

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE  
UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À  
L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

COMME EXIGENCE PARTIELLE  
À L'OBTENTION DE LA  
MAÎTRISE EN GÉNIE DE LA PRODUCTION AUTOMATISÉE  
M.Ing.

PAR  
MUNTEANU, Gheorghe

MODÉLISATION MATHÉMATIQUE, SIMULATION ET OPTIMISATION  
DES RÉSEAUX DE TRANSPORT

MONTRÉAL, LE 20 FÉVRIER 2009  
© Droit réservés de Munteanu Gheorghe

**PRÉSENTATION DU JURY**  
**CE MÉMOIRE A ÉTÉ ÉVALUÉ**  
**PAR UN JURY COMPOSÉ DE**

M. Amar Ramudhin, directeur de mémoire  
Département de génie de la production automatisée à l'École de technologie supérieure

M. Michel Rioux, président du jury  
Département de génie de la production automatisée à l'École de technologie supérieure

M. Marc Paquet, membre du jury  
Département de génie de la production automatisée à l'École de technologie supérieure

**IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC**

**LE 20 FÉVRIER 2009**

**À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE**

## REMERCIEMENTS

Je tiens à remercier très sincèrement toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce travail.

Tout d'abord, M. Amar Ramudhin, mon directeur de mémoire qui m'a offert la chance de travailler avec lui. Je le remercie aussi pour les discussions que nous avons pu avoir et qui ont bien souvent permis d'enrichir mes connaissances.

Je tiens à remercier profondément M. Marc Paquet qui a non seulement accepté de faire partie de ce jury, mais qui a également eu la patience et la sympathie de m'écouter et de m'aider. Son aide a été décisive dans la réalisation de ce travail.

Je tiens à exprimer tout ma gratitude et mon respect à M. Michel Rioux pour le temps qu'il a bien voulu me consacrer en acceptant de présider le jury d'évaluation.

Je suis très reconnaissant à Mme. Doina Precup de l'Université McGill pour ses conseils, sa disponibilité et ses grandes compétences qui m'ont permis d'accomplir ce travail.

Aussi, je tiens à remercier tous mes amis et mes collègues avec qui j'ai partagé des moments très agréables au cours des dernières années.

Finalement, je voudrais remercier mon épouse Carmen et mon garçon Tudor qui m'ont soutenu et aidé tout au long de mon projet. Je leur dédie ce travail.

# **MODÉLISATION MATHÉMATIQUE, SIMULATION ET OPTIMISATION DES RÉSEAUX DE TRANSPORT**

MUNTEANU, Gheorghe

## **RÉSUMÉ**

Les recherches sur différents phénomènes physiques ou sur différents comportements sociaux ont motivé les chercheurs à développer des algorithmes permettant de résoudre des problèmes d'optimisations combinatoires complexes. Ces algorithmes d'optimisation constituent un sujet de recherche de plus en plus important tant du point de vue théorique que pratique.

Ce projet traite le comportement des différents algorithmes sur des problèmes non déterministes polynomiaux difficiles, tel que le problème du voyageur de commerce. Notre objectif vise l'implantation et l'étude de l'efficacité de ces algorithmes dans un contexte d'optimisation difficile.

Cette contribution complète les travaux de recherches sur les études des méta-heuristiques et elle peut s'appliquer tant aux problèmes académiques qu'aux problèmes réels.

# **MATHEMATICAL MODELING, SIMULATION AND OPTIMIZATION OF THE TRANSPORTATION SYSTEMS**

MUNTEANU, Gheorghe

## **ABSTRACT**

The research of different physical phenomena and social behaviour motivated the researchers to develop algorithms to solve complex combinatorial optimization problems. These optimization algorithms constitute an important topic of research from theoretical and practical points of view.

This project deals with the application of different algorithms on polynomial non deterministic hard problems, like the problem of traveling salesman. Our goal is to implement and study the efficiency of these algorithms in an optimization complex context.

This contribution complements the research studies on meta-heuristic methods and it can be applied to academic and real life problems.

## TABLE DES MATIÈRES

	Page
CHAPITRE 1 INTRODUCTION.....	1
1.1 Contexte général.....	1
1.2 Contexte économique.....	1
1.3 Contexte environnemental.....	1
1.4 Problématique et objectifs de la recherche.....	2
1.5 Plan de travail.....	3
1.6 Conclusion.....	5
CHAPITRE 2 REVUE DE LA LITTÉRATURE.....	6
2.1 Problème du voyageur de commerce.....	6
2.1.1 Complexité du problème du voyageur de commerce.....	7
2.1.2 Formulation mathématique du problème du voyageur de commerce.....	8
2.2 Problème du voyageur de commerce symétrique et asymétrique.....	9
2.2.1 Problème du voyageur de commerce symétrique.....	10
2.2.2 Problème du voyageur de commerce asymétrique.....	10
2.3 Extensions au problème du voyageur de commerce.....	10
2.3.1 Problème du voyageur de commerce avec retour à charge.....	10
2.3.2 Problème du voyageur de commerce avec cueillette et livraison.....	12
2.3.3 Problème du voyageur de commerce avec la localisation de la demande sur les arcs.....	13
2.3.4 Autres types de problèmes du voyageur de commerce.....	14
2.4 Problème de tournées de véhicules.....	17
2.4.1 Formulation mathématique du problème de tournées de véhicules.....	19
2.4.2 Problème de tournées de véhicules standard avec capacité.....	21
2.4.3 Problème de tournées de véhicules standard avec flotte hétérogène.....	26
2.4.4 Problème de tournées de véhicules avec fenêtres de temps.....	27
2.4.5 Problème de tournées de véhicules multi-dépôts.....	35
2.5 Autres généralisations du problème de tournées de véhicules.....	36
2.5.1 Problème de tournées de véhicules périodique.....	36
2.5.2 Problème de tournée de véhicules stochastique.....	36
2.5.3 Problème de tournées de véhicules multiples.....	37
2.5.4 Problème de tournées de véhicules avec ravitaillements intermédiaires.....	37
2.6 Problème de la planification de tournées de véhicules scolaires.....	38
2.7 Conclusion.....	40
CHAPITRE 3 MÉTHODOLOGIE ET REVUE DES ALGORITHMES ÉTUDIÉS.....	41
3.1 Méthodologie et procédures proposées.....	41
3.2 Notion d'une méta-heuristique.....	41
3.3 Algorithme du recuit simulé dans la recherche opérationnelle.....	44
3.3.1 Historique du recuit simulé.....	44
3.3.2 Rappel de thermodynamique.....	44

3.3.3	Comment déterminer la température initiale ?.....	46
3.3.3.1	Décroissance de la température .....	47
3.4	Étapes d'un algorithme de recuit simulé.....	49
3.5	Algorithme génétique dans la recherche opérationnelle.....	52
3.5.1	Chronologie des algorithmes évolutionnaires.....	52
3.5.2	Fonctionnement des algorithmes génétiques .....	53
3.5.2.1	Objectif des algorithmes génétiques.....	53
3.5.2.2	Principe des algorithmes génétiques.....	53
3.5.2.3	Codage d'une population .....	55
3.5.2.4	Codage d'un algorithme génétique .....	56
3.5.2.5	Fonction d'évaluation d'un algorithme génétique.....	56
3.5.2.6	Sélection des individus dans un algorithme génétique.....	56
3.5.2.6.1	Méthode de roulette biaisée .....	57
3.5.2.6.2	Méthode de tournoi.....	58
3.5.2.6.3	Méthode de sélection stochastique.....	58
3.5.2.7	Recombinaison d'une population .....	59
3.5.2.8	Mutation d'une population.....	60
3.5.2.8.1	Mutation d'une population avec le codage binaire.....	60
3.5.2.8.2	Mutation stochastique d'une population.....	61
3.5.2.9	Élitisme d'une population.....	61
3.5.2.10	Réinsertion d'une population.....	61
3.6	Algorithmes de fourmis dans l'optimisation combinatoire .....	62
3.6.1	Chronologie des algorithmes chez les fourmis .....	62
3.6.2	Problème du plus court chemin chez les fourmis .....	63
3.6.3	Algorithme de fourmis ( <i>Ant System</i> ).....	64
3.6.4	Modélisation mathématique pour le problème du voyageur de commerce en utilisant l'algorithme de fourmis ( <i>Ant System</i> ).....	65
3.7	Conclusion .....	68
CHAPITRE 4 EXPÉRIMENTATIONS.....		69
4.1	Introduction.....	69
4.2	Instances utilisées pour résoudre le problème du voyageur de commerce .....	69
4.3	Expérimentations pour l'algorithme de recuit simulé.....	72
4.3.1	Améliorations sur la modalité de déterminer la température initiale.....	72
4.3.2	Description du programme qui détermine la température initiale .....	72
4.3.3	Résultats obtenus sur différentes instances utilisées.....	77
4.3.4	Génération des solutions initiales pour l'algorithme de recuit simulé dans le cas du voyageur de commerce.....	80
4.3.5	Procédures d'amélioration des arcs pour l'algorithme de recuit simulé dans le cas du voyageur de commerce.....	81
4.3.6	Principe d'un algorithme de recuit simulé avec une procédure d'améliorations des arcs.....	83
4.4	Expérimentations pour l'algorithme génétique.....	84
4.4.1	Paramétrage d'un algorithme génétique .....	84
4.4.2	Procédures d'améliorations des arcs.....	84

4.4.3	Usage d'un algorithme génétique avec une procédure d'amélioration pour résoudre le problème du voyageur de commerce .....	85
4.5	Expérimentations pour l'algorithme de fourmis .....	86
4.5.1	Résolution des différentes instances avec l'algorithme de fourmis Max Min.....	86
4.5.2	Paramétrage d'un algorithme de fourmis Max Min.....	87
4.5.3	Génération des solutions initiales pour l'algorithme de fourmis Max Min.....	88
4.5.4	État de convergence de l'algorithme Max Min.....	88
4.5.5	Calcul de bornes supérieures pour les instances du réseau de Côte Saint-Luc .....	89
4.5.6	Procédures d'amélioration des arcs .....	90
4.5.7	Usage d'un algorithme de fourmis Max Min avec une procédure d'amélioration pour résoudre le problème du voyageur de commerce.....	90
4.6	Conclusion .....	92
CHAPITRE 5 RÉSULTATS DES EXPÉRIMENTATIONS.....		93
5.1	Introduction.....	93
5.2	Conditions d'expérimentation.....	93
5.3	Présentation des données .....	94
5.3.1	Justification de l'utilisation du critère d'arrêt des algorithmes utilisés .....	94
5.4	Résultats expérimentaux et discussion.....	94
5.4.1	Résultats obtenus pour l'algorithme de recuit simulé avec une amélioration de type 2-opt, 3-opt et 4-opt .....	95
5.4.1.1	Discussions .....	95
5.4.2	Résultats obtenus pour l'algorithme génétique avec une amélioration type 2-opt, 3-opt et 4-opt .....	107
5.4.2.1	Discussions .....	107
5.4.3	Résultats obtenus pour l'algorithme de fourmis Max Min avec une amélioration 2-opt, 3-opt et 4-opt.....	112
5.4.3.1	Discussions .....	112
5.5	Conclusions.....	118
CHAPITRE 6 CONCLUSIONS ET PERSPECTIVES.....		122
6.1	Conclusions.....	122
6.2	Perspectives de recherche .....	123
ANNEXE I RÉSULTATS OBTENUS POUR LES DIFFÉRENTES INSTANCES UTILISÉES DANS LE CAS DU PROGRAMME DE TEMPÉRATURE DU RECUIT SIMULÉ .....		127
ANNEXE II SYNTHÈSE DES RÉSULTATS.....		144
BIBLIOGRAPHIE.....		169

## LISTE DES TABLEAUX

		Page
Tableau 2.1	Temps de calcul et nombre de possibilités (Tirée de Laporte ,1992) .....	8
Tableau 2.2	Comparaison des heuristiques pour le voyageur du commerce avec retour à charge (Extrait de Ghaziri et Osman ,2003).....	12
Tableau 2.3	Résultats obtenus en utilisant les algorithmes 3-opt et 4-opt.....	12
Tableau 2.4	Comparaison avec les meilleures solutions connues (Extrait de Paletta ,2002) .....	15
Tableau 2.5	Résultats du CVRP (Source : Achuthan, Caccetta <i>et al.</i> 1998) .....	24
Tableau 2.6	Comparaison de méta-heuristiques utilisée (Source : Prins, 2003) .....	25
Tableau 2.7	Résultats sur les problèmes de Christofides et Eilon (1989) (Doerner, Hartl <i>et al.</i> 2004).....	26
Tableau 2.8	Meilleures solutions pour les problèmes C1 (Source : Berger et Barkaoui, 2003) .....	29
Tableau 2.9	Meilleures solutions pour les problèmes C2 (Source : Berger et Barkaoui, 2003) .....	29
Tableau 2.10	Meilleures solutions pour les problèmes R1 (Source : Berger et Barkaoui, 2003) .....	30
Tableau 2.11	Meilleures solutions pour les problèmes R2 (Source : Berger et Barkaoui, 2003) .....	30
Tableau 2.12	Meilleures solutions pour les problèmes RC1 (Source : Berger et Barkaoui, 2003) .....	31
Tableau 2.13	Meilleures solutions pour les problèmes RC2 (Source : Berger et Barkaoui, 2003) .....	31
Tableau 2.14	Comparaison des résultats moyens entre les publications (Source : Berger et Barkaoui, 2003) .....	32
Tableau 2.15	Résultats de Li, Guo <i>et al.</i> (2004) .....	33
Tableau 2.16	Résultats de (Berger et Barkaoui, 2004) .....	34
Tableau 4.1	Instances utilisées pour résoudre le problème du voyageur de commerce .....	71

Tableau 4.2	Variation de la température médiane $T_0$ en fonction de la solution initiale $s_0$ .....	77
Tableau 4.3	Condition d'expérimentation pour l'algorithme génétique.....	84
Tableau 4.4	Condition d'expérimentation pour l'algorithme de fourmis Max Min.....	87
Tableau 4.5	Calcul des bornes supérieures pour les instances du réseau de Côte Saint-Luc .....	90
Tableau 5.1	Résultats obtenus pour l'algorithme de recuit simulé avec une amélioration 2-opt, 3-opt et 4-opt (choix de température égale à 100 unités) .....	97
Tableau 5.2	Écart-type obtenu pour les résultats du Tableau 5-1.....	99
Tableau 5.3	Résultats obtenus pour l'algorithme de recuit simulé avec une amélioration 2-opt, 3-opt et 4-opt (choix de température égale à 801 unités) .....	100
Tableau 5.4	Écart-type obtenu pour les résultats du Tableau 5-3.....	102
Tableau 5.5	Temps de simulation pour l'algorithme de recuit simulé avec une amélioration 2-opt, 3-opt et 4-opt (choix de la température égale à 100 unités) .....	103
Tableau 5.6	Temps de simulation pour l'algorithme de recuit simulé avec une amélioration 2-opt, 3-opt et 4-opt (choix de la température égale à 801 unités) .....	104
Tableau 5.7	Temps de simulation - amélioration calculée entre les choix de température à 100 et 801 unités .....	106
Tableau 5.8	Résultats obtenus pour l'algorithme génétique avec une amélioration 2-opt, 3-opt et 4-opt .....	108
Tableau 5.9	Écart-type obtenu pour les résultats du Tableau 5-8.....	110
Tableau 5.10	Temps de simulation pour l'algorithme génétique avec une amélioration 2-opt, 3-opt et 4-opt .....	111
Tableau 5.11	Temps simulation – l'écart de la valeur moyenne calculé entre les trois types d'algorithmes.....	112
Tableau 5.12	Résultats obtenus pour l'algorithme de fourmis Max Min avec une amélioration 2-opt, 3-opt et 4-opt.....	113

Tableau 5.13	Écart-type obtenu pour les résultats du Tableau 5-12.....	115
Tableau 5.14	Temps de simulation pour l'algorithme de fourmi Max Min avec une amélioration 2-opt, 3-opt et 4-opt .....	116
Tableau 5.15	Temps de simulation – écart de la valeur moyenne calculée entre les trois types d'algorithmes.....	117
Tableau 5.16	Comportement des algorithmes – problème du voyageur de commerce KROA100 119	
Tableau 5.17	Écart-type obtenu pour les résultats du Tableau 5-15.....	120
Tableau 5.18	Temps de simulation - problème du voyageur de commerce KROA100 .....	121

## LISTE DES FIGURES

		Page
Figure 1.1	Plan de lecture du mémoire.....	4
Figure 2.1	Réseau de 100 villes pour le voyageur de commerce (KROA100) .....	6
Figure 2.2	Classification des problèmes des tournées des véhicules.....	18
Figure 2.3	Tournée des véhicules (1 dépôt, trois véhicules). .....	21
Figure 3.1	Décroissance géométrique du paramètre $T$ (Extrait de Bowerman, 1995).....	48
Figure 3.2	Algorithme de la méthode de recuit simulé Source : Adapté de Kirkpatrick , Gelett <i>et al.</i> (1983). .....	51
Figure 3.3	Principe d'un algorithme évolutionnaire standard. (Source : Dorigo, 2003) .....	54
Figure 3.4	Exemple d'un codage réel et binaire (Dumas, 2007). .....	55
Figure 3.5	La méthode de la roulette biaisée (Goldberg,1989). .....	55
Figure 3.6	Méthode de tournoi entre deux individus avec rééligibilité. (Source : Goldberg, 1991).....	58
Figure 3.7	Opérateurs de recombinaison les plus fréquemment utilisés. ....	60
Figure 3.8	Mutation d'une population. ....	61
Figure 3.9	Modalité de trouver de la nourriture en prenant le plus court chemin. ....	64
Figure 4.1	Réseau routier de 40 points du secteur Côte Saint-Luc (Montréal). ....	70
Figure 4.2	Algorithme de la méthode de calcul qui détermine la température initiale pour le recuit simulé.....	74
Figure 4.3	Valeur de la température obtenue pour une solution initiale égale à 25 000 unités.....	75
Figure 4.4	Valeur de la température obtenue pour une solution initiale de 100 unités.....	76
Figure 4.5	Perturbation d'une solution pour le programme qui détermine la température. ....	76

Figure 4.6	Temps de simulations et valeurs de la température $T_0$ médiane pour les séries A, B, C et D.....	79
Figure 4.7	Valeurs de la température médiane $T_0$ pour les séries A, B, C et D.....	79
Figure 4.8	Méthode des plus proches voisins appliquée au l'algorithme de Metropolis.....	81
Figure 4.9	Méthode 2-opt.....	82
Figure 4.10	Méthode 3-opt.....	82
Figure 4.11	Méthode 4-opt.....	82
Figure 4.12	Principe d'un algorithme de recuit simulé avec une procédure d'amélioration.....	83
Figure 4.13	Principe d'un algorithme génétique avec une amélioration 2-opt, 3-opt et 4-opt pour résoudre le problème du voyageur de commerce.....	85
Figure 4.14	Principe d'un algorithme de fourmis Max Min avec une amélioration 2-opt, 3-opt et 4-Opt.....	91
Figure 5.1	Recuit simulé avec une amélioration 2-opt, 3-opt et 4-opt (choix de la température égale à 100 unités).....	98
Figure 5.2	Problème du voyageur de commerce KROA100 (algorithme de recuit simulé avec 2-opt, choix de la température égale à 100 unités).....	98
Figure 5.3	Recuit simulé avec une amélioration 2-opt, 3-opt et 4-opt (choix de la température égale à 801 unités).....	101
Figure 5.4	Problème du voyageur de commerce KROA100 (algorithme de recuit simulé avec 2-opt, choix de la température égale à 801 unités).....	101
Figure 5.5	Temps de simulation pour l'algorithme de recuit simulé avec une amélioration 2-opt, 3-opt et 4-opt (choix de la température égale à 100 unités et 801 unités).....	105
Figure 5.6	Algorithme génétique avec une amélioration 2-opt, 3-opt et 4-opt.....	109
Figure 5.7	Temps de simulation de l'algorithme génétique avec 2-opt, 3-opt et 4-opt pour Kroa100.....	109
Figure 5.8	Problème du voyageur de commerce Kroa100 (algorithme de fourmi avec 3-opt).....	114

Figure 5.9	Algorithme de fourmis avec une amélioration 2-opt, 3-opt et 4-opt.....	114
Figure 5.10	Temps de simulation de l'algorithme de fourmis avec 2-opt, 3-opt et 4-opt pour Kroa100.....	117
Figure 6.1	Interface graphique - programme d'optimisation des émissions de gaz à effet de serre.....	124
Figure 6.2	Interface graphique - programme de paramétrage pour les véhicules.....	124
Figure 6.3	Interface graphique - programme d'optimisation des émissions de gaz à effet de serre qui utilise une approche basée sur l'algorithme génétique ...	125
Figure 6.4	Interface graphique – programme d'optimisation du trajet pour le transport scolaire.....	126

## LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

BATA	<i>Backtracking Adaptive Threshold Accepting</i> Méthode adaptative d'acceptation
BWTSP	<i>Black and White Traveling Salesman Problem</i> Voyageur de commerce avec des sommets noirs et blancs
CVRP	<i>Capacited Vehicle Routing Problem</i> Problème de tournées de véhicules avec capacité
CCC-TSP	<i>Cardinality Constrained Traveling Salesman Problem</i> Voyageur de commerce avec contraintes de cardinalité
GI	<i>Generalized Insertion</i> Phase d'insertion
HCPP	<i>Hierarchical Chinese Postman Problem</i> Problème du postier chinois
NPC	<i>Nondeterministic Polynomial Problem</i> Problème non polynomial complet
PVRP	<i>Periodic Vehicle Routing Problem</i> Problème de tournées des véhicules périodique
RPP	<i>Rural Postman Problem</i> Voyageur de commerce rural
SVRP	<i>Stochastic Vehicle Routing Problem</i> Problème stochastique de tournée de véhicules

TSPPC	<i>Travel Salesman Problem with Presence Constrained</i> Voyageur du commerce avec contraintes de présence
TSP	<i>Travelling Salesman Problem</i> Problème du voyageur de commerce
US	<i>Unstringing et Stringing</i> Phase de post-optimisation
VRP	<i>Vehicle Routing Problem</i> Problème de tournées de véhicules
VRPSTW	<i>Vehicle Routing Problem with Soft Time Windows</i> Problème de tournée des véhicules avec fenêtres de temps souples
VRPTW	<i>Vehicle Routing Problem with Time Windows</i> Problème de tournée de véhicules avec fenêtres de temps

## **CHAPITRE 1**

### **INTRODUCTION**

#### **1.1 Contexte général**

Le rythme de croissance de l'économie mondiale a augmenté de façon exponentielle au cours des trois dernières décennies. Le transport est considéré comme un vecteur essentiel dans le développement de l'économie, qui favorise l'intégration sociale, économique et environnementale. La croissance de la population, l'expansion urbaine, les nouvelles notions sur l'environnement, impose des mesures sur la façon d'améliorer les concepts de transport. Que l'on soit intéressé par l'optimisation ou par le transport, un grand défi à l'heure actuelle est d'élaborer des méthodes de résolution rapides et efficaces qui peuvent apporter des bénéfices pour les secteurs intéressés.

#### **1.2 Contexte économique**

Il existe en effet un lien très fort entre le transport et le développement économique. Ainsi, plus on développe des zones économiques, plus on a besoin du transport pour résoudre les problèmes liés au déplacement de personnes et de marchandises. En conséquence, toute amélioration dans les techniques de résolution de ces problèmes a un impact majeur sur le plan de développement économique.

#### **1.3 Contexte environnemental**

Alors que ces dernières années la société s'est préoccupée de la nécessité de réduire les gaz à effet de serre dans le contexte du protocole de Kyoto, peu d'études ont été menées dans le domaine du transport pour satisfaire les enjeux environnementaux.

L'application de différents algorithmes d'optimisation permet de résoudre plusieurs de ces types de problèmes tout en respectant les contraintes liées à l'environnement.

#### **1.4 Problématique et objectifs de la recherche**

Lorsqu'un problème se pose en recherche opérationnelle, il faut parfois faire des analyses ou définir des nouvelles méthodes de résolution car les données existantes ou les techniques sont soit incomplètes, soit ne sont pas bien harmonisées au cas traité. De même, il existe des problèmes pour lesquels la solution finale est difficile à obtenir étant donné leur complexité.

À la lumière de ces éléments, ce mémoire synthétise mes travaux de recherche réalisés durant ces deux dernières années au sein de LRCA (Laboratoire de recherche sur les chaînes d'approvisionnement) de l'École de technologie supérieure de Montréal, dans le cadre de ma maîtrise.

L'optimisation est considérée comme une des branches les plus importantes de la recherche opérationnelle et plusieurs études, théoriques et pratiques, lui sont consacrées. Parmi les nombreuses problématiques issues du domaine de l'optimisation, nous nous intéressons à l'évaluation des méta-heuristiques dans un contexte de résolution des problèmes non déterministes polynomiaux difficiles (NP-difficiles) avec le cas du voyageur de commerce.

Trois (3) types d'algorithmes sont proposés : l'algorithme de recuit simulé, l'algorithme génétique et l'algorithme de fourmis. De plus, trois (3) méthodes d'améliorations sont utilisées pour chacun de ces algorithmes (méthodes 2-Opt, 3-Opt et 4-Opt).

Étant donné qu'on utilise plusieurs algorithmes d'optimisation et que l'on parle de déterminer les différents paramètres de ces algorithmes, on propose de faire une étude sur un de ces paramètres peu étudiés dans la littérature et de donner une explication pour

sélectionner la valeur de ce paramètre. La température du recuit simulé s'avère un paramètre très difficile à déterminer. Puisqu'il n'existe pas à notre connaissance une explication pour la détermination de ce paramètre, on propose de l'étudier. L'analyse de ce paramètre est un objectif de ce mémoire.

Ce mémoire a donc les objectifs suivants :

- a. comparer plusieurs algorithmes méta-heuristiques afin de comprendre leurs comportements et de déterminer les meilleurs choix selon le problème proposé. Les algorithmes de recuit simulé, génétique et fourmi, ainsi que les méthodes d'améliorations 2, 3,4-opt seront utilisées;
- b. proposer et vérifier un algorithme capable de déterminer la température initiale du recuit simulé.

