi-objectif (POM), nous cherchons à optimiser un vecteur de fonctions composé par un ensemble de critères. Nous obtenons par conséquent plusieurs solutions à la fin de la recherche appelées solution de Pareto. Les solutions obtenues représentent la surface de compromis. En effet, toute solution de cet ensemble est optimale dans le sens si aucune solution ne domine cette dernière pour tous les objectifs traités. Une autre définition peut être formulée comme suit : dans cet ensemble, aucune amélioration ne peut être faite sur une composante du vecteur sans dégradation d'au moins une autre composante de ce vecteur [78]. Donc l'objectif ou le but dans la recherche d'une solution d'un POM est d'obtenir l'ensemble des solutions Pareto optimales ou bien, sélectionner des solutions diversifiées dans l'ensemble. En effet, la première phase dans la résolution pratique de POM est la détermination de l'ensemble de solutions Pareto optimales. Puis dans un deuxième temps, le choix d'une solution à partir de cet ensemble suivant des préférences choisies par le décideur. Le choix d'une solution par rapport à une autre nécessite la connaissance du problème et de nombreux facteurs liés à ce dernier. Ainsi, un décideur peut choisir une solution qui ne sera pas acceptée par un autre, elle pourra également ne plus convenir dans un milieu dynamique. Il serait judicieux d'avoir un ensemble de solutions alternatives afin de choisir une solution Pareto optimale.

Dans ce chapitre, nous allons définir et détailler les POM<sub>s</sub>. Puis, nous nous intéressons aux méthodes d'optimisation combinatoire multi-critère permettant d'approximer la frontière Pareto sans s'attarder sur les aspects d'aide à la décision pour le choix d'une solution finale. En effet, nous présentons un état de l'art des approches basées sur des métaheuristiques appliquées pour la résolution de POM. Ces approches permettent de générer une variété de solutions Pareto optimales diversifiées dans l'espace de recherche. Une analyse critique de

chaque classe de méthode est présentée dans ce chapitre. Des réponses à des questions spécifiques aux POM, comme le maintien de la diversité, l'évaluation des performances, et l'étude des paysages des frontières Pareto sont abordées.

### III.2 Définitions et Vocabulaires :

Les problèmes d'optimisation réels sont décrits généralement à l'aide de plusieurs objectifs ou critères souvent contradictoires devant être optimisés d'une façon concurrente. Alors que, pour les problèmes mono-objectifs, l'optimum cherché est clairement défini, celui-ci reste à formaliser pour les problèmes d'optimisation multiobjectif. En effet, pour un problème à deux objectifs contradictoires, la solution optimale cherchée est un ensemble de solutions correspondant aux meilleurs compromis possibles pour les deux objectifs formant un POM.

Prenons le cas d'une personne souhaitant voyager. Le voyage idéal est celui qui est peu chère avec un moyen de transport qui regroupe moins de voyageurs dans le même véhicule. Mais ce voyage idyllique n'existe pas. Notre voyageur va donc devoir identifier les meilleurs compromis possibles correspondant à son budget (voir figure III.1).

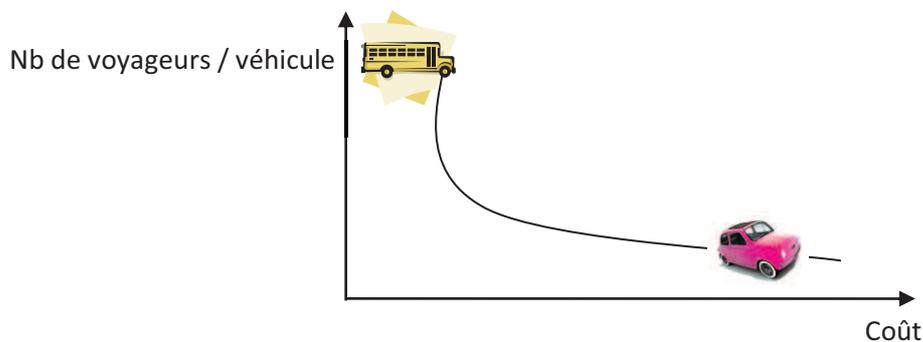


Figure III.1 : Relation entre le type de véhicule et prix.

Le problème d'optimisation multi-objectif est formulé de la façon suivante :

$$\text{POM : } \begin{cases} F(x) = (f_1(x), f_2(x) \dots f_n(x)) \\ \text{s.c } x \in C \end{cases}$$

Avec  $n \geq 2$  représente le nombre de fonctions objectifs.  $x = (x_1, x_2, \dots, x_k)$  est le vecteur représentant les variables de décision,  $C$  représente l'ensemble des solutions réalisables

associé à des contraintes d'égalité, d'inégalité et des bornes explicites (espace de décision), et  $F(x) = (f_1(x), f_2(x) \dots f_n(x))$  est le vecteur des critères à optimiser.

Dans les problèmes d'optimisation multi-objectif, le décideur a besoin d'un ensemble de règles pour évaluer une solution pour chaque critère. En effet,  $y=F(C)$  est l'ensemble représentant les points réalisables dans l'espace des critères et  $y = (y_1 \dots y_n)$  avec  $y_i=f_i(x)$  représente un point de l'espace des critères. Dans les POM, une relation de dominance est imposée sur cet ensemble de points.

La résolution du problème mathématique formulé ci-dessous nous permet de trouver les solutions Pareto optimales.

$$\text{POM}_\alpha \left\{ \begin{array}{l} \min F(x) = \sum_{i=1}^n \alpha_i f_i(x) \\ \text{s.c } x \in C \end{array} \right.$$

Avec  $\alpha_i \geq 0$  pour  $i=1, \dots, n$  et  $\sum_{i=1}^n \alpha_i = 1$

Après la résolution de ce programme, les solutions trouvées sont appelés des solutions supportées. De point de vue complexité un  $(\text{POM}_\alpha)$  a une complexité similaire à celle du problème combinatoire sous-jacent. En effet, si le problème combinatoire sous-jacent est polynômial, l'obtention des solutions supportées est relativement facile. Toutefois, il existe d'autres solutions qui ne peuvent pas être obtenues par la résolution d'un programme  $(\text{POM}_\alpha)$ . En effet, ces solutions, dites *non supportées*, sont dominées par les solutions supportées. Il existe des approches permettant de générer ces solutions non supportées. Parmi ces approches, nous pouvons citer par exemple, les approches de programmation par but fondées sur l'utilisation d'une norme de Tchebycheff. Le principe de fonctionnement de ces approches est la modification de la structure du problème combinatoire qui perd ainsi ses éventuelles propriétés importantes. Et par conséquent, l'obtention de solutions non supportées est en général encore plus difficile [72].

### III.3. Approches de résolution des problèmes d'optimisation multi-objectif

À propos des Approches de résolution des problèmes multi-objectif, plusieurs ouvrages ou articles de synthèse ont été rédigés, des états de l'art plus complets peuvent être consultés notamment dans [43] [44] [45] [46] [47]. En effet, dans la littérature, nous constatons qu'il existe un grand nombre d'approches pour résoudre les problèmes multi-objectif. Parmi ces approches, il existe certaines qui utilisent des connaissances du problème pour fixer des préférences sur les critères et ainsi contourner l'aspect multicritère du problème. Cette méthode exige la bonne connaissance du domaine du problème par le décideur. D'autres approches existent permettant de mettre tous les critères au même niveau d'importance, là aussi, il existe plusieurs façons de réaliser une telle opération.

Dans les approches de résolution des POMs, nous pouvons distinguer deux catégories : les approches non Pareto et les approches Pareto. Les approches non Pareto ne traitent pas le problème comme un véritable problème multi objectif. Cette technique cherche à ramener le problème initial à un ou plusieurs problèmes mono-objectifs. Alors que les approches Pareto ne transforment pas les objectifs du problème mais les traitent sans aucune distinction pendant la résolution.

Dans cette section, Nous allons décrire les principales approches non Pareto et commenter leurs avantages et leurs inconvénients.

### **1. Les approches de transformations des objectifs du POM en un seul objectif**

Dans notre recherche bibliographique, nous avons trouvé des méthodes traditionnelles qui transforment le POM en un problème mono-objectif. Ces méthodes sont des méthodes d'agrégation des objectifs à fin de transmettre le problème vers un problème mono-objectif. Parmi ces méthodes on trouve les méthodes d'agrégation, les méthodes  $\varepsilon$ -contrainte, et les méthodes de programmation par but.

### a. Méthode d'agrégation

La méthode d'agrégation est l'une des premières méthodes utilisée pour la reconstruction de solutions Pareto optimales. Elle consiste à transformer le problème  $POM_\alpha$  en un problème d'Optimisation Uni-objectif (POU). Cette méthode combine les différentes fonctions coût  $f_i$  du problème en une seule fonction objectif  $F$  généralement de façon linéaire (Hwang and Masud, 1979) [48] : La fonction se présente alors sous la forme suivante:

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

Avec les poids des fonctions objectifs  $\alpha_i \in [0, 1]$  pour  $i=1, \dots, n$ , et  $\sum_{i=1}^n \alpha_i = 1$

Les paramètres choisis pour le vecteur de poids  $\alpha$  influe d'une manière directe sur les résultats obtenus dans la résolution du problème ( $POU_\alpha$ ). En effet, les poids  $\alpha_i$  doivent aussi être choisis en fonction des préférences adjointes aux objectifs, cette tâche est assez délicate. Plusieurs tests seront donc nécessaires pour la résolution d'un tel problème.

La méthode d'agrégation présente l'avantage de produire une seule solution et ne nécessite donc pas d'interaction avec le décideur. Par contre, la solution trouvée par cette méthode peut ne pas être acceptable et ceci pour deux raisons ; la première est que l'espace de recherche est considérablement réduit prématurément à cause de la non disponibilité d'informations suffisantes, et la seconde, est la difficultés de choisir les valeurs des poids pour chaque fonction sans connaissance suffisante sur le problème à résoudre [49] [114].

### b. Méthode $\varepsilon$ -contrainte

La méthode  $\varepsilon$ -contrainte consiste à optimiser une seule fonction objectif  $f_k$  du POM possédant des contraintes sur les autres fonctions objectifs.

$$POM_k(\varepsilon) \left\{ \begin{array}{l} \min f_k(x) \\ \text{s.c } x \in C \\ \text{s.c } f_j(x) \leq \varepsilon_j, j = 1, \dots, n, j \neq k \end{array} \right.$$

Avec  $\varepsilon = (\varepsilon_1, \dots, \varepsilon_{k-1}, \varepsilon_{k+1}, \dots, \varepsilon_n)$ .

Dans la méthode  $\varepsilon$ -contrainte, le problème est formulé sous forme d'un problème uni-objectif (objectif  $f_k$ ) sujet à des contraintes sur les autres objectifs. Pour pouvoir générer différentes solutions Pareto optimales, il faut générer différentes valeurs de  $\varepsilon_i$ . Il faut avoir une connaissance a priori des intervalles appropriés pour les valeurs  $\varepsilon_i$  pour tous les objectifs du POM. Pour pouvoir définir les valeurs adéquates pour  $\varepsilon_i$ , le vecteur idéal doit être calculé pour déterminer les bornes inférieures. Nous allons avoir :

$$\varepsilon_i \geq f_i(x^*), i=1,2,k-1,k+1,n$$

L'approche  $\varepsilon$ -contrainte a été expérimentée en utilisant différentes heuristiques. Dans ce contexte, plusieurs travaux ont utilisé les Algorithmes génétiques (AGs) pour résoudre les POMs (Veldhuizen et al., 1997) [50], (Ritzel et al., 1994) [51].

### c. Programmation par but

Dans la méthode de programmation par but, le décideur doit définir des buts ou références qu'il désire atteindre pour chaque objectif. Cette méthode transforme ainsi le problème multi-objectif en un problème uni-objectif. Par exemple, la fonction coût peut intégrer une norme pondérée qui minimise les déviations par rapports aux buts. Le problème peut être formulé de la manière suivante :

$$\text{POM}_{\alpha}(\varepsilon) \begin{cases} \min (\sum_{i=1}^n \alpha_i |f_i(x) - z_i|^p)^{1/p} \\ \text{s.c } x \in C \end{cases}$$

Avec  $1 \leq p \leq \infty$  et  $z$  est le vecteur de référence (but) ou le vecteur idéal. Cette normalisation utilisée est la métrique de Tchebycheff ( $L_p$ -métrique). Généralement  $p$  est égal à 2 ; dans ce cas nous aboutissons à une métrique euclidienne. Si  $p=1$ , l'équation revient à une fonction Min-Max. Lorsque nous sélectionnons une valeur arbitraire pour le vecteur de référence, nous ne pouvons pas avoir une optimisation qualifiée ou désirable, étant donné qu'un mauvais choix du vecteur de référence peut aboutir à une solution qui n'est pas Pareto optimale. En effet, différentes méta-heuristiques ont été utilisées pour résoudre cette classe de problèmes

(Wienke et al., 1992) [52] . Dans (Coello, 1998) [53], une fonction min-max comparant les déviations relatives par rapport à des minimas atteignables (vecteur idéal) est utilisée.

Dans les approches basées sur la transformation du problème multi-objectif en un problème uni-objectif, il faut posséder des connaissances a priori sur le problème traité. En effet, l'optimisation d'un problème uni-objectif peut garantir l'optimalité Pareto de la solution trouvée, mais trouve une seule solution. Dans les situations réelles, les décideurs ont généralement besoin de plusieurs alternatives. En effet, si certains objectifs sont bruités ou des données sont incertaines, ces méthodes ne sont pas efficaces. Ils sont également sensibles au paysage de la frontière Pareto (convexité, discontinuité, etc.). L'autre inconvénient de ces méthodes est leur sensibilité par rapport aux poids, aux contraintes et/ou aux vecteurs de référence. Les solutions obtenues dépendent fortement de ces paramètres. Pour différentes situations, différents paramètres sont utilisés, et le problème doit être résolu plusieurs fois, d'où le coût associé à cette classe d'algorithmes, qui nécessitent parfois l'optimisation indépendante de chacun des objectifs. Dans l'exécution multiple des algorithmes, on espère trouver différentes solutions Pareto optimales.

## **2. Approches non-Pareto**

Dans les approches non-Pareto, nous séparons les différents objectifs non comparables. Dans cette section, nous mettons l'accent sur les algorithmes évolutionnaires comme étant des approches non-Pareto.

### **a. Sélection parallèle dans les algorithmes évolutionnaires**

Concernant l'utilisation des algorithmes évolutionnaires et plus précisément les Algorithmes Génétiques  $AG_s$ , (Schaffer, 1985) [54] a utilisé les  $AG_s$  pour résoudre les POM. L'algorithme développé baptisé VEGA (Vector Evaluated Genetic Algorithm) sélectionne les individus de la population courante selon chaque objectif, indépendamment des autres (sélection parallèle) (voir figure III.2). Dans cette approche, à chaque génération, la population est divisée en plusieurs sous-populations égales au nombre d'objectifs de la fonction coût. Chaque sous-population  $i$  est sélectionnée suivant l'objectif  $f_i$ . L'algorithme VEGA compose la population complète, et applique les opérateurs génétiques (mutation et croisement).

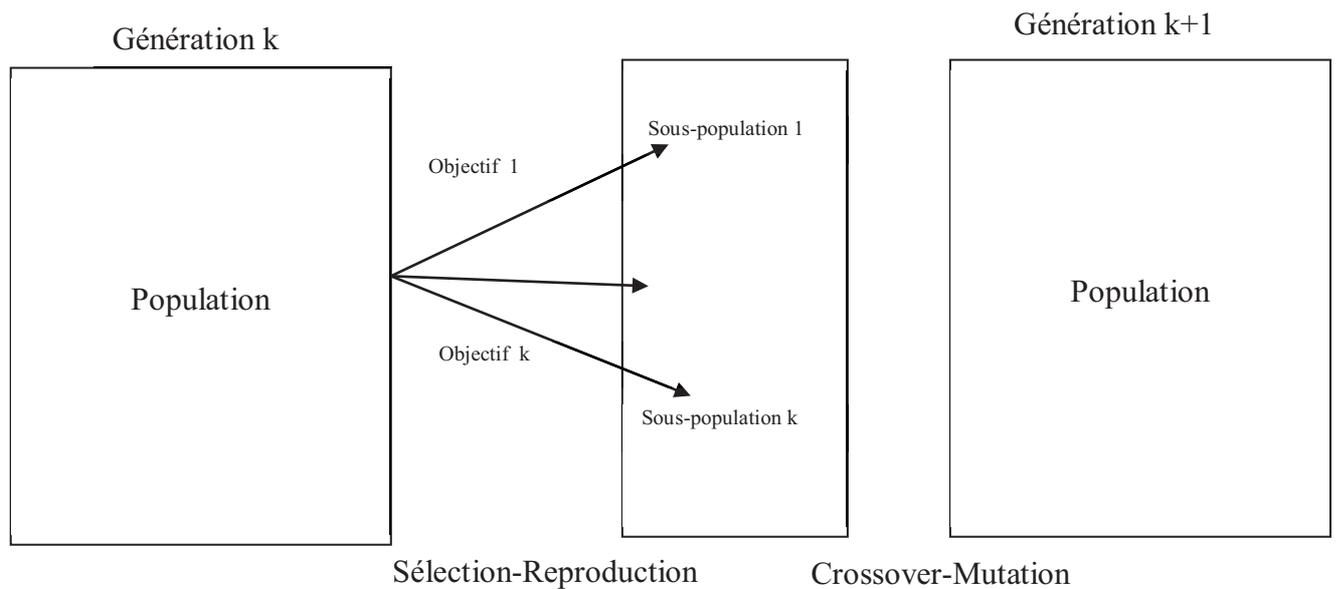


Figure III.2 : Sélection parallèle dans l’algorithme VEGA

Puisque l’algorithme VEGA affecte aléatoirement les individus aux sous-populations à chaque génération, nous pouvons trouver une situation où un même individu peut être évalué différemment suivant l’objectif qui lui est affecté, d’une génération à l’autre. Une analyse a été effectuée sur l’algorithme VEGA et a montré que son comportement est le même qu’un algorithme réalisant une agrégation linéaire (Richardson et al., 1989) [55]. En effet, l’algorithme VEGA a été l’objet de beaucoup de recherches faites par plusieurs auteurs. Surry et Radcliffe l’ont utilisé dans la modélisation des contraintes d’un problème uni-objectif pour éviter l’utilisation des pénalités dans la fonction coût (Surry et al, 1995) [56]. Dans ce travail, les contraintes sont modélisées comme des objectifs. La procédure standard de VEGA a été transformée pour introduire une forme de “ranking” basée sur le nombre de contraintes violées par un individu. D’autres opérateurs de sélection parallèle ont aussi été proposés comme le “n-branch tournament” (Jones et al, 1998) [57].

#### d. Reproduction multi-sexuelle

La méthode de reproduction multi-sexuelle permet l’utilisation de plusieurs classes d’individus dans un algorithme à base de populations. Ces classes sont associées à un des

objectifs. Ce travail a été proposé dans (Allenson, 1992) [58]. Un sexe différent est associé à chaque classe d'individus. Un individu d'un sexe donné est évalué par rapport à la fonction objective associée à son sexe. Dans le cas de la résolution d'un problème bi-critères, nous associons deux sexes différents, mâle et femelle aux deux objectifs. La reproduction est permise seulement entre un individu mâle et un individu femelle, l'attribution du sexe à l'individu étant faite aléatoirement à la création de ce dernier. Dans la phase de l'initialisation de l'approche de reproduction multi-sexuelle, le nombre d'individus par sexe est le même. Par contre c'est n'est pas le cas au cours des générations suivantes. En effet l'auteur utilise les algorithmes évolutionnaires pour implémenter une forme d'attracteurs sexuels.

En 1996 Lis et Eiben ont généralisé cette approche dans les AGs (Lis and Eiben, 1996) [59]. En effet, dans leur approche, l'opérateur de croisement a été changé pour porter sur plusieurs individus et non pas sur deux individus seulement comme dans l'AG standard. La sélection d'un parent de chaque sexe (objectif) permet ainsi à tous les sexes de contribuer à la génération d'un nouvel individu. L'individu créé aura le sexe du parent dont il hérite le plus grand nombre de gènes, et réalise un compromis entre les différents critères. Dans cette approche, l'opérateur de mutation est limité pour éviter à un individu de changer de sexe.

Cette approche présente un inconvénient majeur qui est son aptitude à générer des solutions optimisées pour certains critères et pas du tout pour les autres. Donc, Les solutions présentant des compromis sont négligées.

### **3. Approches Pareto**

Dans les approches Pareto, la notion de dominance est utilisé dans la sélection des solutions générées, contrairement aux autres approches qui utilisent une fonction d'utilité ou traitent séparément les différents objectifs. Cette approche a été introduite dans un premier travail dans les AGs par Goldberg (Goldberg, 1989) [60]. Cette approche présente un avantage principal par rapport aux autres approches, qui est la possibilité de générer des solutions Pareto optimales dans les portions concaves de la frontière Pareto.

Les approches Pareto se basent essentiellement sur la notion de population. Pour cette raison Les AGs ont été largement utilisés pour la résolution Pareto de POM, étant donné qu'ils travaillent sur une population de solutions. En effet, dans cette approche, deux objectifs

doivent être pris en compte dans la résolution d'un POM. La convergence vers la frontière Pareto et la diversification des solutions dans cette frontière [114].

- **La convergence vers la frontière Pareto :**

Les méthodes de ranking permettent de converger vers les solutions Pareto optimales. La plupart des travaux de recherche sur l'application des AGs aux POM se sont concentrés sur l'étape de sélection. En effet, dans cette étape, des méthodes de ranking sont appliquées dont le rôle est d'établir un ordre (rank) entre les individus. Cet ordre dépend de la notion de dominance et donc directement de l'optimalité Pareto.

- **La diversification des solutions dans la frontière Pareto :**

Pour stabiliser les sous-populations multiples le long de la frontière Pareto, il faut utiliser des méthodes de maintien de la diversité. Ces méthodes se basent sur la formation de niches écologiques et d'espèces.

#### **a. Les méthodes de ranking**

Plusieurs travaux de recherche ont été concentrés sur les méthodes de ranking. Elles ont été utilisées dans la littérature tel que : NSGA (Non-dominated Sorting Genetic Algorithm) [60], NDS (Non Dominated Sorting) (Fonseca and Fleming, 1995) [61], WAR (Weighted Average Ranking) (Bentley and Wakefield, 1997) [62]. Dans cette section nous détaillons la méthode NSGA (Non-dominated Sorting Genetic Algorithm) [60] :

- **NSGA (Non-dominated Sorting Genetic Algorithm) :**

Cette méthode a été initialement proposée par Goldberg (Goldberg, 1989) [60] et implémentée par Srinivas et Deb (Srinivas and Deb, 1995) [63]. Cette procédure de ranking donne à tous les individus non dominés de la population le rang 1. Par la suite, ces individus sont enlevés de la population, et l'ensemble suivant d'individus non dominés est identifié et on leur attribue le rang 2. Ce processus est réitéré jusqu'à ce que tous les individus de la population aient un rang. C'est dans les AGs pour la résolution de plusieurs POM que cette méthode de ranking a été utilisée traitant les problèmes suivants : arbre recouvrant minimum (Zhou and Gen, 1999) [64], sac à dos (Abdelaziz et al. 1999) [65], conception de réacteurs (Parks and Miller, 1998)[66], distribution de l'eau (Halhal et al., 1997) [67].

- **Sélection**

Il existe plusieurs travaux de recherches sur la sélection basées sur la dominance. En effet, Horn et al ont proposé une sélection à base de tournoi (Horn and Nafpliotis, 1993) [69]. Chaque tournoi de cette méthode est constitué par deux individus A et B qui entrent en compétition par rapport à un ensemble de  $t_{dom}$  individus. Si l'individu A domine l'ensemble des individus, et le compétiteur B est dominé par au moins un individu de l'ensemble, alors l'individu A est sélectionné. Pour contrôler la pression de sélection, on agit sur le paramètre  $t_{dom}$ : si  $t_{dom}$  est petit, la pression est faible et le nombre de solutions Pareto dans la population sera réduit. Par contre, si  $t_{dom}$  est grand, la pression est forte et une convergence prématurée peut se produire. La méthode de tournoi a été utilisée dans les AG<sub>s</sub> pour un problème de design de polymère dans l'industrie plastique (Cunha et al., 1997)[70].en 1996 (Tamaki et al) ont élaboré une stratégie de réservation Pareto qui correspond à l'élitisme dans le cas uni-objectif. (Tamaki et al., 1996) [71]. En effet, dans cette méthode, les individus non-dominés sont toujours sauvegardés pour la génération suivante. Si le nombre de solutions non-dominées est inférieur à la taille de la population, le reste de la population est produit en appliquant la sélection parallèle de la méthode VEGA. Par contre, si le nombre d'individus non dominés est supérieur à la taille de la population, les individus de la génération suivante sont sélectionnés parmi les individus non dominés en appliquant la sélection parallèle. Ce travail a été amélioré en utilisant une technique de partage de fonction dans la sélection des individus non dominés

## **b. Les méthodes de diversification**

Dans un POM, il est nécessaire que les solutions trouvées soient Pareto optimales, mais aussi qu'elles soient uniformément réparties dans le sous-espace des solutions Pareto optimales. Pour atteindre le premier objectif qui est la convergence vers la frontière Pareto, il faut utiliser les méthodes de ranking présentées dans la section précédente. Mais le deuxième objectif n'est pas pris en compte. Pour maintenir une diversité dans la population, les méthodes de ranking doivent être utilisées en complément avec les techniques de formation de niches écologiques et d'espèces.

Dans l'optimisation multimodale, des méthodes de formation de niches ont d'abord été utilisées. L'optimisation multimodale consiste à localiser les différents optima d'un problème

uniobjectif donné. Dans les AGs, il est connu que la diversité génétique est perdue à cause du processus de sélection stochastique. L'utilisation d'un AG simple converge généralement vers une seule solution ; ce phénomène est connu sous le nom de dérive génétique (genetic drift) (Goldberg, 1989) [60]. Ainsi, un certain nombre de techniques basées sur les algorithmes évolutionnaires ont été proposées pour la conservation de la diversité.