## **1.5 Plan de travail**

Outre le présent chapitre, le corps de ce mémoire se compose de cinq (5) chapitres. Dans le Chapitre 1, le projet de recherche a été positionné dans son contexte. Le Chapitre 2, présente une revue très large de la littérature portant sur le problème du voyageur de commerce et aussi sur d'autres problèmes dérivés. Au Chapitre 3, nous présentons les algorithmes étudiés en décrivant leurs caractéristiques et leurs limites. Le Chapitre 4 décrit les modèles expérimentaux utilisés pour valider les objectifs de recherche. Dans le Chapitre 5, nous rapportons l'ensemble des résultats obtenus lors d'expérimentations numériques avec les différents algorithmes présentés au Chapitre 3. Les conclusions du Chapitre 6 permettront de mettre en valeur les contributions de ce mémoire aux connaissances actuelles et de proposer quelques perspectives de recherche. La Figure 1.1 présente le plan de lecture du présent mémoire.

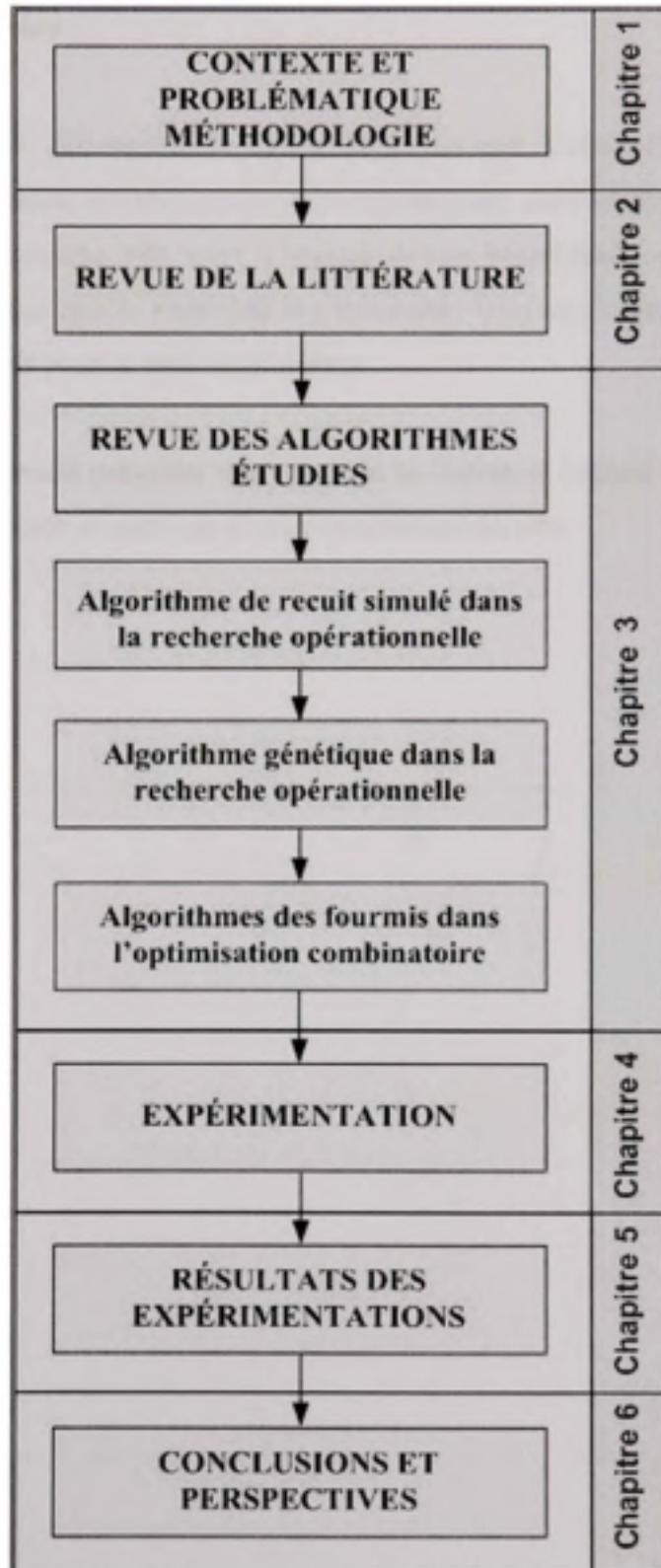


Figure 1.1 Plan de lecture du mémoire.

## **1.6 Conclusion**

Malgré l'importance des travaux de recherche concernant la résolution du problème du voyageur de commerce, en utilisant des méta-heuristiques, celle-ci reste à l'heure actuelle un domaine de recherche très actif. L'impact de ces algorithmes sur la résolution de problèmes réels ainsi que la multitude des difficultés liées aux réglages des paramètres vient élargir l'intérêt pour ce type de problème.

Nous allons maintenant présenter une revue de la littérature portant sur le problème du voyageur de commerce et aussi sur d'autres problèmes dérivés.

## CHAPITRE 2

### REVUE DE LA LITTÉRATURE

#### 2.1 Problème du voyageur de commerce

Le voyageur de commerce doit visiter  $n$  villes en passant par chaque ville une seule fois. On commence en choisissant n'importe quelle ville et on termine en retournant à la ville de départ (voir la figure 2.1). On connaît les distances entre les villes. On doit donc minimiser la distance parcourue. On peut remplacer la notion de distance par le temps ou par le coût.

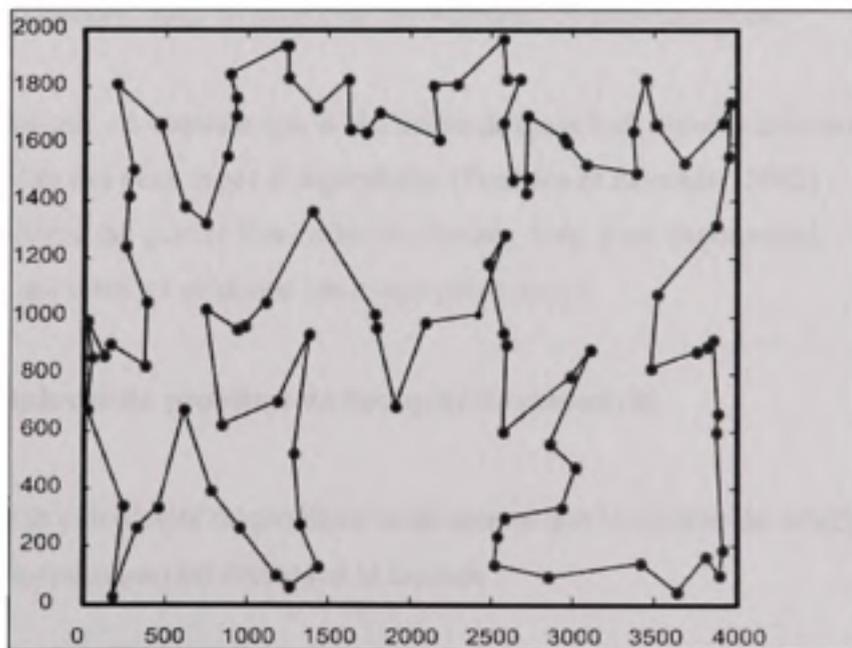


Figure 2.1 Réseau de 100 villes pour le voyageur de commerce (KROA100).

Le problème du voyageur de commerce est un problème NP-Complet (Non déterministe polynomial complet). Ce qui signifie que le temps pour trouver une solution optimale augmente de façon exponentielle en fonction de la grandeur du problème. Un algorithme de complexité polynomiale n'existe pas pour ce type de problème.

D'après Kernighan (1973) , la définition du problème du voyageur de commerce s'énonce comme suit :

Soit un graphe  $G = (N, A)$ , où  $N = \{v_0, v_1, \dots, v_n\}$  représente l'ensemble de nœuds (villes). Et soit  $A = \{(v_i, v_j) / v_i, v_j \in N, i < j\}$  l'ensemble des arrêtes qui relient les nœuds si les distances sont symétriques et  $A = \{(v_i, v_j) / v_i, v_j \in N, i \neq j\}$  représente les arcs si les distances sont asymétriques. En d'autres termes, le problème est symétrique si la distance pour faire le trajet de la ville A à la ville B est la même que celle du retour de la ville B à la ville A. Si le problème est symétrique, le graphe est non orienté et on aura des cycles. Dans le cas contraire, pour un problème asymétrique, on aura un circuit.

Dans la littérature, on constate que la recherche de cycle hamiltonien dans un graphe peut se faire par l'un des deux types d'algorithmes (Perdalos et Resende , 2002) :

- a. un algorithme qui génère l'ensemble des cycles, donc il est exponentiel;
- b. un autre qui teste les solutions (en temps polynomial).

### 2.1.1 Complexité du problème du voyageur de commerce

Un calcul de la complexité du problème nous montre que le nombre de solutions faisables (dans le cas symétrique) est donné par la formule :

$$(n-1)! / 2 \tag{2.1}$$

Où  $n$  est le nombre de villes.

Si on considère le temps de calcul pour un trajet égal à  $1\mu s$ , on constate que le temps de calcul est assez grand tel que le montre le tableau suivant :

Tableau 2.1

Temps de calcul et nombre de possibilités (Tirée de Laporte ,1992)

Nombre des villes	Nombre de possibilités	Temps de calcul
5	12	12 $\mu s$
10	181440	0.18ms
15	43 milliards	12 heures
20	60E+15	1928 ans
25	310E+21	9.8 milliards d'années

### 2.1.2 Formulation mathématique du problème du voyageur de commerce

On détermine les variables nécessaires pour faire l'énonciation mathématique du problème du voyageur de commerce :

$d_{\varepsilon\sigma}$  = distance entre la ville  $\varepsilon$  et la ville  $\sigma$  ;

$n$  = nombre de villes;

$x_{\varepsilon\sigma}$  = variable binaire qui prend la valeur 1 si la ville  $\varepsilon$  est visité immédiatement avant la ville  $\sigma$ . Sinon, cette variable prend la valeur 0.

On doit minimiser la longueur du cycle hamiltonien, donc la fonction-objectif sera :

$$Z = \sum_{\varepsilon=1}^n \sum_{\sigma=1}^n d_{\varepsilon\sigma} x_{\varepsilon\sigma} \quad (2.2)$$

Les contraintes sont les suivantes :

$$\sum_{\varepsilon=1}^n x_{\varepsilon\sigma} = 1 \text{ pour tout nœud } \sigma \quad (2.3)$$

$$\sum_{\sigma=1}^n x_{\varepsilon\sigma} = 1 \text{ pour tout nœud } \varepsilon \quad (2.4)$$

$$\sum_{\varepsilon \in E} \sum_{\sigma \in E} x_{\varepsilon\sigma} \leq |E| - 1, \text{ pour tout } E \subset N \text{ avec } 2 \leq |E| \leq n-1 \quad (2.5)$$

La contrainte (2.3) s'assure qu'on ne sort qu'une seule fois de chacun des points et la contrainte (2.4) vérifie qu'on entre qu'une seule fois à chaque point. Donc, on s'assure que chaque point est visité une seule fois. Ensuite on ajoute la contrainte (2.5) pour que des sous tours ne se forment pas. Dans la contrainte (2.5),  $|E|$  représente la cardinalité de l'ensemble  $E$ .

## 2.2 Problème du voyageur de commerce symétrique et asymétrique

Le voyageur de commerce présente une particularité au niveau des caractéristiques du graphe. Cette particularité est liée aux arcs du graphe, car il peut y avoir une masse inégale d'après le sens dans lequel on le parcourt. La matrice de ce problème est une matrice asymétrique. Si les distances sont égales la matrice du problème sera symétrique. Dans le cas symétrique le problème doit respecter la condition suivante :

$$d_{\varepsilon\sigma} = d_{\sigma\varepsilon} \quad \forall \varepsilon, \sigma \quad (2.6)$$

De plus, les distances peuvent respecter l'inégalité du triangle, donc on applique la condition :

$$d_{\varepsilon k} \leq d_{\varepsilon\sigma} + d_{\sigma k} \quad \forall \varepsilon, \sigma, k \quad (2.7)$$

### 2.2.1 Problème du voyageur de commerce symétrique

Shutler (2001) présente une approche de résolution pour le problème du voyageur de commerce symétrique en améliorant un algorithme de séparation et d'évaluation progressive pour un problème de l'ordre de cent à cinq cents nœuds. Jusqu'à ce moment, la meilleure heuristique est considérée celle de Helsgaun (2000) et l'algorithme de Lin-Kernigan (Glover, Gutin *et al.*, 2001).

### 2.2.2 Problème du voyageur de commerce asymétrique

Glover, Gutin *et al.* (2001) développent trois (3) heuristiques de construction pour un graphe orienté, où chacun des arcs présente des valeurs différentes. Ces heuristiques représentent de nouvelles approches pour la construction du problème de voyageur du commerce asymétrique. L'heuristique basée sur la méthode *Karp-Steele patching* (la première développée par Gutin et Glover) donne des solutions qui sont proches de l'optimum (à 3.36 % de l'optimum).

## 2.3 Extensions au problème du voyageur de commerce

Une autre particularité du problème du voyageur de commerce est donnée par le type d'opérations effectuées. C'est-à-dire qu'on peut avoir différents types de problèmes du voyageur de commerce en modifiant le type d'opérations.

### 2.3.1 Problème du voyageur de commerce avec retour à charge

Gendreau, Hertz *et al.* (1996) proposent une heuristique en respectant l'inégalité du triangle et des coûts symétriques. En appliquant cette heuristique, ils ont obtenu un résultat situé au 3/2 de l'optimum. Les tests ont été effectués sur trente problèmes de 100, 200 et 300 nœuds. Le type d'heuristique utilisé était l'algorithme GENIUS développé par Gendreau, Hertz et Laporte. Cet algorithme est constitué de deux (2) phases : la phase

d'insertion (*GENERALIZED Insertion*) et la phase de post-optimisation (*Unstringing* et *Stringing*). Pour la phase d'insertion (*GENERALIZED Insertion*) ils ont choisi trois (3) nœuds et ils ont ensuite inséré d'autres sommets pour former un cycle hamiltonien.

Pour la phase de post-optimisation (*Unstringing* et *Stringing*), ils ont considéré le retrait de chaque nœud et après la réinsertion de ces nœuds dans le tour pour tenter de réduire le coût de la tournée.

Mladenovic et Hansen (1997) proposent une variante de GENIUS, avec une structure de recherche, appelée GENIUS-VNS. En utilisant cette méthode, ils ont obtenu une croissance de 0,4 % par rapport à la méthode proposée par Gendreau, Hertz et Laporte. Par contre, le temps de calcul avait augmenté de 30 % par rapport à la méthode GENIUS.

Ghaziri et Osman (2003) présentent une nouvelle heuristique basée sur les réseaux de neurones, nommée SOFM. Ils utilisent quatre (4) types d'interactions : la première interfère avec les clients ayant une grande distance alors que la deuxième interfère avec les clients qui nécessitent une cueillette. Les deux (2) autres interactions traitent la modalité d'interagir entre l'ensemble (les deux types des clients) et les dépôts et la modalité d'interaction entre les chaînes. Les tests ont été effectués sur des problèmes ayant entre 100 et 1000 nœuds. Le tableau 2.2 nous montre une comparaison des heuristiques pour le voyageur du commerce avec retour à charge faite par (Ghaziri et Osman, 2003).

Tableau 2.2

Comparaison des heuristiques pour le voyageur du commerce avec retour à charge  
(Extrait de Ghaziri et Osman ,2003)

Nombre de points	SOFM	GENIUS	GENIUS-VNS	SOFM	GENIUS	GENIUS-VNS
100	1072,05	1070,05	1065,62	23,98	4,76	5,54
200	1493,65	1498,56	1491,86	66,86	34,24	36,02
300	1815,18	1818,01	1812,12	308,38	85,02	100,66
500	2328,60	2328,78	2321,09	805,86	342,54	354,66
1000	3269,46	3273,13	3258,44	1453,20	1197,72	1671,82
Tous	1951,87	1953,73	1946,08	844,03	303,03	391,05

### 2.3.2 Problème du voyageur de commerce avec cueillette et livraison

La problématique est de trouver le cycle hamiltonien le plus court permettant de desservir tous les clients d'une façon à ce que chaque cueillette soit effectuée avant sa livraison.

Renaud (2000) présente un algorithme composite qui a été testé sur un ensemble de 108 problèmes dérivés des 36 de Groetschel, Juenger *et al.* (1984) ayant jusqu'à 441 nœuds. Les résultats obtenus sont donnés dans le tableau 2.3.

Tableau 2.3

Les résultats obtenus en utilisant les algorithmes 3-opt et 4-opt

Algorithme	Écart de l'optimum	Temps calcul (en seconde)
4-opt (RBO)	4,48 %	73,1
3-opt (Reinelt)	3,87 %	1314,2

On constate qu'en utilisant l'algorithme 3-opt, le meilleur résultat était situé à 3,87 % au dessus de la meilleure solution connue, tandis que l'algorithme 4-opt avait donné comme meilleur résultat une valeur de 4,48 % situé au dessus de la meilleure solution connue.

Le temps de calcul était beaucoup moins élevé pour l'algorithme 4-opt, mais on ne connaît pas le type d'ordinateur sur lequel les calculs ont été fait en utilisant l'algorithme 3-opt.

Renaud (2000) a proposé sept (7) algorithmes de perturbation en essayant de trouver de meilleures solutions pour les modèles proposés. Il a abordé trois (3) types de perturbations : perturbations du problème, perturbations de l'algorithme et perturbation de la solution. Pour un ensemble de 36 problèmes de Reinelt ayant jusqu'à 441 nœuds (entre 51 et 441), il a obtenu un résultat qui avait amélioré le résultat de Renaud, Boctor et Ouenniche de 4 %. Pour un autre ensemble de 20 problèmes de 101 nœuds à 201 nœuds, il a obtenu un optimum pour 12 problèmes et pour les autres il a obtenu des valeurs très proches de l'optimum (entre 0,98 % et 1,26 %).

### **2.3.3 Problème du voyageur de commerce avec la localisation de la demande sur les arcs**

En général, la demande se situe sur les nœuds, mais il existe de cas quand la demande se trouve sur les arcs. On donne comme exemple le problème du postier chinois (Hierarchical Chinese postman problem-HCPP) pour lequel la demande est située sur les arcs.

Gianpaolo et Gutin (2005) ont adapté le problème du postier chinois pour le problème du voyageur de commerce rural (rural postman problem-RPP). Ils ont séparé les arcs en deux (2) ensembles : les arcs qui nécessitaient un service et les arcs qui ne nécessitaient pas un service. Par contre, tous les arcs du HCPP nécessitaient un service, donc ils devaient être visités au minimum un fois. Le RPP a permis d'appliquer un algorithme exact de génération de coupes sur le problème transformé de HCPP. Ils ont utilisé un graphe proposé par Dorigo (2003) qui contenait 113 nœuds et 171 arcs. L'écart maximum obtenu était de 0,8 % par rapport à la solution trouvée par séparation et évaluation progressive.

#### 2.3.4 Autres types de problèmes du voyageur de commerce

Dans la littérature, plusieurs types de problèmes du voyageur du commerce ont été étudiés. Parmi les plus récents, on mentionne la généralisation du voyageur de commerce avec des sommets noirs et blancs (*Black and white traveling salesman problem*), le problème du voyageur de commerce avec contraintes de présence (TSPPC), le problème du voyageur de commerce dynamique (avec une matrice de distances dynamique), le voyageur de commerce avec contraintes de cardinalité (*Cardinality constrained traveling salesman problem - CCC-TSP*).

Bourgeois, Laporte *et al.* (2003) ont proposé le problème de *Black and white traveling salesman problem*. Le problème est défini par un graphe  $G$  dont le sommet est divisé dans des sommets noirs et blancs. Le but est de concevoir un cycle hamiltonien le plus court sur  $G$  en respectant deux (2) contraintes. La première contrainte restreint le nombre de sommets blancs entre deux (2) sommets noirs, donc il s'agit d'une contrainte de cardinalité. La deuxième contrainte restreint la distance entre deux (2) sommets noirs consécutifs. Pour un ensemble de cinquante sommets, ils ont obtenu des solutions réalisables situées entre 12 % et 13 % de l'optimum. Par contre, pour un ensemble de 200 sommets, les solutions réalisables étaient situées entre 16 % et 23 % de l'optimum. Si les nombres des sommets noirs deviennent plus élevés, il est plus difficile d'identifier une solution réalisable. Ils ont proposé ce problème pour résoudre l'ordonnancement de lignes aériennes, car les séquences de vols (sommets blancs) sont séparées par des périodes de maintenances (sommets noirs).

Paletta (2002) propose une heuristique pour résoudre un problème du voyageur de commerce périodique. Le problème était de trouver le cycle hamiltonien minimal pour un voyageur de commerce qui retourne à chaque jour à la maison en visitant chaque ville un nombre prédéterminé de fois durant une période de  $k$  jours. L'algorithme a été implémenté en Fortran et les résultats obtenus ont été comparés avec les résultats de Chao, Golden *et al.* (1995) et Laporte (1992). Les tests ont été effectués sur un ensemble

de 33 problèmes de 48 à 288 nœuds. L'heuristique proposée par Paletta a permis de trouver une solution pour 25 problèmes, situé à 0,745 % de l'optimum. La pire solution obtenue était situé à 1,035 % de l'optimum. Le Tableau 2.4 nous montre les résultats obtenus par Paletta (2002).

Tableau 2.4

Comparaison avec les meilleures solutions connues (Extrait de Paletta ,2002)

Problème	Sommets	Chao (1995)	Laporte (1997)	Paletta (2002)	BPS	%
1	50	442,1	439,02	436,50	436,50	0,00
2	50	1106,7	1111,93	1122,44	1122,44	1,42
3	50	474,0	469,69	469,16	469,64	0,10
4	75	554,2	556,21	559,68	559,49	0,95
5	75	1394,0	1389,54	1387,90	1384,75	-0,23
6	75	657,3	651,28	643,59	655,06	1,78
7	100	662,4	660,41	-	646,65	-2,08
7 bis	100	-	-	651,05	643,86	-1,10
8	100	1635,2	1634,68	-	1633,92	-0,05
8 bis	100	-	-	1606,34	1594,15	-0,76
9	100	735,3	734,16	-	733,13	-0,14
9 bis	100	-	-	718,05	717,30	-0,14
10	100	1248,8	1240,01	-	1249,14	0,74
10 bis	100	-	-	1231,40	1228,83	-0,21
11	65	491,0	490,97	490,97	490,97	0,00
12	87	664,1	664,10	664,10	664,10	0,00
13	109	830,8	830,80	830,80	830,80	0,00
14	131	994,6	994,60	994,60	994,60	0,00
15	153	1157,1	1157,07	1157,07	1157,07	0,00
16	48	726,8	660,12	660,12	660,12	0,00
17	66	776,5	776,43	776,43	776,43	0,00
18	84	873,7	873,73	876,44	876,44	0,31
19	102	974,6	958,88	958,51	958,51	0,00
20	120	1053,6	1034,51	1033,58	1033,58	0,00
21	77	1379,1	1375,08	-	1375,07	0,00
21 bis	78	--	-	1375,26	1375,07	-0,01
22	154	4323,6	4319,72	-	4323,49	0,09
22 bis	156	-	-	4396,95	4330,75	-1,51
23	231	8753,3	8553,10	8390,53	8498,00	1,28

Problème	Sommets	Chao (1995)	Laporte (1997)	Paletta (2002)	BPS	%
24	48	-	2068,46	2064,84	2064,84	0,00
25	96	-	3293,50	3232,72	3231,50	-0,04
26	144	-	4106,72	4084,75	4118,63	0,83
27	192	-	4661,97	4636,76	4621,36	-0,33
28	240	-	4698,83	4757,90	4682,54	-0,35
29	288	-	5699,96	5688,42	5595,45	-1,63
30	72	-	4453,15	4479,65	4474,17	0,47
31	144	-	5405,40	-	5475,70	1,30
31 bis	144	-	-	5581,02	5565,82	-0,27
32	216	-	7469,73	7405,52	7346,32	-0,80
33	288	-	8493,74	8394,52	8415,31	0,25

Le problème du voyageur de commerce avec contraintes de présence (TSPPC) a été proposé pour résoudre les problèmes d'acheminement des marchandises et les problèmes d'ordonnancement. Dans ce cas, il y a deux (2) types de contraintes :

- a. contraintes sur l'ordre prédéfini entre certains nœuds du graphe;
- b. contraintes sur la visite des nœuds (certains nœuds doivent être visités avant d'autres).

Chung, Jongsoo *et al.* (2002) ont proposé de résoudre ce problème en appliquant un algorithme génétique. Les résultats obtenus montrent qu'une solution optimale a été trouvée pour les problèmes de petite et moyenne tailles tandis que pour les problèmes de grande taille, la solution était positionnée à 3,2 % de plus que l'optimum.

Patterson et Emerson (2003) ont proposé un nouveau problème pour le voyageur de commerce avec contraintes de cardinalité nommé *Cardinality constrained traveling salesman problem (CCC-TSP)*.

Ce problème consiste à diviser un graphe en plusieurs sous-graphes hamiltoniens avec des exigences de tailles minimale et maximale. Les auteurs ont testé leur heuristique sur 30 problèmes tirés de TSPLIB de 56 à 226 nœuds. En effet, pour les problèmes ayant une densité de 15 %, la solution était positionnée à 0,053 % de l'optimum tandis que

pour les problèmes d'une densité de 100 % la solution était positionnée à 0,74 % de l'optimum. Les résultats ont été trouvés dans un intervalle de temps situé entre 6,7 et 132 818 secondes.

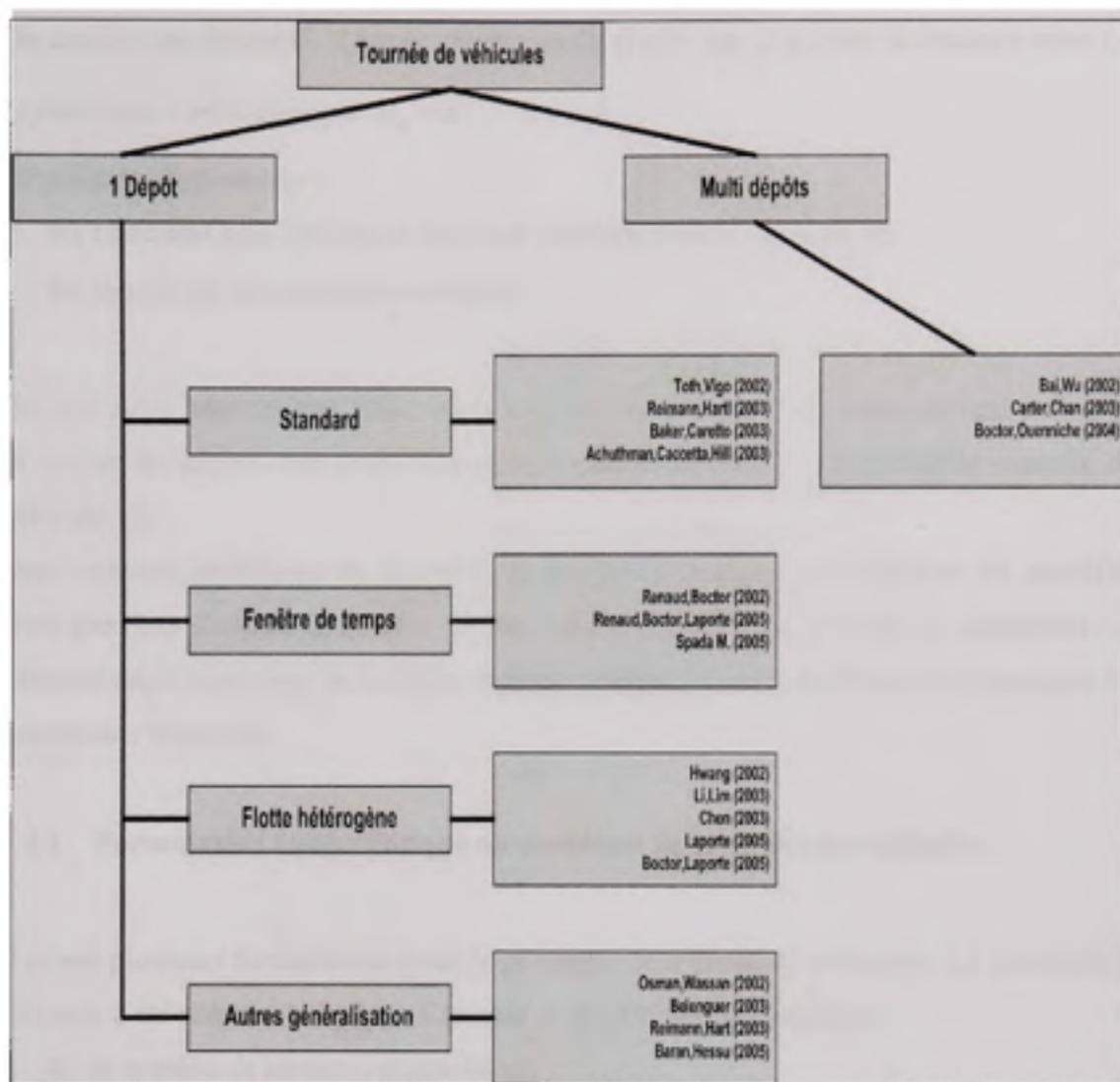
## **2.4 Problème de tournées de véhicules**

Danzig et Ramser (1959) proposent une adaptation du problème du voyageur de commerce nommé problème de tournées de véhicules (*Vehicle routing problem - VRP*).

Ils modélisent le problème de la livraison de produits auprès d'un ensemble de clients répartis géographiquement tout en minimisant la distance totale parcourue.

Ce problème a attiré beaucoup de chercheurs, car le problème est très intéressant au niveau théorique et économique. La littérature propose plusieurs approches pour résoudre ce problème de tournées de véhicules.

La figure 2.2 nous montre une classification des problèmes des tournées des véhicules (Extrait de Laporte, 2005).



**Figure 2.2 Classification des problèmes des tournées des véhicules.**  
(Extrait de Laporte, 2005)

Mathématiquement, on définit le problème comme un graphe simple orienté  $G = (V, A)$ , où  $V = \{v_0, v_1, \dots, v_n\}$  représente l'ensemble des sommets, c'est-à-dire les clients qu'on doit visiter et  $A = \{(v_i, v_j) : v_i, v_j \in V, i \neq j\}$  représente l'ensemble des arcs possibles. Le sommet  $v_0$  représente le dépôt qui est le point de départ et d'arrivée pour tous les arcs (routes).

On associe une distance  $d_{ij}$  pour chaque arc  $(i, j) \in A$ . On considère la distance entre  $i$  et  $j$  symétrique c'est-à-dire que  $d_{ij} = d_{ji}, \forall i, j \in A$ .

On pose des hypothèses :

- les véhicules sont identiques avec une contrainte sur la capacité  $Q$ ;
- les clients ont une demande connue  $q_i$ .

On peut aussi imposer une limite sur la longueur de la route. Les routes doivent permettre de visiter les clients une seule fois et la demande ne peut pas dépasser la capacité du véhicule,  $Q$ .

Dans certains problèmes de tournées, le nombre maximum de véhicules est prédéfini alors que dans d'autres, ce nombre est une variable de décision. En effet, la répartition des véhicules doit minimiser la fonction-objectif (distance, temps, coût) tout en respectant les contraintes imposées.

#### 2.4.1 Formulation mathématique du problème de tournées de véhicules

Il existe plusieurs formulations pour le problème de tournée de véhicules. La formulation suivante a été tirée de Achuthan, Caccetta *et al.* (1998). On considère :

- $K$  : le nombre de camions disponibles;
- $Q_k$  : la capacité du camion  $k$ ;
- $N$  : le nombre de clients à visiter (on donne des numéros pour chaque client entre 1 et  $n$  et pour le dépôt on donne le numéro 0);
- $d_{ij}$  : la distance entre les villes  $i$  et  $j$ ;
- $q_i$  : la demande du client  $i$ .

On considère aussi les variables de décision suivantes :

- $y_{ik}$  : variable de décision binaire égale à 1 si la commande de client  $i$  est livrée par le camion  $k$  et à 0 autrement;

- b.  $x_{ijk}$  : variable de décision binaire égale à 1 si le camion  $k$  parcourt la distance de la ville  $i$  vers la ville  $j$  et à 0 autrement.

On doit minimiser la fonction-objectif :

$$\sum_{i=0}^n \sum_{j=0}^n \sum_{k=1}^K d_{ij} x_{ijk} \quad (2.8)$$

Il faut considérer les contraintes de problèmes d'affectation généralisée :

$$\sum_{i=1}^n q_i y_{ik} \leq Q_k \quad k=1, \dots, K; \quad (2.9)$$

$$\sum_{k=1}^K y_{ik} = \begin{cases} k & i=0 \\ 1 & i=1, \dots, n \end{cases} \quad (2.10)$$

$$y_{ik} = 0 \text{ ou } 1 \quad i=0, \dots, n; k=1, \dots, K; \quad (2.11)$$

Il faut également respecter les contraintes du voyageur de commerce :

$$\sum_{i=0}^n x_{ijk} = y_{jk} \quad j=0, \dots, n; k=1, \dots, K; \quad (2.12)$$

$$\sum_{j=0}^n x_{ijk} = y_{ik} \quad i=0, \dots, n; k=1, \dots, K; \quad (2.13)$$

$$\sum_{i \in S} \sum_{j \in S} x_{ijk} \leq |S| - 1, \quad \text{pour tout } S \subset N \text{ avec } 2 \leq |S| \leq n-1$$

$$2 \leq |S| \leq n-1 \quad k=1, \dots, K; \quad (2.14)$$

$$x_{ijk} = 0 \text{ ou } 1 \quad i=0, \dots, n; j=0, \dots, n; k=1, \dots, K; \quad (2.15)$$

La formule (2.8) permet de minimiser la distance parcourue par l'ensemble des véhicules. La contrainte (2.9) permet d'assurer que le chargement de véhicules respecte leur capacité. La contrainte (2.10) permet de s'assurer que chaque client est desservi par un seul véhicule et que chacune des routes débute et se termine au dépôt. Les contraintes (2.11), (2.12), (2.13), (2.14), (2.15) permettent d'éviter les sous tours, car chaque client doit être visité une seule fois. La figure 2.3 nous montre une tournée pour trois véhicules et un dépôt.

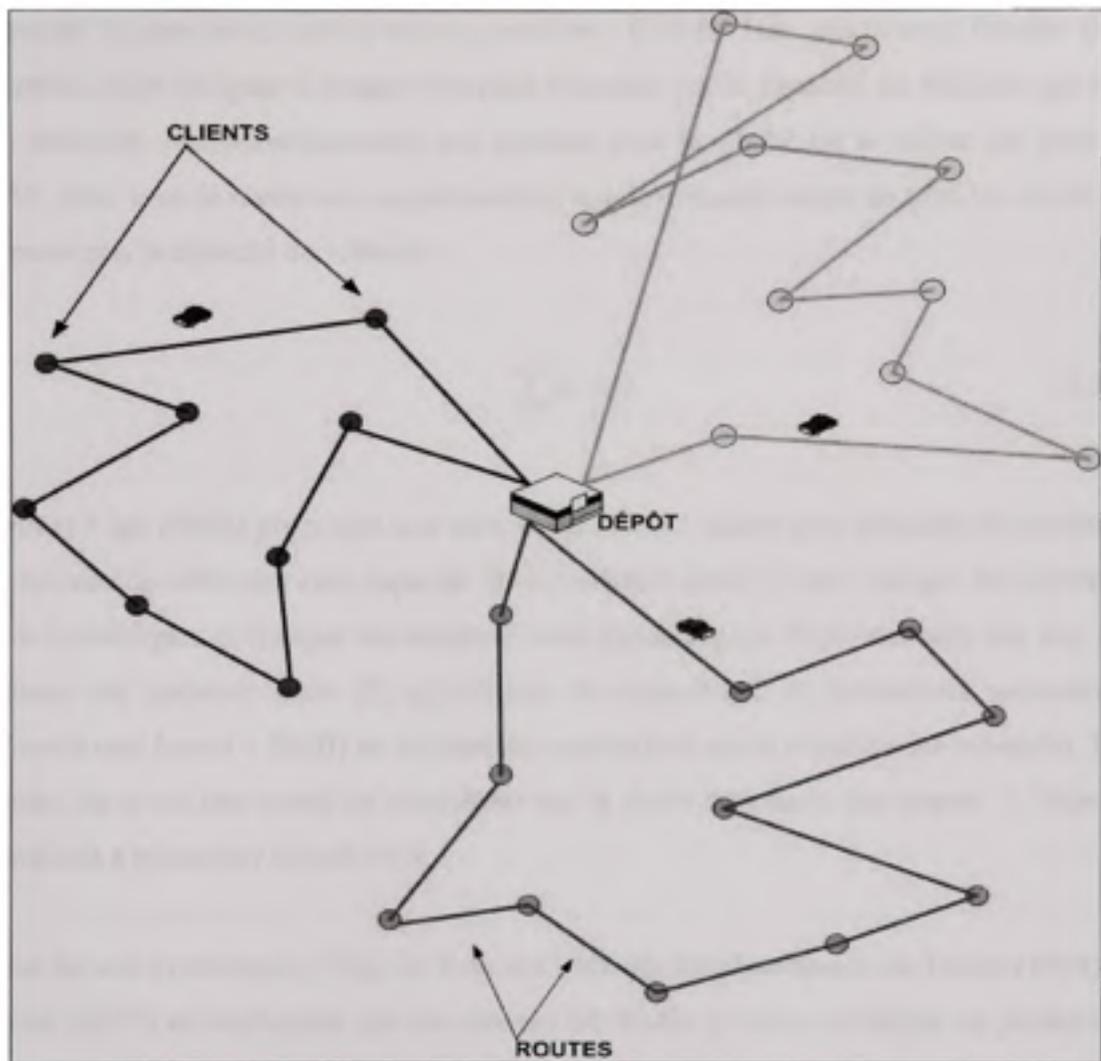


Figure 2.3 Tournée des véhicules (1 dépôt, 3 véhicules).

#### 2.4.2 Problème de tournées de véhicules standard avec capacité

L'objectif consiste à minimiser la distance parcourue d'une flotte de véhicules et la somme du temps de voyage, lorsque l'on connaît les clients, la demande et le type de véhicule. On suppose que les véhicules sont du même type et possède une capacité identique. La demande cumulée des clients ne peut dépasser la capacité totale des véhicules utilisés. Ce problème ne tient pas compte de la limite sur la durée des tournées.

En effet, le problème présenté est nommé problème de tournées de véhicules avec capacité (*Capacitated vehicle routing problem - CVRP*). Une solution est faisable si la quantité totale assignée à chaque itinéraire n'excède pas la capacité du véhicule qui sert cet itinéraire. Mathématiquement, une solution pour le CVRP est la même que pour le VRP, mais avec la restriction supplémentaire que la demande totale de tous les clients ne dépasse pas la capacité du véhicule :

$$\sum_{i=1}^k d_i \leq Q. \quad (2.16)$$

Toth et Vigo (2002) proposent une série d'algorithmes exacts pour résoudre le problème de tournée de véhicules avec capacité. Ils considèrent deux (2) cas : lorsque les distances sont symétriques et lorsque les distances sont asymétriques. Pour résoudre ces cas, les auteurs ont présenté deux (2) algorithmes de séparations et évaluations successives (*Branch and bound – B&B*) en mettant des restrictions sur la capacité des véhicules. Par contre, ils n'ont pas ajouté de contraintes sur la durée maximale des routes. L'objectif consistait à minimiser le coût total.

Pour les cas symétriques, Vigo et Toth ont présenté les algorithmes de Fisher (1994) et Miller (1995) en expliquant que ces auteurs ont résolu le même problème en posant des hypothèses différentes sur l'interdiction où l'acceptation de voyager sur une route qui desservait un seul client.

Le cas asymétrique qu'ils ont présenté était donné pour un ensemble allant jusqu'à 300 points et 4 véhicules. La résolution du cas symétrique en utilisant les algorithmes de B&B a donné une solution (exacte) dans un temps acceptable (Toth and Vigo 2002).

Achuthan, Caccetta *et al.* (1998) ont traité le problème du CVRP en utilisant la méthode de génération de coupes. En effet, ils ont développé un nouveau système de génération de coupes et une procédure de recherche permettant d'identifier la création de sous-tours.

Les auteurs ont expérimenté leur méthode sur 1650 problèmes générés aléatoirement. Ils ont par la suite testé la méthode sur 24 problèmes standards répertoriés dans la littérature. Le temps maximum de résolution a été fixé à 1 heure. C'est pour cette raison que certaines bornes supérieures ne sont pas définies au tableau 2.5.

Pour les problèmes de grande taille, les algorithmes exacts ne donnent pas toujours de solution. En raison de cet inconvénient, les chercheurs ont essayé de développer des heuristiques et méta-heuristiques capables de donner des solutions optimales. Rochat et Taillard (1995) sont parmi les pionniers du développement des heuristiques pour le problème de tournée de véhicules de grande taille. Ils ont proposé de décomposer le problème en sous-problèmes en utilisant deux (2) méthodes distinctes de partitionnement. Ensuite, les sous-problèmes ont été traités avec une approche avec liste de tabous. Finalement, après un certain nombre d'itérations, les sous-problèmes ont été regroupés pour former un seul problème.

Dans le tableau 2.6, on présente une classification selon les meilleures solutions connues. Le tableau 2.7 relève les résultats sur les problèmes de Christofides et Eilon (1989). À partir de ces résultats (les tableaux 2.6 et 2.7) on constate que Taillard (1993) a obtenu les meilleurs résultats pour les problèmes de petites tailles (problèmes tirés de Christofides et Eilon (1989)). En effet, Taillard (1993) a trouvé la meilleure solution pour 12 des 14 problèmes de Christofides et Eilon située à 0.05% de l'optimum.

Tableau 2.5

## Résultats du CVRP

(Source : Achuthan, Caccetta *et al.* 1998)

No du problème	Nombre de clients	Routes à un seul client : permis		Route à un seul client : interdit		Meilleure solution connue	Fisher (1994)		Publication
		Borne inférieure	Borne supérieure	Borne inférieure	Borne supérieure		Borne inférieure	Borne supérieure	
11	50	514.58	524.61	514.58	524.61	525.61	507.09	N/A	Cristofides (1969)
12	75	786.29	N/A	786.29	N/A	835.26	755.05	N/A	Cristofides (1969)
13	100	799.07	N/A	799.07	N/A	826.14	785.86	N/A	Cristofides (1969)
14	150	950.65	N/A	950.57	N/A	1028.14	932.68	N/A	Cristofides (1969)
15	199	1114.17	N/A	1153.99	N/A	1298.79	1096.72	N/A	Cristofides (1979)
16	120	862.98	N/A	862.98	N/A	1042.11	N/A	N/A	Cristofides (1979)
17	100	785.29	819.56	813.73	819.56	819.56	817.77	819.56	Cristofides (1979)
18	44	721.88	723.54	723.54	723.54	723.54	720.76	723.54	Fisher (1994)
19	71	238.51	241.97	241.97	238.51	241.97	237.76	241.97	Fisher (1994)
20	134	1120.16	N/A	1120.16	N/A	1163.60	1133.73	1163.60	Fisher (1994)
21	75	659.17	752.25	752.25	659.17	N/A	N/A	N/A	Reinelt (1981)
22	75	705.25	949.26	949.26	705.25	N/A	N/A	N/A	Reinelt (1981)
23	75	932.41	N/A	N/A	932.41	N/A	N/A	N/A	Reinelt (1981)
24	100	1000.4	N/A	N/A	1003.17	N/A	N/A	N/A	Reinelt (1981)

Tableau 2.6

Comparaison de méta-heuristiques utilisée (Source : Prins, 2003)

Algorithme utilisé	% écart par rapport à la meilleure solution	Nb. de meilleures solutions trouvées	Temps moyen	Publication
Tabou	0.05	12	-	Taillard (1993)
Algorithme génétique	0.08	10	5.2	Prins (2001)
Tabou	0.1	5	103	Xu-Kelly (1996)
Tabou	0.2	9	-	Gendreau (1994)
Tabou	0.39	6	-	Taillard (1993)
Tabou	0.55	6	-	Rego et Rouicard (1996)
Tabou granulaire	0.55	4	3.5	Toth et Vigo (2002)
Tabou	0.68	5	-	Gendreau <i>et al.</i> (1991)
Tabou	0.77	4	-	Rego et Rouicard (1996)
Tabou	0.86	5	46.8	Gendreau <i>et al.</i> (1994)
Algorithme génétique	0.9	5	0.7	Prins (2001)
Tabou	1.01	4	26.1	Osman (1993)
Tabou	1.03	3	34	Osman (1993)
Recuit simulé	2.09	2	-	Osman (1993)

Tableau 2.7

Résultats sur les problèmes de Christofides et Eilon (1989)  
(Doerner, Hartl *et al.*, 2004)

Nom	N. clients	Capacité véhicules	Longueur max route	Temps de service	Meilleure solution	Publication
C1	50	160	$\infty$	0	524.61	Taillard (1993)
C2	75	140	$\infty$	0	835.26	Taillard (1993)
C3	100	200	$\infty$	0	836.14	Taillard (1993)
C4	150	200	$\infty$	0	1028.42	Taillard (1993)
C5	199	200	$\infty$	0	1291.45	Rochat et Taillard (1995)
C6	50	160	200	10	555.43	Taillard (1993)
C7	75	140	160	10	909.68	Taillard (1993)
C8	100	200	230	10	865.94	Taillard (1993)
C9	150	200	200	10	1162.55	Taillard (1993)
C10	199	200	2000	10	1395.85	Rochat et Taillard (1995)

### 2.4.3 Problème de tournées de véhicules standard avec flotte hétérogène

La plupart des articles recensés dans la littérature traite le cas d'une flotte homogène pour les problèmes de tournées de véhicules. En réalité, ce type de problème est rare et il est nécessaire d'étudier ce type de problème quand la flotte est hétérogène.

Renaud (2000) présente le problème de véhicules avec flotte hétérogène. Cette approche a été conçue pour résoudre le problème de flotte composé de véhicules avec des capacités différentes. Pour une flotte de véhicules hétérogène, les coûts fixes et variables peuvent varier d'un véhicule à l'autre. L'avantage d'une flotte hétérogène est donné par sa flexibilité par rapport à la composition de la flotte. L'objectif consiste à minimiser le coût total composé des coûts fixes et des coûts variables. L'auteur a utilisé une heuristique de balayage pour générer différentes routes. Les résultats nous montrent que pour des tests effectués sur 20 problèmes de 50 à 100 points, cette heuristique donne de bons résultats.

L'écart par rapport à la meilleure solution obtenue par recherche avec liste de tabous était en moyenne de 0,49 % et le temps de calcul moyen était de 179 secondes. Les auteurs ont comparé leurs résultats à ceux obtenus par Osman (1993 ) et Gendreau (1999).

Tarantilis, Ioannou *et al.* (2004) ont proposé une méthode basée sur un seuil d'acceptation appelé méthode adaptive d'acceptation (*Backtracking adaptive threshold accepting - BATA*). Dans ce cas, les auteurs ont considéré que la flotte de véhicules est fixe, c'est-à-dire que la flotte comporte plusieurs types de véhicules, mais chaque type est limité. Les résultats, tirés de tests effectués sur des problèmes de 50 à 100 points démontrent que la méthode BATA donne de bonnes solutions. Les auteurs ont comparé leurs résultats à ceux de Taillard (1999) et ils ont obtenu des solutions en moyenne 0,31 % meilleures que Taillard (1999).

#### **2.4.4 Problème de tournées de véhicules avec fenêtres de temps**

Le problème de tournée de véhicules avec fenêtres de temps (*Vehicle routing problem with time windows - VRPTW*) a comme objectif de minimiser la flotte de véhicules et le temps total de voyage et d'attente.

Le problème de VRPTW est un problème VRP modifié, c'est-à-dire qu'on a des restrictions supplémentaires sur le problème de VRP. Les restrictions imposées sont :

- a. une solution est infaisable si un client est fourni après la borne supérieure de sa fenêtre de temps;
- b. chaque itinéraire doit commencer et finir dans la fenêtre de temps associée au dépôt;
- c. un véhicule qui arrive avant la limite inférieure de la fenêtre de temps cause un temps d'attente supplémentaire sur l'itinéraire;
- d. dans le cas de petites fenêtres de temps, un service en retard n'affecte pas la faisabilité de la solution mais, il est pénalisé en ajoutant une valeur à la fonction-objectif.

On considère une flotte de véhicules homogènes ( $V$ ) et un ensemble de clients ( $C$ ) et le graphe associé  $G = (V, C)$  ou  $V = \{v_0, \dots, v_n\}$  représente l'ensemble des points et  $C = \{(v_i, v_j) : v_i, v_j \in V, i \neq j\}$  représente l'ensemble des arcs possibles. Le point  $v_0$  représente le dépôt et correspond au point de départ et au point d'arrivée des routes. On considère que les véhicules sont identiques, les clients ont une demande connue  $q_i$  et il existe un temps de service  $\varepsilon_i$  pour chaque client. Le client  $v_i$  doit être visité dans la fenêtre du temps  $[e_i, f_i]$ . Les fenêtres des temps peuvent être modifiées en fonction des demandes de clients.

Le coût associé lors du VRPTW est composé de deux (2) composants : le coût pour parcourir la distance entre tous les clients et le coût d'attente chez un client jusqu'à ce qu'il soit prêt pour recevoir la livraison.

Les 56 problèmes de Solomon (1987) sont reconnus comme les problèmes tests pour le VRPTW. Les problèmes sont séparés en six (6) classes : C1, C2, R1, R2, RC1 et RC2. Chaque problème a 100 clients, un seul dépôt, des contraintes de capacités pour les véhicules et des fenêtres de temps pour les clients. Solomon a choisi de générer aléatoirement les coordonnées géographiques en utilisant une distribution uniforme pour les problèmes C1 et C2. Les problèmes R1 et R2 utilisent une distribution par graphe pour déterminer les coordonnées géographiques. Par contre, les problèmes RC1 et RC2 sont plus complexes et tiennent compte des restrictions supplémentaire sur la capacité de chargement des véhicules et sur la modalité de desservir les clients. Les tableaux 2.8, 2.9, 2.10, 2.11, 2.12, et 2.13 présentent les meilleurs résultats obtenus pour ces problèmes. Le tableau 2.14 présente une comparaison entre différentes publications pour tous les ensembles de problèmes.