### **III.3 Classification des méthodes d'optimisation multi-objectif :**

La majorité des travaux de recherche ont été concentrés sur les problèmes à deux critères en utilisant les méthodes exactes tel que le "branch and bound" (Sen et al., 1988) [72], l'algorithme A\* (Stewart and White, 1991) [73], et la programmation dynamique (White, 1982) [74]. Les méthodes exactes sont efficaces pour des problèmes de petites tailles. Pour des problèmes à plus de deux critères et/ou de grandes tailles avec une complexité plus importante, il n'existe pas d'algorithmes exacts efficaces qui donnent une solution dans un temps raisonnable, étant donnée les difficultés simultanées de la complexité NP-complet, et le cadre multicritères des problèmes.

### **III.4 Les méta-heuristiques de l'optimisation multi-objectif :**

Dans cette section, nous présentons quelques méthodes de résolution de POM. Ces méthodes sont des méthodes Pareto fondées sur des métaheuristiques. En effet, nous introduisons les méthodes de recherche locale et les approches évolutionnaires. Nous décrivons aussi deux algorithmes évolutionnaires représentatifs adoptant une approche Pareto pour résoudre les problèmes d'optimisation multiobjectif.

#### **1. Les méthodes de recherche locale**

##### **a. La recherche de voisinage**

La notion de voisinage est utilisée dans toutes les approches de recherche locale. Ces approches utilisent un aspect fondamental qui est la détermination de ce voisinage.

Le voisinage est souvent représenté par une fonction  $N$  qui, à un point  $x$ , associe un ensemble de points  $N(x)$ .

Pour choisir  $N$ , Il existe plusieurs manières, il faut cependant, adapter ce choix au problème, c'est-à-dire choisir la meilleure fonction  $N$  selon le problème considéré.

## b. La descente

La descente est une méthode d'amélioration itérative simple permettant d'atteindre le premier optimum local. Nous pouvons décrire la méthode de la descente pour un problème de minimisation de la manière suivante dans la figure III.2 :

:

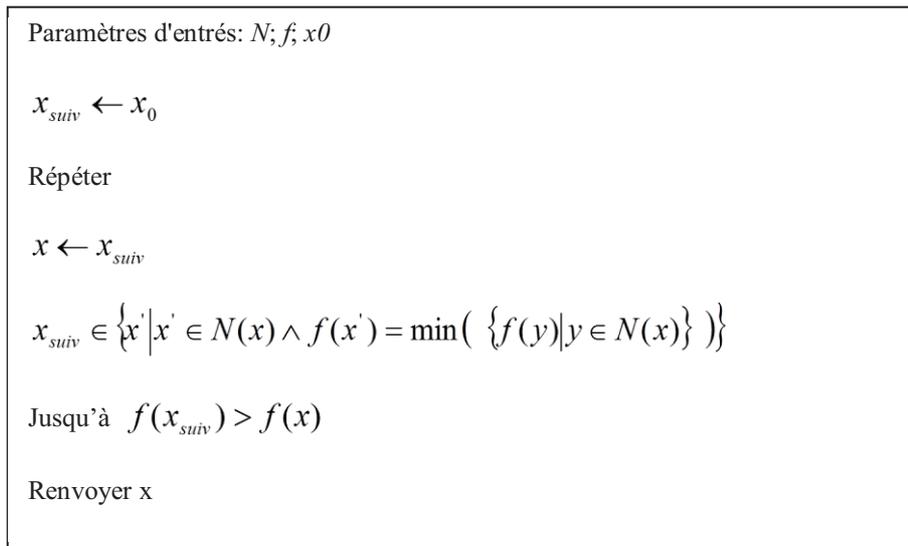


Figure III.2 : Pseudo-code de la méthode de descente.

Où  $N$  est la fonction de voisinage,  $f$  la fonction d'évaluation, et  $x_0$  la configuration initiale formant le point de départ de l'algorithme.

La méthode de descente est largement utilisée dans le domaine de l'optimisation combinatoire. Elle est souvent la première méthode expérimentée sur un nouveau problème. Elle permet, dans un temps de développement assez court, de bien se familiariser avec le problème et de calculer rapidement les premières approximations de l'optimum global.

## 2. Le recuit simulé

Le recuit simulé est un algorithme qui s'inspire du processus de recuit physique. En effet, le processus du recuit simulé répète une procédure itérative qui cherche des configurations de

coût plus faible. Dans cette procédure de recherche des configurations, l'algorithme accepte de manière contrôlée des configurations qui sont moins bonnes pour les solutions intermédiaires trouvées (Kirkpatrick et al., 1983) [81]. Le pseudo code de l'algorithme de recuit simulé est le suivant :

```

Fournir une Solution initiale S

Sopt = S

Fopt = F(Sopt) (F(): Fonction objectif)

Fournir une Température Initiale T0

Tant que (Température T > Température Finale TF) faire
  tant que (Nbitération > NBIterationMax) faire:
    choisir Y dans le Voisinage de (S)
    calculer DF = F(Y) - F(S)
    Si DF < 0 alors
      S = Y
    Si F(S) < F(Sopt) alors
      Sopt = S
    Fopt = F(Sopt)
  Sinon
    Tirer P dans [0; 1]
    Si P ≤ exp(-DF / T) alors S = Y
  T = α * T
Retourner Sopt

```

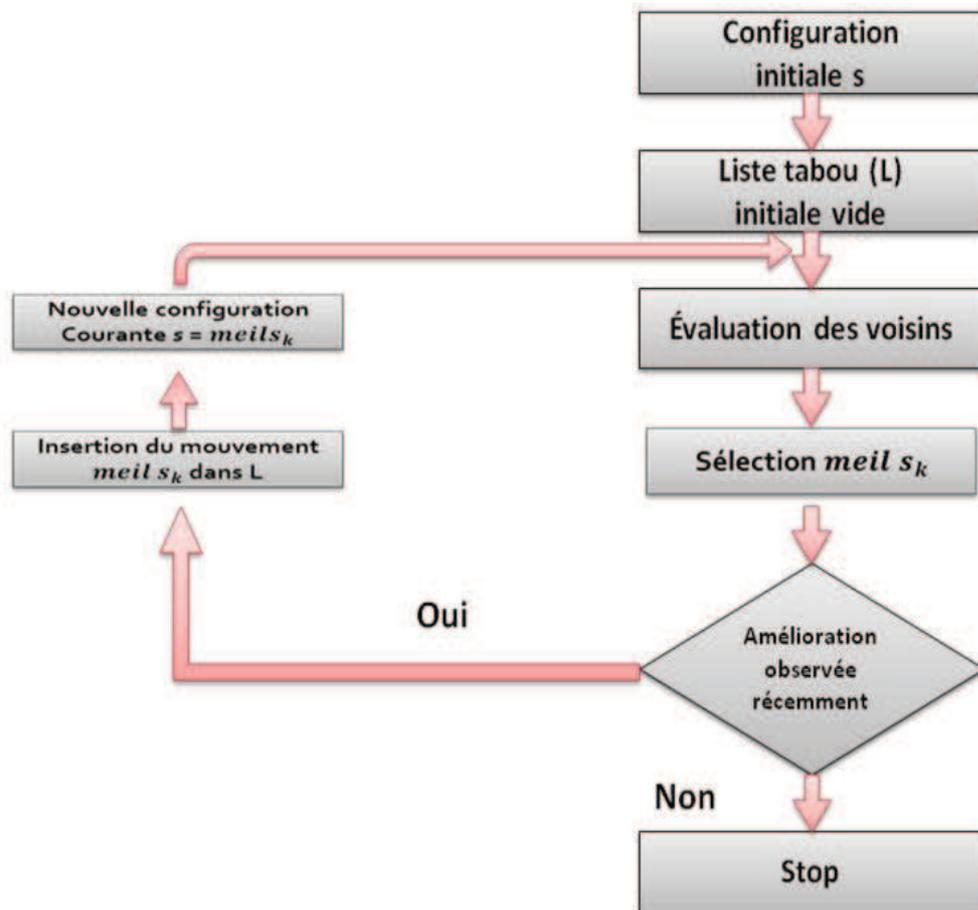
Le recuit simulé a eu un grand succès notamment grâce à des résultats pratiques obtenus sur de nombreux problèmes dans des domaines variés. Des exemples de ces applications sont présentés dans [82] [83] [84].

### 3. La recherche Tabou

En 1986, la recherche Tabou est introduite par Glover (Glover, 1986) [85] et largement détaillé dans (Glover and Laguna, 1997) [86].

La recherche Tabou à un fonctionnement plus proche de la méthode de descente que du recuit simulé. Elle examine un échantillonnage de configurations de  $N(x)$  et retient la meilleure solution  $x_0$  même si  $x_0$  est plus mauvaise que  $x$ . Cependant, cette méthode peut entrainer des cycles. Pour éviter les cycles, on mémorise les  $n$  dernières configurations visitées dans une mémoire à court terme et on interdit de revenir à toute configuration qui en fait déjà partie. Cette mémoire est appelée la liste Tabou. Elle est une des composantes essentielles de la méthode. La liste Tabou nous permet d'éviter les cycles de longueur inférieure à  $n$ .

L'algorithme de la recherche tabou est décrit dans l'organigramme suivant :



L'inconvénient majeur de cette méthode est sa gourmandise en espace mémoire. La solution consiste alors à ne mémoriser que les caractéristiques des configurations par exemple sauvegarder uniquement un attribut de la configuration. Il en résulte que toutes les configurations possèdent cet attribut, y compris celles qui n'ont pas encore été rencontrées,

deviennent elles aussi Tabou. Un mécanisme particulier, appelée l'aspiration, est mis en place pour remédier à ce problème. La fonction d'aspiration la plus simple consiste à enlever le statut Tabou d'une configuration si celle-ci est meilleure que la meilleure configuration déjà trouvée.

Pour améliorer les performances de la méthode Tabou, Il existe aussi d'autres techniques permettant, en particulier, l'intensification et la diversification. En effet, l'intensification permet de se focaliser sur certaines zones de l'espace de recherche en apprenant des propriétés favorables. La diversification a un objectif contraire de l'intensification. Elle cherche à diriger la recherche vers des zones inexplorées, en modifiant par exemple la fonction d'évaluation. L'intensification et la diversification sont donc deux objectifs complémentaires.

#### **4. Les algorithmes évolutionnaires**

Nous pouvons distinguer trois grandes classes d'algorithmes évolutionnaires : les algorithmes génétiques (Holland, 1975 [87]; Goldberg, 1989 [88]), les stratégies d'évolution (Schwefel, 1981) [89] et la programmation évolutive (Fogel, 2000) [90]. Les méthodes évolutionnaires se distinguent par rapport aux autres méta-heuristiques par la possibilité de représenter l'information et par leur façon de faire évoluer la population d'une génération à l'autre. L'algorithme évolutionnaire est composé généralement de trois éléments fondamentaux :

- Une population constituée de plusieurs individus représentant des solutions potentielles (configurations) du problème donné,
- Un mécanisme d'évaluation des individus permettant de mesurer l'adaptation de l'individu à son environnement,
- Un mécanisme d'évolution de la population permettant, grâce à des opérateurs prédéfinis, d'éliminer certains individus et d'en créer des nouveaux.

Dans cette section, nous s'intéressons à la classe des algorithmes génétiques des méthodes évolutionnaires.

##### **a. Les algorithmes génétiques**

Les algorithmes génétiques (AGs) ont été inventés par Holland (Holland, 1975) comme un modèle de méthode adaptative. En effet, les algorithmes génétiques se basent généralement sur un codage de l'information sous forme de chaînes binaires de longueur fixe et d'un ensemble d'opérateurs génétiques : la sélection, la mutation et le croisement. Avec le codage d'un individu, nous appelons ce dernier un chromosome, représentant une configuration du problème.

Un algorithme génétique est un algorithme itératif qui possède une architecture bien définie. Pour résoudre un problème d'optimisation, cet algorithme passe par plusieurs étapes. Nous les présentons dans la figure III.3. En effet, il s'agit des étapes de (Schwefel, 1981) [89]

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

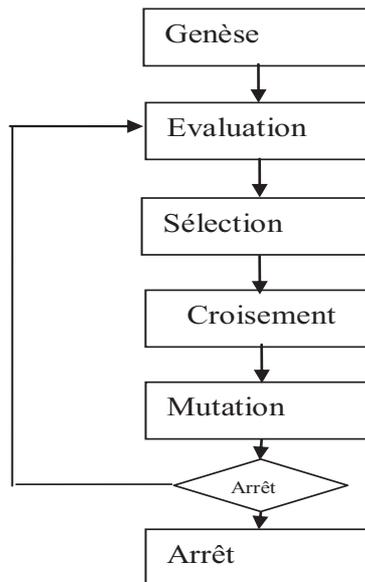


Figure III.3 : Architecture générale d'un algorithme évolutionnaire.

### III.5 Conclusion :

Dans ce chapitre, nous avons présenté quelques méthodes d'optimisation des problèmes multi-objectifs (POM). En effet, nous avons présenté trois classes de méthodes pour la résolution de ces derniers : les approches à base de transformation du problème vers un problème à un seul objectif, les approches non Pareto et les approches Pareto. Nous avons donné une présentation détaillée des approches Pareto vue leur efficacité dans la résolution des POM. Les approches Pareto sont issues des méta-heuristiques pour l'optimisation multi-objectif. Ces méthodes ont montré leur efficacité pour trouver des solutions approchées satisfaisantes pour un grand nombre de problèmes. Cependant, ces méthodes ont aussi leurs défauts, le plus important d'entre eux concernant l'impossibilité de juger de la qualité de l'approximation par rapport au front Pareto optimal.

Dans cette thèse, nous nous intéressons à l'application d'un algorithme de recuit simulé multi-objectif pour la résolution d'un Problème de Transport à la Demande (PTD). Cette méthode, nous a permis de trouver des solutions approchées du front Pareto.

## **Chapitre IV :**

# **Une Approche Centralisée de Résolution basée sur l'algorithme du Recuit Simulé pour le Problème de Transport à la Demande**

### **IV.1. Introduction :**

Un PTD consiste à répondre aux demandes de transport relatives à un ensemble de voyageurs dispersés géographiquement. Chaque demande de transport est modélisée par une requête contenant des informations sur cette dernière. Ces informations sont le nombre de passagers, les points de départ et d'arrivée, ainsi que les intervalles des heures de passages désirés. Le problème du transport à la demande est un système soumis à de nombreuses contraintes et doit répondre à plusieurs attentes. Ces attentes et/ou objectifs peuvent être contradictoires, comme par exemple la réduction du temps de parcours, la réduction du coût engendré, maximisation de la qualité de service. En plus, le PTD est un problème appartenant à la classe NP-difficile (Inge Li, 2006) [91]. Par conséquent, les méthodes exactes ne sont pas capables de résoudre un tel problème dans un temps raisonnable surtout quand la taille du problème est importante. Dans ce cas, nous avons souvent recours à des méthodes qui permettent de trouver une solution approchée en un temps acceptable ; il s'agit des heuristiques et méta-heuristiques. Comme par exemple celles basées sur les algorithmes génétique, recuit simulé, recherche tabou etc.

Dans ce chapitre, nous présentons la modélisation du problème de transport à la demande multicritère. Nous proposons notre contribution pour la résolution du PTD en utilisant l'algorithme du recuit simulé selon deux approches. La première méthode est basée sur une méthode agrégative. Par contre, dans la deuxième nous développons une méthode basée sur le recuit simulé multi-objectif pour trouver des solutions Pareto optimales. Nous détaillons

également les résultats numériques obtenus prouvant l'efficacité de l'approche proposée. Nous finissons ce chapitre en présentant les perspectives pour cette approche.

## IV.2. Formalisme mathématique du PTD :

Le PTD a été modélisé mathématiquement dans plusieurs travaux de recherche. Il est modélisé généralement par un programme mathématique multi-objectif. Dans cette section nous présentons notre modélisation mathématique du PTD. Ce modèle est caractérisé par deux principaux objectifs. Le premier concerne l'objectif économique. Le second s'intéresse à la qualité de service rendue aux voyageurs. A cet effet, nous résolvons un PTD multi-objectif en utilisant la méthode d'agrégation détaillée dans (Boudali et al, 2004) [92]. Dans ce qui suit nous présentons notre formalisation mathématique du problème:

- **Variables du PTD:**

- $n$  : Nombre de demandes de transport.
- $D = \{1, \dots, n\}$  : Ensemble de points de récupération des voyageurs.
- $A = \{n+1, \dots, 2n\}$  : Ensemble de points d'arrivée des voyageurs.
- $M = \{0, \dots, m\}$  : Ensemble de dépôts contenant les véhicules.
- $N = D \cup A \cup M$  : Ensemble de tous les nœuds du graphe  $G(N, Ar)$  où  $Ar$  est l'ensemble des arrêtes du graphe.
- $V = \{0, \dots, v\}$  : Ensemble de véhicules.
- $Q_v$  : Capacité du véhicule  $v$ .
- $q_{vi}$  : Nombre de personnes qui ont été pris par la véhicule  $v$  à la station  $i$  tel que  $i \in D$ .
- $[a_i \ b_i]$  : Fenêtre de temps associée au point de départ  $i$  tel que  $i \in D$ .
- $[a_{i+n} \ b_{i+n}]$  : Fenêtre de temps associée au point d'arrivée  $i+n$  tel que  $i+n \in A$
- $C_{ijv} = C_{ij} \times C_v$  : Coût de transport de  $i$  vers  $j$  avec le véhicule  $v$  tel que  $C_v$  est le coût kilométrique d'utilisation du véhicule.
- $T_{ijv}$  : Durée de transport de  $i$  vers  $j$  avec le véhicule  $v$ .
- $K_{siv}$  : Heure de début de service pour la demande  $i$  avec le véhicule  $v$ .
- $t_{aiv}$  : Heure d'arrivée de la demande  $i$  à destination avec le véhicule  $v$ .
- $NSV_i$  : Le nombre de stations visités par une demande de transport  $i$ .
- $L_{iv}$  : Nombre de voyageurs dans le véhicule après avoir visité  $i$  tel que  $i \in N$
- $X_{ijv}$  : Variable de décision du problème,  $X_{ijv} = 1$  si le véhicule emprunte un chemin direct de  $i$  vers  $j$ , sinon  $X_{ijv} = 0$

- **La fonction objectif**

La fonction objectif globale étant la somme de la fonction économique et celle de la qualité de service ;  $F = ECO + QS$

- **Fonction objectif économique :**

$$ECO = \sum_{i \in N} \sum_{j \in N} \sum_{v \in V} X_{ijv} C_{ijv} \quad (IV.1)$$

- **Fonction objectif qualité de service :**

$$QS = TP + NSV \quad (IV.2)$$

TP : Temps de parcours

NSV : Nombre de Stations Visitées

Tel que :

$$TP = \sum_{i \in D} \sum_{v \in V} (t_{aiv} - K_{siv}) \quad (IV.3)$$

$$NSV = \sum_{i \in D} NSV_i \quad (IV.4)$$

$$\text{Donc } F = \left( \sum_{i \in N} \sum_{j \in N} \sum_{v \in V} X_{ijv} C_{ijv} \right) + \left( \sum_{i \in D} \sum_{v \in V} (t_{aiv} - K_{siv}) \right) + \sum_{i \in D} NSV_i$$

- **Modèle mathématique :**

Dans cette section nous présentons notre modélisation mathématique du PTD ainsi que la description des contraintes formant le modèle mathématique

$$\text{Minimiser } F (X_{ijv}) \quad (IV.5)$$

⇒ La fonction objectif du problème de transport à la demande avec prise en compte de la qualité de service rendue aux voyageurs,

Sujet à

$$\sum_{v \in V} \sum_{j \in N} X_{ijv} = 1 \quad \forall i \in D \quad (\text{IV.6})$$

⇒ Un véhicule  $v$  quitte une seule fois la station de départ  $i$ ,

$$\sum_{j \in D \cup A} X_{ijv} - \sum_{j \in D \cup A} X_{j,n+i,v} = 0 \quad \forall v \in V, \forall i \in D \quad (\text{IV.7})$$

⇒ Si un véhicule  $v$  sert une demande de départ  $i$ , il arrive certainement au point d'arrivée  $i+n$ ,

$$\sum_{i \in N} X_{ijv} - \sum_{i \in N} X_{jiv} = 0 \quad \forall j \in A \cup D, \forall v \in V \quad (\text{IV.8})$$

⇒ Si un véhicule  $v$  arrive à un nœud  $i$ , il quitte ce dernier,

$$X_{ijv}(T_{siv} + T_{ijv} - T_{sjv}) \leq 0 \quad \forall v \in V, (i, j) \in Ar \quad (\text{IV.9})$$

⇒ Un véhicule  $v$  commence le service en  $j$  après avoir fini le service en  $i$  et emprunté l'arête  $(i, j)$ ,

$$a_i \leq T_{siv} \leq b_i \quad \forall i \in N, v \in V \quad (\text{IV.10})$$

⇒ Un véhicule  $v$  doit respecter la fenêtre de temps d'un nœud  $i$  afin de le servir à temps,

$$a_{i+n} \leq T_{aiv} \leq b_{i+n} \quad \forall i \in N, v \in V \quad (\text{IV.11})$$

⇒ Un véhicule  $v$  doit respecter la fenêtre de temps d'un nœud d'arrivée  $i+n$ ,

$$X_{ijv}(L_{iv} + q_{vj} - L_{jv}) = 0 \quad \forall v \in V, (i, j) \in Ar \quad (\text{IV.12})$$

⇒ Conservation du nombre de personnes transmis sur un chemin  $(i, j)$  par un véhicule  $v$ ,

$$q_{iv} \leq L_{iv} \leq Q_v \quad \forall i \in D, v \in V \quad (\text{IV.13})$$

⇒ Le nombre de personnes présentes dans le véhicule  $v$  après avoir visité  $i$  est supérieur à celui collecté en  $i$  et inférieur à la capacité maximale du véhicule.

$$0 \leq L_{n+i,v} \leq Q_v - q_{iv} \quad \forall n+i \in D, v \in V \quad (\text{IV.14})$$

⇒ Le nombre de personnes présentes dans le véhicule  $v$  après avoir visité  $i+n$  est inférieur ou égale à la capacité du véhicule moins le nombre de personnes prises dans la demande  $i$ ,

$$L_{mv} = 0 \quad \forall m \in M, v \in V \quad (\text{IV.15})$$

⇒ Un véhicule  $v$  sort vide d'un dépôt  $m$ ,

$$X_{ijv} \in [0,1] \quad \forall v \in V, (i,j) \in Ar \quad (\text{IV.16})$$

⇒ La variable binaire du problème

### **IV.3. Une approche agrégative basée sur l'algorithme de recuit simulé pour la résolution du PTD statique :**

Pour des problèmes NP-complets d'optimisation (comme le problème de transport à la demande), ne nous disposons pas d'algorithmes polynomiaux permettant leur résolution d'une façon optimale dans un délai raisonnable. L'utilisation d'une méthode approchée s'avère quasiment obligatoire. Dans ce chapitre, nous avons appliqué l'algorithme de recuit simulé pour résoudre le PTD. L'algorithme du recuit simulé est une méthode issue du processus utilisé en métallurgie. Ce processus alterne des cycles de refroidissement lent et de réchauffage (recuit) qui tendent de minimiser l'énergie du matériau. Il est classé parmi les méthodes de recherche opérant localement. En effet, il permet d'effectuer des changements sur la solution courante pour sortir d'un optimum locale. Il a été proposé comme algorithme d'optimisation par (Kirkpatrick et al, 1983) [81]. Pour résoudre le PTD avec l'algorithme de recuit simulé, il faut définir la méthode de génération de la solution initiale " $S$ ", la méthode de recherche du voisinage " $N(S)$ " et la fonction objectif à optimiser " $f(S)$ ".

#### **1. L'algorithme du recuit simulé :**

Le recuit est le processus par lequel un refroidissement lent du métal produit la cristallisation à faible niveau d'énergie, tandis que le refroidissement rapide produit une mauvaise cristallisation. La procédure d'optimisation du recuit simulé imite la cristallisation pour atteindre un minimum ou un maximum globale par le refroidissement lent. Le recuit simulé

est classée parmi les méthodes de recherche locale, il peut apporter des modifications à la solution actuelle pour sortir d'un optimum local. En général, la réduction rapide de la température élevée (trempe) ne permet pas d'obtenir cet état cristallin. En revanche, le matériel doit être refroidi lentement (recuit) pour obtenir l'état cristallin. En effet, l'algorithme de recuit simulé est composé par la recherche locale qui tente de trouver en permanence une meilleure solution que celle en cours pendant le processus de recherche. L'algorithme est présenté dans la section III-4, paragraphe 2.

#### a. La solution initiale

Dans notre approche, la solution initiale a été générée par une heuristique de distribution. Cette heuristique permet l'affectation de  $m$  véhicules à un ensemble de  $n$  demandes de transport. Cette méthode de recherche de solution initiale se base sur le principe de l'affectation aléatoire des véhicules aux demandes de transport. Après avoir affecté les véhicules aux demandes, ces derniers seront ordonnés aléatoirement dans la tournée de chaque véhicule.

L'heuristique de distribution proposée dans notre approche est présentée dans la figure IV.1 :

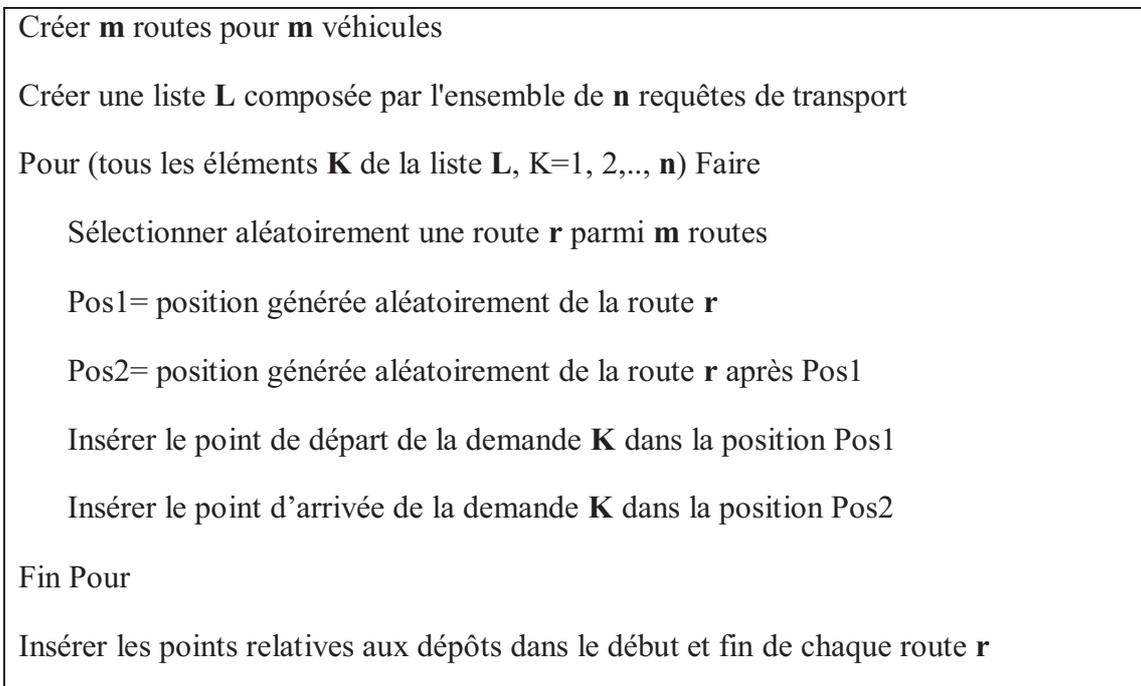


Figure IV.1 : Heuristique de distribution

Dans la figure IV.2, nous présentons un exemple de composition de la solution initiale. Nous désignons par « i » le point de ramassage de la demande de transport «i» et « i + n » désigne le point de livraison. Notre heuristique de distribution peut produire une violation de certaines contraintes comme la capacité du véhicule et le respect des fenêtres de temps associés aux points de départ et de livraison. En effet, dans notre approche nous partons d'une solution initiale qui ne satisfait pas nécessairement les contraintes du problème. Ces contraintes seront satisfaites par le processus de recherche des solutions de voisinage pour la solution initiale en utilisant la structure de voisinage.