Tableau 2.8

Meilleures solutions pour les problèmes C1  
(Source : Berger et Barkaoui, 2003)

Tests	No. véhicules	Distances totales	Références
C101	10	828,94	Rochat, Taillard (1995)
C102	10	828,94	Rochat, Taillard (1995)
C103	10	828,06	Rochat, Taillard (1995)
C104	10	824,78	Rochat, Taillard (1995)
C105	10	828,94	Rochat, Taillard (1995)
C106	10	828,94	Rochat, Taillard (1995)
C108	10	828,94	Rochat, Taillard (1995)
C109	10	828,94	Rochat, Taillard (1995)

Tableau 2.9

Meilleures solutions pour les problèmes C2  
(Source : Berger et Barkaoui, 2003)

Tests	No. véhicules	Distances totales	Références
C201	3	591,56	Rochat, Taillard (1995)
C202	3	591,56	Rochat, Taillard (1995)
C203	3	591,17	Rochat, Taillard (1995)
C204	3	590,60	Rochat, Taillard (1995)
C205	3	588,88	Rochat, Taillard (1995)
C206	3	588,49	Rochat, Taillard (1995)
C207	3	588,29	Rochat, Taillard (1995)
C208	3	588,32	Rochat, Taillard (1995)

Tableau 2.10

Meilleures solutions pour les problèmes R1  
(Source : Berger et Barkaoui, 2003)

Tests	No. véhicules	Distances totales	Références
R101	19	1650,80	Rochat, Taillard (1995)
R102	17	1486,12	Rochat, Taillard (1995)
R103	13	1292,85	Homberger, Gehring (1999)
R104	9	1013,32	Li, Lim (2003)
R105	14	1377,11	Rochat, Taillard (1995)
R106	12	1252,03	Rochat, Taillard (1995)
R107	10	113,69	Cordeau, Laporte, Mercier (2000)
R108	9	963,99	Berger, Barkaoui (2003)
R109	11	1194,73	Homberger, Gehring (1999)
R110	10	1124,40	Rousseau, Gendreau, Pesant (2002)
R111	10	1099,46	Homberger, Gehring (1999)
R112	9	1003,73	Homberger, Gehring (1999)

Tableau 2.11

Meilleures solutions pour les problèmes R2  
(Source : Berger et Barkaoui, 2003)

Tests	No. véhicules	Distances totales	Références
R201	4	1252,37	Homberger, Gehring (1999)
R202	3	1191,70	Rousseau, Gendreau, Pesant (2002)
R203	3	942,70	Homberger, Gehring (1999)
R204	2	849,62	Cordeau, Laporte, Mercier (2000)
R205	3	994,42	Li, Lim (2003)
R206	3	912,97	Rousseau, Gendreau, Pesant (2002)
R207	2	914,39	Rochat, Taillard (1995)
R208	2	730,71	Baker et al (2000)
R209	3	909,86	Rousseau, Gendreau, Pesant (2002)
R210	3	939,37	Baker et al (2000)
R211	2	910,09	Homberger, Gehring (1999)

Tableau 2.12

Meilleures solutions pour les problèmes RC1  
(Source : Berger et Barkaoui, 2003)

Tests	No. véhicules	Distances totales	Références
RC101	14	1696,94	Taillard et al (1997)
RC102	12	1554,75	Taillard et al (1997)
RC103	11	1262,02	Rochat, Taillard (1995)
RC104	10	1135,48	Cordeau, Laporte, Mercier (2000)
RC105	13	1633,72	Rousseau, Gendreau, Pesant (2002)
RC106	11	1427,13	Cordeau, Laporte, Mercier (2000)
RC207	11	1230,54	Taillard et al (1997)
RC108	10	1139,82	Taillard et al (1997)

Tableau 2.13

Meilleures solutions pour les problèmes RC2  
(Source : Berger et Barkaoui, 2003)

Tests	No. véhicules	Distances totales	Références
RC201	4	1046,94	Cordeau, Laporte, Mercier (2000)
RC202	3	1389,57	Homberger, Gehring (1999)
RC203	3	1060,45	Li, Lim (2003)
RC204	3	799,12	Homberger, Gehring (1999)
RC205	3	1302,42	Homberger, Gehring (1999)
RC206	3	1153,93	Rousseau, Gendreau, Pesant (2002)
RC207	3	1062,05	Cordeau, Laporte, Mercier (2000)
RC208	3	829,69	Rousseau, Gendreau, Pesant (2002)

Tableau 2.14

Comparaison des résultats moyens entre les publications  
(Source : Berger et Barkaoui, 2003)

Publications	C1		C2		R1		R2		RC1		RC2	
	Véh.	Dist.	Véh.	Dist.	Véh.	Dist.	Véh.	Dist.	Véh.	Dist.	Véh.	Dist.
Rochat, Taillard (1995)	10	828,38	3	589,96	12,25	1208,50	2,91	961,72	11,88	1377,39	3,38	1119,59
Chiang, Russell (1997)	10	828,38	3	591,42	12,17	1204,19	2,73	986,32	11,87	1397,44	3,25	1229,54
Taillard et al (1997)	10	828,38	3	589,96	12,17	1209,35	2,82	980,27	11,50	1389,22	3,38	117,44
Homberger, Gehring (1999)	10	828,38	3	589,96	11,92	1220,97	2,73	968,55	11,50	1388,24	3,25	1140,43
Cordeau, Laporte, Mercier (2000)	10	828,38	3	589,96	12,08	1210,14	2,73	958,69	11,50	1389,78	3,25	1134,52
Baker et al (2000)	10	828,38	3	589,96	12,41	1200,54	3	936,51	12	1383,21	3,38	1116,51
Liu, Shen (1999)	10	830,06	3	591,03	12,17	1249,57	2,82	1016,58	11,88	1412,87	3,25	1204,87
Rousseau, Gendreau, Pesant (2000)	10	828,38	3	589,86	12,08	1210,21	3	941,08	11,63	1382,78	3,38	1105,22
Li, Lim (2003)	10	828,38	3	589,86	12,08	1215,14	2,91	953,43	11,75	1385,47	3,25	1142,48
Berger, Barkaoui (2003)	10	828,38	3	589,93	11,92	1221,11	2,73	975,43	11,50	1389,89	3,25	1159,37

Dans la littérature, on trouve plusieurs méthodes pour résoudre le problème de tournée de véhicules avec fenêtres de temps. En effet les auteurs étudient les différentes facettes de ce problème dans le but de trouver de meilleures solutions. Parmi les articles les plus intéressantes se trouve celui de Heung-Suk (2002), de Li, Guo *et al.* (2004), de Berger, Barkaoui *et al.* (2003) et de Chen (2007).

L'article de Heung-Suk (2002) propose une approche basée sur l'algorithme génétique. L'auteur a développé un programme, avec une interface graphique, capable de résoudre des problèmes de 10 à 99 villes.

Li, Guo *et al.* (2004) proposent une approche basée sur le recuit simulé. Ils ont obtenu sept (7) nouvelles meilleures solutions pour les problèmes de Solomon. Les résultats sont donnés dans le tableau 2.15.

Tableau 2.15

Résultats de Li, Guo *et al.* (2004)

Problèmes	Meilleure solution connue			Nouvelle meilleure solution	
	Nb. véhicule	Distance	Référence	Nb. véhicule	Distance
<b>R103</b>	13	1292,85	Homberger, Gehring (1999)	13	1292,68
<b>R104</b>	9	1013,32	Homberger, Gehring (1999)	9	1007,31
<b>R107</b>	10	1113,69	Cordeau, Laporte, Mercier (2000)	10	1114,36
<b>R204</b>	2	849,62	Cordeau, Laporte, Mercier (2000)	2	849,05
<b>R207</b>	2	914,39	Rochat, Taillard (1995)	2	905,13
<b>RC106</b>	11	1427,13	Cordeau, Laporte, Mercier (2000)	11	1424,73

Berger et Barkaoui (2004) ont développé une approche basée sur un nouveau type d'algorithme génétique. Dans ce cas, les auteurs ont considéré une évolution simultanée de deux (2) populations des solutions et une relaxation de quelques contraintes. Les résultats obtenus par Berger et Barkaoui sont présentés dans le tableau 2.16.

Tableau 2.16

Résultats de (Berger et Barkaoui, 2004)

Problèmes	Meilleure solution connue			Nouvelle meilleure solution	
	Nb. véhicule	Distance	Référence	Nb. véhicule	Distance
<b>R108</b>	9	963,99	Shaw (1998)	9	960,88
<b>R110</b>	10	1125,40	Cordeau, Laporte, Mercier (2000)	10	1119,00
<b>RC105</b>	13	1633,72	Rousseau, Gendreau, Pesant (2002)	13	1629,44
<b>RC106</b>	11	1427,13	Cordeau, Laporte, Mercier (2000)	11	1424,73
<b>R221</b>	2	910,09	Homberger, Gehring (1999)	2	906,19

L'introduction des nouvelles contraintes sur la tournée de VRPTW vient d'élargir l'ensemble de ces types de problèmes. Si la fenêtre de temps est relaxée, on parle d'un problème de tournée de véhicules avec fenêtres de temps souples (*Vehicle Routing Problem with Soft Time Windows - VRPSTW*). En effet, si on relaxe la fenêtre de temps on permet que la livraison soit plus flexible, c'est-à-dire qu'on puisse dépasser la fenêtre de temps nécessaire pour faire la livraison. Par contre, une pénalité vient s'ajouter à la fonction-objectif.

Ioannou, Kritikos *et al.* (2003) proposent un VRPSTW dans lequel ils ont considéré qu'une partie des clients sont desservis en dehors de leurs fenêtres de temps. La méthode utilise l'approche du plus proche voisin pour faire la construction et la sélection des routes.

Chen (2007) a proposé un problème de VRPTW avec une contrainte *body clock*, qui prend en considération les temps d'arrêt des voyageurs. Effectivement, il a considéré des fenêtres de temps sur les nœuds et sur les arcs. En pratique, l'introduction de ces contraintes vient pénaliser la fonction-objectif d'un VRPTW classique.

#### **2.4.5 Problème de tournées de véhicules multi-dépôts**

Une compagnie peut avoir plusieurs dépôts pour desservir ses clients. Si les clients sont groupés autour des dépôts, alors le problème de distribution devrait être modélisé comme le problème de VRP. Cependant, si les clients et les dépôts sont mélangés, alors un problème de cheminement de véhicules multi-dépôts devrait être résolu.

Le problème de tournée de véhicules multi-dépôts exige l'attribution des clients aux dépôts. Chaque dépôt a sa propre flotte des véhicules. Chaque véhicule sert les clients et retourne ensuite au même dépôt. L'objectif est de réduire au minimum la flotte de véhicules et la somme du temps de voyage quand les clients sont desservis par plusieurs dépôts. Une solution faisable est obtenue si les conditions de VRP sont respectées.

Tai-Hsi, Chinyao *et al.* (2002) présentent une méthode pour résoudre le problème de tournées de véhicules multi-dépôts. La méthode permet de trouver le nombre optimal de centres de distributions et sélectionne les routes de distributions dans le but de minimiser le coût total du système. En raison de la complexité du problème, les auteurs ont proposé une décomposition en deux (2) sous-problèmes : un problème de VRP et un problème de localisation-allocation (LAP). Pour résoudre les sous-problèmes, les auteurs ont utilisé

un algorithme de recuit simulé sur des problèmes des tailles différentes (de 75 à 150 nœuds).

Cordeau, Gendreau *et al.* (1997) ont proposé un algorithme de recherche avec liste de tabous pour le problème de tournées de véhicules avec plusieurs dépôts. La taille des problèmes étudiés varie de 100 à 150 nœuds.

## 2.5 Autres généralisations du problème de tournées de véhicules

### 2.5.1 Problème de tournées de véhicules périodique

Dans le cas de tournée des véhicules périodique (PVRP) le problème de VRP est généralisé en prolongeant la période de planification sur une durée de  $m$  jours. L'objectif est de desservir tous les clients tout en minimisant la flotte de véhicules et la somme du temps sur une durée de  $m$  jours. Une solution est faisable si toutes les contraintes de VRP sont respectées. Autrement dit, on ne peut pas retourner au dépôt dans la même journée, car la période de voyage est de  $m$  jours.

Si la période de planification  $m=1$ , le PVRP devient un cas de VRP classique. Dans le problème de PVRP, la demande quotidienne d'un client est toujours fixée.

### 2.5.2 Problème de tournée de véhicules stochastique

Les problèmes de tournées de véhicules stochastiques (SVRP) sont des problèmes VRP où un ou plusieurs composants du problème sont aléatoires. On donne l'exemple de trois (3) différents types de SVRP :

- a. clients stochastiques – chaque client  $v_i$  est présent avec une probabilité  $p_i$  et absent avec une autre probabilité  $1 - p_i$  ;
- b. demandes stochastiques – la demande  $d_i$  de chaque client est aléatoire;

- c. périodes stochastiques – les temps de voyage  $t_{ij}$  et les temps de service  $\varepsilon_i$  sont des variables aléatoires.

Pour obtenir une solution dans SVRP, on doit parcourir deux (2) étapes :

- a. on détermine une première solution avant de savoir les réalisations des variables aléatoires;
- b. quand les valeurs des variables sont connues, on reprend les calculs.

L'objectif est de réduire au minimum la flotte de véhicules et la somme de temps de voyage nécessaires pour desservir tous les clients en tenant compte des valeurs aléatoires qui interviennent dans le système.

### **2.5.3 Problème de tournées de véhicules multiples**

Le problème de tournées de véhicules multiples (SDVRP) est une variante de VRP car le même client peut être desservi par des véhicules différents. L'objectif est de servir tous les clients tout en optimisant la flotte des véhicules et le temps total.

### **2.5.4 Problème de tournées de véhicules avec ravitaillements intermédiaires**

Le problème de tournées de véhicules avec équipements satellites est relativement nouveau, on utilise des équipements satellites pour remplir de nouveaux véhicules pendant un itinéraire. Le remplissage satellite permet aux conducteurs de continuer à faire des livraisons sans nécessairement retourner au dépôt central. On retrouve cette situation principalement dans la distribution de carburants et de certains articles au détail.

## 2.6 Problème de la planification de tournées de véhicules scolaires

Pour la planification de véhicules scolaires, plusieurs approches sont proposées dans la littérature. En pratique, il y a deux (2) approches de base, une sur les écoles et l'autre sur les enfants.

La première approche vise le point de vue de l'école, c'est-à-dire qu'on permet une planification seulement pour une école. On embarque dans un autobus les élèves qui appartiennent à une seule école. Cette approche a été traitée par nombreux auteurs, Clarke et Wright (1964), Bennett et Gazis (1972), Gavish (1979) ainsi que Thangiah et Nygard (1992).

La deuxième approche vise un embarquement mixte, c'est-à-dire qu'on peut embarquer dans un autobus des enfants allant à plusieurs écoles. Concrètement, cette approche s'ajoute à la première, ce qui permet de mieux gérer la planification de tournées de véhicules scolaires. Cette approche proposée par Braca, Bramel *et al.* (1997) est caractérisée par sa flexibilité par rapport à l'embarquement des élèves.

On peut affirmer que la plupart des auteurs ont choisi comme objectif la diminution des coûts, c'est-à-dire qu'ils ont étudié la possibilité d'utiliser un nombre minimum d'autobus pour faire le routage. Bodin (1979), Swersey (1984), Braca, Bramel *et al.* (1997) ont étudié cette possibilité.

Gavish et Shlifer (1979) et Thangiah (1992) ont proposé de minimiser le nombre d'autobus de même que le temps total des itinéraires. La mesure d'optimiser le niveau de service fourni aux enfants, et donc d'optimiser la perte de temps que subit chaque enfant par rapport à un résultat, n'est pas une approche courante dans la littérature. Dans un article de Bennett et Gazis (1972) on trouve une approche basée sur la somme des temps passés par les enfants dans les véhicules.

D'autres auteurs proposent de minimiser le temps en mettant des contraintes sur la planification des trajets des véhicules scolaires. Par exemple Thangiah (1992) propose de limiter le temps de trajet de chaque véhicule et Braca (1997) introduit une contrainte sur l'heure de départ.

Newton et Thomas (1969), proposent un routage des véhicules scolaires en traitant le problème avec une (1) seule école et 90 arrêts. Ils donnent une solution pour le problème de voyageur de commerce en divisant les routes, dont chaque route est couverte par un seul autobus.

Angel, Caudle *et al.* (1972) proposent une approche basée sur la résolution d'un ensemble formée de 1500 écoliers et de cinq (5) écoles, dont les arrêts sont partitionnés et elles génèrent des tournées sur chaque élément de la partition - algorithme de groupement (clustering algorithm).

Bennett et Gazis (1972) utilisent un modèle modifiée de Clarke et Wright (1964) en minimisant le trajet total (256 arrêts, 30 et 1 seule école).

L'approche de Bodin et Berman (1979) est basée sur les travaux de Newton et Thomas (1969), mais ils tentent d'améliorer la solution en appliquant la procédure Lin-3-opt développée par Lin (1965). L'algorithme développé par Bodin et Berman utilise deux (2) composants additionnels : le regard en avant et le diviseur d'arrêt d'autobus. Le regard en avant permet à l'ordre initial d'être légèrement modifié, alors que le diviseur d'arrêt d'autobus permet à un arrêt d'être fractionné en plusieurs.

Swersey et Ballard (1984) ont étudié l'approche concernant 38 autobus, 100 routes et une (1) seule école. Concrètement, ils traitent le problème de la planification de véhicules scolaires en donnant une méthode afin de déterminer le nombre minimum d'autobus qui peuvent couvrir les routes.

Pour Desrosier (1986), le problème de planification des véhicules scolaires est un programme de nombres entiers et résolu en utilisant une approche de génération de colonnes.

Braca (1997) propose une approche basée sur les enfants, c'est-à-dire qu'il ne décompose pas le problème de génération de tournées des véhicules de planification. Il donne une tournée initiale entre la maison et l'école et après il ajoute d'autres enfants tout en minimisant la longueur totale de la tournée.

## **2.7 Conclusion**

Dans ce chapitre, nous avons présenté une revue de la littérature du voyageur de commerce. D'autres problèmes dérivés (différents types de routages des véhicules) ont été aussi étudiés.

Le chapitre suivant présente la méthodologie poursuivie d'une revue des algorithmes étudiés.

## CHAPITRE 3

### MÉTHODOLOGIE ET REVUE DES ALGORITHMES ÉTUDIÉS

#### 3.1 Méthodologie et procédures proposées

La recherche scientifique a permis d'accumuler une somme de connaissances sur différentes méthodes de résolution du problème de voyageur de commerce. Les dernières années ont apporté beaucoup de nouveautés concernant la recherche sur cette famille de problèmes. Son importance se justifie d'une part par sa grande difficulté et d'autre part par de nombreuses applications pratiques pouvant être formulées. Malgré le fait que ces problèmes sont faciles à définir, ils demeurent encore difficile à résoudre, surtout pour des instances de grande taille.

Suite à la revue de la littérature présentant le problème du voyageur de commerce et d'autres problèmes dérivés, on décrira les algorithmes utilisés pour résoudre ce problème. En conséquence, chacun de ces algorithmes sera exposé d'une manière précise. Finalement, une analyse des résultats obtenus sur des instances réelles provenant de la littérature permettra d'évaluer les méta-heuristiques et de tirer les conclusions qui s'imposent.

#### 3.2 Notion d'une méta-heuristique

Depuis une vingtaine d'années, les recherches ont connu des progrès importants avec l'arrivée d'une nouvelle génération de méthodes de résolution, souvent appelées méta-heuristiques. Une méta-heuristique est constituée d'un ensemble de concepts fondamentaux qui permettent d'aider à la conception de méthodes heuristiques pour un problème d'optimisation (Aarts et Lenstra ,1997).

Le mot méta-heuristique est dérivé de la composition de deux (2) mots grecs :

- a. **heuristique** qui vient du verbe heuriskein et qui signifie « trouver »;
- b. **meta** qui est un suffixe signifiant « au-delà », « dans un niveau supérieur ».

Plusieurs définitions ont été proposées pour expliquer ce qu'est une méta-heuristique, mais aucune de ces définitions n'est en totalité acceptée. Nous donnons ensuite quelques exemples de définitions telles qu'on les a trouvées dans la littérature :

" A metaheuristic is formally defined as an iterative generation process which guides a subordinate heuristic by combining intelligently different concepts for exploring and exploiting the search space, learning strategies are used to structure information in order to find efficiently near-optimal solution " (Osman et Laporte, 1966).

" Metaheuristic are typically high level strategies which guide an underlying more problem specific heuristic, to increase their performance. The main goal is to avoid the disadvantages of iterative improvement and, in particular, multiple descents by allowing the local search to escape for local optima. This is achieved by either allowing worsening moves or generating new starting solutions for the local search in a more « intelligent » way than just providing random initial solutions. Many of the methods can be interpreted as introducing a bias such that high quality solutions are produced quickly. This bias can be of various forms and can be cast as descent bias (based on the objective function), memory bias (based on previously made decision) or experience bias (based on prior performance). Many of the metaheuristic approaches rely on probabilistic decision made during the search. But the main difference to pure random search is that in metaheuristic algorithms, randomness is not used blindly but in an intelligent biased form " (Stützle, 1999).

" A metaheuristic is an iterative master process that guides and modifies the operations of subordinate heuristics to efficiently produce high-quality solutions. It may manipulate a complete (or incomplete) single solution or a collection of solutions at each iteration. The subordinate heuristics may be high (or low) level procedures, or a simple local search, or just a constructive method " (Voß, Martello *et al.* 1999).

Pour faire un résumé à ces définitions, on peut dire que les propriétés fondamentales des méta-heuristiques sont les suivantes :

- a. le but est d'explorer l'espace de recherche efficacement afin de trouver des solutions quasi-optimales;
- b. les méta-heuristiques sont des stratégies qui permettent de guider la recherche d'une solution optimale;
- c. les méta-heuristiques sont en général non déterministes, donc elles ne donnent aucune garantie d'optimalité;
- d. les méta-heuristiques peuvent contenir des mécanismes qui permettent d'éviter d'être bloqué dans des régions de l'espace de recherche;
- e. les méta-heuristiques peuvent faire appel à des heuristiques qui tiennent compte de la spécificité du problème traité, mais ces heuristiques sont contrôlées par une stratégie de niveau supérieur;
- f. les méta-heuristiques peuvent faire usage de l'expérience accumulée durant la recherche de l'optimum pour mieux guider la suite du processus de recherche;
- g. les techniques qui constituent des algorithmes de type méta-heuristiques vont de la simple procédure de recherche locale à des processus d'apprentissage complexes.

Ces propriétés, tirées des plusieurs ouvrages de Laporte (1992), Osman (1993) ,Voß, Martello *et al.*,(1999), permettent d'encadrer d'un façon assez précis la notion d'une méta-heuristique.

### 3.3 Algorithme du recuit simulé dans la recherche opérationnelle

La méthode du recuit simulé tire son origine de la physique statique. Ce processus utilisé en métallurgie consiste à porter un matériau à température élevée, puis à le refroidir d'une manière contrôlée jusqu'à l'obtention d'une structure stable. Durant ce processus, l'énergie du matériau est minimisée, en évitant de rester piégé dans des optimums locaux (Aarts et Korst, 1989).

#### 3.3.1 Historique du recuit simulé

- Kirkpatrick (1983), propose l'algorithme de recuit simulé en faisant une analogie avec le processus de recuit des matériaux.
- Geman et Geman (1984), proposent un algorithme de recuit simulé conventionnel. Szu et Hartley (1987), présentent un algorithme de recuit simulé rapide basé sur la méthode de Cauchy.
- Ingber et Rosen (1992), proposent un algorithme de recuit simulé très rapide.
- Tsallis et Stariolo (1996), utilisent la théorie de micros fractales comme base pour un algorithme de recuit simulé généralisé.
- Acton et Bovik (1996), proposent un algorithme de recuit simulé déterministe. Salazar et Toral (1997), utilisent la théorie de Monte Carlo comme base pour un algorithme de recuit simulé hybride.
- Xi-Huai et Jun-Jun (2004), proposent un algorithme hybride de recuit simulé avec un algorithme d'essaim de particules.

#### 3.3.2 Rappel de thermodynamique

On considère un système physique pouvant prendre un certain nombre d'états  $C \in C_\Omega$ . Si on fait une analogie avec la mécanique statique on pourrait dire qu'à l'équilibre, le système tend à la fois à minimiser son énergie (moyenne) :

$$e[p] := \sum_{C \in C_0} P(C)E(C) \quad (3.1)$$

(où  $P(C)$  est la probabilité d'occuper l'état et  $E(C)$  l'énergie associée)

Le système tend également à maximiser son entropie :

$$u(p) := -\sum_{C \in C_0} P(C) \ln P(C) \quad (3.2)$$

Les deux tendances contradictoires sont gérées par la température  $T > 0$  du système.

D'autre part, le système doit minimiser globalement son énergie libre :

$$W := e - Tu = \sum_{C \in C_0} P(C)E(C) + T \sum_{C \in C_0} P(C) \ln P(C) \quad (3.3)$$

dont le minimum ( $W_{\min} = -T \ln Z(\beta)$ ) est atteint par la distribution de Gibbs :

$$P(A) = \frac{\exp(-\beta E(C))}{Z(\beta)} \quad (3.4)$$

$$\beta := \frac{1}{T}$$

$$Z(\beta) := \sum_{C \in C_0} \exp(-\beta E(C)) \quad (3.5)$$

À basse température  $\beta \gg 0$ , l'énergie libre  $W$  est contrôlée par son composant énergétique. En conséquence, le système est essentiellement figé dans son état fondamental :

$$\begin{aligned} \min_{C \in C_0} E(C) &= E(C_0) \\ e &\cong E(C_0) \\ u &\cong 0 \end{aligned} \quad (3.6)$$

L'état fondamental est défini par l'expression  $\min_{C \in C_0} E(C) = E(C_0)$

$e \cong E(C_0)$  - minimal

$u \cong 0$  - minimal

A la haute température  $\beta \ll 1$ , l'entropie domine est le système et distribué de façon uniforme, donc on va avoir :

$$P(C) \cong \text{constante} \quad (3.7)$$

pour lequel  $u \cong \ln(C)$  est maximal. On peut écrire l'énergie moyenne  $e[\beta]$  comme :

$$e[\beta] = \sum_{C \in C_n} P(C)E(C) = -\frac{\partial \ln Z(\beta)}{\partial \beta} \quad (3.8)$$

### 3.3.3 Comment déterminer la température initiale ?

Pour le calcul de la température de départ, la littérature ne fournit que très peu d'information. Peu d'études ont été menées ayant comme objectif de déterminer la bonne température de départ. Dans la plupart des cas, la température a été choisi d'une manière empirique (la valeur de 100 degrés comme température de départ est souvent utilisée) ce qui pourra affecter le temps de calcul et aussi la valeur de la fonction-objectif.

Une des ces méthodes empiriques est basée sur l'observation de la variation moyenne de la fonction objective. En prenant une solution initiale  $s_0$ , on génère, par transformations élémentaires aléatoires, un certain nombre de solutions  $s'_0$  (environ 50 à 100) telles que  $f(s'_0) > f(s_0)$ , et on calcule la variation moyenne  $\langle \Delta f \rangle_m$ . Une température initiale  $T_{mor}$  est calculée de façon à accepter au départ une certaine proportion  $p_{mor}$  de mouvements dégradant la fonction  $f$ . Pour une température initiale « moyenne », la valeur de la probabilité est  $p_{mor} = 0.5$ . La valeur de  $T_{mor}$  est déduite de la formule suivante:

$$p_{mit} = e^{-\frac{|\langle \Delta f \rangle_m|}{T_{mit}}} \Rightarrow T_{mit} = -\frac{|\langle \Delta f \rangle_m|}{\ln(p_{mit})} \quad (3.9)$$

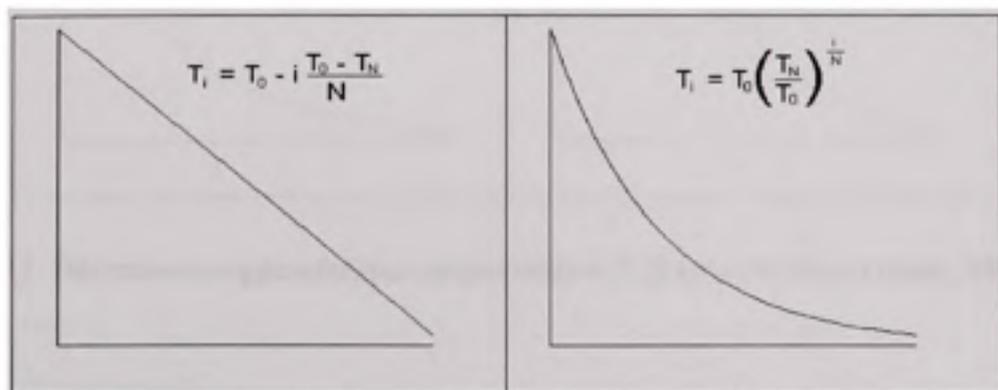
Le rôle de la température  $T$  au cours du processus de recuit simulé est très important. Une forte décroissance de température risque de piéger l'algorithme dans un minimum local, alors qu'une faible décroissance au début du processus entraîne une convergence très lente de l'algorithme. Un compromis pour adapter la décroissance de la température à l'évolution du processus consiste à adopter une décroissance géométrique

$$T_{\alpha+1} = k \cdot T_{\alpha}, \text{ avec } (0 < k < 1) \quad (3.10)$$

Pour le changement de palier de température, on peut simplement spécifier un nombre de transformations, acceptées ou non, au bout duquel la température sera abaissée.

### 3.3.3.1 Décroissance de la température

Au niveau de la décroissance du paramètre  $T$ , on peut prendre une décroissance géométrique de la forme suivante (voir la Figure 3.1) :



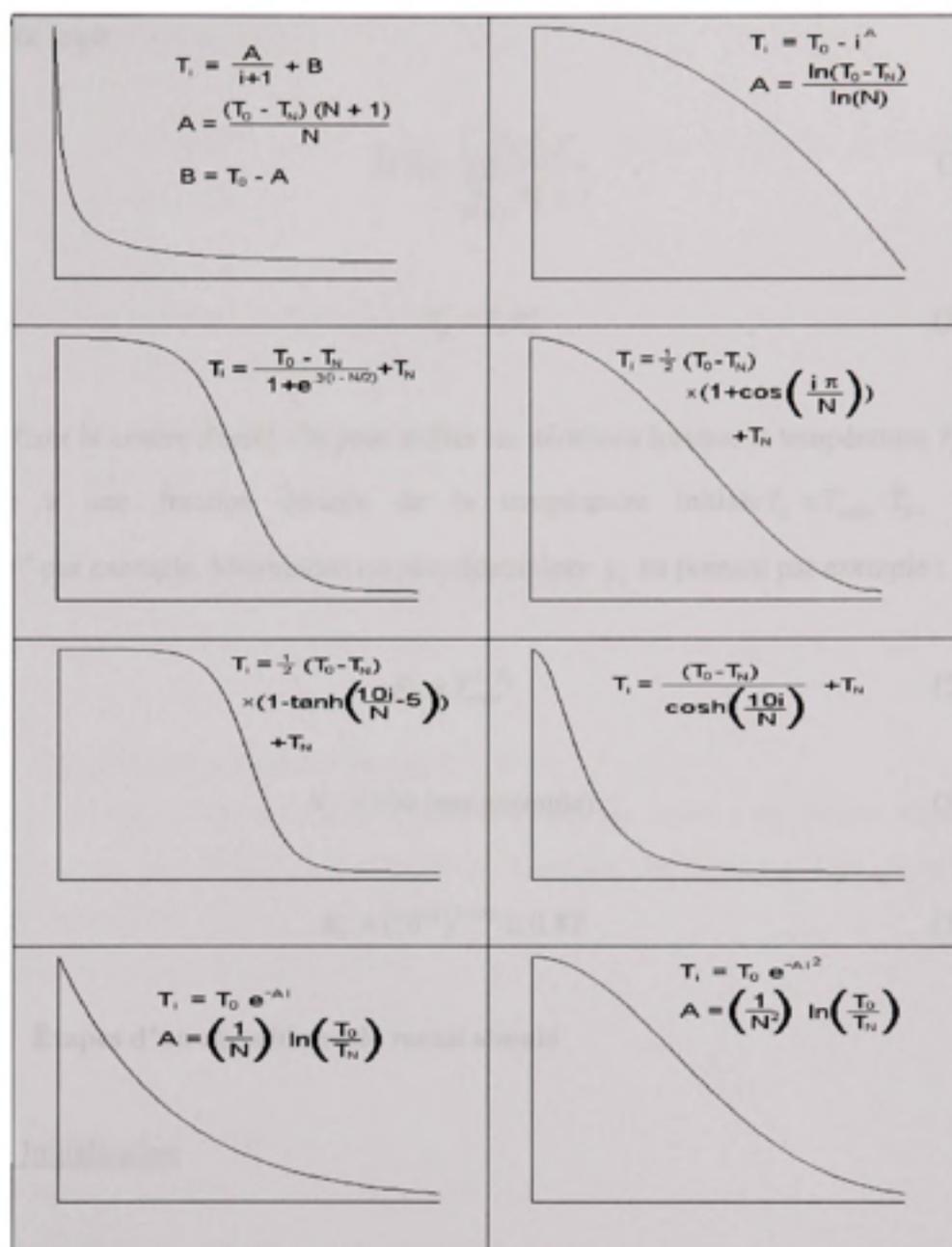


Figure 3.1 Décroissance géométrique du paramètre  $T$  (Extrait de Bowerman, 1995).

Donc on va avoir :

$$f(\bar{y}_i) : \frac{f(\bar{y}_i)}{\sum_j^n f(\bar{y}_i)} \quad (3.11)$$

$$T_n = T_0 R_T^n \quad (3.12)$$

Il reste à fixer le critère d'arrêt. On peut arrêter les itérations lorsque la température  $T_n$  est inférieure à une fraction donnée de la température initiale  $T_n < T_{ratio} \cdot T_0$ , avec  $T_{ratio} = 10^{-6}$  par exemple. Maintenant on peut déterminer  $R_T$  en prenant par exemple :

$$R_T = T_{ratio}^{1/N_T} \quad (3.13)$$

$$N_T = 100 \text{ (par exemple)} \quad (3.14)$$

$$R_T = (10^{-6})^{1/100} \cong 0.87 \quad (3.15)$$

### 3.4 Étapes d'un algorithme de recuit simulé

#### Étape 1 : Initialisation

- définir un paramètre (la température  $T$ ) et l'initialiser à une valeur élevée  $T_i$ ;
- choisir l'état initial ( $d$ ) selon une méthode adéquate;
- initialiser :  $E_{min} = \Delta \varepsilon_0$  ( $\Delta \varepsilon_0 = \Delta \varepsilon$  de la première itération) et  $t_{d_{opt}} = d_0$  ( $d_0$ , de la première itération).

**Étape 2 : Méthode de Metropolis**

- a. choisir aléatoirement un état  $d'$  qui est la perturbation de l'état  $d$  et calculer la variation d'énergie :  $\Delta\varepsilon = \varepsilon_c(d') - \varepsilon_c(d)$ ;
- b. Règle d'acceptation de Metropolis :
  - a. si  $\Delta\varepsilon < 0$  , remplacer  $d$  par  $d'$  et aller à l'étape 3;
  - b. si  $\Delta\varepsilon \geq 0$  , remplacer  $d$  par  $d'$  avec la probabilité :  $\exp(-\Delta\varepsilon/T)$  , et aller à l'étape 3;
- c. Règle de sauvegarde de l'état d'énergie minimal :
  - c. si l'état  $d$  est accepté et  $\Delta\varepsilon < E_{\min}$  ;  $E_{\min} = \Delta\varepsilon$  et  $d_{opt} = d$  .

**Étape 3 : Test d'équilibre thermodynamique local**

- a. si dans l'étape 2 le nombre de diminution d'énergie dépasse un nombre requis ou si trop de perturbation qui n'engendrent aucune diminution d'énergie, aller à l'étape 4;
- b. sinon on retourne à l'étape 2.

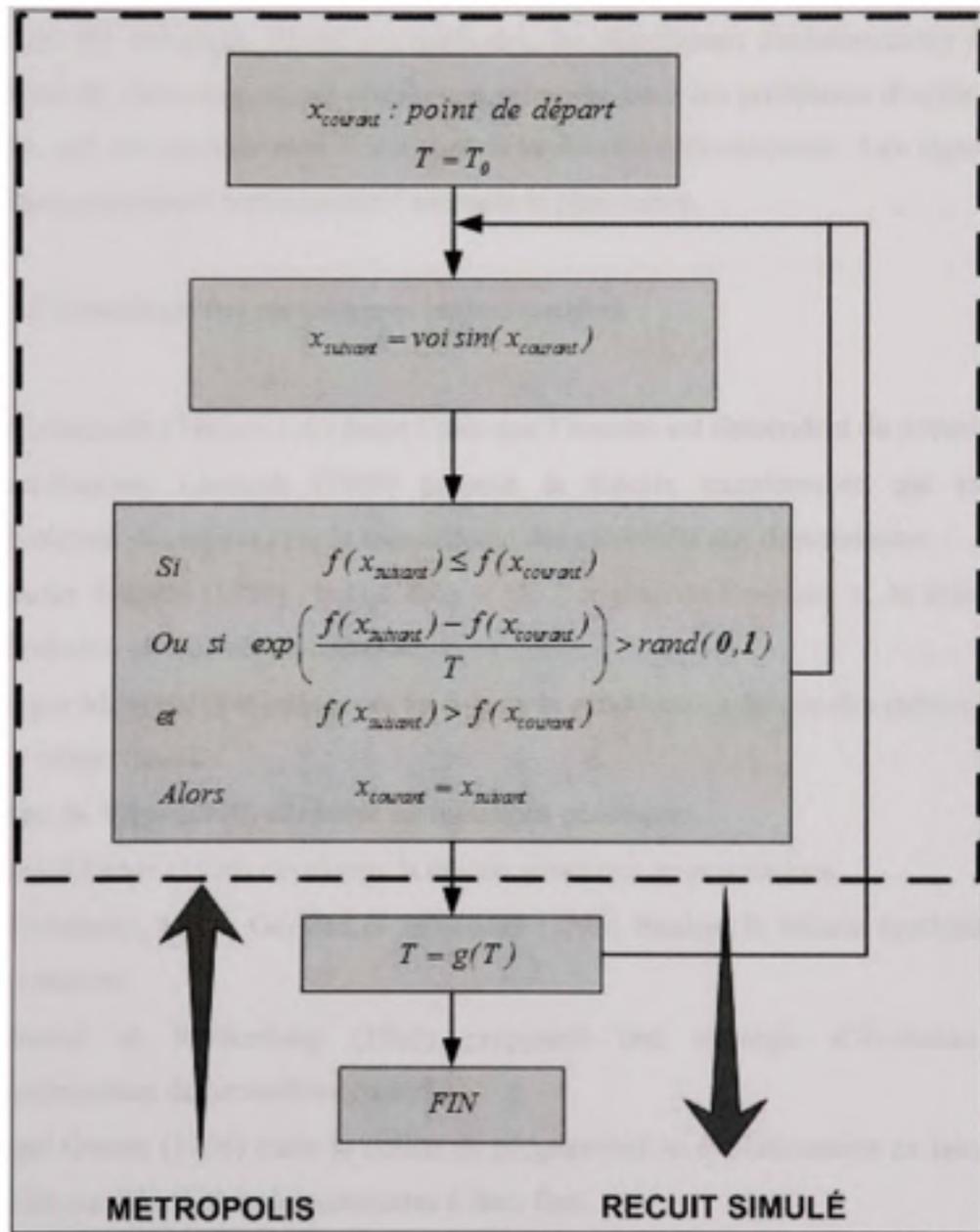
**Étape 4 : Programme du recuit**

- a. diminuer lentement  $T : T_k = \alpha T_{k-1}, 0 < \alpha < 1$  .

**Étape 5 :**

- a. si  $T < T_c$  ( $T_c$ , est la température de congélation prescrite) ou s'il apparaît qu'un état stable est obtenu, on va avoir la fin du programme avec  $d_{opt}$  qui décrira l'état stable du système;
- b. sinon on retourne à l'étape 2.

Le diagramme de la Figure 3.2 nous donne l'algorithme de recuit simulé tel que nous l'avons présenté.



**Figure 3.2** L'algorithme de la méthode de recuit simulé  
Source : Adapté de Kirkpatrick , Gelett *et al.* (1983).

### 3.5 Algorithme génétique dans la recherche opérationnelle

Dans les dernières décennies, les chercheurs se sont trouvés confrontés à des problèmes d'optimisation plus complexes. Pour résoudre ces problèmes, plusieurs algorithmes de calcul ont été envisagés. Parmi ces méthodes, les algorithmes évolutionnaires forment une classe de méta-heuristique récemment proposée pour les problèmes d'optimisation difficile, qui ont pratiquement bouleversé la recherche opérationnelle. Les algorithmes génétiques constituent certainement l'exemple le plus connu.

#### 3.5.1 Chronologie des algorithmes évolutionnaires

- Anaximandre (546 a.v J.-C) lance l'idée que l'homme est descendant du poisson.
- Jean-Baptiste Lamarck (1809) propose la théorie transformiste qui explique l'évolution des espèces par la transmission des caractères aux descendantes.
- Charles Darwin (1859) publie dans « De l'origine de l'espèces », la théorie sur l'évolution et la sélection naturelle.
- Gregor Mendel (1896) découvre les lois de la génétique en faisant des recherches sur des plants (pois).
- Hugo de Vries (1910) découvre les mutations génétiques.
- Ronald Fisher (1930) développe la théorie génétique de populations.
- Dobzhansky, Mayr, Gaylord et JulHuxley (1946) fondent la théorie synthétique de l'évolution.
- Schewel et Rechenberg (1965) proposent une stratégie d'évolution pour l'optimisation de paramètres continus.
- Fogel Owens (1966) traite la notion de programmation évolutionnaire en faisant des études sur l'évolution des automates à états finis.
- Motoo Kimura (1970) introduit la notion de hasard au niveau de l'évolution des gènes.

- John Holland (1975) propose l'algorithme génétique comme modalité de se rapprocher de l'optimum d'une fonction.

### 3.5.2 Fonctionnement des algorithmes génétiques

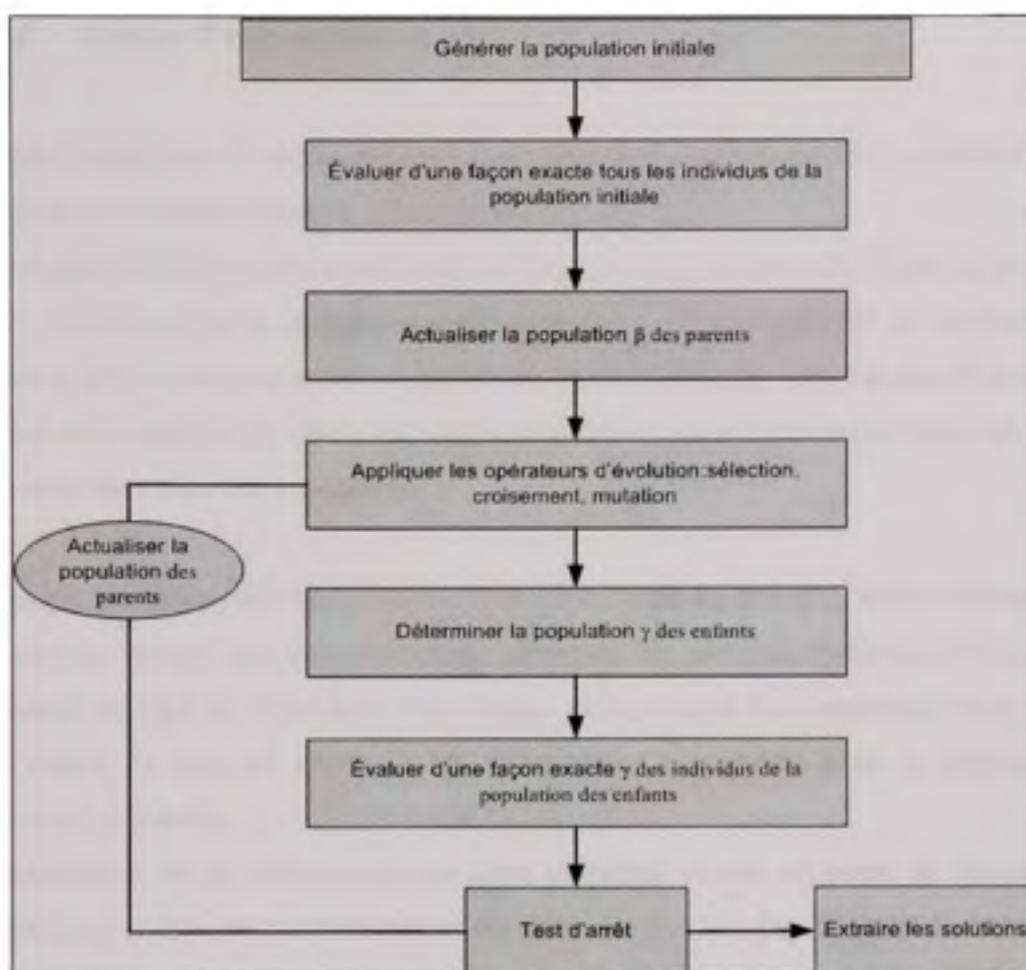
#### 3.5.2.1 Objectif des algorithmes génétiques

L'objectif des algorithmes génétiques est de minimiser ou maximiser une fonction

$F: Q \rightarrow R$ , où  $Q$  est un ensemble quelconque nommé espace de recherche et  $F$  est la fonction d'évaluation ou la fonction de *fitness*.

#### 3.5.2.2 Principe des algorithmes génétiques

L'algorithme génétique proposé par Holland (1975) fait partie de la classe des algorithmes évolutionnaires. La programmation évolutionnaire introduite par Fogel (1966) et les stratégies d'évolutions développées par Schwefel (1981) sont d'autres approches qui font partie de la classe des algorithmes évolutionnaires. La Figure 3.3 nous montre le principe d'un algorithme évolutionnaire standard donnée par (Dorigo , 2003).



**Figure 3.3 Principe d'un algorithme évolutionnaire standard.**  
(Source : Dorigo, 2003)

Le principe d'un algorithme génétique se décrit de la manière suivante :

Soit un ensemble de  $N$  points dans un espace de recherche qu'on peut le choisir au hasard et qui forme la population initiale. On suppose que la population initiale est constituée de  $\beta$  individus. Chaque individu possède une certaine performance d'adaptation à l'objectif visé. Par exemple, dans le cas de minimisation d'une fonction objectif  $F$ ,  $\beta$  est autant plus performant que  $F(\beta)$  est plus petit. En effet, nous pouvons dire que les algorithmes génétiques agissent sur l'ensemble de la population et non pas sur un individu ou sur une partie de la population.

### 3.5.2.3 Codage d'une population

Le codage représente la modalité d'associer une structure de donnée aux variables d'état à l'aide d'un alphabet approprié (Dumas, 2007).

Le codage d'une population d'individus se fait par analogie avec la biologie. Le point de départ est d'associer un chromosome ou génotype à chaque individu de la population (Holland, 1975). On a donc un ensemble de chromosomes qui code chacun un point de l'espace de recherche. Le choix du codage d'un chromosome déterminera donc en partie l'efficacité de l'algorithme génétique.

En théorie, on utilise différentes sortes d'alphabets, mais en pratique, seul le codage réel et le codage binaire sont majoritairement appliqués. En pratique, Holland (1975) utilise un codage de type bit, c'est-à-dire que chaque chromosome était représenté sous forme d'un chaîne de bits qui contenait les informations nécessaires pour le définir dans l'espace de recherche.

L'inconvénient de ce codage c'est que deux individus voisins en terme de distance de Hamming ne codent pas nécessairement des éléments proches dans l'espace de recherche (Dumas, 2007). Pour résoudre le problème, le code Gray (codage binaire) a été adopté. Celui la fait en sorte qu'on permet de ne faire changer qu'un seul bit à la fois quand un nombre est augmenté d'une unité (Dumas, 2007). La Figure 3.4 nous montre les différences entre un codage réel et un codage binaire.

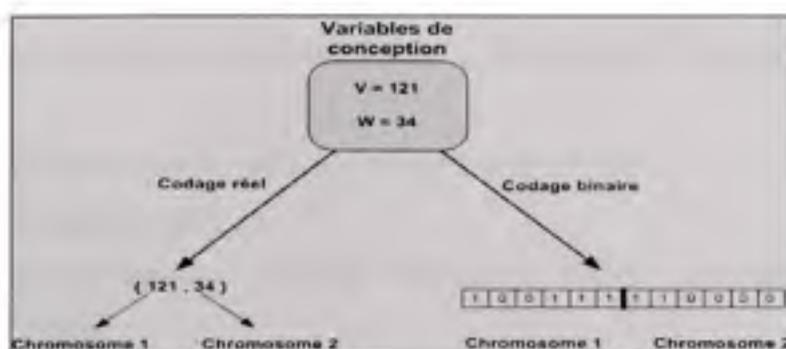


Figure 3.4 Exemple d'un codage réel et binaire (Dumas, 2007)

#### 3.5.2.4 Codage d'un algorithme génétique

Dans le problème du voyageur de commerce, chaque chromosome représente un élément de l'espace de recherche. Ces éléments désignent pratiquement des chemins possibles pour le voyageur de commerce. Cependant, chaque ville est représentée par un point situé dans l'espace de recherche et porte un numéro d'identification. On trouvera que chaque chemin représente une succession de villes, donc chaque chromosome représente une succession de nombres.

#### 3.5.2.5 Fonction d'évaluation d'un algorithme génétique

La fonction d'évaluation est utilisée pour déterminer la valeur d'un point situé dans l'espace de recherche. Dans le cas de l'algorithme génétique, le résultat de cette fonction permet d'accepter ou non un individu, donc de retenir seulement les individus pour laquelle la fonction d'évaluation est la meilleure. Cela fait en sorte que les individus moins performants seront progressivement éliminés.

#### 3.5.2.6 Sélection des individus dans un algorithme génétique

L'opérateur de sélection permet d'identifier les meilleurs individus d'une population et d'éliminer les autres. Cela fait en sorte que seulement les meilleurs individus pourront participer à la recombinaison.

On trouve dans la littérature plusieurs méthodes de sélection. Les méthodes les plus connues sont :

- a. la roulette biaisée (*roulette wheel selection*) (Goldberg ,1989) ;
- b. le tournoi (Goldberg ,1989) ;
- c. la sélection stochastique (*stochastic remainder without replacement selection*) (Goldberg ,1989).

### 3.5.2.6.1 Méthode de roulette biaisée

La méthode de roulette consiste à utiliser le concept de la roulette de casino pour tirer les individus les mieux adaptés. En outre, ce processus de sélection utilise la division d'une roulette de casino en secteurs, qui dépend du grade d'adaptation de chaque individu. Le largeur du secteur d'un individu  $\bar{y}_i$  est proportionnel à sa performance :

$$f(\bar{y}_i) = \frac{f(\bar{y}_i)}{\sum_j^n f(\bar{y}_j)} \quad (3.16)$$

En faisant tourner la roue, on arrive à sélectionner un individu qui est mieux adapté donc qui a plus de chance pour être choisi. Cette méthode favorise les individus meilleurs adaptés, mais elle donne une chance aux autres. L'espérance  $n_i$  de nombre de copies d'un élément  $\bar{y}_i$  de la population est donnée par :

$$n_i = \frac{N}{\sum_j^n f(\bar{y}_j)} f(\bar{y}_i) \quad (3.17)$$

La Figure 3.5 illustre la méthode de la roulette biaisée (Goldberg, 1989).

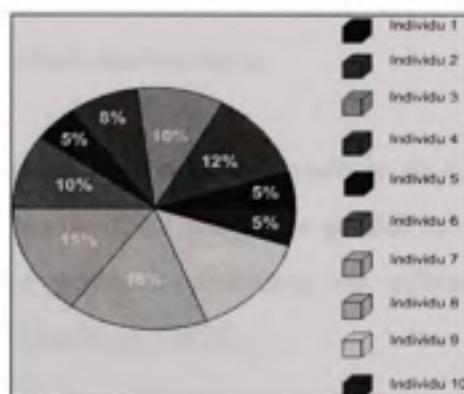
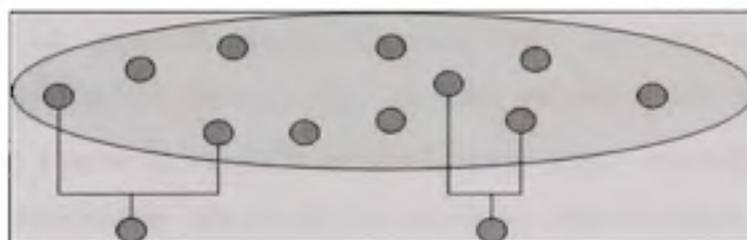


Figure 3.5 La méthode de la roulette biaisée (Goldberg, 1989)

### 3.5.2.6.2 Méthode de tournoi

Le principe de tournoi consiste à utiliser les individus d'une population dans une confrontation pour déterminer les meilleurs. On peut y avoir plusieurs variantes pour déterminer les individus les plus puissants. Goldberg (1991) propose une variante de tournoi entre deux individus avec rééligibilité, variante qui est présentée dans la Figure 3.6.

Baker, Edward *et al.* (1986) proposent un modèle de tournoi qui s'assure que la distribution des valeurs d'adaptation d'une population soit proche d'une distribution de Boltzmann.



**Figure 3.6** La méthode de tournoi entre deux individus avec rééligibilité.  
(Source : Goldberg, 1991)

### 3.5.2.6.3 Méthode de sélection stochastique

Lorsque on utilise les méthodes de la roulette biaisée ou du tournoi, on risque d'avoir un résultat final après un nombre limité de tirages, ce qui fait en sorte qu'on écarte plusieurs possibilités. Pour éviter ce genre de problème, on utilise la méthode de sélection stochastique développé par Goldberg, (1991).