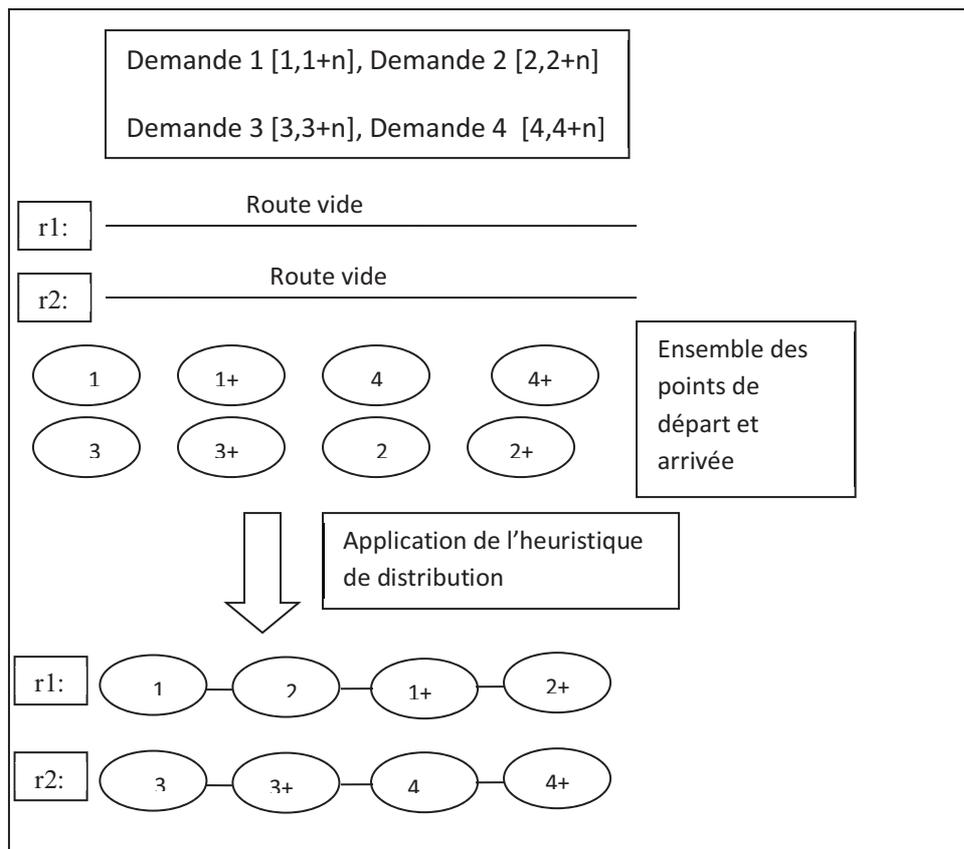


Figure IV.2 : Exemple de composition de la solution initiale

## b. La méthode de recherche de voisinage

Pour améliorer la solution du PTD, il faut effectuer des changements sur la solution courante. Ces changements sont effectués par une méthode de recherche de voisinage. Dans l'algorithme de recuit simulé, on peut accepter des changements même s'ils dégradent la qualité de la solution. La méthode de recherche de voisinage est issue du travail présenté par (Gendreau et al, 2006) [93]. Cette méthode est décrite comme suit :

(Soit  $r_i$  et  $r_j$  deux routes qui servent les demandes  $i$  et  $j$ )

1. sélectionner les deux demandes  $i$  et  $j$ ;
2. éjecter + (supprimer) la demande  $j$  de la route  $r_j$ ;
3. insérer le point de départ de la demande  $j$  dans la meilleure position (coût minimum) de la route  $r_i$ ;
4. insérer le point d'arrivée de la demande  $j$  après le point de départ de cette dernière dans la meilleure position (coût minimum) de la route  $r_i$ ;
5. éjecter + (supprimer) la demande  $i$  de la route  $r_i$ ;
6. insérer le point de départ de la demande  $i$  dans la meilleure position (coût minimum) de la route  $r_j$ ;
7. insérer le point d'arrivée de la demande  $i$  après le point de départ de cette dernière dans la meilleure position (coût minimum) de la route  $r_j$ ;

Dans notre structure de voisinage, nous avons sept étapes et la complexité de cette structure est la suivante: dans l'étape 1, il y a une complexité de  $O(n^2)$ . L'étape 2 est effectuée en temps constant, puisque l'impact de la suppression de chaque demande de son propre itinéraire est calculé qu'une seule fois au début de l'évaluation de voisinage. Cela se fait en une complexité de  $O(n^2)$ . En effet, l'insertion des lieux de ramassage et de livraison ne sont pas faites simultanément, mais plutôt de façon séquentielle. Autrement dit, le meilleur endroit d'insertion pour la collecte est d'abord identifié. Ensuite, le meilleur endroit d'insertion pour la livraison est choisi (après l'insertion du point de ramassage). Dans la figure IV.3, nous présentons un exemple de l'évolution de la solution par la structure de voisinage utilisée dans notre approche. Chaque point de ramassage pour une demande de " $i$ " est modélisé par le nombre " $i$ " et le point de livraison par le nombre " $n + i$ ".

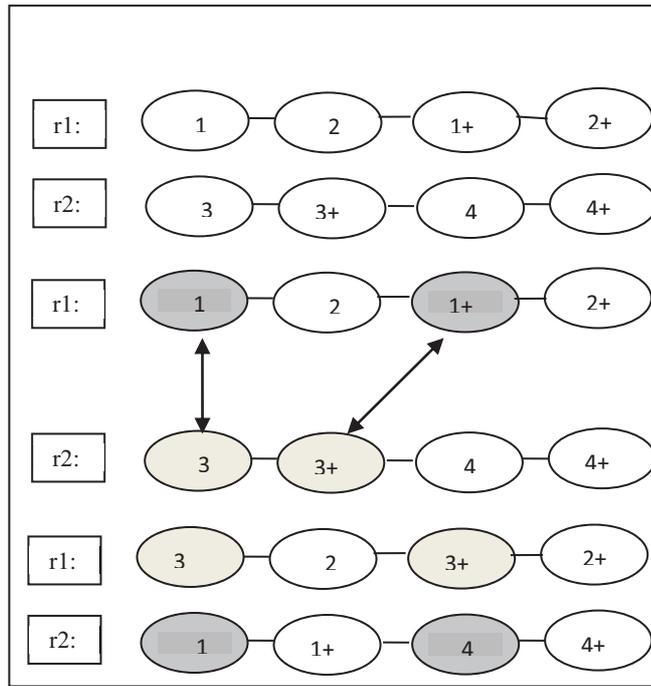


Figure IV.3 : La structure de voisinage

### c. La méthode de planification temporelle

L'heuristique de distribution et la structure de voisinage sont utilisées pour programmer les itinéraires des véhicules. Mais la programmation temporelle de ces routes de véhicules doit encore être effectuée pour déterminer l'heure d'arrivée dans les lieux (point de départ et d'arrivée). En effet, une autre heuristique de programmation ou une méthode de planification temporelle est utilisée. L'heuristique de programmation est adaptée de celle présentée dans (Cordeau et Laporte, 2003). Cette heuristique tente de réduire les violations des fenêtres de temps, la durée des parcours. Considérons une route  $k = (v_0, \dots, v_i, \dots, v_q)$  effectuée par le véhicule  $k$  ( $\forall k \in K$ ), où  $v_0$  et  $v_q$  représentent le dépôt. On note par  $A_i$  l'heure d'arrivée d'un véhicule au sommet  $v_i$ , par  $B_i \geq \max \{a_i, A_i\}$  le début du service au sommet  $v_i$ , et par  $D_i$  l'heure de départ du sommet  $v_i$ . Nous supposons ici que l'attente à tout point  $v_i$  est autorisée avant que le service démarre, mais n'est plus autorisée après la fin du service. Le temps d'attente de la demande  $i$  est  $W_i = B_i - A_i$ . Le temps de parcours relatif à la demande  $i$  est calculé par  $L_i = B_{i+n} - D_i$ , avec  $i+n$  le point de livraison de la demande  $i$ . Ainsi,  $D_0 = a_0$  et  $B_i = \max \{A_i, a_i\}$  et  $a_i$  est la borne inférieure de la fenêtre de temps d'une demande  $i$  est optimale en termes de minimisation des violations de fenêtre de temps parce que le véhicule quitte le dépôt le plus tôt possible et le service à chaque sommet commence aussi tôt que

possible. Pour toute route  $v_k$ , le délai  $F_i$  est calculé tel qu'il est présenté dans les équations (IV.17) et (IV.18).

$$F_i = \min_{i \leq j \leq q} \left\{ b_j - \left( B_j - \sum_{i \leq p \leq j} W_p \right) \right\} \quad (\text{IV.17})$$

$$= \min_{i \leq j \leq q} \left\{ \sum_{i \leq p \leq j} W_p + (b_j - B_j) \right\} \quad (\text{IV.18})$$

## 2. Résultats Numériques

Dans cette contribution, nous avons choisi de tester notre approche sur les données présentés dans le benchmark de (Cordeau et Laporte, 2003) [34] (téléchargeable du net à partir du site suivant: « <http://www.hec.ca/chairedistributique/data/darp/>»). Nous pouvons comparer nos résultats avec ceux des travaux déjà publiés. Nous trouvons dans ce benchmark 20 instances varies en terme de requêtes de transport et en terme de nombre de véhicules mis en jeux. Nous trouvons dans ces instances des problèmes de taille allant de 24 à 144 demandes. Le nombre de véhicules utilisés pour servir les demandes de transport varie de 3 à 13 véhicules. Ce benchmark est formé par des véhicules homogènes.

Dans (Mauri et al, 2006) [95], l'algorithme du recuit simulé a été appliqué au problème avec des méthodes de recherche de solution initiale et du voisinage différents de ceux appliquées dans notre approche. Nous procéderons à la comparaison de nos résultats avec ceux obtenus par Mauri et al. En effet, dans cette contribution nous avons testé notre approche sur 4 instances du benchmark présenté par (Cordeau et Laporte, 2003). Les instances choisies sont les suivants : R1a (24 demandes et 3 véhicules), R5a (120 demandes et 11 véhicules), R1b (144 demandes et 13 véhicules) et en fin R4b (108 demandes et 8 véhicules).

Dans le tableau IV.1, nous donnons les valeurs des paramètres de l'algorithme de recuit simulé utilisé dans les tests effectués dans la résolution du problème. Ces valeurs sont choisies après plusieurs tests appliqués aux différentes instances de ce benchmark. Les valeurs des poids associées à la fonction multi-objectifs "F" déjà détaillée dans la section IV.2 sont respectivement : [0.7, 0.2, 0.1].

Paramètre	Valeur
T0	3000
$\alpha$	0,975
TF	0.001
NBIterationMax	800

Tableau IV.1: Paramètres de l'algorithme du recuit simulé

Dans le tableau IV. 2, nous présentons les résultats obtenus par notre approche appliquée sur les instances motionnés ci-dessus.

Instance	Distance parcourue	Durée de la route (min)	Temps d'exécution (min)
R1a	260,85	982,06	0,18
R5a	989,56	3537,19	0,29
R1b	308,89	877,62	0,24
R5b	1006,52	2947,38	0,28
<b>Totale</b>	<b>2560,82</b>	<b>8344,25</b>	<b>0,99</b>

Tableau IV.2: Les résultats obtenus par notre approche (Zidi et al, 2010)<sup>a</sup> [113]

Dans le tableau IV.3, nous présentons les résultats obtenus par (Mauri et al, 2006).

Instance	Distance parcourue	Durée de la route (min)	Temps d'exécution (min)
R1a	252,79	831,3	1,00
R5a	1085,45	3920,25	1,79
R1b	251,85	738,42	0,92
R5b	1010,09	3654,02	1,95
<b>Totale</b>	<b>2600,18</b>	<b>9143,99</b>	<b>5,66</b>

Tableau IV.3: Les résultats obtenus par (Mauri et al, 2006)

Dans la figure IV.4, nous montrons la comparaison de nos résultats en termes de distance parcourue avec les résultats obtenus par (Mauri et al, 2006).

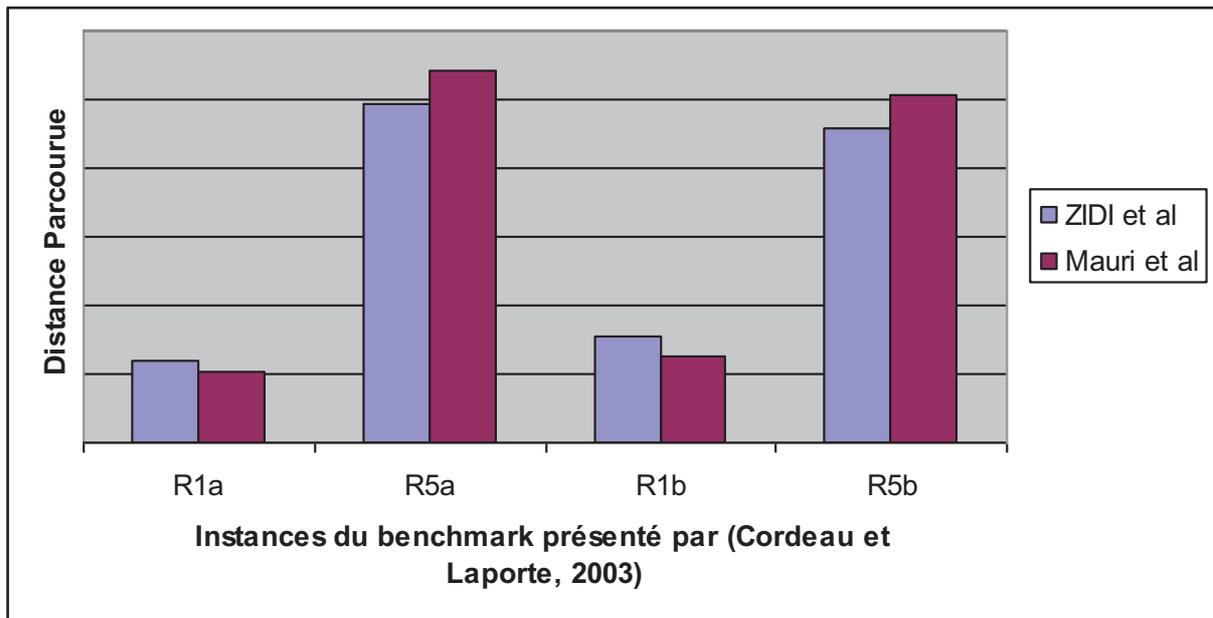


Figure IV.4 : Comparaison de nos résultats avec ceux obtenus par (Mauri et al, 2006)

Après avoir présenté nos résultats et les résultats obtenus dans (Mauri et al, 2006) [95], nous remarquons que dans les instances « R5a » et « R5b » notre approche est plus efficace que l'approche implémentée dans (Mauri et al, 2006). Par contre dans les instances « R1a » « R1b » l'approche développée par (Mauri et al, 2006) est plus efficace que le nôtre. En conclusion, notre approche montre des résultats meilleurs que ceux de (Mauri et al, 2006) pour les instances de grande taille.

Nous avons testé notre approche sur un ordinateur portable Dell B14DEE640C ayant un processeur Intel Core 2 Duo 2Ghz et une mémoire vive RAM de 2GO. Notre approche est implémentée en langage JAVA. Les résultats présentés dans (Mauri et al, 2006), sont obtenus par des tests effectués sur un ordinateur portable Toshiba A10 S127 avec un processeur Intel Celeron 2Ghz et une mémoire vive RAM de 256MO. Mauri et al, ont utilisés le langage C++ dans le développement de leur approche.

#### **IV.4. Une approche Pareto basée sur l'algorithme du Recuit Simulé Multi-Objectif (RSMO) pour la résolution du PTD statique :**

##### **1. L'algorithme du recuit simulé multi-objectif:**

L'algorithme du Recuit Simulé Multi-Objectif (RSMO) représente une méthode d'optimisation multicritère Pareto. En effet, l'algorithme utilise la notion de dominance pour trouver une solution aux POM.

Définition : Le vecteur  $F(U)$  domine un autre vecteur  $F(V)$ , notée  $F(U) < F(V)$  (dans le cas de la minimisation), si et seulement si :

$$f_i(U) \leq f_i(V) \text{ pour tout } i \in \{1, 2, \dots, n\} \quad \text{et il existe } f_j(U) < f_j(V) \text{ pour } j \in \{1, 2, \dots, n\}$$

L'algorithme de RSMO a un inconvénient dans l'optimisation multi-objectif qui se résume dans son incapacité à trouver un ensemble de solutions. Toutefois le RSMO peut faire le même travail en répétant les essais car il converge vers l'optimum global avec une distribution de probabilité uniforme dans l'optimisation mono-objectif. Par contre RSMO a des avantages par rapport aux Algorithmes Evolutionnaires (AE), car il n'a pas besoin d'une grande mémoire pour stocker la population, et l'algorithme RSMO n'utilise pas des algorithmes supplémentaires pour diversifier les solutions pour couvrir une partie étendue de la frontière de Pareto. En effet, RSMO trouve un ensemble de solutions Pareto dans un court laps de temps ce qui est appréciable quand on a besoin d'une réponse rapide, il peut également trouver plus de solutions en répétant la routine d'optimisation.

Nous présentons le pseudo-code de l'algorithme de Recuit Simulé Multi-Objectif (RSMO) dans la figure IV5.

```

1. Pour ( $\alpha$ , Iter-MAX, T0 ,TC) faire
2. Créer (Une solution initiale S en utilisant la méthode de distribution);
3. IterT  $\leftarrow$  0; {le numéro d'itération pour la température T}
4. T  $\leftarrow$  T0; {la température courante}
5. Tant que (T > TC) faire
6.     Tant que (IterT < Iter-MAX) faire
7.         IterT  $\leftarrow$  IterT + 1;
8.         Créer (une solution voisine S' pour S en utilisant la méthode de
voisinage);
9.         Si (C(S') domine C(S)) S  $\leftarrow$  S';
10.        Si non Si ( $x < Pt$ ) S  $\leftarrow$  S';
11.        Si non si (C(S') ne domine pas C(S) et C(S) ne domine pas C( S' )
S  $\leftarrow$  S';
12.        Fin si ;
13.    Fin Tant que
14. T  $\leftarrow$   $\alpha$  * T; IterT  $\leftarrow$  0;
15. Fin tant que
16. Returner (S);

```

Figure IV.5 : Pseudo-code de l'algorithme de Recuit Simulé Multi-Objectif (RSMO)

$0 < \alpha < 1$  est une vitesse de refroidissement, Iter-MAX: le nombre d'itérations maximum pour chaque température, T0: la température initiale, TC: la température finale, C: le vecteur objectif à optimiser, Pt: La probabilité de transition de la solution S au voisinage S'. Les règles de transition générales tels que la métropole ou de la méthode logistique ne peut pas être appliqué directement sur les problèmes multi-objectif car ils ne supportent que la fonction de coût scalaire. La règle de transition proposée dans cette thèse est très similaire à la méthode de Metropolis, sauf qu'ils ont utilisé un critère de coût différent pour la fonction de coût mono-objectif. La probabilité de transition de la solution i à la solution j est :

$$Pt(i, j) = \exp(c(i, j)) / T \quad (IV.18)$$

Avec  $c(i, j)$  est le critère de coût pour la transition de la solution de i vers j, et T est la température de recuit. Six critères sont proposés pour le recuit simulé multi-objectif (RSMO). Ces critères sont :

- Critère de coût minimum :

$$c(i, j) = \min_k (c_k(j) - c_k(i))$$

Où  $c_k(i)$  est la  $k^{\text{ième}}$  valeur de coût dans le vecteur objectif de l'état  $i$

- Critère de coût maximum :

$$c(i, j) = \max_k (c_k(j) - c_k(i))$$

- Critère de coût aléatoire :

$$c(i, j) = \sum_{k=1}^D a_k (c_k(j) - c_k(i))$$

Où  $D$  est la dimension du vecteur objectif et  $a_k$  est une variable aléatoire de distribution de probabilité uniforme.

- Critère de coût autonome :

$$c(i, j) = \sum_{k=1}^D c_k(i) \quad \text{ou} \quad \sum_{k=1}^D c_k(j)$$

- Critère de coût moyen :

$$c(i, j) = \frac{\sum_{k=1}^D (c_k(j) - c_k(i))}{D}$$

- Critère de coût fixe :

$$c(i, j) = \text{valeur fixe}$$

Dans ce travail, nous utilisons le critère du coût moyen, car nous avons testé les six critères ci-dessus et nous avons constaté que le critère moyen donne de bonnes performances.

L'algorithme de RSMO de cette recherche est décrit comme suit:  $S$  représente la position de recherche en cours (ou de l'état actuel dans un problème de recherche à états finis) et  $T$  est le paramètre de température qui est progressivement diminué au cours de l'exécution de l'algorithme. Une nouvelle solution  $S'$  est générée par la fonction  $N(S)$ , son coût est évalué et comparé au coût de la solution actuelle. Si le test de la domination est concluant, la nouvelle solution est acceptée même si elle n'est pas bonne (ce qui signifie que la nouvelle solution est dominée par la solution actuelle, ceci permettra éventuellement de sortir d'un optimum local). Elle est acceptée avec une certaine probabilité dite 'probabilité d'acceptation'. Lorsque

la nouvelle solution a le même niveau de valeur que la solution actuelle, dans ce cas, soit on passe à la nouvelle solution ou on garde la solution courante. En effet, dans (Suman et Kumar, 2006) [94] les auteures ont montré que le déplacement vers le nouvel état est meilleur que de rester dans l'état actuel. Alors, quand il n'y a pas de supériorité entre l'état actuel et l'état suivant, le nouvel état est accepté et remplace l'état actuel, car le déplacement dans la situation de non-dominé contribue à accroître la performance des solutions.

## 2. Résultats Numériques

Les meilleurs résultats obtenus par notre approche de RSMO (tableau IV.2) sont comparés à ceux obtenus par (Claudio et al, 2009) [35] et (Cordeau et Laporte, 2003) [34]. Dans le travail de (Claudio et al, 2009), les auteurs ont appliqué l'algorithme génétique (AG) pour résoudre le PTD. Alors que dans le travail de (Cordeau et Laporte, 2003) c'est l'algorithme de recherche tabou qui est appliqué pour résoudre le problème concerné. En effet, il est important de mentionner que, dans (Claudio et al, 2009), les auteures considèrent une restriction dans les solutions trouvées ce qui n'est pas le cas dans notre approche. Cette restriction est le fait de considérer fenêtres de temps souples (des fenêtres de temps qui peuvent être violées) dans la résolution du PTD.

Dans le tableau IV. 4, nous présentons les résultats obtenus par notre approche et dans les tableaux IV.5 et IV.6 nous montrons les résultats obtenus dans les travaux de recherches mentionnées précédemment. Puisque les modèles n'ont pas les mêmes caractéristiques, la comparaison a été faite d'une part sur la durée totale de la route et d'autre part sur la durée de voyage d'un client (durée total y compris le temps d'attente) qui est associé à la qualité de service offerte.

Instance	Durée de la route (min)	Durée du voyage (min)	Temps d'exécution (min)
Pr01	1020,43	582,06	0,98
Pr02	1808,99	820,19	4,70
Pr03	2730,70	3347,64	11,07
Pr05	4020,75	2895,37	9,61
Pr11	1014,38	604,12	0,88
Pr12	1462,65	743,14	4,71
Pr15	4088,85	3102,19	15,34
Pr16	4652,64	2300,06	18,90

Pr17	1407,56	904,58	2,32
Pr019	3415,79	2913,36	12,03

Tableau IV.4: les résultats obtenu par l'algorithme de RSMO(Zidi et al, 2011<sup>a</sup>)[115]

Instance	Durée de la route (min)	Durée du voyage (min)	Temps d'exécution (min)
Pr01	955,25	524,59	1,36
Pr02	1839,06	838,41	4,08
Pr03	2787,18	1597,95	7,96
Pr05	4068,05	2935,48	18,43
Pr11	902,18	449,91	1,58
Pr12	1503,34	744,93	4,49
Pr15	4057,08	3152,67	22,09
Pr16	4658,64	2348,48	17,48
Pr17	1223,68	612,40	3,13
Pr019	3427,06	2515,53	25,43

Tableau IV.5 : les résultats obtenus par l'algorithme génétique (Claudio et al, 2009)

Instance	Durée de la route (min)	Durée du voyage (min)	Temps d'exécution (min)
Pr01	881	1095	1,9
Pr02	1985	1977	8,06
Pr03	2579	3587	17,18
Pr05	3870	6154	46,24
Pr11	965	1042	1,93
Pr12	1565	2393	8,29
Pr15	3596	6105	54,33
Pr16	4072	7347	73,7
Pr17	1097	1762	4,23
Pr019	3249	5581	51,28

Tableau IV.6 : Les résultats obtenus par l'algorithme de recherche tabou (Cordeau & Laporte, 2003)

Après avoir présenté nos résultats et ceux obtenus dans (Claudio et al, 2009) et (Cordeau et Laporte, 2003), nous notons que dans la majorité des cas, notre approche est plus efficace que l'approche mise en œuvre dans (Claudio et al, 2009). Les valeurs énoncées dans le tableau des

résultats présentés ci-dessus, montrent bien que notre approche est meilleure que (Claudio et al, 2009). En effet, lorsque nous nous sommes focalisé sur la durée de route, notre approche RSMO donne de meilleurs résultats que l'Algorithme Génétique (AG) dans 6 cas sur 10 et uniquement dans 2 cas pour la Recherche Tabou (RT) (surligné en vert). Par contre concernant la durée du voyage, notre approche RSMO est meilleure que les deux autres pour toutes les instances testées.