### 3.5.2.7 Recombinaison d'une population

La notion de recombinaison a été introduite pour expliquer la modalité des nouveaux chromosomes de garder les meilleurs sections des chromosomes anciens (Holland ,1975). À partir de deux (2) chromosomes parents, on obtiendra par recombinaison un nouveau chromosome enfant qui héritera de certaines caractéristiques de ces parents, avec une certaine probabilité de recombinaison. Cette probabilité de recombinaison représente la fréquence à laquelle la recombinaison est appliquée. L'opérateur de recombinaison des algorithmes génétiques permet de construire une nouvelle solution avec une mémoire de type population. Parmi les opérateurs de recombinaison les plus fréquemment utilisés, on cite ceux de croisement à un ou plusieurs points et celui de croisement uniforme. L'opérateur de recombinaison à un point permet de obtenir une nouvelle solution  $r$  à partir de deux (2) autres solutions  $x$  et  $y$ . En pratique, on génère un nombre aléatoire  $j$  compris dans un intervalle  $[1,n]$ , déterminé par une permutation de type  $r_i = x_i (i = 1, \dots, j-1)$  et  $r_i = y_i (i = j, \dots, n)$ . En effet, on obtient une nouvelle solution (solution enfant) à partir des deux (2) solutions parents. Cette nouvelle solution enfant présente les caractéristiques des parents (les premières caractéristiques de la solution  $r$  sont identiques au premier parent  $x$  et les dernières sont identiques au deuxième parent  $y$ ). On peut aussi obtenir une solution de type  $r'$  en permutant les rôles de deux vecteurs  $x$  et  $y$ . On procède de cette manière pour s'assurer qu'on obtiendra toutes les caractéristiques dans la nouvelle solution.

L'opérateur de recombinaison à deux points consiste à générer deux (2) nombres aléatoires  $j$  et  $p$  comprises dans l'intervalle  $[1,n]$  et déterminés par une permutation de type  $r_i = x_i (i = 1, \dots, j-1, i = p, \dots, n)$  et  $r_i = y_i (i = j, \dots, p-1)$ . L'opérateur de recombinaison peut être généralisé pour  $n$  points. La méthode de recombinaison uniforme consiste à obtenir une nouvelle solution  $r$  à partir de caractéristiques identiques qu'on le trouve dans les solutions parents ( $x$  et  $y$ ). La Figure 3.7 nous montre les différents types de recombinaison qu'on peut y avoir pour un algorithme génétique. Dans le cadre de

cette thèse on utilise le procédé de recombinaison à deux (2) points en permutant aussi les vecteurs qui représente les parents.

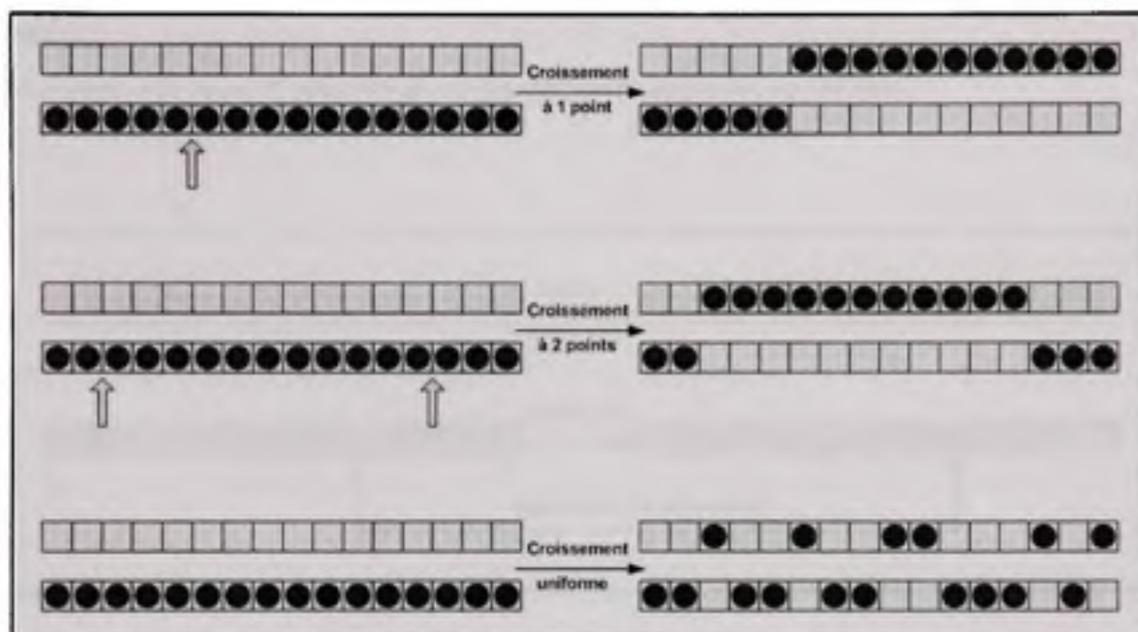


Figure 3.7 Les opérateurs de recombinaison les plus fréquemment utilisés.

### 3.5.2.8 Mutation d'une population

L'opérateur de mutation permet d'éviter la dégénérescence d'une population en enrichissant les caractéristiques des chromosomes, c'est-à-dire qu'on essaie de contourner une convergence des individus d'une population vers un optimum local. Techniquement, seulement quelques caractéristiques de chromosomes sont modifiées d'une manière aléatoire, avec un faible taux de probabilité.

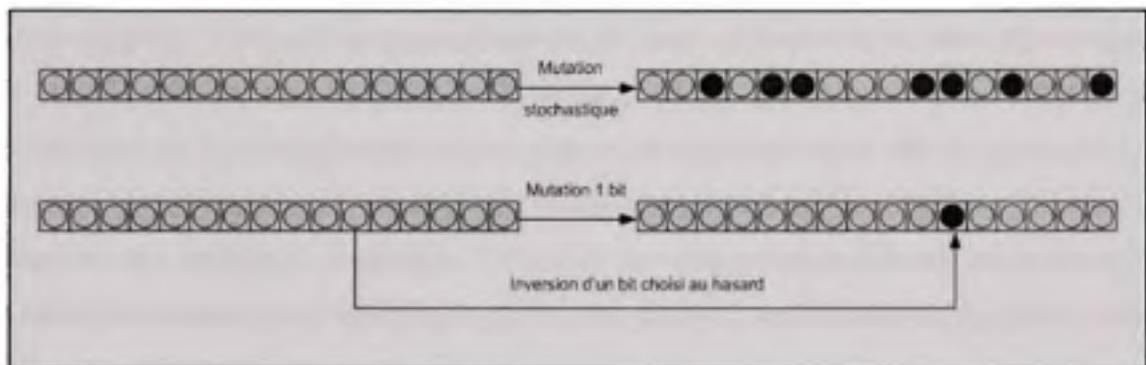
#### 3.5.2.8.1 Mutation d'une population avec le codage binaire

Cette méthode propose une inversion de caractéristiques de chromosomes. En pratique, la méthode consiste à faire une inversion d'un bit choisi au hasard.

### 3.5.2.8.2 Mutation stochastique d'une population

Cette méthode propose l'inversion de plusieurs caractéristiques des chromosomes qui est choisi au hasard, avec un taux de mutation plus faible que dans le cas d'un codage binaire.

La Figure 3.8 nous montre les deux (2) méthodes de mutation traitées.



**Figure 3.8 La mutation d'une population.**

### 3.5.2.9 Élitisme d'une population

L'opérateur d'élitisme permet de sélectionner l'individu le plus puissant d'une génération à la suivante. Malgré le fait qu'on arrive à avoir un individu mieux adapté, il y a la possibilité que l'opérateur de sélection ne le garde pas, parce que la recombinaison entre cet individu et un autre peut donner des individus faibles au niveau de caractéristiques. Pour cette raison l'opérateur d'élitisme est optionnel.

### 3.5.2.10 Réinsertion d'une population

La réinsertion d'une solution a comme objectif d'insérer le nouveau chromosome dans le groupe. Les moyens de faire la réinsertion sont :

- a. éliminer le chromosome le moins adapté;

- b. éliminer le chromosome qui a les caractéristiques semblables au nouveau chromosome;
- c. remplacer un des deux parents.

### 3.6 Algorithmes de fourmis dans l'optimisation combinatoire

Depuis plus de 100 millions d'années, les fourmis forment la plus vieille société basée sur l'interaction sociale. L'étude des fourmis a longtemps été négligée par les entomologistes alors qu'elles représentent un des stades d'évolution les plus aboutis dans le monde des insectes (Hölldobler et Wilson, 1990). Le comportement collectif et l'utilisation de l'environnement comme support de communication fait en sorte que les fourmis ont survécu aussi longtemps. Ce comportement collectif a permis aux fourmis de résoudre des problèmes complexes. L'étude de ce comportement a donné naissance à de nouvelles méthodes de résolution de problèmes, connues sous le nom d'algorithmes des colonies de fourmis.

#### 3.6.1 Chronologie des algorithmes chez les fourmis

- Pierre-Paul Grassé (1959) explique la communication indirecte (stigmergie) dans un environnement émergent auto-organisé, où les individus communiquent entre eux en modifiant leur environnement, chez les termites.
- Manderick et Manson (1988) présentent un article expliquant les méthodes d'auto-organisation chez les fourmis.
- Goss, Glover *et al.* (1989) présentent des travaux sur le comportement collectif chez les fourmis argentines.
- Colomi et Dorigo (1991) proposent l'algorithme de fourmis, *Ant System (AS)* dont la première implémentation est l'algorithme de fourmis.
- Hoos et Stützle (1996) proposent l'algorithme de fourmis max-min, *Max-Min Ant System*.

- Dorigo et Gambardella (1997) proposent une hybridation entre l'algorithme de fourmis et la recherche locale de type 3-Opt nommée *ACS & 3-Opt*.
- Bilchev et Parmee (1997) présente un algorithme nommé algorithme de fourmis continu (*Continuous Ant Colony Algorithm - CACO*). Il s'agit d'une approche basée sur le déplacement régional des fourmis et sur la quantité de phéromone déposée sur le trajet.
- Stützle (1998) propose une approche basée sur l'implémentation parallèle entre l'algorithme de fourmis, l'algorithme de fourmis Max-Min et la méthode 3-Opt.
- Application de l'algorithme de fourmis pour le routage de véhicules (1999).
- Martinoli, Bruno *et al.* (2000) utilisent les algorithmes de colonies de fourmis pour résoudre les problèmes de contrôle des robots.
- Gutjahr (2000) prouve, pour la première fois, la convergence d'un algorithme de colonie de fourmis.
- Iredi (2001) utilise l'algorithme de colonie de fourmis pour résoudre les problèmes multi-objectifs.
- L'algorithme de colonie de fourmis est utilisé aux problèmes biochimiques (2004).

### 3.6.2 Problème du plus court chemin chez les fourmis

Deneubourg, Goss *et al.* (1989) ont constaté que lorsque plusieurs sources de nourriture sont disponibles, les fourmis choisissent toujours la plus proche. Il a pu tirer quelques idées de base :

- a. le déplacement des fourmis isolées se fait au hasard;
- b. les fourmis laissent des traces de phéromone sur leur trajet;
- c. la phéromone se dégrade avec le temps;
- d. les fourmis sont influencées par les pistes de phéromones qui ont été déjà déposées, c'est-à-dire que lorsqu'il aura une bifurcation de chemin, le chemin le plus marqué va être choisi;

- e. la modalité de prendre une décision pendant le déplacement se fait d'une manière probabiliste.

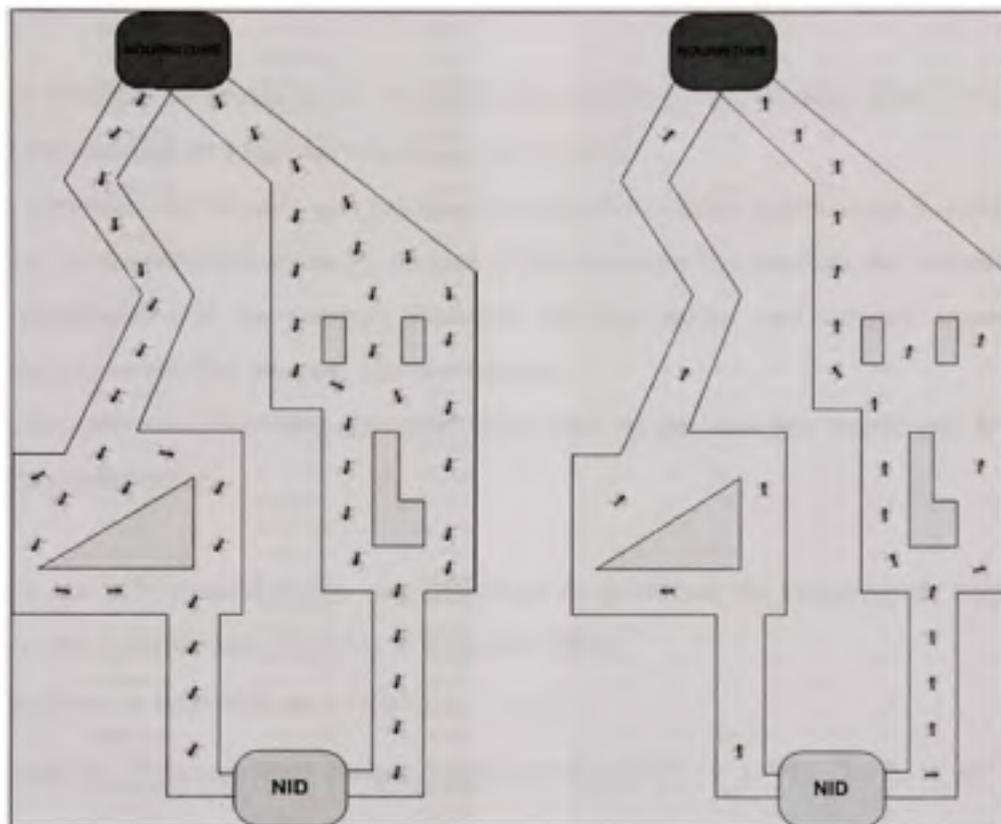


Figure 3.9 Modalité de trouver de la nourriture en prenant le plus court chemin.

### 3.6.3 Algorithme de fourmis (*Ant System*)

L'algorithme proposé par Dorigo et Colomi en 1991 dans l'article « Distributed Optimization by Ant Colonies » est le premier qui vise à résoudre le problème du voyageur de commerce en utilisant le comportement des fourmis. D'une manière générale, on peut définir l'algorithme *Ant System* comme un algorithme d'optimisation stochastique distribué, c'est-à-dire qu'il peut être défini comme une combinaison de calcul distribué, de récompense retardée et d'heuristique gloutonne (Dorigo, 2004) .

### 3.6.4 Modélisation mathématique pour le problème du voyageur de commerce en utilisant l'algorithme de fourmis (*Ant System*)

Afin de résoudre le problème du voyageur de commerce, qui revient donc à trouver le circuit hamiltonien de longueur minimale, on considère :

- un ensemble de fourmis qui parcourent chacune un trajet parmi ceux possibles. On place les fourmis aux sommets du graphe (les sommets représentent des villes);
- la possibilité que les fourmis déposent, sur leur trajet, une certaine quantité de phéromone, qu'elles peuvent détecter ensuite;
- les fourmis possèdent une mémoire, donc elles ne peuvent pas passer par la même ville plusieurs fois.

Dans ce qui suit, nous donnons une définition du problème du voyageur de commerce telle qu'elle apparaît dans Perdalos et Resende (2002).

On considère un ensemble de  $n$  villes  $\{v_1, \dots, v_n\}$ .

On connaît les distances entre chaque couple de villes  $d(v_i, v_j)$ ,  $\forall (i, j) \in \{1, \dots, n\}^2$ .

On doit trouver la permutation  $\varphi$  telle que la séquence  $r = \{v_{\varphi(1)}, \dots, v_{\varphi(n)}\}$  minimise la distance  $D(\varphi)$  parcourue par le voyageur de commerce.

$$\text{Avec } D(\varphi) = \sum_{i=1}^{n-1} d(v_{\varphi(i)}, v_{\varphi(i+1)}) + d(v_{\varphi(n)}, v_{\varphi(1)}) \quad (3.1)$$

Dans l'algorithme de fourmis (*Ant System*), chaque fourmi  $f$  ( $f = 1, \dots, m$ ) parcourt le graphe en construisant un trajet complet. La construction d'un trajet entre une ville  $i$  et une ville  $j$ , dépend des éléments suivants :

- le **regroupement** des villes déjà visitées par les fourmis, qui met en évidence les mouvements de chaque fourmi  $f : L_f^t$ ;

- b. la **visibilité** entre les villes, ce qui signifie que pour éviter les villes situées trop loin et choisir le bon trajet, on utilise l'inverse de la distance entre une ville  $i$  et une ville  $j$  :

$$\psi_{ij} = \frac{1}{d_{ij}};$$

- c. l'**intensité** de la piste, c'est-à-dire la quantité de phéromone qui est déposée sur un trajet qui lie deux villes .

Kube and Bonabeau (1998) expliquent la règle de déplacement en définissant le paramètre aléatoire de transaction proportionnelle comme :

$$p_{ij}^f(t) = \begin{cases} \frac{\lambda_{ij}(t)^\alpha (\psi_{ij})^\beta}{\sum_{k \in L_i^f} (\lambda_{ik}(t))^\alpha (\psi_{ik})^\beta} & \text{si } j \in L_i^f \\ 0 & \text{si } j \notin L_i^f \end{cases} \quad (3.2)$$

En désignant par :

$\alpha$  , comme paramètre contrôlant l'importance relative de l'intensité d'un arrêt.

$\beta$  , comme paramètre contrôlant la visibilité d'un arrêt.

L'**intensité** de la piste  $\lambda_{ij}(t)$  est contrôlée par les deux paramètres  $\alpha$  et  $\beta$  et aussi par le niveau de la visibilité  $\psi_{ij}$  .

Par exemple, pour  $\alpha=0$  on aura :

$$p_{ij}^f(t) = \begin{cases} \frac{\lambda_{ij}(t)^0 (\psi_{ij})^\beta}{\sum_{k \in L_i^f} (\lambda_{ik}(t))^0 (\psi_{ik})^\beta} & \text{si } j \in L_i^f \\ 0 & \text{si } j \notin L_i^f \end{cases}, \quad (3.3)$$

donc dans le calcul, on établit seulement la visibilité de la ville. Cela signifie qu'on peut choisir, à chaque pas, seulement la ville la plus proche.

Par contre si on prend  $\beta=0$ , on aura :

$$p_{ij}^f(t) = \begin{cases} \frac{\lambda_{ij}(t)^\alpha (\psi_{ij})^0}{\sum_{i \in L'_j} (\lambda_{ij}(t))^\alpha (\psi_{ij})^0} & \text{si } j \in L'_i \\ 0 & \text{si } j \notin L'_i \end{cases}, \quad (3.4)$$

ce qui signifie que les pistes avec une quantité élevée de phéromone ont plus de chance d'être empruntées.

La sélection d'un trajet possible va donc être influencée par les deux (2) facteurs  $\alpha$  et  $\beta$ . C'est pour cette raison qu'on doit faire un compromis entre les facteurs pour éviter d'avoir une sélection trop rapide d'un trajet.

Chaque fourmi dépose une quantité de phéromone sur son trajet complet (tour)  $\Delta\lambda_{ij}(t)$ , quantité qui dépend de la qualité de la solution trouvée. On aura donc :

$$\Delta\lambda_{ij}(t) = \begin{cases} \frac{S}{L^f(t)} & \text{si } (i, j) \in T^f(t) \\ 0 & \text{si } (i, j) \notin T^f(t) \end{cases} \quad (3.5)$$

Où :

$S$  est un paramètre fixé.

$L^f(t)$  est la longueur du trajet.

$T^f(t)$  est le trajet choisi par la fourmi  $f$  à l'instant  $t$ .

On complète l'algorithme en choisissant un facteur d'évaporation  $\mu$ , qui représente le taux de disparition de la phéromone. Mathématiquement, cela signifie que l'algorithme évite d'être piégé dans des solutions sous-optimales en acceptant des solutions moins bonnes pour pouvoir sortir d'un éventuel optimum local.

Nous introduisons la formule de mise à jour des pistes :

$$\lambda_y(t+1) = (1-\gamma)\lambda_y(t) + \sum_1^m \Delta\lambda_y^f(t) \quad (3.6)$$

Où :

$m$  est le nombre de fourmis utilisées pour l'itération  $t$ ,

$\gamma$  est un paramètre de réglage.

### 3.7 Conclusion

Tout au long de ce chapitre, nous avons présenté la méthodologie et une revue des algorithmes étudiés. Nous nous sommes ainsi plus particulièrement intéressés aux algorithmes de recuit simulé, génétique et fourmis.

Dans la suite, nous allons nous intéresser à la résolution du problème proposé (problème du voyageur de commerce).

## CHAPITRE 4

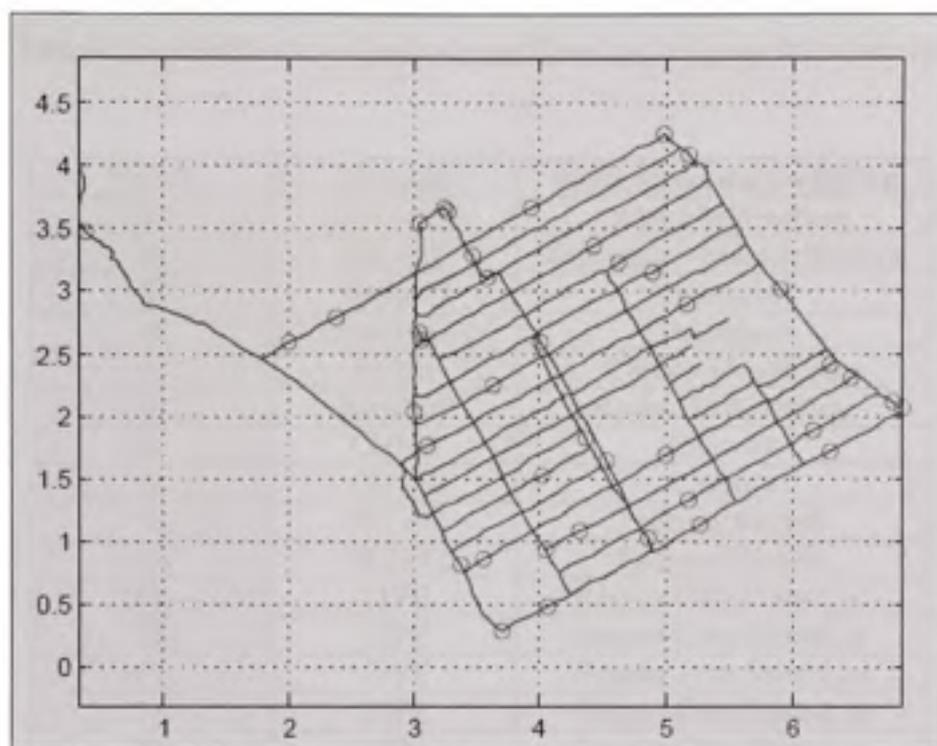
### EXPÉRIMENTATIONS

#### 4.1 Introduction

Avant de connaître les résultats d'un problème il est important de bien connaître le contexte des expérimentations. Ce chapitre décrit les expérimentations sur différentes méthodes de résolutions. On trouve également les améliorations et les instances proposées pour résoudre le problème de voyageur de commerce.

#### 4.2 Instances utilisées pour résoudre le problème du voyageur de commerce

Dans notre étude de cas, nous considérons 30 instances sur un réseau routier de 40 points du secteur Côte Saint-Luc (Montréal). Nous étudierons également 10 autres instances connues dans la littérature du problème de voyageur de commerce allant de 22 points à 2392 points à visiter. Le réseau Côte Saint-Luc a été implémenté à partir d'un réseau des routes MapPoint. Pour refaire ce réseau, nous avons pris environ 7 000 points qui ont été liées pour former le secteur de routes spécifié. Dans le but de concevoir les 30 instances nécessaires pour notre problème, nous avons choisi une quarantaine de points du secteur qu'on a créés. Les 40 points du réseau Côte Saint-Luc (Montréal) ont été sélectionnés d'une manière exacte, c'est-à-dire que les points correspondent aux codes postaux des différents résidents de ce secteur. Les autres 10 instances sélectionnées ont été adoptée d'une manière diversifié, pour nous permettre d'étudier un grand nombre de situations (dans notre cas allant d'un petit nombre à un grand nombre de points) et pour nous permettre d'avoir un vision assez large de l'impact de différentes méta-heuristiques sur le problème de voyageur de commerce. La Figure 4.1 nous montre le réseau routier du secteur Côte Saint-Luc (Montréal) qu'on l'utilise comme base pour le problème du voyageur de commerce.



**Figure 4.1 Réseau routier de 40 points du secteur Côte Saint-Luc (Montréal).**

Le réseau routier du secteur Côte Saint-Luc (Montréal) a été sauvegardé dans des fichiers Excel. Le raison pour laquelle on a choisi ce type de fichier est la facilité de créer d'autres couches d'information (exemple : contraintes de circulation, points de charge, développement urbaine, etc.) sur la même base de données. Cette façon de procéder a permis d'analyser d'autres types de problèmes dérivés du problème du voyageur de commerce (exemple : routage de véhicules) et de faciliter l'intégration avec le langage de programmation Matlab. Le tableau 4.1 présente les instances proposées pour résoudre le problème du voyageur de commerce.