Nous présentons, dans la figure IV.6 la représentation graphique de la solution d'une instance de 48 demandes de transport et 5 véhicules. Cette solution est présentée en 5 parties, chacune montre l'itinéraire d'un véhicule. Le dépôt est marqué en rouge et les autres nœuds en vert. L'itinéraire du véhicule est présenté par des flèches.

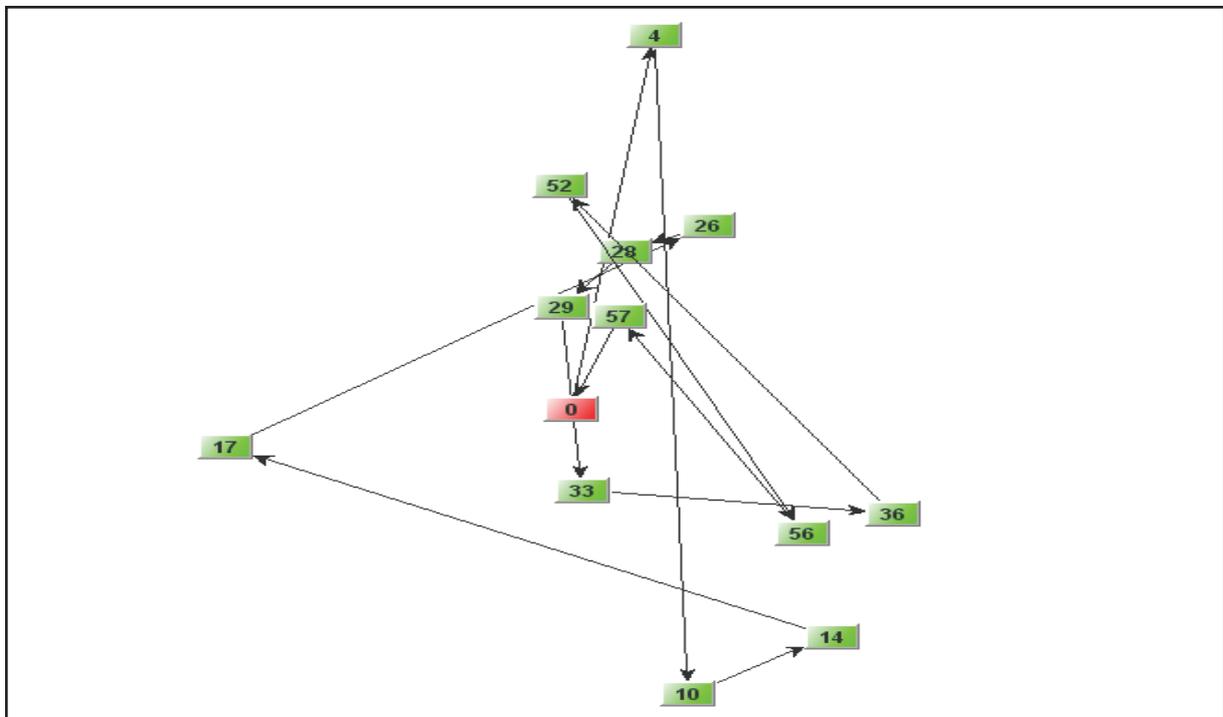


Figure IV.6 (a) : Itinéraire du véhicule numéro 1

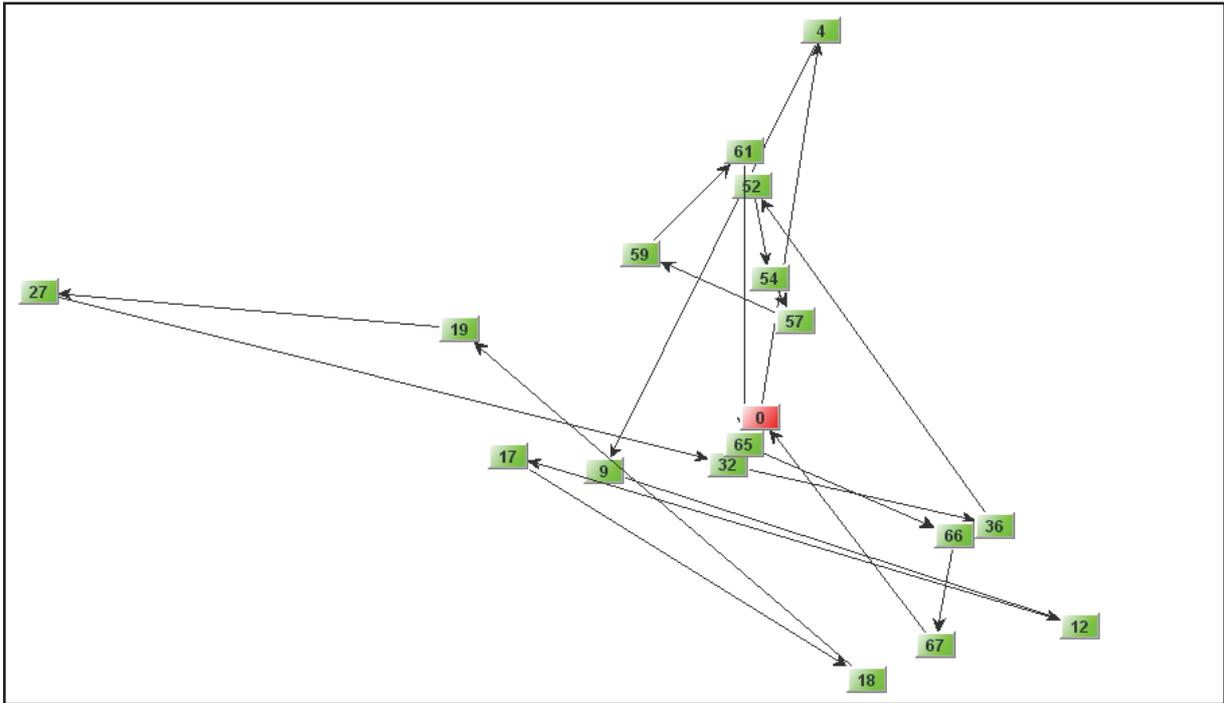


Figure IV.6 (b) : Itinéraire du véhicule numéro 2

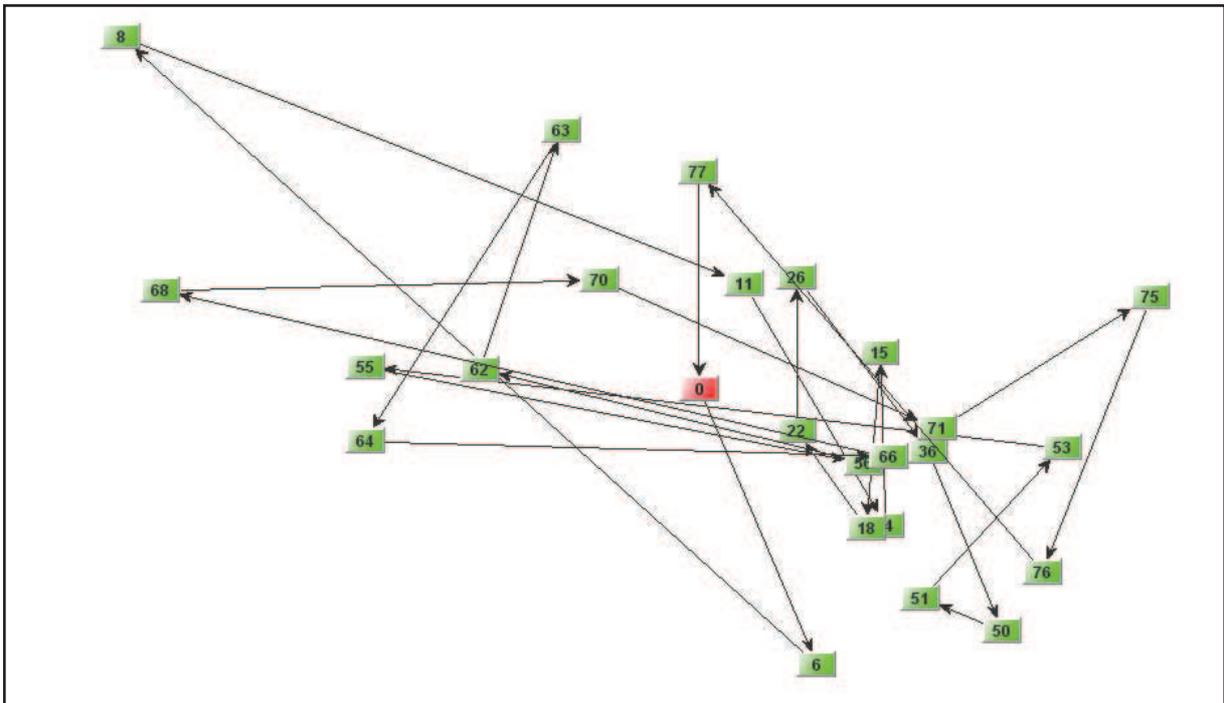


Figure IV.6 (c) : Itinéraire du véhicule numéro 3

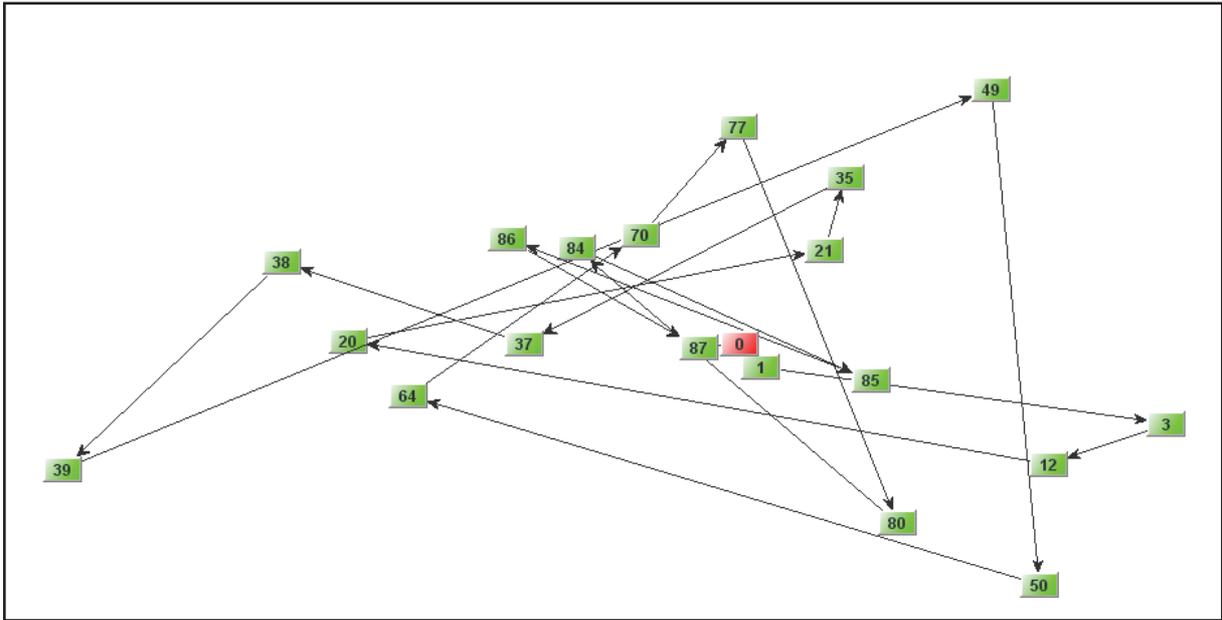


Figure IV.6 (d) : Itinéraire du véhicule numéro 4

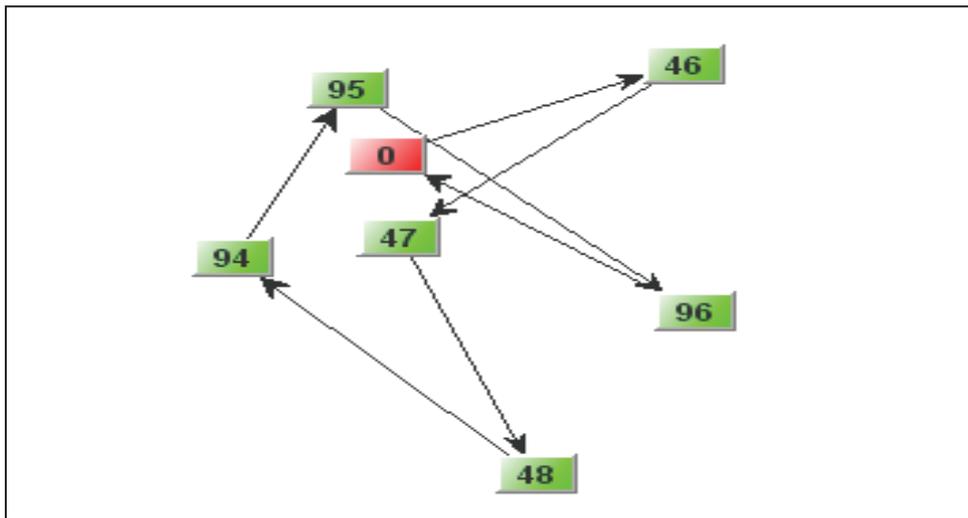


Figure IV.6 (e) : Itinéraire du véhicule numéro 5

Comme indiqué dans les tableaux ci-dessus, la mise en œuvre de notre RSMO présente de meilleurs résultats que ceux obtenus dans (Claudio et al, 2009) pour la durée de voyage. Ceci est principalement dû à l'utilisation des fenêtres temps exigeantes (hard window-time) et l'utilisation du concept de dominance pour l'optimisation multi-objectif. Mais notre approche

RSMO montre de moins bons résultats que ceux de RT (Cordeau et Laporte, 2003) pour la durée de la route. En effet, ceci est probablement dû à l'utilisation d'une méthode d'agrégation avec un grand poids attribué à l'objectif durée de la route pour résoudre le PTD multi-Objectif dans la mise en œuvre de la RT.

En se concentrant sur la durée du trajet présentée, notre solution a montré des temps égaux à ceux de l'algorithme génétique de Claudio et al dans 5 fois sur 10. Nous pensons que ceci est dû à l'utilisation de fenêtres de temps souples dans la mise en œuvre de l'AG.

Dans la figure IV.7 et IV.8, nous présentons une comparaison de notre approche avec l'Algorithme Génétique (AG) et la Recherche Tabou (RT) à l'aide des histogrammes

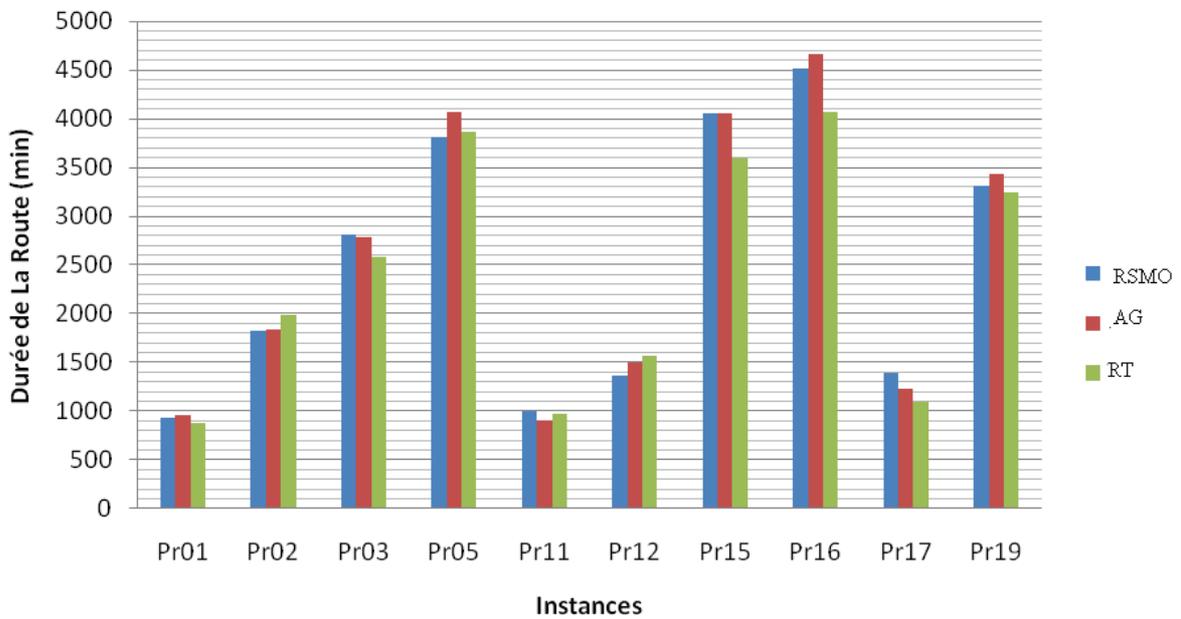


Figure IV.7 : Comparaison sur la durée de la route (RSMO, AG et RT)

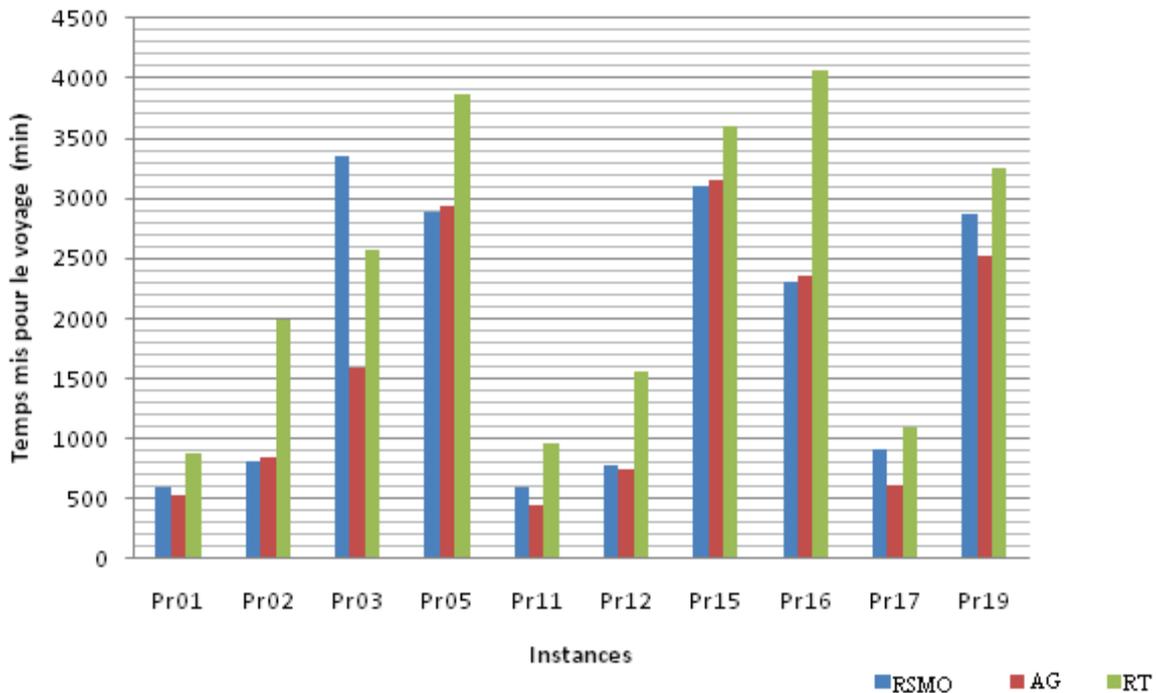


Figure IV.8 : Comparaison sur le temps mis pour le voyage (RSMO, AG et RT)

Il est intéressant de mentionner que les tests ont été effectués en utilisant un ordinateur portable Dell B14DEE640C avec Intel Core 2 Duo 2,0 GHz comme processeur et 2 Go de mémoire RAM. La mise en œuvre de notre travail a été développée en langage Java, tandis que dans (Claudio et al, 2009), les tests ont été faits sur un Intel Pentium 2,66 GHz. Les tests de Cordeau et Laporte ont été faits en utilisant un Intel Celeron 2,0 GHz.

## IV.5. Une approche Pareto basée sur l’algorithme de recuit simulé multi-objectif pour la résolution du PTD dynamique :

### 1. Le PTD dynamique

En réalité le transport des passagers ou la mise en place d’un service de transport à la demande est sujet à des variations ; le problème est donc dynamique parce que la compagnie offrant le service de TAD reçoit en permanence de nouveaux appels de transport au cours du temps. Pour chaque nouvelle demande, une décision doit être prise rapidement afin de l’affecter à un véhicule particulier. Par la suite un ordonnancement au sein de la route doit être prévu pour chaque véhicule. Dans cette thèse, la dynamique du problème de transport à la demande est décrite comme suit : tous les clients sollicitant le service de transport à la

demande doivent définir leur lieu de départ et d'arrivé, ainsi que les horaires de voyages préférées. Quand une nouvelle demande de transport est reçue par le centre de répartition de la compagnie offrant le service de TAD, certaines demandes de transports précédents qui ont été déjà satisfaites seront par conséquent ré-planifiées.

Dans notre travail, il s'agit d'affecter en temps réel chaque nouvelle demande de transport à l'un des véhicules disponibles et d'élaborer un nouvel itinéraire. Nous supposons que nous avons un seul dépôt. En début de journée tous les véhicules se trouvent dans le dépôt, et la capacité des véhicules est supérieur à un. On admet aussi que tous les véhicules desservant les passagers sont du même type et ont la même capacité.

Dans notre environnement de simulation, les demandes de transports doivent être reçues avant la date limite fixée pour le jour de départ du service jour. Toutefois, ceux qui sont reçues après la date limite, peuvent être conservées pour le lendemain. Ainsi, il peut y avoir un certain nombre de demandes en instance ou "statique" (pour lesquels une solution pourrait être construite en avance). La dynamique du système provient d'une source unique, à savoir l'arrivée de nouvelles demandes de transport. En effet, il n'y a pas d'autres sources d'incertitudes associées aux demandes des clients, comme les annulations ou des informations erronées. De plus, les temps de voyage sont connus avec certitude, aucune perturbation inattendue n'est supposée, comme la congestion soudaine sur le réseau de transport causée par un accident, la panne de véhicule, etc.

Dans la figure IV.9, nous donnons un exemple de situation pour un véhicule dans le contexte dynamique, où  $+i$  et  $-i$  représentent respectivement les points de départ et d'arrivé du client  $i$ . A  $t = t_0$ , le véhicule se trouve au point  $X$  et se dirige vers son point suivant de service. Son itinéraire actuel prévu est le suivant : ramassage du client 1, ramassage du client 2, livraison du client 2, livraison du client 1. En supposant qu'un appel du client 3 arrive, le problème est donc d'insérer les nouveaux points de départ et d'arrivé de la demande 3 sur la route en minimisant les objectifs du PTD.

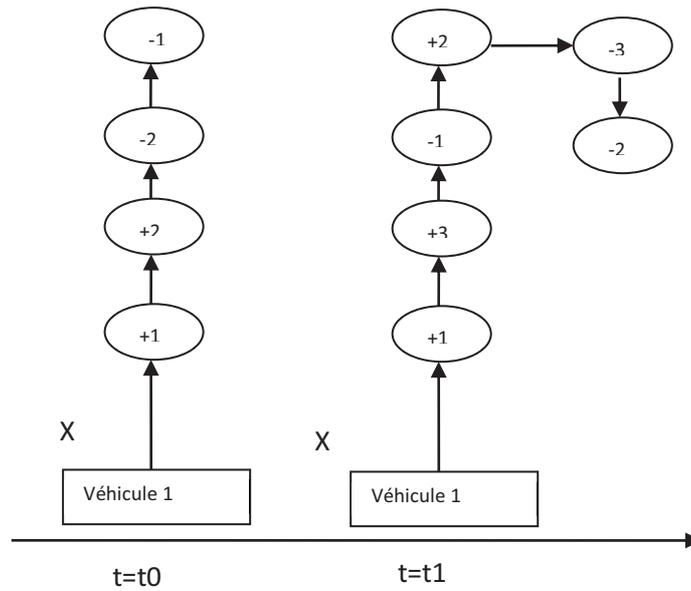


Figure IV.9: Le problème de transport à la demande dynamique

### 1.1. Résolution du PTD dynamique avec le recuit simulé multi-objectif

Dans la résolution du PTD dynamique, nous appliquons l'algorithme de recuit simulé multi-objectif déjà détaillé dans la section (IV.4.1). En effet, dans cet algorithme nous utilisons les mêmes méthodes de génération de solution initiale, méthode de recherche de voisinage et la méthode de planification temporelle qui ont été détaillées dans la section (IV.3.1).

Dans la figure IV.10, nous présentons le fonctionnement de l'algorithme de RSMO pour la résolution du PTD dynamique.