Tableau 4.1

Instances utilisées pour résoudre le problème du voyageur de commerce

Nr.crt.	Instance	Provenance du problème
1.	Ulysses22	Groetschel/Padberg
2.	Bayg29	Groetschel/ Juenger/Reinelt
3.	Berlin52	Groetschel
4.	Thompson70	Thompson
5.	Eil101	Christofides/Eilon
6.	Kroa100	Krolak/Felts/Nelson
7.	CH130	Churriz
8.	CH150	Churriz
9.	PR1002	Padberg/Rinaldi
10.	PR2392	Padberg/Rinaldi
11.	11PE	Réseau Côte Saint-Luc
12.	12PE	Réseau Côte Saint-Luc
13.	13PE	Réseau Côte Saint-Luc
14.	14PE	Réseau Côte Saint-Luc
15.	15PE	Réseau Côte Saint-Luc
16.	16PE	Réseau Côte Saint-Luc
17.	17PE	Réseau Côte Saint-Luc
18.	18PE	Réseau Côte Saint-Luc
19.	19PE	Réseau Côte Saint-Luc
20.	20PE	Réseau Côte Saint-Luc
21.	21PE	Réseau Côte Saint-Luc
22.	22PE	Réseau Côte Saint-Luc
23.	23PE	Réseau Côte Saint-Luc
24.	24PE	Réseau Côte Saint-Luc
25.	25PE	Réseau Côte Saint-Luc
26.	26PE	Réseau Côte Saint-Luc
27.	27PE	Réseau Côte Saint-Luc
28.	28PE	Réseau Côte Saint-Luc
29.	29PE	Réseau Côte Saint-Luc
30.	30PE	Réseau Côte Saint-Luc
31.	31PE	Réseau Côte Saint-Luc
32.	32PE	Réseau Côte Saint-Luc
33.	33PE	Réseau Côte Saint-Luc
34.	34PE	Réseau Côte Saint-Luc
35.	35PE	Réseau Côte Saint-Luc
36.	36PE	Réseau Côte Saint-Luc
37.	37PE	Réseau Côte Saint-Luc

Nr.crt.	Instance	Provenance du problème
38.	38PE	Réseau Côte Saint-Luc
39.	39PE	Réseau Côte Saint-Luc
40.	40PE	Réseau Côte Saint-Luc

### 4.3 Expérimentations pour l'algorithme de recuit simulé

#### 4.3.1 Améliorations sur la modalité de déterminer la température initiale

Pour la plupart de chercheurs, le choix de la température représente un inconnu qui est difficile à gérer. Des nombreuses questions se posent sur la façon de choisir la bonne température de départ, température qui influencera les résultats du recuit simulé.

Pour essayer de répondre à cette question on propose un model d'algorithme qui peut déterminer la température de départ pour le recuit simulé.

#### 4.3.2 Description du programme qui détermine la température initiale

Le model proposé repose sur l'idée originale d'utiliser une analogie avec la théorie des perturbations d'une solution pour générer d'autres solutions dans un espace de recherche. En mécanique quantique, la théorie des perturbations est une méthode d'approximation mathématique utilisée pour décrire un système quantique d'une façon simplifiée (Cohen-Tannoudji, 1997). On commence donc par déterminer une solution initiale et après on applique un hamiltonien pour perturber cette solution (énergie). On associe la solution initiale avec un hamiltonien non perturbé  $H_0$ , hamiltonien qui possède des niveaux d'énergie connus, niveaux qui sont donnés par l'équation de Schrödinger (Cohen-Tannoudji, 1997).

$$\hat{H}|\Psi(t)\rangle = i\hbar \frac{d}{dt}|\Psi(t)\rangle = \frac{\hat{p}^2}{2m}|\Psi(t)\rangle + V(\vec{r},t)|\Psi(t)\rangle \quad (4.1)$$

Où

$\Psi(t)$  représente les probabilités de résultats de toutes les mesures possibles d'un système.

$\hbar$  est la constante de Planck, utilisé pour décrire les niveaux d'énergie.

$i$  est un nombre complexe qui sert à définir des solutions supplémentaires pour certaines équations.

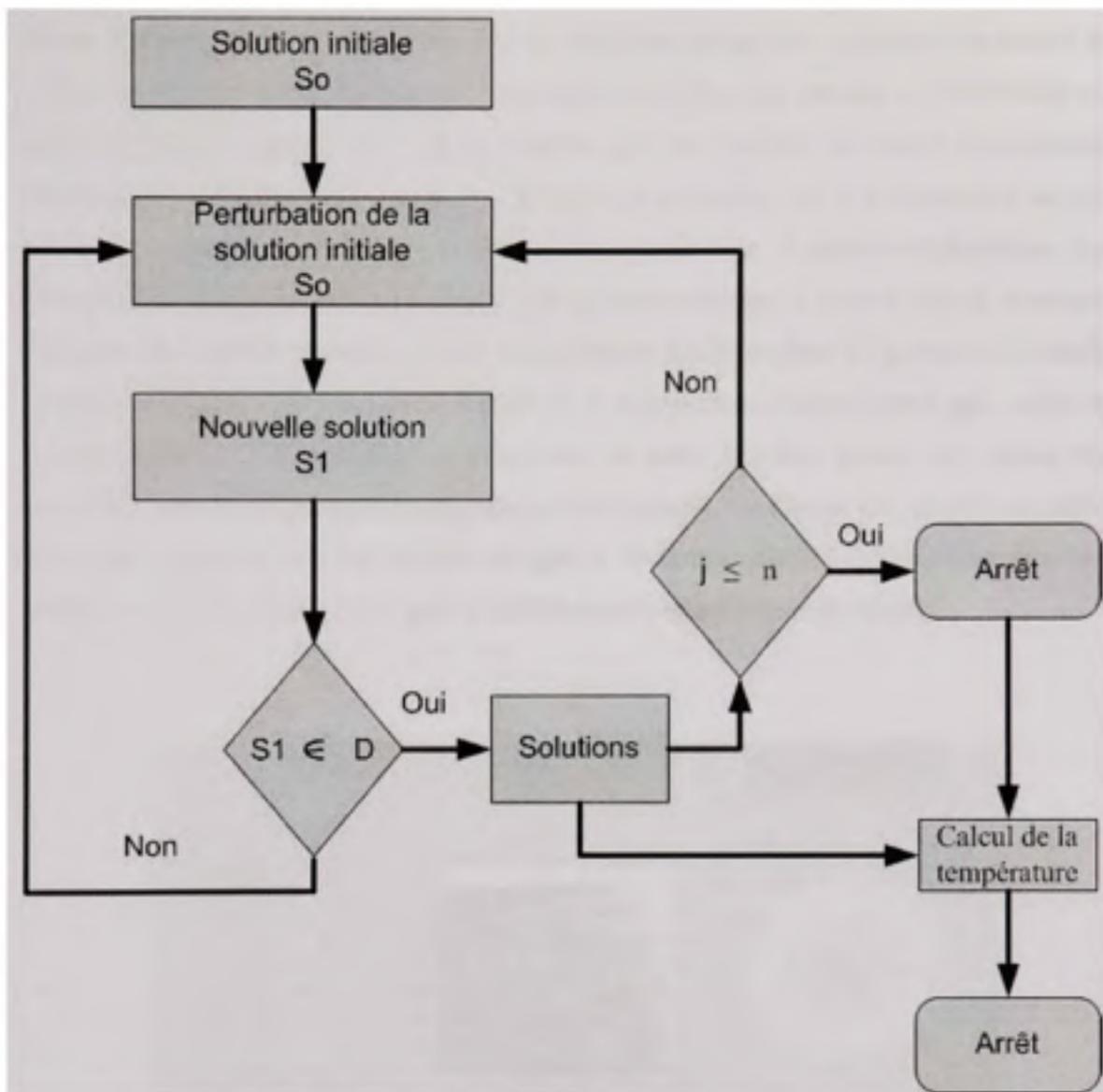
$\hat{H}$  est l'hamiltonien.

$\vec{r}$  sont des vecteurs qui déterminent la position.

$\vec{p}$  sont des vecteurs qui donnent l'impulsion.

On suppose que les énergies sont discrètes et l'équation de Schrödinger est indépendante du temps pour réduire le problème. Étant donné que les énergies sont associée à un système non perturbé, on peut introduire un hamiltonien  $V$  qui représente un petite perturbation. Un paramètre  $\lambda$  pouvant prendre des dimensions entre 0 (pas de perturbation) et 1 (perturbation au maximum) est aussi considérée.

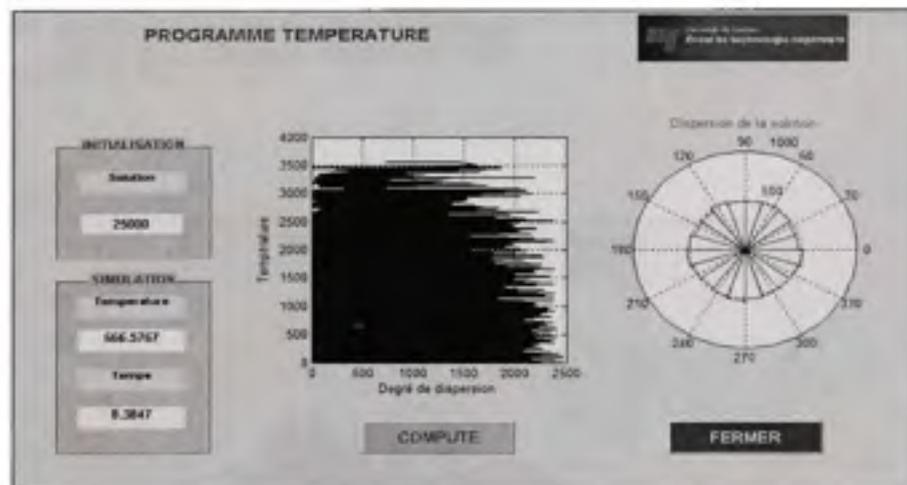
On obtient donc un hamiltonien perturbé  $\hat{H} = H_0 + \lambda * V$ , qui donnera la plage des valeurs perturbées. Dans notre cas, l'hamiltonien non perturbé correspond à la solution initiale et l'hamiltonien perturbé aux solutions perturbées. La solution initiale a été obtenue à l'aide d'un algorithme de recuit simulé de base. Une valeur de base de 100 degrés a été choisie comme point de départ pour la température du recuit. On a choisi aussi une valeur de 0.978 pour le taux de refroidissement, valeur qu'on gardera après. Plusieurs instances allant de 10 points à 2 359 points ont été utilisées pour montrer la viabilité de cette logique. La Figure 4.2 montre l'algorithme utilisé pour déterminer la température initiale pour le recuit simulé.



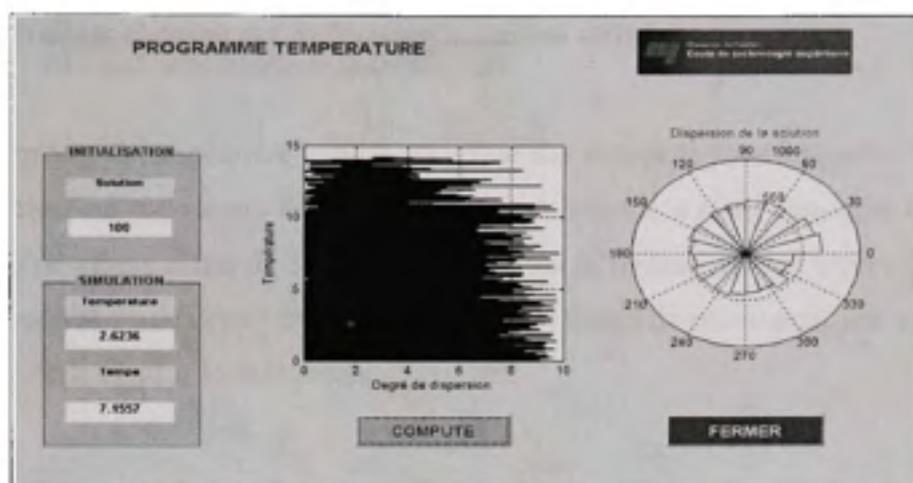
**Figure 4.2** L'algorithme de la méthode de calcul qui détermine la température initiale pour le recuit simulé.

On doit déterminer la température initiale  $T_0$ . Au début des itérations, on est éloigné du minimum; il faut donc choisir une probabilité d'acceptation élevée,  $p = 0.5$  (par exemple). En théorie des probabilités et en statistiques, la médiane  $M$  est un nombre qui divise la population ou la distribution de probabilités. Chaque partie contient le même nombre de valeurs. On génère un certain nombre de vecteurs  $X$  aléatoire et pour chacun

de ces vecteurs on détermine  $F(X)$ . En les ordonnant en un ordre croissant, on choisit la valeur au milieu comme médiane  $M$ , c'est à dire la valeur qui partage la distribution en deux (2) parties égales. S'il y a un nombre pair de valeurs, on prend la moyenne arithmétique entre les deux valeurs qui se trouvent au milieu. S'il y a un nombre impair de valeurs, il suffit de prendre la valeur qui partage en deux (2) parties l'échantillon. En procédant de cette manière, on s'assure que la valeur obtenue se trouve dans le domaine de recherche et qu'on choisit une bonne température de départ pour l'algorithme de recuit simulé. La Figure 4.3, montre la valeur de la température obtenue pour une solution initiale égale à 25 000 unités. La résolution de cette instance trouve une valeur de 666,5767 pour la température de départ du recuit simulé. La Figure 4.4, montre un autre essai pour lequel la solution initiale est égale à 10 unités. Suite à la résolution de cette instance on obtient une valeur égale à 2,6236 pour la température de départ.

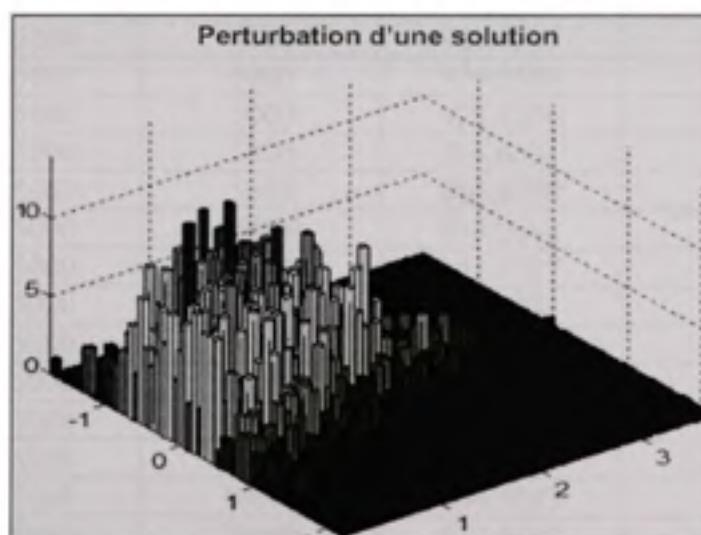


**Figure 4.3** Valeur de la température obtenue pour une solution initiale égale à 25 000 unités.



**Figure 4.4** Valeur de la température obtenue pour une solution initiale de 100 unités.

La Figure 4.5, montre les perturbations d'une solution obtenues à l'aide du programme qui détermine la température de départ pour l'algorithme du recuit simulé. On peut constater que les perturbations d'une solution respectent la distribution gaussienne dans un espace de recherche.



**Figure 4.5** La perturbation d'une solution pour le programme qui détermine la température.

### 4.3.3 Résultats obtenus sur différentes instances utilisées

Pour déterminer la température initiale, on utilise des instances pour lesquelles les valeurs de la solution initiale varient entre 35 000 et 30 unités. On a aussi considéré un nombre aléatoire d'itérations allant de 5 000 à 20 000. Dans le tableau 4.2, on trouve les valeurs pour la température de départ du recuit simulé et les temps de calcul pour une valeur de la solution initiale égale à 35 000 unités.

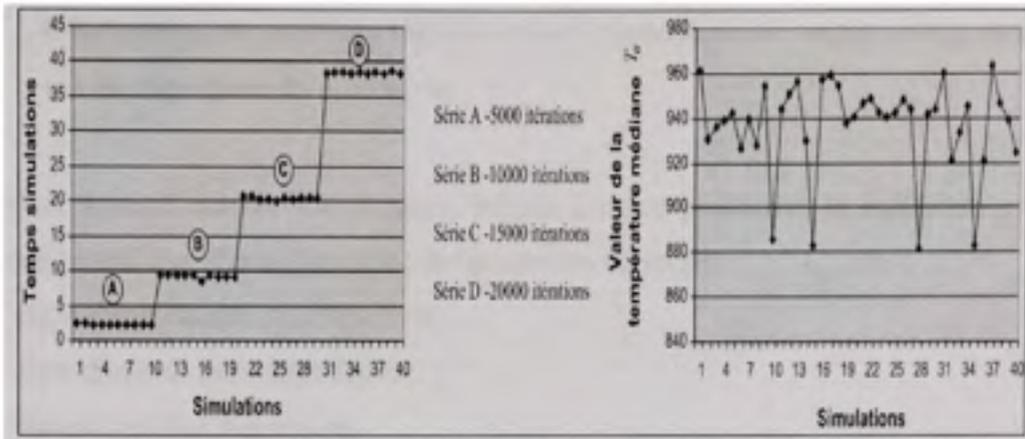
Tableau 4.2

La variation de la température médiane  $T_0$  en fonction de la solution initiale  $s_0$

Essais	Valeur de la solution initiale $s_0$	Nombre aléatoire d'itérations	Valeur de la température médiane $T_0$	Temps (s) de simulation
1	35000	5000	961.1826	2.469
2	35000	5000	931.1370	2.501
3	35000	5000	936.6709	2.223
4	35000	5000	939.4393	2.188
5	35000	5000	942.3018	2.203
6	35000	5000	926.6395	2.203
7	35000	5000	939.5589	2.223
8	35000	5000	928.3356	2.326
9	35000	5000	954.6631	2.228
10	35000	5000	885.6758	2.250
11	35000	10000	944.0662	9.297
12	35000	10000	951.1707	9.297
13	35000	10000	956.2241	9.326
14	35000	10000	930.3395	9.447
15	35000	10000	883.2271	9.336
16	35000	10000	957.0188	8.551
17	35000	10000	958.9595	9.296
18	35000	10000	954.6883	9.188
19	35000	10000	937.5502	9.229
20	35000	10000	940.2018	9.222
21	35000	15000	946.9109	20.782
22	35000	15000	948.5547	20.669
23	35000	15000	942.3374	20.335

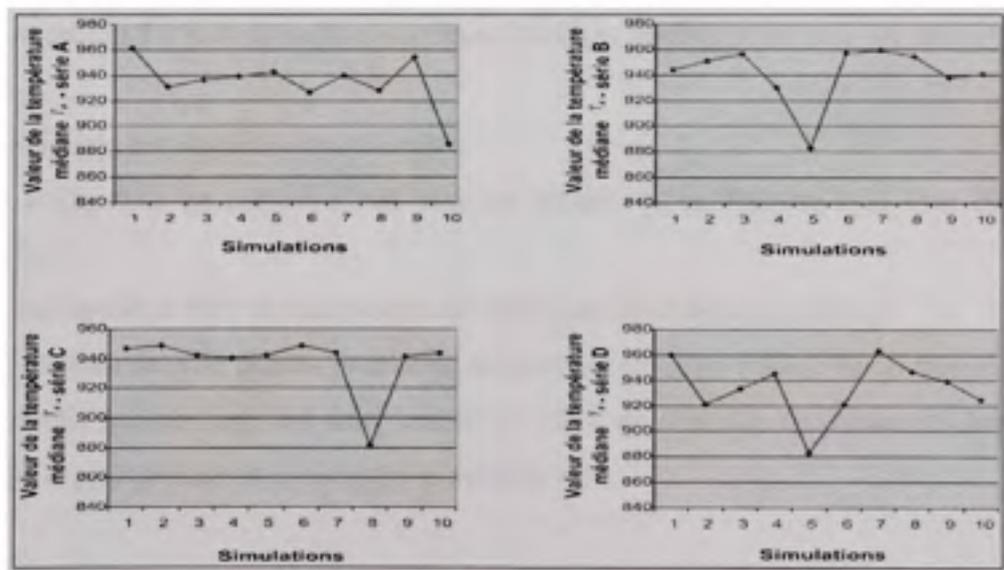
Essais	Valeur de la solution initiale $s_0$	Nombre aléatoire d'itérations	Valeur de la température médiane $T_0$	Temps (s) de simulation
24	35000	15000	940.6621	20.335
25	35000	15000	942.4474	20.005
26	35000	15000	948.4465	20.557
27	35000	15000	944.2336	20.221
28	35000	15000	881.3362	20.599
29	35000	15000	941.6623	20.557
30	35000	15000	944.1101	20.552
31	35000	20000	959.8067	38.359
32	35000	20000	921.2656	38.449
33	35000	20000	933.4121	38.448
34	35000	20000	945.3398	38.227
35	35000	20000	883.0544	38.556
36	35000	20000	921.2656	38.344
37	35000	20000	963.2144	38.500
38	35000	20000	946.7702	38.221
39	35000	20000	939.3356	38.662
40	35000	20000	924.4478	38.338

On observe que pour une valeur de la solution initiale égale à 35 000 unités et pour un nombre aléatoire d'itérations compris entre 5 000 et 20 000, les valeurs de la température médiane varient entre 881,3362 et 885,6758. Par contre, le temps de simulation varie dans un intervalle plus grand, soit entre 2,203 et 38,662 secondes. Cette variation s'explique par le nombre croissant d'itérations utilisées. La Figure 4.6 montre la valeur de la température médiane et le temps de simulation pour des différentes séries d'itérations versus le nombre de simulations.



**Figure 4.6** Temps de simulations et les valeurs de la température  $T_0$  médiane pour les séries A, B, C et D.

La Figure 4.7 présente les valeurs de la température médiane reliées aux séries A, B, C, D pour un nombre de 10 simulations.



**Figure 4.7** Valeurs de la température médiane  $T_0$  pour les séries A, B, C et D.

À l'Annexe I, on trouve les autres résultats concernant le choix de la température pour le recuit simulé.

#### 4.3.4 Génération des solutions initiales pour l'algorithme de recuit simulé dans le cas du voyageur de commerce

En général, pour trouver une solution initiale à notre problème, la littérature propose plusieurs méthodes. Parmi les méthodes proposées on retient :

- a. l'algorithme du plus proche voisin;
- b. l'algorithme du plus petit détour;
- c. l'algorithme de Christofides.

L'algorithme du plus proche voisin a été choisi comme méthode de départ pour déterminer la solution initiale à notre problème. On parle d'un algorithme simple et rapide, mais il n'est efficace que pour un nombre réduit de villes, selon Dorigo (2003). Initialement on l'utilise pour déterminer un trajet acceptable et après on se sert de lui comme base pour un autre algorithme. En augmentant le nombre de villes on obtient une solution qui est 50 % de la solution optimale (Dorigo, 2003).

Le principe de cet algorithme est simple :

- a. on commence en partant d'une ville au hasard, qu'on l'ajoute à la liste de villes visitées;
- b. on recherche la ville la plus proche qu'on l'ajoute aussi dans cette liste;
- c. on cherche la ville la plus proche de la nouvelle solution (ville). Si on trouve que la nouvelle solution est déjà dans la liste de villes, on continue en prenant la deuxième ville la plus proche ou la suivante et ainsi de suite.

La Figure 4.8, présente l'application de l'algorithme de plus proche voisin comme base pour l'algorithme de Metropolis.

Dans ce mémoire, on utilise l'algorithme des plus proches voisins comme base pour l'algorithme de recuit simulé, l'algorithme génétique et l'algorithme de fourmi avec les méthodes d'améliorations 2-opt, 3-opt et 4-opt.



**Figure 4.8** La méthode des plus proches voisins appliquée au l'algorithme de Metropolis.

#### **4.3.5 Procédures d'amélioration des arcs pour l'algorithme de recuit simulé dans le cas du voyageur de commerce**

La procédure d'amélioration des arcs est utilisée pour essayer d'améliorer la possibilité de trouver une bonne solution à notre problème. Dans une première étape, on part d'une solution initiale  $S$  et on applique une transformation locale pour déterminer l'ensemble des voisins  $K(S)$  de cette solution. Une telle transformation peut être un échange de sous chemins, l'inversion d'un élément, *etc.* Plusieurs études montrent que les méthodes qui utilisent les échanges de type  $k$ -opt sont la base des heuristiques les plus performantes pour le voyageur de commerce. En général, on utilise une transformation de type 2 ou 3 opt, mais on peut aussi appliquer un voisinage global, qu'on l'appelle *Meil-Opt*, qui va évaluer les voisinages 2-opt et 3-opt et va utiliser la transformation qui va apporter l'amélioration la plus grande. Dans les Figures 4.9, 4.10 et 4.11, on montre les procédures 2-opt, 3-opt et 4-opt d'amélioration des arcs pour un réseau de 9 points, réseau qui a été choisi d'une manière aléatoire.

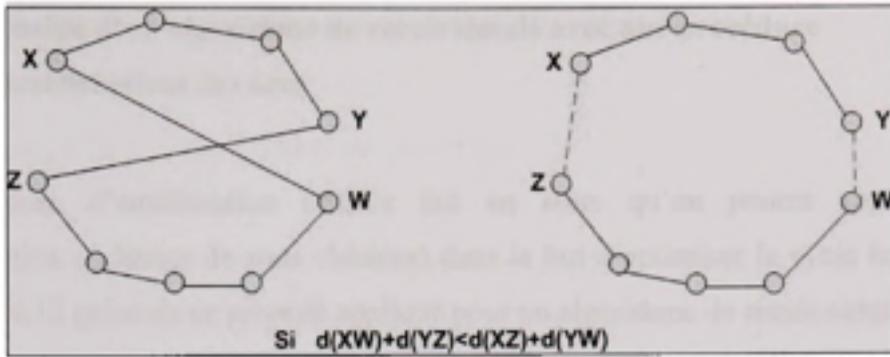


Figure 4.9 Méthode 2-opt.

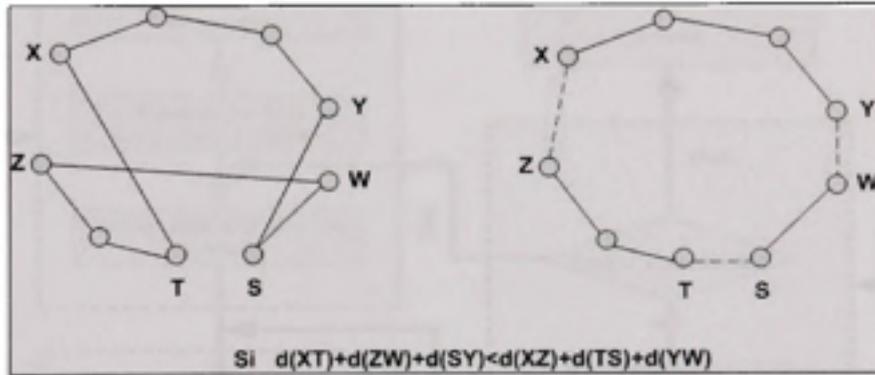


Figure 4.10 Méthode 3-opt.