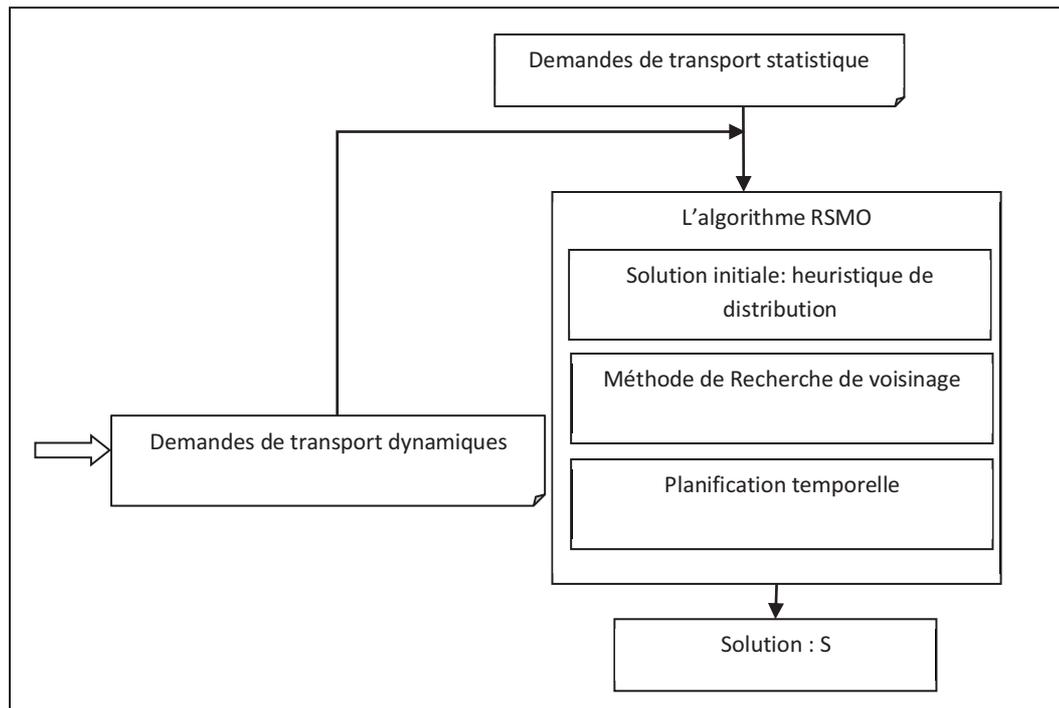


Figure IV.10 : Fonctionnement de l'algorithme de RSMO pour la résolution du PTD dynamique.

### a. Résultats Numériques

Dans ce travail, nous avons choisi de comparer notre approche basée sur l'algorithme RSMO avec une méthode d'insertion et une procédure de recherche locale pour montrer l'efficacité de l'algorithme RSMO dans le contexte dynamique. Nous avons pour cela générés 5 exemples pris au hasard formant des PTD dynamiques. Ces problèmes générés sont diversifiés par la localisation des demandes de transport et le nombre de demandes statiques et dynamiques dans chaque exemple. Ces situations reflètent assez fidèlement ce qui est observé dans le monde réel. Tous ces exemples ont un horizon d'une journée de travail de sept heures et de quatre heures. Par exemple, une journée qui se déroule entre 09h00 et 16h00, a un horizon temporel de 420 min soit 7h. Dans cet horizon, des événements en temps réel sont produits par notre simulation pour tenir compte de l'apparition de nouvelles demandes de transport. Le temps de service est égal à 0 min à chaque point de service et un appel est accepté que s'il y a au moins 30 min entre l'appel et le temps de ramassage souhaité.

Dans cette section, l'algorithme RSMO est comparé avec d'autres heuristiques (l'heuristique d'insertion et la méthode de recherche locale). L'heuristique d'insertion insère simplement une nouvelle demande en choisissant le meilleur emplacement possible. Un algorithme de

recherche locale commence à partir d'une solution candidate, puis se déplace de manière itérative vers une solution voisine. Pour produire une solution voisine nous utilisons la structure de voisinage indiqué dans la section IV.3.1. La procédure de recherche locale remplace la solution actuelle par la meilleure solution trouvée dans toutes les itérations de la procédure.

Numéro d'instance	heuristique d'insertion	Recherche Locale	RSMO	
1	Distance(Km)	384	430	281
	TMV (min)	19	16	11
	NBSV (nombre)	520	536	508
2		415	458	343
		18	16	12
		530	540	528
3		325	298	267
		22	17	15
		533	512	507
4		313	259	233
		24	18	17
		566	548	530
5		460	376	343
		27	21	18
		589	552	531
Total		1897	1821	1457
		110	89	75
		2738	2688	2604

Tableau IV.7 : Simulation de 4 h avec 10 véhicules et 20 demandes par heure

**TMV : Temps moyen mis pour le voyage**

**NBSV : nombre de stations visitées**

Dans ce tableau, le temps est en minutes et la distance parcourue en kilomètres, arrondies à l'entier le plus proche. Dans le tableau IV.7, nous présentons les résultats obtenus par l'application de l'algorithme de RSMO, l'heuristique d'insertion et la procédure de recherche locale sur cinq exemples de PTD.

L'application utilisée pour obtenir les résultats dans le tableau IV.7, correspond à des simulations de 4 h, avec 10 véhicules et l'intensité du nombre des demandes dynamiques est de 20 requêtes par heure, avec une simulation en temps réel. Ainsi, chaque véhicule sert deux demandes de transport par heure en moyenne. Pour chacune des méthodes (algorithme de RSMO, l'heuristique d'insertion et la recherche locale) le tableau possède trois valeurs différentes qui sont : la distance totale parcourue, temps moyen de voyage et le nombre des stations visitées.

D'après les résultats obtenus, RSMO surclasse les deux autres méthodes, tandis que la recherche locale est meilleure que l'heuristique d'insertion dans la majorité des cas. Les mauvais résultats produits par la procédure de recherche locale sont expliqués par le fait qu'à chaque fois qu'une nouvelle demande arrive, la procédure de recherche locale perd les gains déjà obtenus par la recherche locale précédente. L'apparition de chaque nouvelle demande de transport active la reconstruction de la solution, sans aucune considération de la solution précédente.

Les résultats présentés dans le tableau IV.7, sont obtenus par l'application d'un scénario pas très intense en termes de nombre de demandes par heure. En effet, la plupart des écarts entre ces méthodes est identifiable sur la distance parcourue.

Les résultats de l'algorithme de RSMO et de la recherche locale sont proches, avec une différence de 11, 1% pour la distance parcourue en faveur de l'algorithme de RSMO. Aussi l'algorithme de RSMO donne de bons résultats par rapport à l'heuristique d'insertion en termes de temps de trajet total avec une différence de 18, 9%. Il est nécessaire de mentionner que lors de nos tests, nous avons généré différents types de PTD diversifiés par les fenêtres de temps des demandes de transport, par conséquent, nos tests ne produisent pas nécessairement la même solution lorsqu'elle est appliquée deux fois ou plus sur le même problème. Nous avons appliqué l'algorithme de RSMO 10 fois sur une instance de problème particulier et on a obtenu un écart très faible de l'ordre de 1%.

#### **IV.6. Conclusion :**

Notre vision du PTD nous a conduits à considérer les deux aspects de ce problème, l'aspect statique et l'aspect réel ou dynamique. Pour le PTD statique les demandes de transport sont connues à l'avance et donc une résolution non instantanée est appliquée pour fournir les

itinéraires des véhicules afin de servir ces derniers. Par contre dans le PTD dynamique, des demandes de transport arrivent au fur et à mesure que le système est en train d'effectuer son service aux voyageurs. Dans ce cas, il faut un système qui répond en temps réel pour fournir des solutions afin de faire face aux demandes arrivant d'une manière dynamique. Dans ce chapitre, nous avons détaillé nos contributions qui sont basées sur l'utilisation de l'algorithme de recuit simulé pour la résolution du PTD statique et dynamique. Cette approche est basée sur une architecture centralisée.

L'architecture centralisée présente beaucoup de limites tel que la qualité de la solution lorsque il ne s'agit pas d'une bonne répartition des tâches d'optimisation. Pour remédier à ces limites, nous avons conçu une architecture décentralisée basée sur les Systèmes Multi-Agent (SMA) pour la résolution du PTD statique et dynamique. Cette approche sera détaillée dans le chapitre suivant.

# Chapitre V :

## Une approche distribuée basée sur l'algorithme du Recuit Simulé pour la Résolution du Problème de Transport à la Demande

### V. 1. Introduction :

Le PTD est un problème appartenant à la classe NP- difficile (Inge Li, 2006) [91]. Les méthodes exactes ne sont pas capables de résoudre un tel problème dans un temps raisonnable surtout lorsque la taille du problème est importante. Dans ce cas, on a souvent recours à des méthodes qui permettent de trouver une solution approchée en un temps acceptable ; il s'agit des heuristiques et méta- heuristiques. C'est pour ces raisons que nous avons appliqué une de ces méthodes ; à savoir le recuit simulé. Par contre, la résolution centralisée du PTD reste toujours un processus compliqué au regard du temps de calcul, de la complexité des tâches et de leurs nombres pendant la résolution. Plusieurs techniques existent permettant plus au moins une résolution approchée des problèmes complexes, nous avons choisi celle basée sur la distribution du processus. Elle permettrait d'aboutir à des résultats de bonne qualité en un temps raisonnable. Nous avons utilisé pour notre approche basée sur le recuit simulé multi-objectif, les Systèmes Multi-Agents (SMA) permettant la répartition des tâches sur différents agents autonomes. La distribution est un avantage pour les SMA. Cet avantage se manifeste par la réduction de la complexité des problèmes à résoudre en décomposant le traitement en plusieurs tâches réparties sur des agents. Elle peut être caractérisée comme un élément essentiel des systèmes évolutifs dans le sens où elle permet aussi le remplacement d'un système par plusieurs sous-systèmes, ce qui conduit à une meilleure adaptation aux changements de l'environnement.

Dans ce chapitre, nous présentons en premier lieu les SMA comme une solution de distribution. Par la suite nous détaillons notre approche décentralisée basée sur l'algorithme de recuit simulé multi-objectif. Ensuite, Nous présentons l'application de l'approche dans les

cas statique et dynamique. C'est dans cette partie que nous détaillons les résultats numériques obtenus prouvant l'efficacité de l'approche proposée. Nous finissons ce chapitre par une conclusion sur l'approche décentralisée proposée.

## **V. 2. Les systèmes multi-agents :**

### **1. Caractéristiques des agents**

Dans la littérature il existe plusieurs définitions attribuées à la notion d'agent. Dans [97], nous trouvons une définition de Jennings et Wooldridge où ils proposent une définition commune que nous citons dans ce qui suit. Avec cette dernière, nous pouvons relever les notions de situation, autonomie et flexibilité.

**Définition V.1 [97] :** Un agent est un système informatique encapsulé qui est situé dans un environnement et qui est capable d'effectuer des actions autonomes et flexibles afin d'atteindre les objectifs pour lesquels il a été conçu.

Dans cette thèse, nous avons vu que c'est intéressant de noter une autre définition plus détaillée du terme agent, elle est proposée par [98] et basée sur les différentes définitions établies par, entre autres, Jennings, Wooldridge et Ferber.

**Définition V.2 [98] :** 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.

Les agents sont caractérisés par un ensemble de propriétés. Ces propriétés sont les suivantes :

- *Adaptabilité* : capacité à s'adapter aux différents environnements.
- *Autonomie* : actions proactives et dirigées par des buts.
- *Comportement collaboratif* : c'est la notion de la sociabilité dans le monde réel, la capacité à collaborer avec d'autres agents pour un objectif commun.
- *Capacité inférentielle* : capacité à agir avec des spécifications abstraites des tâches.
- *Capacité de communication au niveau des connaissances* : l'agent doit être capable de communiquer avec les autres agents du système.

- *Réactivité* : capacité à détecter et à réagir.
- *Continuité temporelle* : persistance d'une identité et d'un état sur une longue période.

## 2. Caractéristiques des systèmes multi-agents

Les systèmes multi-agents sont des systèmes qui regroupent un ensemble d'agents qui coopèrent dans une société. La plupart des auteurs définissent les SMA de la manière suivante [99] :

**Définition V.3 :** Un système multi-agents est un réseau d'agents faiblement couplés qui coopèrent pour résoudre des problèmes qui dépassent les capacités ou les connaissances individuelles de chaque agent. Les agents sont autonomes et peuvent être de nature hétérogène.

Dans la suite de cette section, nous présentons les différentes caractéristiques des systèmes multi-agents.

### 2.1. Coordination

Pour assurer une communauté d'agents individuels agissant avec cohérence et harmonie, nous avons tendance à développer un processus de coordination entre les agents [99]. En appliquant le processus de coordination, nous pouvons empêcher les comportements désordonnés des agents. En effet, la coordination doit être faite d'une manière globale pour assurer l'association des agents. Il existe deux situations dans lesquelles la coordination est utile. La première étant la situation la plus simple, le traitement s'apparente à une routine, les agents peuvent être parfaitement coordonnés car on connaît ce qu'ils sont en train de faire et prévoir ce qu'ils vont faire. La deuxième situation correspond à des situations non familières et à des traitements inattendus.

Dans un processus de coordination, nous attribuons des tâches aux agents. L'allocation des tâches consiste à affecter des responsabilités et des ressources nécessaires à la résolution de problèmes à un agent. En effet, le concepteur du système d'agents peut allouer toutes les tâches d'avance en engendrant ainsi une procédure de résolution des problèmes qui est non adaptable. Par contre, nous pouvons concevoir une allocation dynamique et flexible des

tâches. Cette planification est une partie de la coordination. Pour un agent, elle constitue un processus de construction d'une séquence d'actions en tenant compte seulement des objectifs, des capacités et des contraintes environnementales. Cette organisation nous permet d'éviter les chevauchements et les conflits. Elle peut être centralisée ou distribuée.

## **2.2. Négociation**

La négociation dans les SMA est un processus utilisé par les agents d'un même groupe. Ce processus a pour objectif d'établir un accord partagé pour accepter une affaire donnée. Donc, la négociation est considérée comme une méthode de coordination et de résolution de problèmes. Dans ce processus, nous trouvons des échanges d'informations, des relaxations des buts initiaux, des concessions mutuelles, des menaces, etc. le principal but de ce processus est la recherche d'un consensus. Il existe deux type de négociations ; le compétitive et le coopérative. La négociation compétitive est utilisée dans les situations où des agents d'intérêts différents tentent de faire un choix de groupe avec des alternatives bien définies. Les agents sont néanmoins compétitifs et non coopératifs. Tandis que la négociation coopérative concerne les situations où les agents ont un but commun et global envisagé pour le système. La négociation coopérative a le même aspect des systèmes distribués conçus pour réaliser un seul objectif global. Les agents de ce type de négociation sont aussi appelés collaboratifs.

## **2.3. Communication**

Pour que les agents puissent interagir et échanger de l'information, il faut que ces derniers communiquent pour pouvoir accomplir leurs tâches [97]. Il existe deux approches de langage de communication. La première est l'approche procédurale ; la communication est basée sur un contenu exécutable (Java, etc.). La deuxième est l'approche déclarative où la communication est basée sur des phrases déclaratives (définitions, hypothèses, etc.). Cette dernière est préférée à la première car le contenu exécutable est difficile à coordonner et à contrôler. La plupart des langages basés sur l'approche déclarative reposent sur des demandes ou encore des requêtes construites avec des locutions comme demander ou commander. Le plus connu parmi ces langages est le KQML (*Knowledge Query Manipulation Language*).

## **2.4. Formation de coalitions**

La formation de coalitions est un modèle obtenu avec une organisation dynamique des agents d'un système multi-agents et d'une méthode de coopération entre ces agents. Le principe de la formation des coalitions est le fait de partitionner l'ensemble des agents définis dans un problème et qui peuvent communiquer et négocier, en sous-groupes dont chacun est appelé coalition.

Dans l'article de Vauvert et Seghrouchni [100], les auteures ont mentionné qu'une coalition est une organisation à court terme basée sur des engagements spécifiques et contextuels. Dans ce cas, les agents peuvent coexister tout en bénéficiant de leurs compétences respectives. Nous pouvons conclure que les coalitions permettent aux agents de satisfaire des besoins qui demandent l'ensemble de leurs compétences. En effet, nous pouvons montrer ceci par l'exemple dans un contexte économique distribué (problèmes de tournées de véhicules, problème d'allocation de tâches, etc.). Dans cet exemple, les acteurs qui agissant seuls seraient incapables ou moins efficaces pour aboutir à un but. C'est pour cette raison, nous avons tendance à établir des coalitions pour résoudre un problème qui nécessite des compétences variées.

La formation des coalitions présente plusieurs avantages. Les plus intéressants sont synthétisés dans les points suivants :

- Le concept d'engagement ponctuel permet aux agents de réagir de façon opportuniste et de réviser dynamiquement leurs intérêts et par conséquent leurs objectifs.
- La formation des coalitions est dépendante du contexte, ce qui permet aux agents d'adapter leur objectif et comportement.

## **3. Les SMA et leur application dans le domaine de transport**

Il existe plusieurs travaux de recherche basés sur les SMA pour la résolution des problèmes de transport. Parmi les premières applications développées à l'aide de la technologie multi-agents, nous trouvons une application dans le contrôle du trafic aérien. En 1991, Durfee a développé une application pour la surveillance de véhicules motorisés [103]. Une autre application d'aide au pilotage des avions militaires a été faite par Chaib-draa en 1996. Cette

application permet la surveillance des capteurs, la collecte, la vérification et l'analyse des données, proposition de mesures correctives ou de plans optionnels au pilote pour qu'il puisse atteindre le but de sa mission [104]. En se focalisant sur le trafic aérien, Ljungberg propose un SMA, nommé OASIS (*Optimal Aircraft Sequencing using Intelligent Scheduling*), pour le contrôle du trafic aérien de la région de Sydney en Australie [105]. Ce système permet de réduire la congestion du trafic aérien en maximisant l'utilisation des pistes en ordonnant les atterrissages. Dans ce travail, on trouve deux types d'agents ; des agents « globaux », qui traitent les relations entre les avions, la coordination et le raisonnement (*coordinateur, séquenceur, vérificateur de trajectoire, modèle du vent, interface utilisateur*) et des agents avion, responsables des calculs et du raisonnement associés à chaque avion (*position, vitesse, heure d'atterrissage, etc.*). Restant dans le trafic aérien, une étude des stratégies de coopération pour pouvoir résoudre les conflits entre les plans d'un groupe d'agents a été faite par (Cammarata 83) [106]. Ces stratégies sont ensuite appliquées dans le cadre d'un système de contrôle du trafic aérien. Dans ce système, chaque agent (représentant un avion) cherche à construire un plan de vol qui devrait le garder à une distance de sécurité des autres agents.

Dans le cadre de transport de marchandises, Fischer présente une plate-forme appelée « MARS » pour la simulation basé sur les multi-agents pour le transport des marchandises [107]. Elle décrit un scénario lié à des compagnies de transport géographiquement distribuées, qui gèrent des commandes arrivant en temps réel. Le système comprend deux types d'agents correspondant aux compagnies de transport et aux camions. Les agents *compagnies* doivent affecter les ordres aux camions ou coopérer avec les autres agents du même type pour qu'ils se chargent du ou des ordres. La coopération se base sur une négociation à l'aide d'une version évoluée du « Contract Net Protocole ». Le comportement et la cohérence du système sont évalués à partir des coûts associés aux ordres ordonnancés.

En ce qui concerne la gestion du trafic urbain, (Chaib-draa, 96) aborde le problème via une approche agent [108]. Les agents peuvent correspondre à des hommes (*conducteurs, policiers, piétons*) ou des machines (*véhicules, feux de circulation, etc.*) et doivent perpétuellement ajuster leurs actions pour éviter les bouchons et les accidents. Dans ce SMA, les agents ont le comportement de la loi sociale. L'intelligence artificielle des agents est guidée par des modèles stockés de procédures prédéfinies, qui vont de la perception ou de l'observation à l'action. Les situations non familières sont adaptées à des situations familières en utilisant des raisonnements basés sur des cas. Le modèle d'agent est ainsi basé sur trois niveaux :

compétences (pour la routine), règles (situations familières) et connaissances (situations non familières). En 2001, Gruer a proposé une approche basé sur un SMA pour la modélisation et la simulation des systèmes de transport [109]. Ce système est composé des agents *bus*, *arrêt* et *section*. En 2005, (El Hmam et al 05) ont adopté le paradigme agent pour modéliser le flux de trafic au niveau microscopique [110].

Pour assister le régulateur dans la régulation des correspondances Saussol 00 a modélisé un réseau de transport urbain par un SMA [111]. Dans ce SMA, un agent modélise une entité active du processus de transport. Il peut être un agent *bus*, *station*, *section dépôt* et *interface*. Restant toujours dans le cadre de la régulation des correspondances, (Laïchour 01) présente un modèle multi-agents basé sur trois types d'agents [112]:

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

Un autre SMA, a été proposé aussi par Balbo en 2000 pour l'aide à la décision afin de réguler un réseau de transport par bus. Ce système est basé sur un module d'interaction qui utilise l'environnement comme support de communication. Sa particularité réside dans son fonctionnement sans horaires préétablis pour les véhicules, mais suivant une gestion dynamique du planning. Il est composé d'agents [113] :

- *Arrêt*, qui assure dans les conditions normales la gestion des horaires en temps réel. Il définit les horaires de passage, à travers la distance du parcours et le temps nécessaire, pour que les véhicules les suivent ;
- *Bus*, qui a pour rôle d'exécuter les courses qui lui sont affectées et de proposer des mesures de régulation ;
- *Incident*, qui est responsable de la gestion des perturbations. Il synthétise les informations, afin d'évaluer la difficulté du problème et l'adéquation des solutions proposées par les agents Bus.

### **V. 3. Approche distribuée basée sur le recuit simulé multi-objectif pour la résolution du PTD :**

L'objectif que nous avons fixé au début de ce chapitre est de contribuer à la résolution distribuée du PTD. Après avoir parcouru les différents travaux de recherche effectués dans cette partie, nous avons noté que l'approche multi-agents est peu utilisée dans la résolution des PTD.

#### **1. L'architecture multi-agents dans notre approche**

Dans le cadre de cette thèse, le problème d'optimisation considéré est un problème multicritère. Nous avons proposé une approche non agrégative pour résoudre ce problème. Cette approche est basée sur l'algorithme de Recuit Simulé Multi-Objectif (RSMO). Nous avons déjà détaillé l'algorithme dans le chapitre IV. L'approche que nous présentons dans ce chapitre se base sur les SMA.

L'approche basée sur le système multi-agents développée dans cette recherche est composée de trois classes d'agents majeurs. Le premier est l'agent *client* qui envoie une requête à l'agent *centre d'appel*. Cet agent diffuse cette demande à tous les agents *véhicules*. L'agent *véhicule* exécute une routine d'optimisation basée sur l'algorithme RSMO détaillée dans le chapitre 4. Ce processus d'optimisation est exécuté à chaque fois que cet agent (*véhicule*) reçoit une demande du centre d'appels. Ce processus d'optimisation consiste à trouver une solution pour le PTD multicritère ayant les objectifs suivants : la durée du trajet, la distance parcourue et le nombre de stations visitées. Ainsi, l'agent *centre d'appel* récupère toutes les solutions proposées par les agents *véhicules*. Le meilleur itinéraire est ensuite choisi en utilisant le concept de dominance en fonction des objectifs du système. Dans la figure V.1, nous présentons l'architecture globale de notre modèle multi-agents.

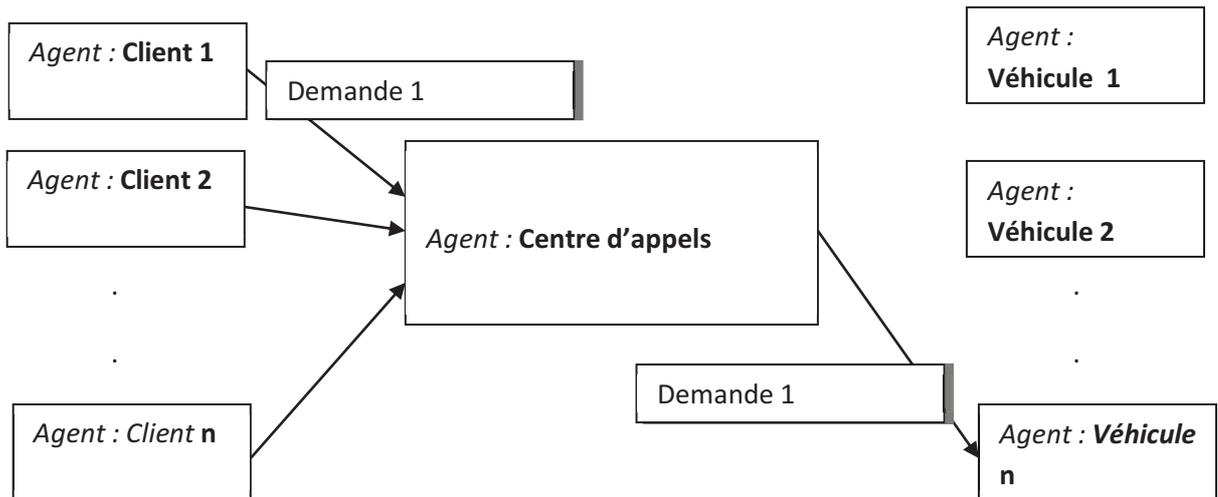


Figure V.1 : L'architecture globale du modèle multi-agents

Après avoir présenté le modèle de notre système multi-agents, nous présentons dans la figure V.2 un scénario de traitement d'une demande de transport.

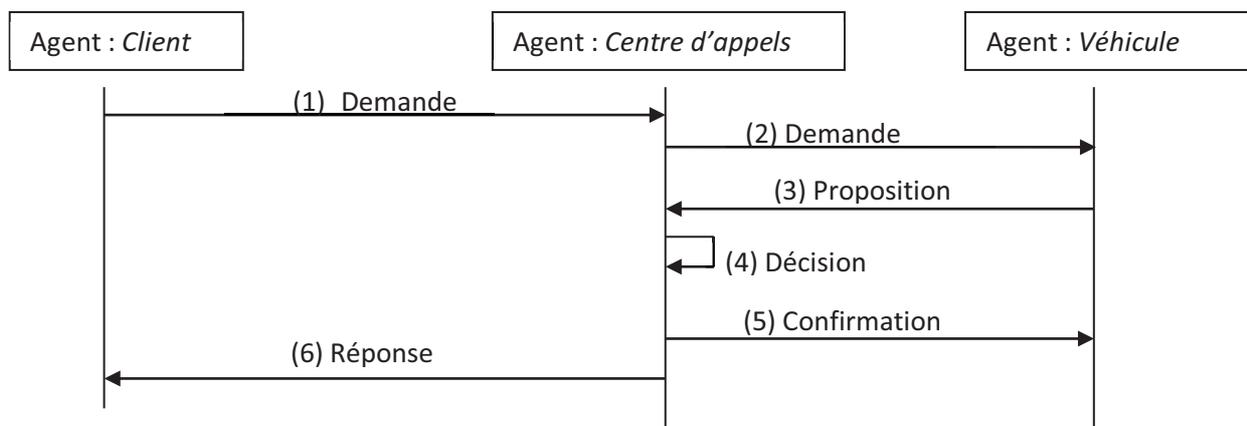


Figure V.2 : Le scénario de traitement d'une demande de transport

Nous détaillons ici, les différentes étapes du scénario de la figure V.2 présentant le traitement d'une demande de transport :

- (1) L'étape où un client (voyageur) envoie une demande de transport caractérisée par les coordonnées point de départ, point d'arrivée et les fenêtres de temps associés à ces points. Cette étape est faite par l'agent « *client* » dans notre SMA.
- (2) Dans cette étape, l'agent « *centre d'appels* » diffuse la demande de transport reçue du client (voyageur) à tous les agents « *véhicule* ».

- (3) Dans cette étape qui concerne l'agent « *véhicule* », un processus d'optimisation sera exécuté par ce dernier afin de trouver le meilleur itinéraire pour la demande de transport. Ensuite l'agent « *véhicule* » envoie sa proposition (meilleur itinéraire) à l'agent « *centre d'appels* ».
- (4) Cette étape concerne l'agent « *centre d'appels* », il se charge de récolter toutes les propositions des véhicules. Ensuite, il exécute un processus d'optimisation de recherche locale basé sur la notion de dominance pour trouver la solution convenable en termes de distance parcourue, de temps de voyage et du nombre de stations visitées. Après cette recherche, l'agent « *centre d'appels* » prendra la décision concernant la demande de transport reçue.
- (5) Le centre d'appels envoie une confirmation pour le véhicule qui a envoyé la meilleure proposition concernant la demande de transport.
- (6) Le centre d'appels envoie une réponse à l'agent « *client* » en lui indiquant le véhicule qui va le prendre en charge ainsi que les heures de départ et d'arrivée estimées.

## 2. Le processus d'optimisation

Le PTD est un problème classé parmi les problèmes d'optimisation NP-complets. Pour les problèmes appartenant à cette catégorie, il n'existe pas d'algorithmes polynomiaux permettant leur résolution d'une façon optimale dans un temps raisonnable. Pour cette raison, nous sommes amenés à appliquer une méthode approchée qui donne une solution acceptable dans un délai raisonnable. Dans notre SMA, il existe deux modules d'optimisation. Le premier module sera exécuté par l'agent *véhicule* et le deuxième par l'agent *centre d'appels*. L'agent *véhicule* exécute un processus d'optimisation à chaque fois qu'il reçoit une demande de transport. Ce processus est basé sur l'algorithme de Recuit Simulé Multi-Objectif (RSMO).

### 2.1. L'agent Véhicule

L'agent *véhicule* doit donner une proposition de solution à chaque demande de transport reçue de la part de l'agent *centre d'appels*. Dans la figure V. 3, nous présentons le comportement de l'agent *véhicule*.

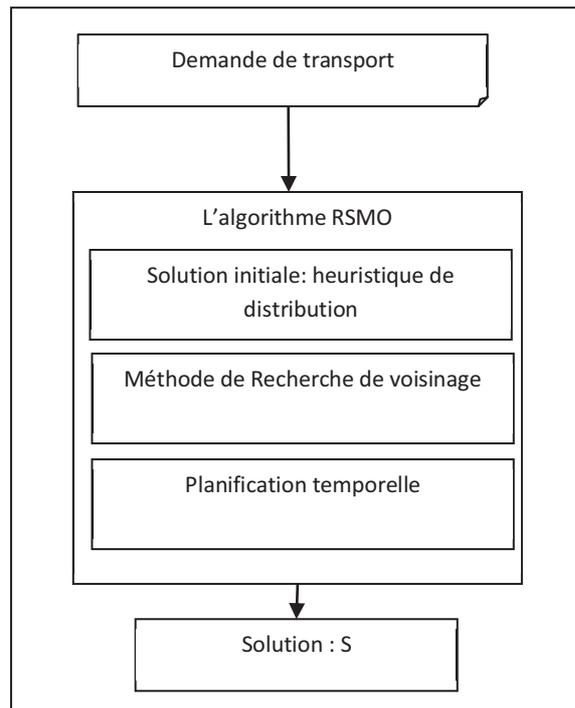


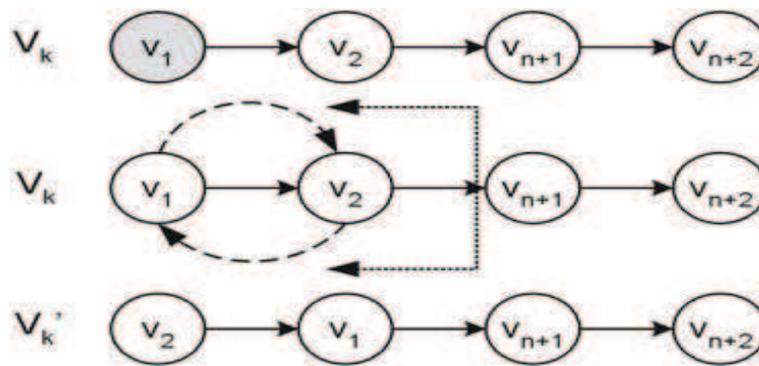
Figure V.3 : L'architecture comportementale de l'agent véhicule

La construction d'une solution pour le PTD se fait de la manière suivante : A chaque fois qu'une demande de transport arrive, l'agent *véhicule* utilise l'algorithme de RSMO. Il représente une méthode d'optimisation multi-objectif Pareto utilisant la notion de dominance pour trouver une solution optimale (proche de l'optimum). L'algorithme est présenté en détails dans le chapitre IV. Nous rappelons également que les méthodes de recherche de la solution initiale ainsi que la méthode de planification temporelle sont détaillées dans le chapitre IV précisément dans la section de l'algorithme de RSMO. Dans cette section nous ne présentons que la méthode de recherche de voisinage qui est déférente de celle présentée dans le chapitre précédent.

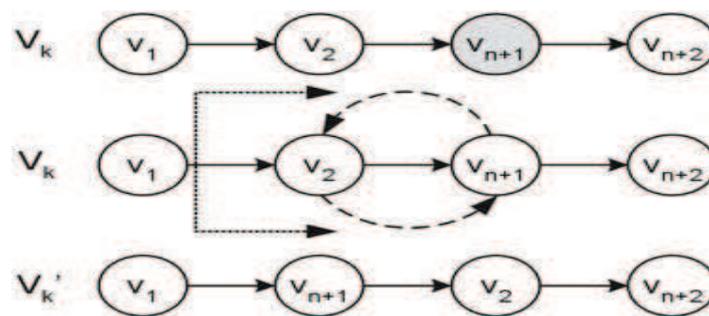
- **La méthode de recherche de voisinage**

Pour améliorer la solution du PTD, il faut effectuer des changements sur la solution courante. Ces changements sont effectués par une méthode de recherche de voisinage. En effet, dans l'algorithme de recuit simulé on peut accepter des changements même s'ils dégradent la qualité de la solution. Notre méthode de recherche de voisinage est basée sur la méthode « re-order Route » qui est issue du travail présenté par (Bergvinsdottir et al, 2004) [102]. Il est intéressant de mentionner que dans les mouvements de cette structure de voisinage les dépôts

ne sont pas considérés, car ils sont "fixés" dans tous les itinéraires, et donc, leurs positions ne peuvent pas être modifiées. La méthode « re-order route » consiste essentiellement à choisir une route de la solution, sélectionner n'importe quel point dans cette route, sélectionner une nouvelle position pour ce point et puis changer la position de ce point avec la nouvelle position. Ce mouvement est montré dans la figure V.4. Le point sélectionné peut être un point de départ ou un point de livraison. Dans le premier cas (figure V.4.a), la nouvelle position sera obligatoirement avant son point de livraison. Dans le second cas (figure V.4.b), la nouvelle position doit être après le point de ramassage. Ces changements sont présentés par les lignes pointillées dans la figure V.4.



(a)



(b)

Figure V.4 : Structure de voisinage

## 2.2. L'agent Centre d'appels

L'agent *centre d'appels* se comporte de la manière suivante :

Chaque demande de transport reçue sera automatiquement transmise à tous les agents *véhicule*. Le centre d'appels récolte toutes les réponses auprès des véhicules. Ensuite, il fait appel à une procédure de recherche locale basée sur la notion de dominance pour trouver la meilleure solution. Avec cette architecture de SMA, nous pouvons explorer plus l'espace de recherche parce qu'il y a plusieurs véhicules qui proposent des solutions, le centre d'appels choisit alors la meilleure.

La procédure de recherche locale de l'agent *centre d'appels* est décrite dans la figure V.5 :

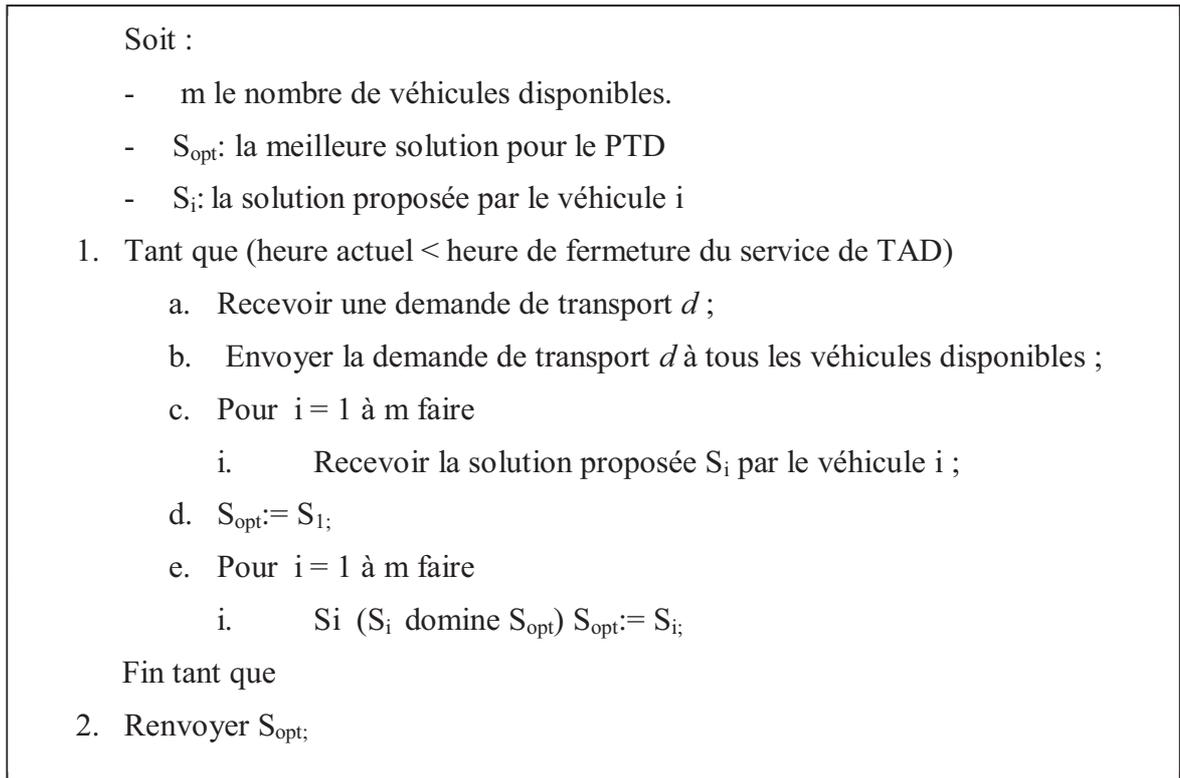


Figure V.5 : comportement de l'agent centre d'appels

#### V. 4. Applications et Résultats Numériques :

Dans ce chapitre, nous présentons les différents résultats numériques obtenus par des simulations et des tests sur le benchmark (Cordeau et Laporte, 2003) [34]. Nous commençons par présenter les outils informatiques utilisés pour développer notre approche basée sur l'algorithme RSMO pour le PTD, afin de valider nos méthodes de résolution. Concernant la partie PTD dynamique, nous générons des scénarii formés par des demandes de transport arrivant en temps réel. Alors que pour la partie du PTD statique, nous appliquons notre approche sur le benchmark de (Cordeau et Laporte, 2003) [34].

## 1. L'outil informatique utilisé

Pour développer l'algorithme de RSMO utilisé dans notre approche d'optimisation, nous avons utilisé le langage de programmation orienté objet JAVA. En effet, ce dernier est similaire en grande partie dans sa syntaxe au langage C++. Le langage Java vient pour donner des solutions aux inconvénients du C++ tels que l'héritage multiple. Les concepteurs ont privilégié l'approche orientée objet, de sorte qu'en Java, tout est objet à l'exception des types primitifs (nombres entiers, nombres à virgule flottante, etc.).

Dans cette thèse, notre choix du langage de programmation est basé sur les caractéristiques de ce langage. En effet, Les applications Java peuvent être exécutées sur tous les systèmes d'exploitation pour lesquels a été développée une plate-forme Java. Cette plate-forme permet d'exécuter les applications faite par JAVA dont le nom technique est JRE (*Java Runtime Environment*). Le JRE JAVA est constitué d'une Machine Virtuelle Java JVM (*Java Virtual Machine*), Mais le JRE est surtout constitué d'un package standard, à partir du quel doivent être développés tous les programmes en Java. C'est la garantie de portabilité qui a fait la réussite de Java dans le domaine de développent d'applications et logiciels informatique.

Un deuxième principal avantage pour JAVA, c'est l'indépendance vis-à-vis de la plate-forme. En effet, tous les programmes écrits en Java peuvent fonctionner sur plusieurs types de matériel informatique sans réécriture, selon le slogan « *write once, run anywhere* » (écrire le programme une seule fois, et le faire fonctionner sur n'importe quel ordinateur).

Pour obtenir la portabilité du langage Java il faut compiler le code source d'un programme dans un pseudo-code intermédiaire, (appelé "*bytecode Java*", des instructions machine standardisées et destinées à un processeur virtuel). Après avoir compilé le code, le programme sera exécuté par une implémentation de machine virtuelle (JVM) java. Le JVM va produire un code natif qui sera exécuté sur le matériel cible en traduisant le pseudo-code Java en code utilisable par le matériel concerné. De plus, des packages sont offertes pour fournir l'accès à des fonctionnalités propres à la machine cible, comme le graphisme ou le réseau, d'une manière unifiée. Le langage Java offre aussi le support des programmes avec plusieurs processus légers (*threads*).

## 2. Le système multi-agents

Les SMA nécessitent une plate-forme de développement. En effet, avec une plate-forme de SMA, nous pouvons organiser les agents d'un tel système et de leurs attribuer des tâches. A l'heure actuelle, plusieurs outils ont été développés pour la réalisation d'agents. Etant donné que ces plates-formes sont nombreuses, et ne répondent pas toutes aux critères que nous nous sommes fixés. Nous allons dans cette partie présenter notre démarche dans le choix d'une plate-forme de développement d'agents.

Vu que l'application de notre approche sur le PTD dynamique n'est pas réalisée sur des données réelles, nous allons simuler le fonctionnement de ce système de transport à la demande. Nous allons générer des demandes aléatoires distingués par leur localisation et les horaires de voyage souhaitées. Après une étude détaillée des plates-formes existantes, nous avons procédé par élimination pour ne garder que trois, à savoir, Magique, MadKit et Jade. Après des tests d'implémentation, nous avons éliminé la première, parce qu'elle demande beaucoup de développement pour définir les agents. Pour les deux restantes, le choix était plus difficile. Nous avons fini par choisir la plate-forme JADE, cette décision été collectif, vu que dans notre équipe, on travaille depuis longtemps afin de développer une plate-forme générique regroupant tous les services liés au transport et après plusieurs réunions de travail avec mes collègues, le choix a été définitivement entériné.

### 2.1. La plate-forme multi-agent JADE

La plate-forme JADE (*Java Agent Development Framework*) est un environnement de conception et de programmation d'agents implanté totalement en langage JAVA. En effet, la plate-forme JADE facilite la mise en place d'un système multi-agents répondant aux spécifications de FIPA (*Foundation for Intelligent Physical Agent*) à travers un ensemble d'outils.

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

La plate-forme JADE donne la possibilité de distribution des agents sur différentes stations (hôtes) d'un réseau. Une seule application JAVA et par conséquent une seule machine virtuelle Java est exécutée sur chaque hôte. Chaque machine virtuelle Java est un conteneur des agents fournissant un environnement complet d'exécution de ces agents et permettant à plusieurs agents concurrents de s'exécuter sur le même hôte (Figure V.6)

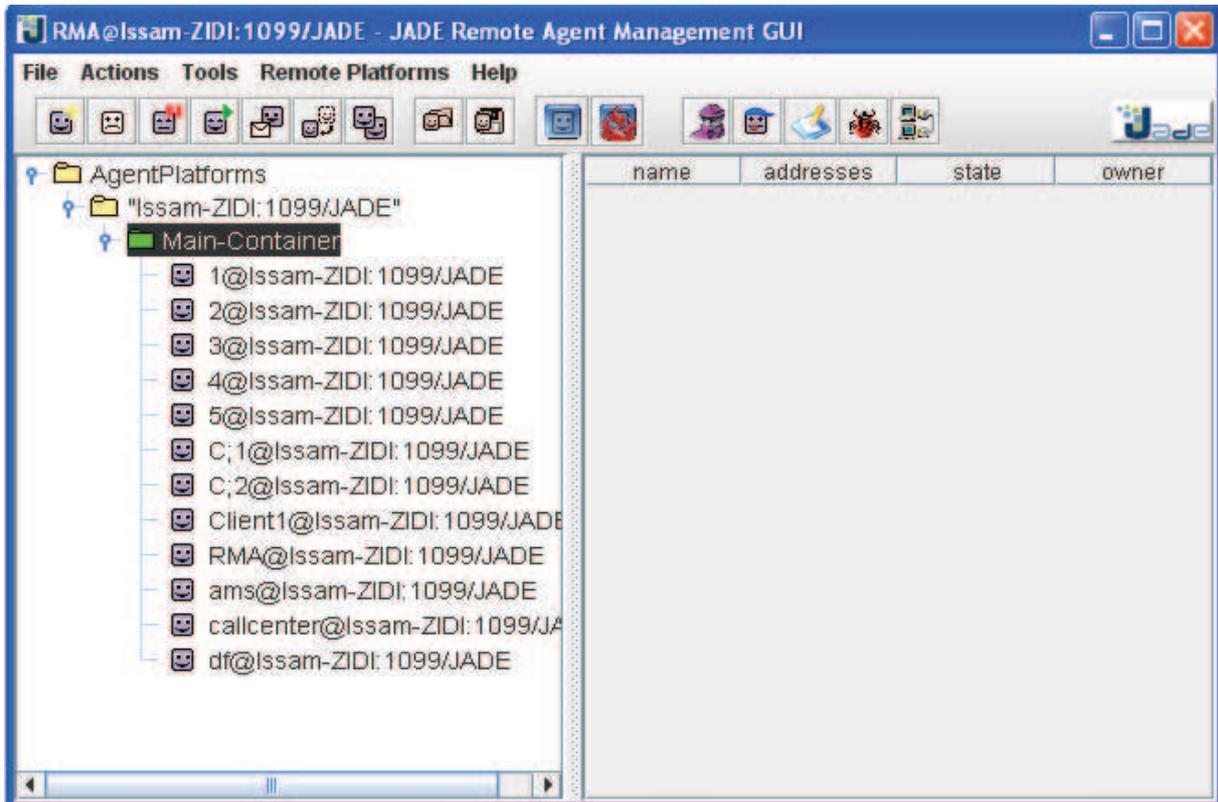


Figure V.6 : le conteneur des agents dans la plate-forme JADE

Dans cette thèse, JADE est la plate-forme la plus adéquate pour satisfaire nos objectifs. En effet, JADE C'est une plate-forme de création d'agents qui prend en compte les spécifications de la FIPA pour l'interopérabilité des systèmes multi-agents. Les agents communiquent à travers des messages représentés en FIPA-ACL. Sous JADE, le concept d'agent est comme un processus autonome et indépendant qui a une identité, et peut effectuer des communications (collaboration, compétition...) avec les autres agents, dans le but de résoudre un problème donné.

Dans le début de ce chapitre, nous avons motionné l'importance de l'autonomie d'un agent : Il ne doit pas se limite seulement à réagir vis-à-vis des événements venant de son environnement externe. Il doit être aussi capable de prendre l'initiative d'accomplira de nouveaux actes

communicatifs d'une façon autonome. Dans la figure V.7, nous présentons la communication inter-agents de notre SMA via l'envoi de messages entre ces derniers.

Généralement, le développeur des SMA doit étendre la classe Agent par une ou plusieurs classes Comportement pour effectuer des tâches spécifiques. Après avoir implémenté des comportements, il faut les instancier et les ajouter à l'agent. La classe Agent représente une super classe commune pour tous les agents définis par l'utilisateur. Un agent JADE est simplement une classe Java qui étend la classe de base Agent. La conséquence étant cette relation d'héritage permettant à l'agent d'hériter un comportement fondamental caché (qui traite toutes les tâches liées à la plate-forme, telles que l'enregistrement, la configuration, la gestion à distance, etc.). L'agent hérite également un ensemble de méthodes qui peuvent être appelées pour implémenter les tâches spécifiques à l'agent, par exemple envoi des messages, utilisation des protocoles d'interaction standard, enregistrement sur plusieurs domaines, etc.

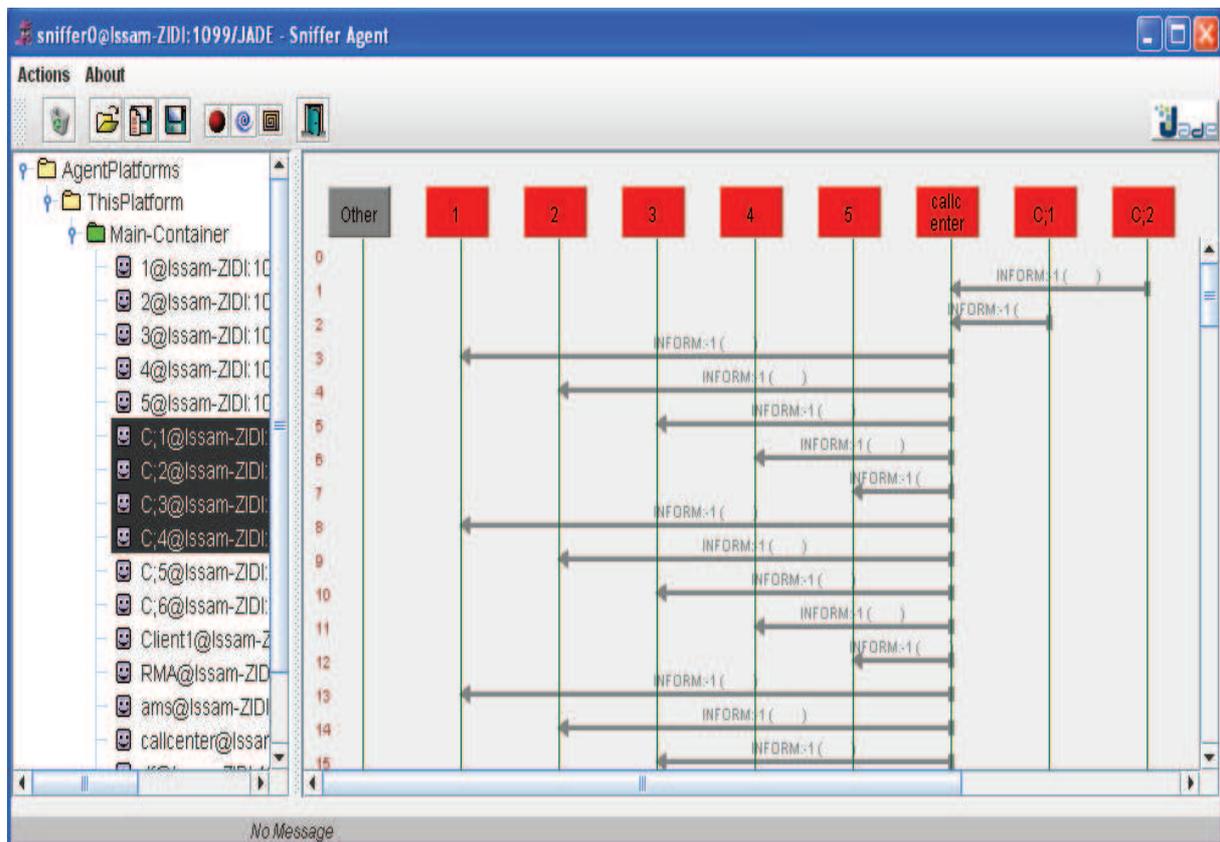


Figure V.7 : La communication des agents via l'envoi de messages

### 3. Application de l'approche sur un PTD statique

Pour cette partie, nous testons notre approche également avec les données présentés dans le benchmark de (Cordeau et Laporte, 2003) [34] (téléchargeable du net à partir du site suivant: « <http://www.hec.ca/chairedistributique/data/darp/> ». Voir chapitre IV, section 2.

Nous rappelons que dans ce benchmark il y a 20 instances de problèmes de transport à la demande. Le nombre de demandes varie entre 24 et 144 et le nombre de véhicule varie entre 3 et 13. Les résultats obtenus par notre approche sont présentés dans tableau V.1 et sont comparés à ceux obtenus par (Claudio et al, 2009) [35] et (Cordeau et Laporte, 2003) [34] déjà présentés dans la section 2 du chapitre précédent. Les paramètres de l'algorithme du recuit simulé restent les mêmes.

Dans le tableau V. 1, nous présentons les résultats obtenus par l'application de notre approche sur les instances mentionnés ci-dessus.

<b>Instance</b>	<b>Distance parcourue</b>	<b>Durée de la route (min)</b>	<b>Temps d'exécution (min)</b>
Pr01	<u>932,73</u>	<u>516,19</u>	1,12
Pr02	<u>1825,39</u>	<u>731,2</u>	3,29
Pr03	2806,70	3546,64	12,30
Pr05	<u>3809</u>	<u>2785</u>	10,56
Pr11	1003,77	598,13	1,53
Pr12	<u>1366,43</u>	811,50	5,62
Pr15	<u>4051,60</u>	<u>3001,2</u>	16,20
Pr16	<u>4512,30</u>	<u>2247</u>	15,21
Pr17	13397,50	933,58	3,60
Pr019	<u>3312.70</u>	2894	14

Tableau V.1: Les résultats obtenus par l'approche multi-agents basé sur le RSMO (Zidi et al, 2011) [101]

Dans le tableau V.2, nous présentons les résultats obtenus par (Claudio et al, 2009).

<b>Instance</b>	<b>Distance parcourue</b>	<b>Durée de la route (min)</b>	<b>Temps d'exécution (min)</b>
Pr01	955,25	524,59	1,36
Pr02	1839,06	838,41	4,08
Pr03	2787,18	1597,95	7,96
Pr05	4068,05	2935,48	18,43
Pr11	902,18	449,91	1,58
Pr12	1503,34	744,93	4,49
Pr15	4057,08	3152,67	22,09
Pr16	4658,64	2348,48	17,48
Pr17	1223,68	612,40	3,13
Pr019	3427,06	2515,53	25,43

Tableau V.2: Les résultats obtenus par l'Algorithme Génétique (AG) (Claudio et al, 2009)

Dans le tableau V.3, nous présentons les résultats obtenus par (Cordeau & Laporte, 2003) en appliquant l'algorithme de recherche Tabou.

<b>Instance</b>	<b>Distance parcourue</b>	<b>Durée de la route (min)</b>	<b>Temps d'exécution (min)</b>
Pr01	881	1095	1,9
Pr02	1985	1977	8,06
Pr03	2579	3587	17,18
Pr05	3870	6154	46,24
Pr11	965	1042	1,93
Pr12	1565	2393	8,29
Pr15	3596	6105	54,33
Pr16	4072	7347	73,7
Pr17	1097	1762	4,23
Pr019	3249	5581	51,28

Tableau V.3: Les résultats obtenus par l'algorithme de Recherche Tabou (RT) (Cordeau & Laporte, 2003)

### a. Analyse des Résultats

Nous avons effectués 5 tests pour chacune des instances du benchmark, Le tableau V.1 présente les meilleurs résultats obtenus des 5 essais. Pour l'ensemble de ces tests, le nombre de véhicules utilisé était fixe et était le même pour tous les essais, et toutes les demandes de transport ont été servi. En effet, le plus important est que toutes les solutions obtenues sont valides (toutes les contraintes du problème ont été satisfaites).

Après avoir présenté nos résultats et ceux obtenus dans (Claudio et al, 2009) et (Cordeau et Laporte, 2003) voir tableaux IV5 et IV6, nous notons que dans la majorité des cas, notre approche est plus efficace que l'approche mise en œuvre dans littérature citée. Les valeurs énoncées dans le tableau des résultats sont sensiblement meilleures. En effet, lorsque nous nous sommes focalisé sur la distance parcourue, notre approche RSMO est meilleure que l'Algorithme Génétique (AG) dans 7 cas sur 10. Par contre notre approche est uniquement meilleure que la Recherche Tabou (RT) que dans 4 fois sur 10.

Lorsque nous nous sommes basés sur le total de la durée de la route, notre approche RSMO est équivalente aux résultats donnés par l'AG (une différence de à 0,6% en faveur des AG) par contre notre approche est nettement mieux que celle de la RT avec un gain de 50%.

#### **4. Application de l'approche sur un PTD dynamique**

Nous constatons clairement, que dans le monde réel de transport des passagers où la fourniture d'un service de TAD, le problème est dynamique. La dynamique du système vient du fait que la compagnie offrant ce service, reçoit en permanence de nouveaux appels de transport de petite ou de grande taille. Donc, pour chaque nouvelle demande, une décision doit être prise rapidement pour son attribution à un véhicule particulier. Après, un ordonnancement au sein de la route doit être prévu pour chaque véhicule.

Dans notre travail, le problème traité est d'affecter en temps réel chaque nouvelle demande de transport à l'un des véhicules, et d'élaborer un nouvel itinéraire. Nous supposons en outre que nous avons un seul dépôt pour les véhicules, et en début la journée tous les véhicules se trouvent dans le dépôt, et leurs capacité est supérieur à un. Tous les véhicules desservant les passagers ne sont pas de même type (parc hétérogènes) mais ayant la même capacité. Donc avec les SMA, nous pouvons simuler ce problème de PTD avec des véhicules hétérogènes à fin de dégager un nouveau critère qui est la satisfaction du voyageur en termes de véhicule.

Ce nouvel objectif rentre dans les critères de qualité de service et nous l'avons modélisé de la façon suivante :

- Satisfaction du voyageur : SV

$$SV = \sum_{i \in D} SV_i$$

Avec

$$SV_i = \sum NB - CR - SATSF_i / NB - CR - PROF_i \quad (\text{eq V.1})$$

Avec  $NB - CR - SATSF_i$  : Le nombre des critères satisfaits pour une demande  $i$

Et  $NB - CR - PROF_i$  : Le nombre des critères souhaités par une demande  $i$

Dans notre environnement de simulation, les demandes des transports doivent être reçues avant la date limite fixée pour la fin de service d'une journée. Ceux qui sont reçus après cette date limite peuvent être conservés pour le lendemain. Ainsi, il est possible d'avoir un certain nombre de demandes en instance ou "statique" (pour lesquels une solution a pu être construite à l'avance). La dynamicité provient d'une source unique, à savoir l'arrivée de nouvelles demandes de transport. En effet, il n'y a pas d'autre source d'incertitude associée aux demandes des clients, comme les annulations ou des informations erronées. De plus les durées de voyage sont connues avec certitude, aucune perturbation inattendue est supposée, comme la congestion soudaine sur le réseau causée par un accident, panne de véhicule, etc.

Dans la figure V.8, nous présentons une situation pour le Problème de Transport à la Demande Dynamique (PTDD).

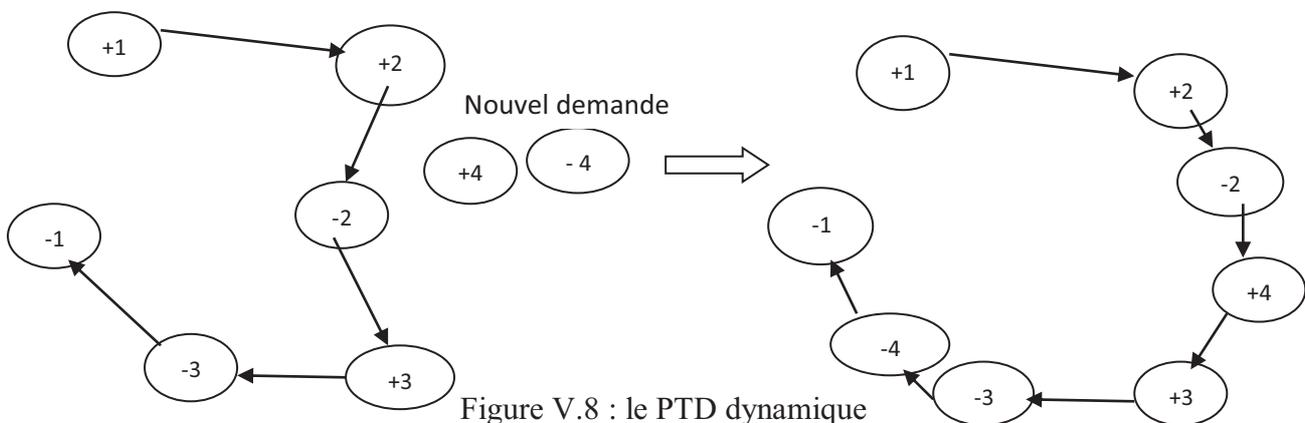


Figure V.8 : le PTDD dynamique

Dans cette contribution de thèse, nous avons choisi de comparer notre approche basée sur l'algorithme RSMO implanté dans l'agent *véhicule* avec une méthode d'insertion et une

procédure de recherche locale. Cette comparaison a pour but de monter l'efficacité de l'algorithme RSMO dans le contexte dynamique. Nous avons générés 5 exemples pris au hasard formant des PTD<sub>s</sub> dynamique en utilisant notre SMA. Ces problèmes générés sont diversifiées par la localisation des demandes de transport et le nombre de demandes statiques et dynamiques dans chaque exemple. Ces exemples reflètent le plus fidèlement possible ce qui est observé dans le monde réel. Tous ces exemples sont générés sur deux horizons de temps d'une journée de travail de sept heures et de quatre heures. Dans cet horizon, des événements en temps discret sont produits par notre simulation pour tenir compte de l'apparition de nouvelles demandes de transport. Par la suite, à chaque demande de transport reçue par l'agent *véhicule* il exécutera la procédure d'optimisation (RSMO). Le temps de service est égal à 0 min (négligeable) à chaque point de service, et un appel est accepté que s'il y a au moins 30 min entre l'appel et le temps de ramassage souhaité.

Dans cette section, l'algorithme RSMO est comparé avec d'autres heuristiques (l'heuristique d'insertion et la méthode de recherche locale). L'heuristique d'insertion insère simplement une nouvelle demande à sa meilleure place dans la route concernée. L'algorithme de recherche locale commence à partir d'une solution initiale, puis se déplace de manière itérative vers une solution voisine. Pour produire une solution voisine nous utilisons la structure de voisinage détaillée dans à la section IV.3.1. La procédure de recherche locale remplace la solution actuelle par la meilleure solution trouvée dans toutes les itérations de la procédure.

Le nombre de véhicules disponibles pour servir les demandes de transport est variable d'un test à l'autre. Notre simulation génère les caractéristiques de chaque nouvelle demande, à savoir, son temps d'apparition ainsi que, les emplacements de départ et de livraison avec leurs fenêtres de temps et le profil de cette dernière. Toutes ces caractéristiques sont générées par l'agent *client*. Ces valeurs doivent être réalisables. Par exemple, les fenêtres de temps doivent laisser suffisamment de temps pour que le véhicule puisse atteindre le point de ramassage, et d'aller directement au point de livraison. En effet, l'apparition des demandes varie dans le temps (par exemple, les heures de pointe) et de l'espace (par exemple, zone densément peuplée). Dans cette contribution, le cas des véhicules hétérogènes a été examiné. Ainsi, chaque véhicule est caractérisé par un ensemble de catégories. Dans notre cas, nous ne

donnons pas des valeurs réalistes pour le profil des clients ainsi que pour les catégories de véhicules.

Ces problèmes générés aléatoirement représentent des problèmes avec un seul dépôt et le modèle proposé dans (section IV.2) s'adapte facilement à ce genre de problème. La distance entre deux lieux  $i$  et  $j$  est calculée par la distance euclidienne entre les coordonnées des lieux  $i$  et  $j$ . Les profils des clients et les catégories de véhicules sont choisis au hasard.

Des Problèmes avec 15 et 25 véhicules ont été générés avec une vitesse moyenne constante de 30 km / h. La zone de tests est de taille 5 km x 5 km et le dépôt est situé au coordonnées (2,5 km, 2,5 km).

### a. Les résultats numériques

Dans cette section, l'algorithme RSMO est comparé avec d'autres heuristiques (l'heuristique d'insertion et la méthode recherche locale).

Numéro d'instance	heuristique d'insertion	Recherche Locale	RSMO
1	399 <sup>a</sup> 19 <sup>b</sup> 586 <sup>c</sup> 51 <sup>d</sup>	495 <sup>a</sup> 17 <sup>b</sup> 585 <sup>c</sup> 63 <sup>d</sup>	267 <sup>a</sup> 12 <sup>b</sup> 516 <sup>c</sup> 72 <sup>d</sup>
2	525 18 590 57	503 16 622 62	333 11 518 85
3	418 19 593 60	376 18 561 65	252 14 487 72

4	583 21 589 60	429 18 568 68	230 16 519 69
5	583 24 601 51	401 19 591 68	332 17 527 71
Total	<b>1833</b> <b>97</b> <b>2719</b> <b>351</b>	<b>1767</b> <b>79</b> <b>2645</b> <b>366</b>	<b>1414</b> <b>70</b> <b>2567</b> <b>369</b>

Tableau V.4: simulation de 4 heures avec 10 véhicules et 20 demandes par heure (Zidi et al, 2011<sup>b</sup>) [116]

**a distance totale parcourue**

**b temps moyen mis pour le voyage**

**c nombre de stations visitées**

**d niveau de satisfaction des clients en termes de véhicule**

Dans nos tests, le temps est en minutes et la distance parcourue en kilomètres et la satisfaction des clients en pourcentage, arrondie à l'entier le plus proche. Dans le tableau V.4, nous présentons les résultats obtenus par l'application de l'algorithme de RSMO, l'heuristique d'insertion et la procédure de recherche locale sur cinq exemples de PTD.

Les tests utilisés pour obtenir les résultats dans le tableau ci-dessus, correspondent à des simulations de 4 h, avec 10 véhicules et l'intensité du nombre des demandes dynamiques est de 20 requêtes par heure. Ainsi, chaque véhicule sert deux demandes de transport par heure en moyenne. Pour chacune des méthodes (algorithme de RSMO, heuristique d'insertion, et la

recherche locale) il existe quatre valeurs différentes dans le tableau : la distance totale parcourue, temps de trajet moyen, le nombre des stations visitées et le niveau de satisfaction des clients.

Comme indiqué dans le tableau V.4, les méthodes utilisant une recherche locale (RSMO, la recherche locale) sont beaucoup mieux que l'heuristique d'insertion. En effet, le RSMO est meilleure que la recherche locale. Les mauvais résultats produits par la procédure de recherche locale sont dus au fait que, à chaque fois qu'une nouvelle demande arrive, la procédure de recherche locale perd les gains déjà obtenus. En effet, l'apparition de chaque nouvelle demande de transport active la reconstruction de la solution, sans aucune considération pour la solution précédente.

Les résultats présentés dans le tableau ci-dessus, sont obtenues par application d'un scénario pas très intense en termes de demandes par heure. En effet, la plupart des écarts entre ces méthodes provient de la distance parcourue. Les résultats de l'algorithme de RSMO et de recherche locale sont proches, avec une différence de 11,09 % pour la distance parcourue en faveur de l'algorithme de RSMO. Concernant le temps mis pour le voyage le RSMO à des bons résultats par rapport à l'heuristique d'insertion avec une différence de 6,04%. Il est nécessaire de mentionner que nos tests sont caractérisés par une variété de problème en changeant les fenêtres de temps pour les demandes, par conséquent, nous ne produisent pas nécessairement la même solution lorsqu'elle est appliquée deux fois ou plus sur le même problème.

Si nous nous concentrons sur les objectifs de la qualité de service, l'algorithme RSMO est meilleur que l'heuristique d'insertion en termes de nombre de stations visitées avec une différence de 2,87%. Pour la satisfaction du client en termes du véhicule, les trois méthodes ont des résultats très proches, cela est principalement dû à ce que cet objectif ne dépend pas de la méthode utilisée dans l'agent *véhicule*. Il dépend de la phase d'affectation faite par l'agent *centre d'appels*. Nous avons exécuté l'algorithme de RSMO 10 fois sur une instance de problème particulier et nous avons trouvé un écart type de 0,7%.

---

Numéro d'instance	heuristique d'insertion	Recherche Locale	RSMO
-------------------	-------------------------	------------------	------

1	468 <sup>a</sup> 19 <sup>b</sup> 592 <sup>c</sup> 58 <sup>d</sup>	529 18 583 54	377 14 549 60
2	483 17 533 63	578 15 567 82	422 13 551 85
3	480 19 594 52	389 17 581 65	321 14 499 51
4	412 20 575 65	381 17 587 72	317 15 531 70
5	548 21 559 61	440 19 588 75	412 16 512 76
Total	<b>2391</b> <b>96</b> <b>2853</b> <b>299</b>	<b>2317</b> <b>86</b> <b>2906</b> <b>348</b>	<b>1849</b> <b>72</b> <b>2642</b> <b>342</b>

Tableau V.5: simulation de 7 heures avec 20 véhicules et 20 demandes par heure

Dans le tableau V.5 nous présentons les résultats obtenus pour un horizon de 7h et en augmentant la taille de la flotte de véhicules de 10 à 20. Le nombre de demandes reste le même c'est-à-dire 20 demandes par heure, et chaque véhicule sert actuellement une demande par heure en moyenne. Comme présenté dans le tableau V.5, nous constatons que la distance parcourue et le temps de parcours ont augmenté par rapport aux résultats présentés dans le tableau V.4. Ceci nous amène à dire que la réduction du nombre de véhicules conduit à un environnement d'optimisation plus difficile. Nous notons que l'algorithme RSMO et la méthode de recherche locale présente quasiment la même distance parcouru (en réalité une très faible différence de l'ordre de 0,1% en faveur de la recherche locale), par contre une différence de près de 9% dans le temps mis pour le voyage. De plus, nous notons que le total du nombre de stations visitées est plus réduit dans les résultats du tableau V.5. Cette réduction peut être expliquée par le fait d'utiliser un grand nombre de véhicules.

Dans le tableau V.6, nous présentons les résultats obtenus par l'application de notre approche pour un scénario plus dynamique que celui présenté dans les tableaux V.4 et V.5. Dans ce scénario, nous augmentons le taux d'arrivée des demandes, nous passons de 20 à 35 demandes par heure, tout en gardant le même horizon de temps utilisé c'est-à-dire 4h et le même nombre de véhicules c'est-à-dire 10. Les résultats obtenus sont reportés dans le tableau ci-dessous. Dans cette simulation, chaque véhicule prend en charge 3,5 demandes par heure en moyenne. Cette augmentation du nombre de demandes a un impact remarquable sur la qualité des solutions, comme on l'a observé dans le tableau V.5. Ce scénario est un exemple illustratif. En effet, l'écart entre l'algorithme RSMO et les deux autres algorithmes (recherche local, l'heuristique d'insertion) pour la distance parcourue et le nombre de stations visitées est passé de 11,09% à 3,25% et de 2,87% à 5%. Par conséquent, lorsque le taux d'arrivée des demandes augmente, nous constatons que la différence en termes de performances entre le RSMO et les autres méthodes augmente.

Numéro d'instance	heuristique d'insertion	Recherche Locale	RSMO
1	359	359	327
	18	18	14
	543	543	579
	65	65	65

2	405 16 520 77	448 14 530 82	491 15 593 71
3	310 19 523 73	283 15 502 70	367 15 582 70
4	310 22 555 61	256 17 538 67	346 15 581 70
5	449 22 578 75	365 18 549 78	482 17 589 61
Total	<b>1833</b> <b>97</b> <b>2719</b> <b>351</b>	<b>1767</b> <b>79</b> <b>2645</b> <b>366</b>	<b>2013</b> <b>76</b> <b>2924</b> <b>337</b>

Tableau V.6 : Simulation de 4 heures avec 10 véhicules et 35 demandes par heure

Dans le contexte dynamique, le temps d'exécution est intéressant pour évaluer la performance de la méthode d'optimisation lors d'un événement survenu dans l'horizon temporel d'une journée de service. Dans le tableau V.7, nous présentons les temps de calcul pour chaque méthode d'optimisation (heuristique d'insertion, la recherche locale et l'algorithme RSMO). Nous remarquons que l'heuristique d'insertion est plus rapide, mais elle est susceptible d'exploiter seulement une petite partie de l'espace de recherche entre deux événements.

Tandis que la procédure de recherche locale et l'algorithme RSMO présentent des temps de calcul importants. Mais, ils présentent des résultats de bonne qualité. Nous mentionnons que les tests ont été effectués sur un ordinateur portable Dell B14DEE640C avec Intel Core 2 Duo 2,0 GHz comme processeur et 2Go de mémoire RAM. Ces méthodes ont été développées sur la plate-forme multi agent JADE en utilisant le langage JAVA.

L'heuristique d'insertion	Recherche Locale	RSMO
19s	38s	52s

Tableau V.7 : Temps de calcul en seconde

## V.5 Conclusion :

Les PTD statique et dynamique sont des problèmes de complexité importante (NP-Difficile). Les méthodes exactes ainsi que les méthodes heuristiques centralisées sont incapables de résoudre ce problème dans un temps raisonnable. Il faut concevoir des systèmes distribués basés sur des heuristiques et des méta-heuristiques pour décentraliser le traitement des approches de résolution des problèmes complexes. En effet, la distribution des tâches pour résoudre le PTD permet non seulement de diminuer la complexité du problème mais aussi permet une bonne présentation du monde réel.

Dans ce chapitre, nous avons présenté en premier lieu le concept de la distribution avec la méthodologie des systèmes multi-agents. Une définition du SMA ainsi que l'aspect comportemental des agents ont été détaillés dans ce chapitre. Après avoir décrit les SMA, nous avons présenté notre approche basée sur l'algorithme de RSMO pour la résolution du PTD. Puis nous avons présenté les résultats de l'application de notre approche distribuée sur le PTD statique et dynamique.



# Chapitre VI :

## Conclusion et Perspectives

### VI. 1. Conclusion :

Dans ce travail de thèse, nous avons traité le Problème de Transport à la Demande. Ce problème présente un intérêt majeur à la fois pour les compagnies de transport ainsi que pour les utilisateurs des moyens de transport. Cet intérêt vient du fait que le transport à la demande est classé parmi les transports collective-individualisé. En effet, Le PTD remplace le transport public et le véhicule individuel. Ce système a un impact direct sur la qualité du service offert aux usagers. En effet, en cas de congestion des systèmes de transport public, ces derniers souhaitent avoir des informations fiables et des solutions alternatives, afin de faire leurs voyages.

Pour implanter des systèmes de transport à la demande, plusieurs projets ont été lancés partout dans le monde pour la conception de ce dernier. Ces projets ont des impacts économique et écologique. En effet, après une étude bibliographique détaillée faite dans cette thèse, nous avons pu constater que les systèmes de transport à la demande existants ne respectent pas tous les objectifs fixés par la compagnie de transport et les voyageurs. Il n'existe pas une stratégie d'optimisation des objectifs du problème. Notre contribution dans cette thèse est la résolution du PTD dans le contexte statique et dynamique d'une manière optimisé.

Avant de contribuer dans la résolution du PTD, nous avons étudiés les différentes méthodes et approches présentent dans la littérature et analysés les travaux effectués sur le problème qui nous intéresse. Nous avons pu positionner notre approche par rapport aux approches existantes. Ces méthodes sont généralement des méta-heuristiques permettant de trouver une solution aux problèmes dans un temps raisonnable et applicable à la situation temps réel contrairement aux méthodes exactes qui en sont incapable. En effet, la complexité du PTD vient du fait que le nombre des combinaisons des itinéraires de voyage est important, les méthodes exactes sont donc à éviter pour la résolution de ce type de problème.

Dans ce travail de recherche, nous avons contribué en premier lieu à la modélisation mathématique du PTD. Cette modélisation est composée principalement de deux grands objectifs ou critères. Le premier est l'objectif économique, il modélise le total de la distance parcourue ainsi que la durée des routes planifiées. Cet objectif est en faveur de la compagnie

de transport prenant en charge le transport de personnes. Le deuxième objectif que nous avons modélisé est la qualité du service rendue aux voyageurs. Cet objectif est composé de trois critères (temps mis pour le voyage, le nombre de stations visitées et le niveau de satisfaction du client en termes de véhicule). Il est à noter que ces deux objectifs sont antagonistes. Cette modélisation a été décrite dans le chapitre III.

Après avoir, modélisé le PTD nous avons développé une approche basée sur l'algorithme de recuit simulé pour la résolution de ce problème. En effet, nous avons suivi des étapes pour améliorer notre approche. Notre première contribution a été la résolution du PTD statique en utilisant l'algorithme de Recuit Simulé (RS) avec une méthode agrégative. Cette méthode présente plusieurs limites telles que (la favorisation d'un objectif par rapport à un autre). Pour remédier à cette lacune, nous avons appliqué l'algorithme de Recuit Simulé Multi-Objectif (RSMO) pour la résolution du PTD. Nous notons que nous sommes les premiers à avoir appliqué cet algorithme sur le PTD. Nous avons obtenu des résultats très intéressants dans le contexte statique et dynamique en comparaison avec les résultats obtenus par les autres méthodes. Ces bons résultats viennent du fait que l'algorithme de RSMO utilise la notion de dominance pour trouver des solutions Pareto optimales. Aussi notre choix de l'algorithme de RSMO est justifié par sa capacité de trouver des résultats en temps réel dans le contexte dynamique. Ces contributions sont détaillées dans le chapitre IV.

Nous avons amélioré la performance de notre approche pour s'adapter aux critères de dynamicité et de l'utilisation des véhicules hétérogènes. Cette adaptation est faite via la distribution du traitement de notre approche sur un Système Multi-Agents (SMA). Ce SMA permet à la fois, de modéliser et de résoudre le PTD. Pour cela, nous avons développé trois principaux agents ; qui sont, l'agent *client*, l'agent *centre d'appels* et l'agent *véhicule*. L'algorithme de RSMO est implémenté comme un comportement pour l'agent *véhicule*. Ce comportement lui permet de planifier les routes tout en tenant compte des différents critères. Ensuite, l'agent *centre d'appels*, choisi parmi les solutions proposés par les agents *véhicules* la meilleure solution en termes de coût et de qualité de service. Nos solutions ont été comparées avec l'algorithme génétique et la recherche tabou dans le cas statique. Par contre dans le cas dynamique, il n'existe pas de données standards pour effectuer des comparaisons. Nous avons développé une méthode de recherche locale et une méthode d'insertion pour qu'on puisse connaître la position de notre approche par rapport à ces méthodes dans le contexte dynamique. Ces contributions sont détaillées dans le chapitre V.

## VI. 2. Perspectives :

Dans cette thèse, nous avons pu dégager des perspectives pour des futurs travaux de recherche. En effet, nous envisageons en premier lieu d'effectuer une hybridation de l'algorithme de RSMO avec d'autres méta-heuristiques comme la recherche tabou. Ce choix est justifié par le fait que la recherche tabou permet de mémoriser les solutions et donc, de permettre une bonne exploration de l'espace de recherche. Il est à noter également que l'algorithme de recherche tabou est caractérisé par un temps de calcul réduit d'où son adaptation à l'environnement dynamique.

D'un autre point de vue, nous envisageons la modification de notre architecture SMA afin de décomposer la tâche de l'agent *véhicule*. Nous espérons qu'avec cette décomposition nous aurons une amélioration des résultats. Le principe général de cette décomposition est le suivant : la répartition des critères à optimiser sur des sous-agents véhicule où chaque sous-agent optimise séparément l'objectif accordé. Puis une négociation entre les agents est établie pour générer la meilleure solution. Cette négociation peut être également utilisée entre les agents véhicules eux même pour collaborer afin d'insérer une nouvelle demande, voir l'échange des demandes pour agir comme une méthode de recherche locale.

En ce qui concerne les données utilisés pour valider notre approche, nous avons utilisé deux types de données. Ces données sont le benchmark de (Cordeau et Laporte, 2003) [34] pour le PTD statique et des données aléatoires pour le PTD dynamique. En effet, ces données ne reflètent pas le cas réel, nous envisageons donc d'appliquer notre approche sur des données réelles. Nous espérons avoir des données réelles au pré du service *handipole* de la compagnie de transport *transpole* à Lille (France).



# Les Références Bibliographiques

- [1] ZIDI K, Système Interactif d'aide au déplacement Multimodal, thèse de doctorat, Ecole Centrale de Lille, 13 décembre 2006.
- [2] [http://www.insee.fr/fr/themes/document.asp?ref\\_id=BDF06](http://www.insee.fr/fr/themes/document.asp?ref_id=BDF06)
- [3] Institut de la Ville en Mouvement – Paris, Séminaire du 21 novembre 2007  
[http://www.ville-en-mouvement.com/mobilite\\_des\\_salaries/indexarticle.html](http://www.ville-en-mouvement.com/mobilite_des_salaries/indexarticle.html)
- [4] Le covoiturage en France et en Europe – État des lieux et perspectives, Certu, 2007  
[http://www.certu.fr/fr/Systèmes\\_de\\_transports-26/Technologies\\_des\\_transportsn84/catalogue/product\\_info.php?products\\_id=1956&language=fr](http://www.certu.fr/fr/Systèmes_de_transports-26/Technologies_des_transportsn84/catalogue/product_info.php?products_id=1956&language=fr)
- [5] Cordeau J-F, Laporte G, Savelsbergh MWP, Vigo D (2007) Vehicle Routing, In: C. Barnhart and G. Laporte (eds.), Transportation, Amsterdam: Elsevier
- [6] Cordeau J-F (2006) A Branch-and-cut algorithm for the dial-a-ride problem, Operations Research 54:573–586
- [7] Savelsbergh MWP (1992) The vehicle routing problem with time windows: Minimizing route duration, ORSA Journal on Computing 4:146–154
- [8] Sexton T, Bodin LD (1985) Optimizing single vehicle many-to-many operations with desired delivery times: I. Scheduling, Transportation Science 19:378–410
- [9] Dumas Y, Desrosiers J, Soumis F (1989) Large scale multi-vehicle dial-a-ride problems, Les Cahiers du GERAD, G–89–30, HEC Montréal
- [11] Psaraftis HN (1980) A dynamic programming approach to the single-vehicle, many-to-many immediate request dial-a-ride problem, Transportation Science 14:130–154
- [12] Psaraftis HN (1983) An exact algorithm for the single-vehicle many-to-many dial-a-ride problem with time windows, Transportation Science 17:351–357
- [13] Sexton T (1979) The single vehicle many-to-many routing and scheduling problem, Ph.D. dissertation, SUNY at Stony Brook
- [14] Sexton T, Bodin LD (1985a) Optimizing single vehicle many-to-many operations with desired delivery times: I. Scheduling, Transportation Science 19:378–410
- [15] Sexton T, Bodin LD (1985b) Optimizing single vehicle many-to-many operations with desired delivery times: II. Routing, Transportation Science 19:411–435

- [16] Desrosiers J, Dumas Y, Soumis F (1986) A dynamic programming solution of the large-scale single vehicle dial-a-ride problem with time windows, *American Journal of Mathematical and Management Sciences* 6:301–325
- [17] Psaraftis HN (1988) Dynamic vehicle routing problems, In: B.L. Golden, and A.A. Assad (eds.), *Vehicle Routing: Method and Studies*. Amsterdam: North-Holland, 223–248
- [18] Mitrović-Minić S, Krishnamurti R, Laporte G (2004) Double-horizon based heuristics for the dynamic pickup and delivery problem with time windows, *Transportation Research B* 38:669–685
- [19] Jaw J, Odoni AR, Psaraftis HN, Wilson NHM (1986) A heuristic algorithm for the multi-vehicle advance-request dial-a-ride problem with time windows, *Transportation Research B* 20:243–257
- [20] Bodin LD, Sexton T (1986) The multi-vehicle subscriber dial-a-ride problem, *TIMS Studies in Management Science* 2:73–86
- [21] Desrosiers J, Dumas Y, Soumis F, Taillefer S, Villeneuve D (1991) An algorithm for mini-clustering in handicapped transport, *Les Cahiers du GERAD*, G-91-02, HEC Montreal
- [22] Ioachim I, Desrosiers J, Dumas Y, Solomon MM (1995) A request clustering algorithm for door-to-door handicapped transportation, *Transportation Science* 29:63–78
- [23] Toth P, Vigo D (1996) Fast local search algorithms for the handicapped persons transportation problem In: I.H. Osman, and J.P. Kelly (eds.), *Meta-heuristics: Theory and applications*. Boston: Kluwer, 677–690
- [24] Borndorfer R, Klostermeier F, Grötschel M, Küttner C (1997) *Telebus Berlin: Vehicle scheduling in a dial-a-ride system*, Technical Report SC 97-23, Konrad-Zuse-Zentrum für Informationstechnik, Berlin
- [25] Brotcorne L, Laporte G, Semet F (2003) Ambulance location and relocation models, *European Journal of Operational Research* 147:451–468
- [26] Aldaihani M, Dessouky MM (2003) Hybrid scheduling methods for paratransit operations, *Computers & Industrial Engineering* 45:75–96
- [27] Rekiek B, Delchambre A, Saleh HA (2006) Handicapped person transportation: An application of the grouping genetic algorithm, *Engineering Application of Artificial Intelligence* 19:511–520

- [28] Xiang Z, Chu C, Chen H (2006) A fast heuristic for solving a large-scale static dial-a-ride problem under complex constraints, *European Journal of Operational Research* 174:1117–1139
- [29] Wong KI, Bell MGH (2006) Solution of the dial-a-ride problem with multi-dimensional capacity constraints, *International Transactions in Operational Research* 13:195–208
- [30] Wolfler Calvo R, Colomi A (2006) An effective and fast heuristic for the dial-a-ride problem, *4OR: A Quarterly Journal of Operations Research*, forthcoming
- [31] Ropke S, Cordeau J-F, Laporte G (2006) Models and branch-and-cut algorithms for pickup and delivery problems with time windows, *Networks*, forthcoming
- [32] Melachinoudis E, Ilhan AB, Min H (2007) A dial-a-ride problem for client transportation in a health-care organization, *Computers & Operations Research* 34:742–759
- [33] Jørgensen RM, Larsen J, Bergvinsdottir KB (2007) Solving the dial-a-ride problem using genetic algorithms, *Journal of the Operational Research Society*, forthcoming
- [34] Cordeau J-F, Laporte G (2003) A tabu search heuristic for the static multi-vehicle dial-a-ride problem, *Transportation Research B* 37:579–594
- [35] Claudio Cubillos, Nivaldo Rodríguez, Enrique Urra (2009), *Application of Genetic Algorithms for the DARPTW Problem*. *International Journal of Computers, Communications & Control*, June 2009
- [36] Zidi I. Zidi K. , Ghedira K, . Mesghouni K. (2010), *A Multi-Objective Simulated Annealing for the Multi-Criteria Dial a Ride Problem* 11th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design, and Evaluation of Human-Machine Systems , Valenciennes France, 2010.
- [37] Madsen OBG, Ravn HF, Rygaard JM (1995) A heuristic algorithm for the a dial-a-ride problem with time windows, multiple capacities, and multiple objectives, *Annals of Operations Research* 60:193–208
- [38] Jaw J, Odoni AR, Psaraftis HN, Wilson NHM (1986) A heuristic algorithm for the multi-vehicle advance-request dial-a-ride problem with time windows, *Transportation Research B* 20:243–257.
- [39] Teodorovic D, Radivojevic G (2000) A fuzzy logic approach to dynamic dial-a-ride problem, *Fuzzy Sets and Systems* 116:23–33.
- [40] Colomi A, Righini G (2001) Modeling and optimizing dynamic dial-a-ride problems, *International Transactions in Operational Research* 8:155–166

- [41] Coslovich L, Pesenti R, Ukovich W (2006) A two-phase insertion technique of unexpected customers for a dynamic dial-a-ride problem, *European Journal of Operational Research* 175:1605–1615
- [42] Brotcorne L, Laporte G, Semet F (2003) Ambulance location and relocation models, *European Journal of Operational Research* 147:451–468
- [43] Ulungu E.L and Teghem J (1994). The two-phases method: an efficient procedure to solve biobjective combinatorial optimization problems. *Foundations of Computing and Decision Sciences*, 20(2):149-165, 1994.
- [44] Ehrgott M. and Gandibleux X (2000). A survey and annotated bibliography of multiobjective combinatorial optimization. *OR Spektrum*, 22:425-460, 2000.
- [45] Deb K, Pratap A, and Meyarivan T (2001) Constrained test problems for multi-objective evolutionary optimization. In *Proceedings of Evolutionary Multi-Criterion Optimization*, pages 284-298, 2001.
- [46] Miettinen K (1999) *Nonlinear multiobjective optimization*. Kluwer Academic Publishers, 1999.
- [47] Collette Y et Siarry P (2002) *Optimisation multiobjectif*. Eyrolles, 2002.
- [48] Hwang, C. and Masud, A (1979). Multiple objective decision making - methods and applications. In *Lectures Notes in Economics and Mathematical Systems*, volume 164. Springer-Verlag, Berlin.
- [49] Ishibuchi, H. and Murata, T. (1998). A multi-objective genetic local search algorithm and its application to flow shop scheduling. *IEEE Transactions on Systems, Man, and Cybernetics Part C: Applications and Reviews*, 28(3) :392–403.
- [50] Veldhuizen, D. V., Sandlin, B., Marmelstein, R., Lamont, G., and Terzuoli, A. (1997). Finding improved wire-antenna geometries with genetic algorithms. In Chawdhry, P., Roy, R., and Pant, P., editors, *Soft Computing in Engineering Design and Manufacturing*, pages 231–240, London. Springer Verlag.
- [51] Ritzel, B., Eheart, J., and Ranjithan, S. (1994). Using genetic algorithms to solve a multiple objective groundwater pollution problem. *Water Resources Research*, 30(5) :1589–1603.
- [52] Wienke, P., Lucasius, C., and Kateman, G. (1992). Multicriteria target optimization of analytical procedures using a genetic algorithm. *Analytical Chimica Acta*, 265(2): 211–225.

- [53] Coello, C. (1998). Using the min-max method to solve multiobjective optimization problems with genetic algorithms. In IBERAMIA'98, LNCS. Springer-Verlag.
- [54] Schaffer, J. (1985). Multiple objective optimization with vector evaluated genetic algorithms. In Grefenstette, J., editor, ICGA Int. Conf. on Genetic Algorithms, pages 93–100. Lawrence Erlbaum.
- [55] Richardson, J., Palmer, M., Liepins, G., and Hilliard, M. (1989). Some guidelines for genetic algorithms with penalty functions. In Third Int. Conf. on Genetic Algorithms ICGA'3, pages 191–197.
- [56] Surry, P., Radcliffe, N., and Boyd, I. (1995). A multi-objective approach to constraint optimization of gas supply networks: The COMOGA method. In Fogarty, T., editor, Evolutionary Computing, AISB Workshop, LNCS, pages 166–180, Sheffield, U.K. Springer-Verlag.
- [57] Jones, B., Crossley, W., and Lyrintzis, A. (1998). Aerodynamic and aeroacoustic optimization of airfoils via a parallel genetic algorithm. In Proc. of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, number AIAI-98-4811, pages 1–11.
- [58] Allenson, R. (1992). Genetic algorithms with gender for multi-function optimisation. Technical Report EPCC-SS92-01, Edinburg Parallel Computing Center, Edinburg, Scotland.
- [59] Lis, J. and Eiben, A. (1996). A multi-sexual genetic algorithm for multi-objective optimization. In Fukuda, T. and Furuhashi, T., editors, Int. Conf. on Genetic Algorithms ICGA, pages 59–64, Nagoya, Japan.
- [60] Goldberg, D. E. (1989). Genetic algorithms in search, optimization, and machine learning. Addison-Wesley.
- [61] Fonseca, C. and Fleming, P. (1995b). An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation*, 3(1):1–16.
- [62] Bentley, P. and Wakefield, J. (1997). *Soft Computing in Engineering Design and Manufacturing*, chapter Finding acceptable Pareto-optimal solutions using multiobjective genetic algorithms, pages 231–240. Springer Verlag, London.
- [63] Srinivas, N. and Deb, K. (1995). Multiobjective optimisation using non-dominated sorting in genetic algorithms. *Evolutionary Computation*, 2(8) :221–248.

- [64] Zhou, G. and Gen, M. (1999). Genetic algorithm approach on multi-criteria minimum spanning tree problem. *European Journal of Operational Research*, 114:141–152
- [65] Abdelaziz, F., Krichen, S., and Chaouachi, J. (1999). Meta-heuristics: Advances and trends in local search paradigms for optimization, chapter A hybrid heuristic for multi-objective knapsack problems, pages 205–212. Kluwer Academic Publishers.
- [66] Parks, G. and Miller, I. (1998). Selective breeding in a multiobjective genetic algorithm. In *Parallel Problem Solving from Nature PPSN'98*, pages 250–259, Amsterdam. Springer-Verlag.
- [67] Halhal, D., Walters, G., Ouazar, D., and Savic, D. (1997). Water network rehabilitation with a structured messy genetic algorithm. *Journal of Water Resources Planning and Management*, 123(3) :137–146.
- [68] Baker, J. (1985). Adaptive selection methods for genetic algorithms. In Grefenstette, J., editor, *Int. Conf. on Genetic Algorithms and their Applications*, pages 101–111, Pittsburg. Lawrence Erlbaum.
- [69] Horn, J. and Nafpliotis, N. (1993). Multiobjective optimization using the niched pareto genetic algorithm. Technical Report 93005, University of Illinois, Urbana-Champaign.
- [70] Cunha, A., Oliveira, P., and Covas, J. (1997). Use of genetic algorithms in multicriteria optimization to solve industrial problems. In Back, T., editor, *Seventh Int. Conf. on Genetic Algorithms ICGA'97*, pages 682–688, San Mateo, California. Morgan Kaufmann.
- [71] Tamaki, H., Kita, H., and Kobayashi, S. (1996). Multi-objective optimization by genetic algorithms : A review. In *IEEE Int. Conf. on Evolutionary Computation ICEC'96*, pages 517–522.
- [72] Sen, T., Raiszadeh, M., and Dileepan, P. (1988). A branch and bound approach to the bicriterion scheduling problem involving total flowtime and range of lateness. *Management Science*, 34(2) :254–260.
- [73] Stewart, B. and White, C. (1991). Multiobjective A\*. *Journal of the ACM*, 38(4):775–814.
- [74] White, D. (1982). The set of efficient solutions for multiple-objectives shortest path problems. *Computers and Operations Research*, 9 :101–107.

- [75] Gandibleux, X., Mezdaoui, N., and Freville, A. (1996). A tabu search procedure to solve multiobjective combinatorial optimization problems. In Caballero, R., Ruiz, F., and Steuer, R., editors, Second Int. Conf. on Multi-Objective Programming and Goal Programming MOPGP'96, pages 291–300, Torremolinos, Spain. Springer-Verlag.
- [76] Ulungu, E. (1993). Optimisation combinatoire multicritère : détermination de l'ensemble des solutions efficaces et méthodes interactives. PhD thesis, Université de Mons-Hainaut.
- [77] Gandibleux, X., Libert, G., Cartignies, E., and Millot, P. (1994). SMART : étude de la faisabilité d'un solveur de problèmes de mobilisation de réserve tertiaire. *Revue des systèmes de Décision*, 3(1) :45–67.
- [78] Srinivas, N. and Deb, K. (1995). Multiobjective optimisation using non-dominated sorting in genetic algorithms. *Evolutionary Computation*, 2(8) :221–248.
- [79] Fonseca, C. and Fleming, P. (1995a). Multiobjective genetic algorithms made easy: Selection, sharing and mating restrictions. In *IEEE Int. Conf. on Genetic Algorithms in Engineering Systems: Innovations and Applications*, pages 45–52, Sheffield, UK.
- [80] Wienke, P., Lucasius, C., and Kateman, G. (1992). Multicriteria target optimization of analytical procedures using a genetic algorithm. *Analytical Chimica Acta*, 265(2) :211–225.
- [81] Kirkpatrick S, Gelatt C.D, and Vecchi M.P. Optimization by simulated annealing. *Science*, 220:671–680, 1983.
- [82] Aarts E.H.L and Korst J. Simulated annealing and boltzmann machines: a stochastic approach to combinatorial and neural computing. Wiley, Chichester, 1989.
- [83] Vidal R.V. (1993). Applied simulated annealing. *Lecture Notes in Economics and Mathematical Systems*, 396, 1993. Springer-Verlag.
- [84] Koulamas C., Anthony S.R., and Jean R. (1994). A survey of simulated annealing application to operations research problems. *OMEGA*, 22:41-56, 1994.
- [85] Glover F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, 13(5):533-549, 1986.
- [86] Glover F. and Laguna M.(1997) Tabu search. Kluwer Academic Publishers, 1997.

- [87] Back T., Fogel D.B., Michalewicz Z., and Baeck T (1997). Handbook of Evolutionary Computation. Institute of Physics Publishing and Oxford University Press, 1997.
- [88] Goldberg D.E. (1989) Genetic algorithms for search, optimization, and machine learning. Reading, MA: Addison-Wesley, 1989
- [89] Schwefel H-P. (1981) Numerical optimization of computer models. Wiley, Chichester, 1981.
- [90] Fogel D. (2000) Evolutionary Computation: Toward a New Philosophy of Machine Intelligence (second edition). IEEE Press, 2000.
- [91] Inge Li Gørtz, (2006). Hardness of Preemptive Finite Capacity Dial-a-Ride. Institut for Matematik og Datalogi Syddansk Universitet, <http://www.imada.sdu.dk> Preprints 2006 No. 4 February 2006 ISSN No. 0903-3920.
- [92] Boudali I., Fki W., Ghédira K., (2004). How to deal with the VRPTW by using multi-agent coalitions. Fourth International Conference on Hybrid Intelligent Systems (HIS'04), Kitakyushu, Japan, 2004
- [93] Cordeau, J.F. and Laporte G., (2006). The Dial-a-Ride Problem: Models and Algorithms. Les Cahiers du GERAD G-2006-78 Copyright c 2006 GERAD
- [94] Suman B. and Kumar P. (2006), .A survey of simulated annealing as a tool for single and multiobjective optimization,. Journal of the Operational Research Society, vol. 57, no. 10, pp. 1143-1160, 2006.
- [95] Mauri G.R, Lorena L.A.N, (2006). A Multiobjective Model and Simulated Annealing Approach for a Dial-a-Ride Problem. Workshop dos Cursos de Computação 2006.
- [96] Ferber J. (1995) Les Systèmes Multi-agents : Vers un intelligence collective. Informatique et Intelligence Artificielle, InterEditions, Paris, 1995.
- [97] Labrie M.A (2004), Langage de communication agent basé sur les engagements par l'entremise des jeux de dialogue, Mémoire présenté à la Faculté des études supérieures de l'Université Laval pour l'obtention du grade de maître des sciences, Janvier 2004
- [98] Florez-Mendez R.A (1999). Towards a standardization of Multi-Agent System Frameworks. ACM Crossroads Student Magazine, Canada, 1999
- [99] Sycara K.P. (1998). Multi-Agent Systems. American Association for Artificial Intelligence, AI Magazine, pages 79-92, 1998.

- [100] Vauvert G. and Seghrouchni A.E (2000). Formation de coalitions pour agents rationnels. In : Proceedings des JLIPN'2000, pages 11-12, Villetaneuse, France, Septembre, 2000.
- [101] Zidi I. , Zidi K. , Mesghouni K., Ghedira K. (2011) ,A Multi-Agent System based on the Multi-Objective Simulated Annealing Algorithm for the Static Dial a Ride Problem, 18th World Congress of the International Federation of Automatic Control (IFAC).
- [102] Bergvinsdottir K.B., (2004) . The genetic algorithm for solving the dial-a-ride problem. Master Thesis of Science in Engineering. Department of Informatics and Mathematical Modelling (IMM), Technical University of Denmark (DTU).
- [103] Durfee E. H. and Lesser V. R. (1991), Partial global planning: A coordination framework for distributed hypothesis formation, IEEE Transactions on Systems, Man, and Cybernetics, 21(5), pp.1167-1183, September-October 1991.
- [104] Chaib-draa B. (1996) Interaction between agents in routine, familiar and unfamiliar situations. International Journal of Intelligent and Cooperative Information Systems, 1(5):7-20, 1996.
- [105] Ljungberg M. and Lucas A. (1992), “The oasis air-traffic management system”, In Proceedings of the Second Pacific Rim International Conference on Artificial Intelligence, PRICAI '92, Seoul, Korea, 1992.
- [106] Cammarata S. McArthur D. and Steeb R. (1983) Strategies of cooperation in distributed problem solving. In Proceedings of the Eighth International Joint Conference on Artificial Intelligence (IJCAI-83), Karlsruhe, Germany, 1983.
- [107] Fischer K., Müller J.P, Pischel M.and Schier D. (1995) A Model for Cooperative Transportation Scheduling, In Proceedings of the First International Conference on MAS, AAAI Press/MIT Press, pp.109-116, Menlo Park, California,1995.
- [108] Chaib-draa B. (1994) Distributed Artificial Intelligence: An overview. In A. Ken, J. G. Williams, C. M. Hall, and R.Kent, editors, Encyclopedia Of Computer Science And Technology, volume 31, pages 215-243. Marcel Dekker, Inc, 1994.
- [109] Gruer P., Hilaire V. and Koukam A. (2001) Multi-Agent Approach to Modelling and Simulation of Urban Transportation Systems, Proceedings of the 2001 IEEE SMC Conference, 6-10 October 2001, Tucson, Arizona, USA, pp.2499-2504.

- [110] Said El Hmam M., Jolly D., Abouaissa H., Benasser A. (2005) Modélisation Hybride du Flux de Trafic. Workshop avec école intégrée Méthodologies et Heuristiques pour l'Optimisation des Systèmes Industriels 24-26 Avril 2005, Hammamet, Tunisie. Pages193-198.
- [111] Sausso B., Maouche S., Hayat S., Dekokere A. Dumont A (2000), Elaboration et mise au point d'un système d'aide à la décision pour la gestion du réseau de transport collectif de Montbéliar, Rapport d'étape INRETS-I3D-LAIL, Appui à la modélisation du système multi-agents, juin 2000.
- [112] Laïchour H., Maouche S. and Mandiau R. (2001) Traffic Control Assistance in Connection nodes, Proceedings of the 2001 IEEE SMC Conference, 6-10 October 2001, Tucson, Arizona, USA.
- [113] Zidi I. , Zidi K. , Mesghouni K., Ghedira K. (2010) , Application de l'Algorithme de Recuit Simulé pour la Résolution d'un Problème de Transport à la Demande Bi-objectif, La Cinquième Conférence Internationale en Recherche Opérationnelle, Marrakech Maroc,2010.
- [114] Talbi E-G., Rahoual M., Mabed M.H, and Dhaenens C. (2001) New genetic approach for multicriteria optimization problems: Application to the flow shop. In Evolutionary Multi-criterion Optimization (EMO), volume LNCS 1993, pages 416–428. Zurich, Switzerland, 2001.
- [115] Issam Zidi, Kamel Zidi, Khaled Ghedira, Khaled Mesghouni, 11th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design, and Evaluation of Human-Machine Systems ,Valenciennes France,2010. IEEE.
- [116] Issam Zidi, Kamel Zidi, Khaled Ghedira, Khaled Mesghouni, special issue on engineering management of the International journal on engineering management and economics

## **Modélisation et Optimisation du Problème de Transport à la Demande Multicritère et Dynamique**

**Résumé :** Le Problème de Transport à la Demande (PTD), consiste à prendre en charge le transport des personnes à partir d'un lieu de départ vers un lieu d'arrivée. Il est caractérisé par un ensemble de demandes de transport et d'un nombre de véhicules disponible. L'ultime objectif dans ce travail de thèse est d'offrir une alternative optimisée au déplacement individuel et collectif.

Le PTD est classé parmi les problèmes NP-difficile, la majorité des travaux de recherche ont été concentrés sur l'utilisation des méthodes approchées pour le résoudre.

En plus, il s'avère également multicritère, la solution proposée dans ce travail permet à la fois une réduction du temps de voyage concernant les demandes de transport ainsi que la réduction de la distance parcourue. Dans ce rapport de thèse, nous proposons notre contribution à l'étude et à la résolution du problème de transport à la demande multicritère et dynamique en appliquant l'algorithme de recuit simulé multi-objectif. Une grande partie de notre travail concerne la conception, le développement et la validation des approches qui permettent de donner des solutions optimales ou quasi optimales, pour un PTD. Ces approches utilisent une méthode multicritère qui s'appuie sur l'algorithme de recuit simulé. La modélisation du PTD est représentée par une architecture multi-acteurs. Cette architecture nous montre l'aspect distribué du système, les interactions et les relations qui peuvent avoir lieu entre les différents acteurs. Nous présentons dans ce travail un Système Multi-Agents pour la planification des itinéraires des véhicules affectées au transport des voyageurs. Les agents de ce système utilisent le module d'optimisation développé dans la première partie.

**Mots-clefs :** Heuristiques, transport à la demande, optimisation multi-objectif, algorithme de recuit simulé multi-objectif, système multi-agents.

### **Modeling and Optimization a Dynamic and multicriteria Dial a Ride Problem.**

**Abstract:** The Dial a Ride Problem (DRP) is to take passengers from a place of departures to places of arrivals. Different versions of the dynamic Dial a Ride Problem are found in every day practice; transportation of people in low-density areas, transportation of the handicapped and elderly persons and parcel pick-up and delivery service in urban areas. In the DRP, customers send transportation requests to an operator. A request consists of a specified pickup location and destination location along with a desired departure or arrival time. The ultimate aim is to offer an alternative to displacement optimized individually and collectively. The DRP is classified as NP-hard problem that's why most research has been concentrated on the use of approximate methods to solve it. Indeed the DRP is a multi-criteria problem, the proposed solution of which aims to reduce both route duration in response to a certain quality of service provided. In this thesis, we offer our contribution to the study and solving the DRP in the application using a multi agent system based on the Multi-Objective Simulated Annealing Algorithm.

**Keywords:** Heuristics, Dial a Ride Problem, Passenger Transportation, Multi-Criteria Optimization, Multi-Objective Simulated Annealing Algorithm, Multi-Agent System