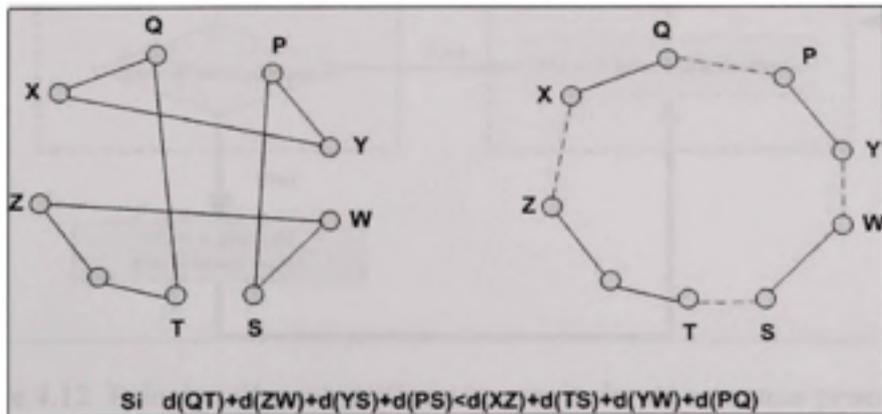


Figure 4.11 Méthode 4-opt.

#### 4.3.6 Principe d'un algorithme de recuit simulé avec une procédure d'améliorations des arcs

La procédure d'amélioration utilisée fait en sorte qu'on pourra appliquer une transformation (échange de sous chemins) dans le but d'optimiser le cycle hamiltonien. La Figure 4.12 présente ce procédé appliqué pour un algorithme de recuit simulé.

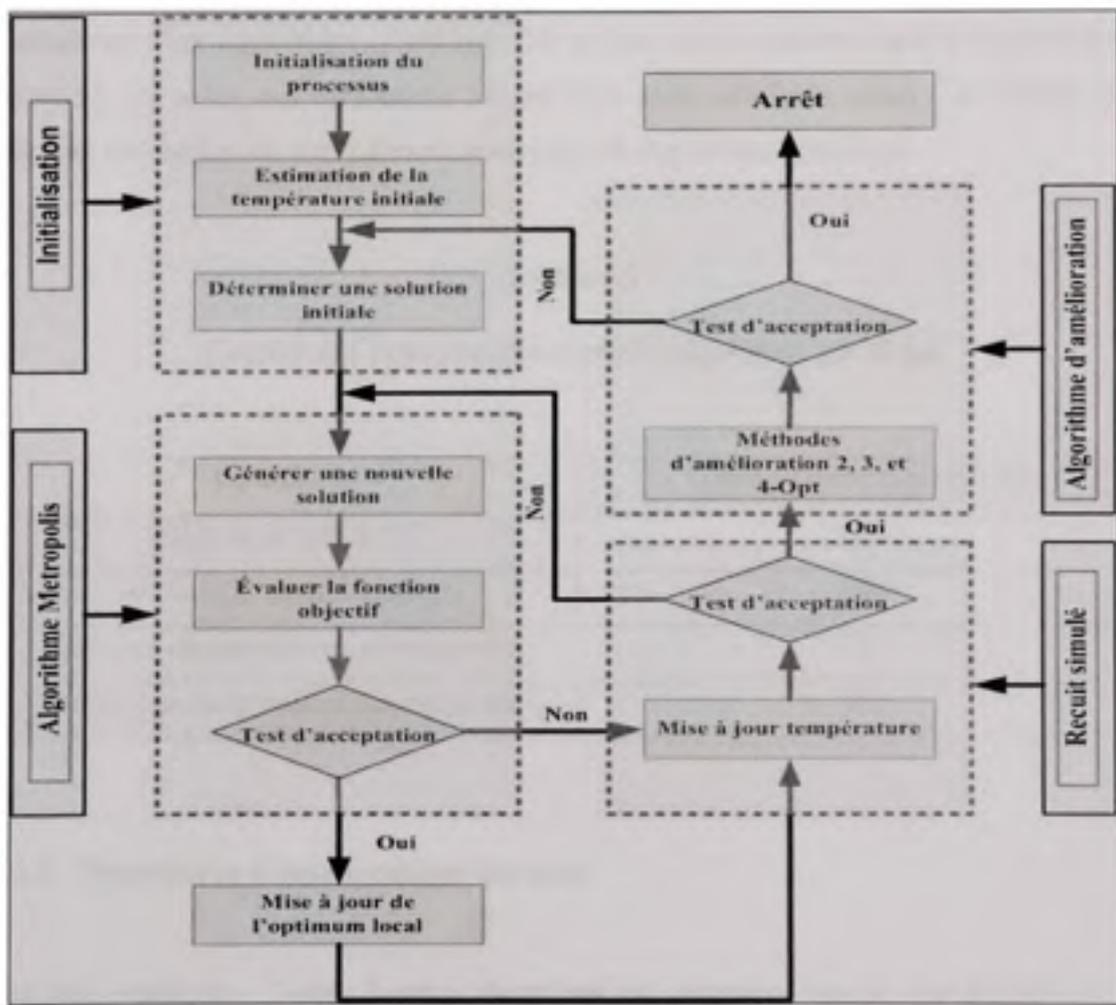


Figure 4.12 Principe d'un algorithme de recuit simulé avec une procédure d'amélioration.

## 4.4 Expérimentations pour l'algorithme génétique

### 4.4.1 Paramétrage d'un algorithme génétique

En un premier temps, il s'agit de choisir les différents paramètres de l'algorithme génétique. C'est une tâche difficile car la plupart des paramètres sont déterminés d'une façon aléatoire. Par exemple, choisir la taille d'une population initiale s'avère assez complexe. On doit faire plusieurs essais pour arriver à un bon résultat. Malheureusement, dans la littérature on ne propose pas efficacement la façon de choisir les différents paramètres d'un algorithme génétique. En ce qui nous concerne dans le cadre de ce mémoire, les choix des paramètres ont été faits après plusieurs essais. Le Tableau 4.3 présente les conditions d'expérimentation pour un algorithme génétique.

Tableau 4.3

Condition d'expérimentation pour l'algorithme génétique

Paramètres	Valeur d'expérimentations
Taille de la population	100
Nombre de recombinaisons	100
Nombre de paramètres sélectionnés	3
Écart-type de la mutation gaussienne	0.1

### 4.4.2 Procédures d'améliorations des arcs

Les trois méthodes 2-opt, 3-opt et 4-opt ont été adoptées dans le but d'améliorer la solution finale. Les méthodes ont été expliquées dans le chapitre précédent.

#### 4.4.3 Usage d'un algorithme génétique avec une procédure d'amélioration pour résoudre le problème du voyageur de commerce

Le principe de résolution du problème du voyageur de commerce en utilisant un algorithme génétique avec une amélioration 2-opt, 3-opt et 4-opt est présenté à la Figure 4.13.

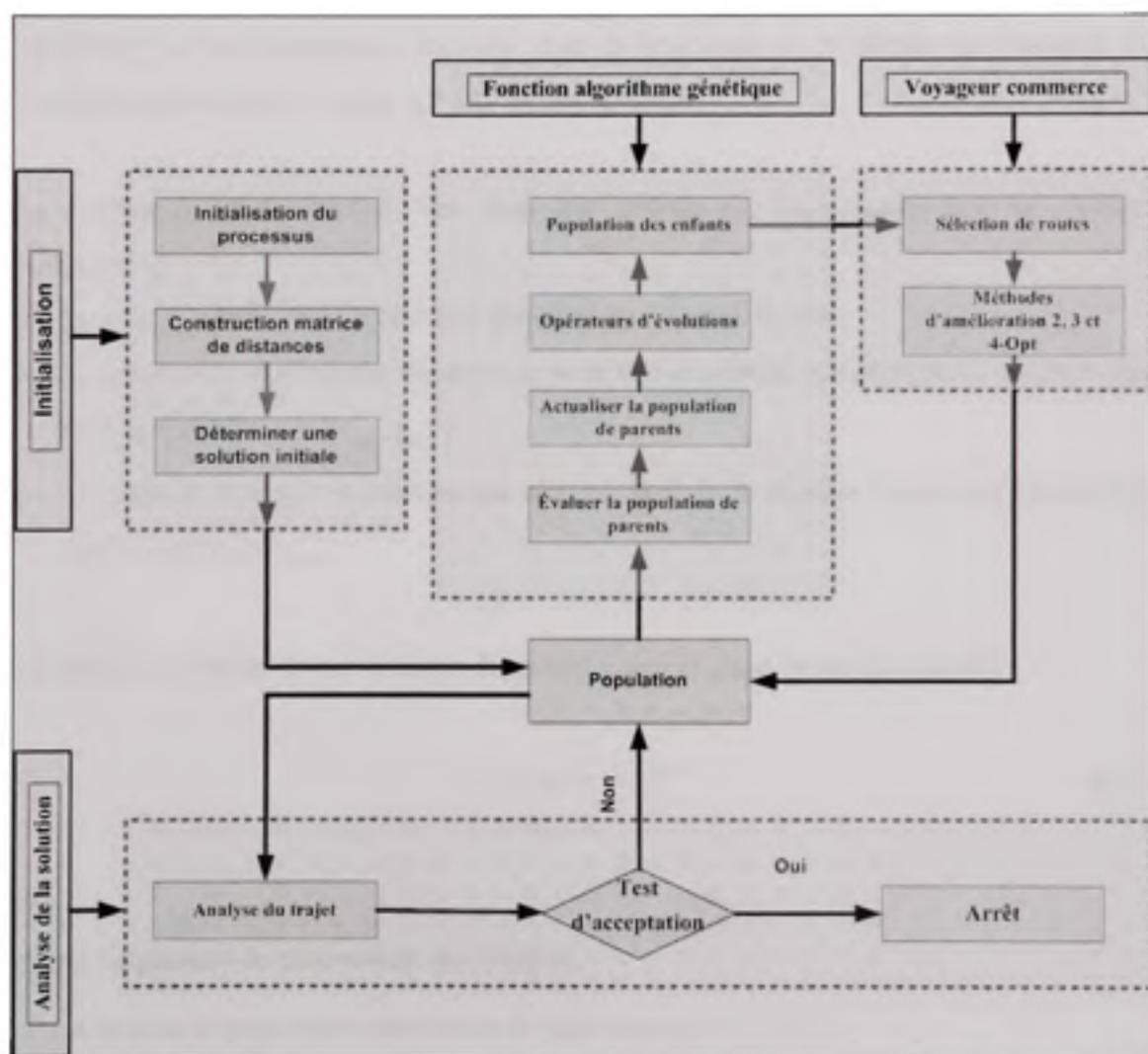


Figure 4.13 Principe d'un algorithme génétique avec une amélioration 2-opt, 3-opt et 4-opt pour résoudre le problème du voyageur de commerce.

## 4.5 Expérimentations pour l'algorithme de fourmis

### 4.5.1 Résolution des différentes instances avec l'algorithme de fourmis Max Min

Parmi les algorithmes de colonies de fourmis, l'algorithme de fourmis Max Min (*Max Min Ant System*) avec 2-opt, 3-opt et 4-opt a été utilisé pour résoudre les différentes instances qu'on a choisies. Dans notre étude de cas, nous considérons 15 instances sur un réseau routier réel de 40 points du secteur Côte Saint-Luc (Montréal). Nous étudierons également 10 autres instances connues dans la littérature du problème du voyageur de commerce allant de 22 points à 2 382 points à visiter.

La technique d'optimisation *Max Min Ant System* est caractérisée par les critères suivants :

- la mise à jour de phéromone sera attribuée à une seule fourmi;
- la quantité de phéromone déposée au bord d'une solution sera limitée à un intervalle d'encadrement noté  $[\tau_{\min}, \tau_{\max}]$ ;
- les phéromones qui se trouvent sur chaque bord de la solution seront initialisées à la valeur maximale  $\tau_{\max}$ .

La formule suivante donne la façon de mettre à jour la quantité de phéromone :

$$\tau_{ij} = \rho * \tau_{ij} + \Delta_{ij}^{best} \quad (5.5)$$

Où :

$\tau_{ij}$  est la quantité de phéromone sur l'arc  $ij$ .

$\rho$  est le taux de persistance des traces de phéromone ( $0 \leq \rho \leq 1$ ).

$\Delta_{ij}^{best}$  est la plus grande quantité de phéromone déposée sur l'arc  $ij$ .

#### 4.5.2 Paramétrage d'un algorithme de fourmis Max Min

Le choix de la valeur des paramètres d'un algorithme de fourmis s'avère très difficile car les valeurs des éléments sont fixées de façon expérimentale. Afin d'obtenir les meilleures valeurs en moyenne, nous avons effectuée plusieurs essais sur les instances retenues. Le tableau 4.4 présente les conditions d'expérimentation pour l'algorithme de fourmis Max Min.

Tableau 4.4

Condition d'expérimentation pour l'algorithme de fourmis Max Min

Conditions d'expérimentations	Valeur d'expérimentations
$\beta$ - pois de la visibilité locale	10
$\rho$ -taux de persistance des traces de phéromone	0.1
$f$ - nombre de fourmis	50

La valeur pour le poids de la visibilité plus petite que 10, fait en sorte que les fourmis trouvent des chemins hamiltoniens de mauvaise qualité de façon à limiter la possibilité de trouver rapidement une solution qui se rapproche de la solution optimale. Une valeur plus grande influence la sensibilité aux traces de phéromones, donc les fourmis ne sont pas capables d'exploiter les expériences passées.

Le taux de persistance des traces de phéromones a été choisi de telle sorte que si aucune fourmi ne passe pas sur un de ces arcs, la quantité de phéromone devient imperceptible et l'arc est abandonné.

Le choix pour déterminer le nombre de fourmis qu'on utilise est aussi important. En choisissant un nombre plus petit de fourmis, on pénalise la solution finale. Donc, pour les problèmes difficiles, l'algorithme converge vers des solutions non optimales.

Par contre, le choix d'un nombre plus grand de fourmis influencera la résolution collective du problème en donnant des résultats qui sont positionnés assez loin de la solution optimale.

#### **4.5.3 Génération des solutions initiales pour l'algorithme de fourmis Max Min**

En général, les solutions initiales d'un algorithme *Max Min Ant System* sont trouvées d'une façon aléatoire. En effet, on génère aléatoirement des solutions qui se trouvent dans l'espace de recherche. Les solutions obtenues de cette manière sont en général d'une qualité médiocre. Au fil des années, plusieurs heuristiques ont été utilisées pour trouver de bonnes solutions initiales. Nous pouvons citer à titre d'exemple la méthode Augment-Merge (Wang, Cui *et al.* 2006), la méthode Path-Scanning (Golden, DeArmon *et al.* 1983) et la méthode d'Ulusoy (Ulusoy 1985). Le programme *Max Min Ant System* qu'on a conçu utilise la méthode du plus proche voisin comme heuristique pour trouver la solution initiale. Cette méthode a comme avantage qu'on démarre d'une solution de bonne qualité. Elle a aussi l'avantage d'être parmi les plus simples à utiliser.

#### **4.5.4 État de convergence de l'algorithme Max Min**

L'algorithme de type *Max Min Ant System* est considéré parmi les algorithmes les plus performants de type *Ant Colony Optimisation (ACO)*. Les solutions produites par cet algorithme sont considérées parmi les meilleures, en terme de qualité (Blum, 2005). La convergence de cet algorithme a été prouvée par Gutjahr (2002), mais le problème reste ouvert en ce qui concerne les autres types d'ACO. Les formules mathématiques qui déterminent les bornes inférieure et supérieure d'un algorithme *Max Min Ant System* ont été tirées de l'ouvrage fait par Gutjahr (2002).

On a donc :

- a. La borne inférieure  $\tau_{\min}$  qui est donnée par la formule :

$$\tau_{\min} = \frac{\tau_{\max} (1 - \sqrt[n]{P_{best}})}{(avg - 1) \sqrt[n]{P_{best}}} ; \quad (5.6)$$

et

- b. La borne supérieure  $\tau_{\max}$  qu'on trouve d'une façon dynamique en appliquant la formule suivante :

$$\frac{1}{1 - \rho} * \frac{1}{f(S_{bs})} \quad (5.7)$$

Où:

$\rho$  est un paramètre de réglage.

#### 4.5.5 Calcul de bornes supérieures pour les instances du réseau de Côte Saint-Luc

L'une de méthodes pour déterminer la qualité d'une solution d'une méta-heuristique est le rapport entre la valeur trouvée et la valeur optimale. Étant donné que très souvent la valeur optimale est inconnue, on doit accepter d'utiliser une borne inférieure. Cette valeur est obtenue à partir d'une relaxation du problème initial. Dans notre cas, on ne connaît pas la valeur des solutions optimales pour les instances du réseau de Côte Saint-Luc. Pour résoudre le problème, des bornes supérieures ont été calculées à partir de la formule (5.7). Le Tableau 4.5 présente ces valeurs pour les bornes supérieures.

Tableau 4.5

Calcul des bornes supérieures pour les instances du réseau de Côte Saint-Luc

Nr.crt.	Instance	Provenance du problème	Borne supérieure
1.	11PE	Réseau Côte Saint-Luc	14.6408
2.	15PE	Réseau Côte Saint-Luc	14.9728
3.	18PE	Réseau Côte Saint-Luc	16.8226
4.	25PE	Réseau Côte Saint-Luc	18.8802
5.	30PE	Réseau Côte Saint-Luc	20.0043
6.	31PE	Réseau Côte Saint-Luc	20.0621
7.	32PE	Réseau Côte Saint-Luc	20.4460
8.	33PE	Réseau Côte Saint-Luc	20.4611
9.	34PE	Réseau Côte Saint-Luc	21.2005
10.	35PE	Réseau Côte Saint-Luc	21.8047
11.	36PE	Réseau Côte Saint-Luc	22.3381
12.	37PE	Réseau Côte Saint-Luc	22.3970
13.	38PE	Réseau Côte Saint-Luc	22.5196
14.	39PE	Réseau Côte Saint-Luc	22.8422
15.	40PE	Réseau Côte Saint-Luc	23.2203

#### 4.5.6 Procédures d'amélioration des arcs

Les trois méthodes 2-opt, 3-opt et 4-opt ont été adoptées dans le but d'améliorer la solution finale. Les méthodes ont été expliquées dans le chapitre 3.

#### 4.5.7 Usage d'un algorithme de fourmis Max Min avec une procédure d'amélioration pour résoudre le problème du voyageur de commerce

La Figure 4.14 présente le principe d'un algorithme de fourmi Max Min avec une amélioration 2-opt, 3-opt ou 4-opt.

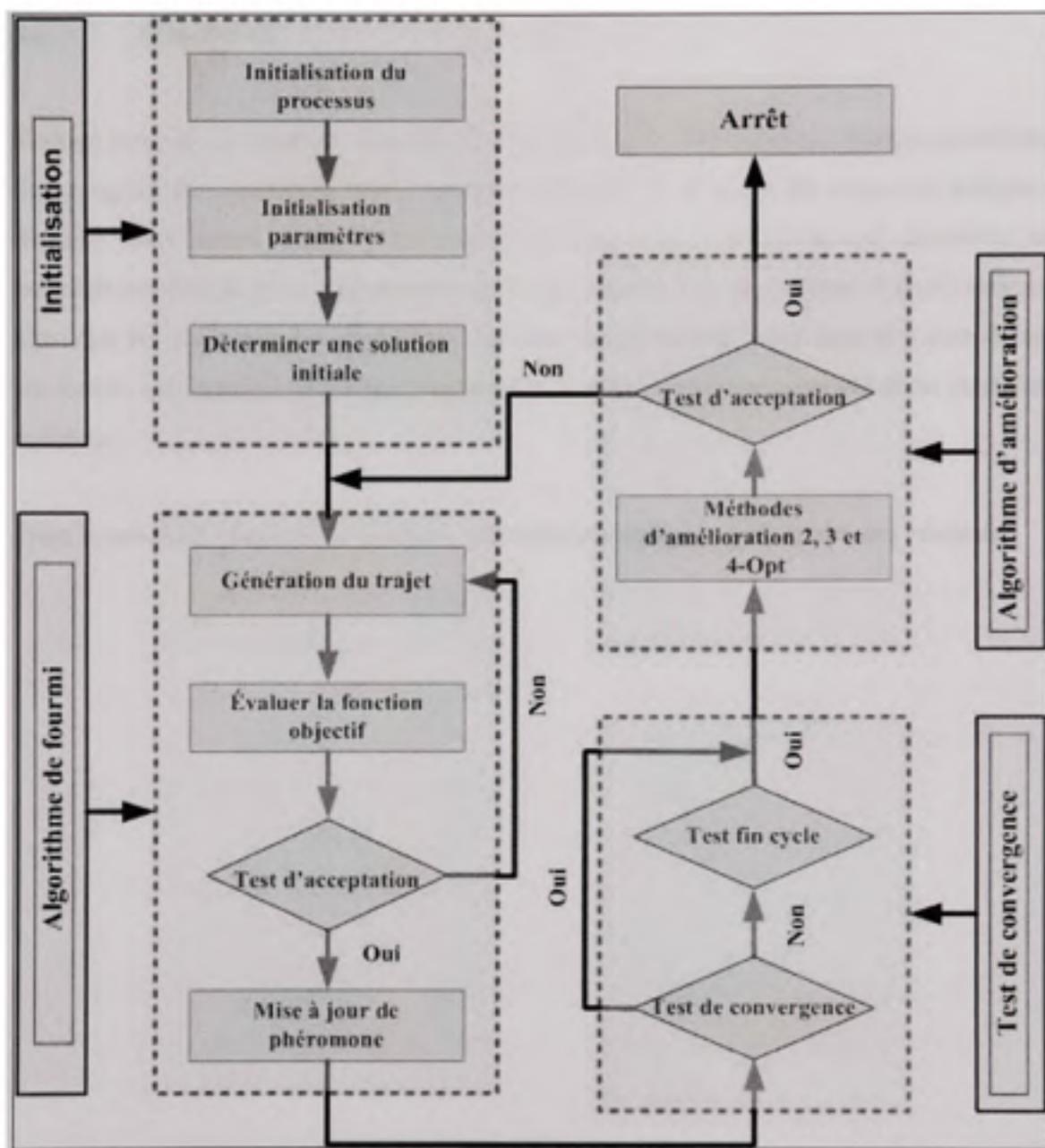


Figure 4.14 Principe d'un algorithme de fourmi Max Min avec une amélioration 2, 3 ou 4-Opt

## **4.6 Conclusion**

Tout au long de ce chapitre, nous avons présenté les expérimentations liées au problème du voyageur de commerce. Dans un premier temps on a défini les instances utilisées. Ensuite, nous avons présenté les expérimentations sur la méthode qui détermine la température initiale pour l'algorithme du recuit simulé. Les procédures d'améliorations ainsi que les principes des algorithmes utilisés (recuit simulé, génétique et fourmis) ont été traités. Les conditions d'expérimentations pour les algorithmes ont été aussi mises en évidence.

Dans le prochain chapitre, nous allons présenter une analyse approfondie des résultats.

## CHAPITRE 5

### RÉSULTATS DES EXPÉRIMENTATIONS

#### 5.1 Introduction

Dans le chapitre précédent, plusieurs algorithmes de résolution et d'amélioration de la solution du problème du voyageur de commerce ont été présentés. Afin d'évaluer ces algorithmes, nous les avons testés sur différentes instances. La totalité de ces expérimentations est présentée dans ce chapitre.

Premièrement, les conditions dans lesquelles nous avons mené les expérimentations sont indiquées. Les caractéristiques des instances de problème utilisées sont notamment précisées. Ensuite, les résultats obtenus avec les méthodes de résolution et d'amélioration (recuit simulé, algorithme génétique et algorithme de fourmis avec les méthodes d'amélioration 2-opt, 3-opt et 4-opt) sont rapportés et analysés.

#### 5.2 Conditions d'expérimentation

Les programmes que nous avons développés à partir des algorithmes du chapitre 4 (recuit simulé, algorithme génétique et algorithme de fourmi avec les méthodes d'amélioration 2-opt, 3-opt et 4-opt) ont été programmés en langage Matlab.

Toutes les expérimentations rapportées dans ce chapitre ont été réalisées avec un ordinateur possédant les caractéristiques suivantes :

- a. CPU : Intel Pentium IV à 660 MHz;
- b. Mémoire vive (RAM) : 1 Gb;
- c. Système d'exploitation : Microsoft Windows XP.

### **5.3 Présentation des données**

Nous avons basé notre évaluation sur des instances que nous avons générées et aussi sur des instances qu'on a choisies dans la littérature. Les instances utilisées ont été mentionnées au chapitre 4. Les algorithmes utilisés et le choix des paramètres ont été spécifiés aux chapitres 3 et 4.

#### **5.3.1 Justification de l'utilisation du critère d'arrêt des algorithmes utilisés**

Utiliser le nombre d'itérations, comme critère d'arrêt pour les algorithmes utilisés, est une pratique qui n'est pas courante dans la littérature quand on parle d'algorithme de recuit simulé, d'algorithme génétique ou d'algorithme de fourmis. Pour illustrer le comportement de ces algorithmes, on a choisi de faire 20 simulations pour un nombre de 2 000 itérations. Ce type de limitation peut influencer d'une certaine manière les résultats obtenus. Malgré ça, nous avons décidé d'utiliser ce type de critère d'arrêt pour les raisons suivantes :

- a. étant donné qu'on utilise les méthodes d'améliorations 2, 3,4-opt, chaque itération est influencée d'une certaine manière, c'est-à-dire que à chaque itération effectuée en simulation lui correspond une valeur améliorée de la solution obtenue;
- b. les expérimentations effectuées sur différentes instances ont démontré que pour toutes les algorithmes utilisées avec une amélioration type 2, 3,4-opt les 2000 itérations sont suffisantes pour obtenir de bons résultats;

### **5.4 Résultats expérimentaux et discussion**

Afin de présenter le comportement de ces algorithmes, nous exposons l'instance qu'on a choisi d'analyser en détails dans ce chapitre. L'instance KROA100 a été sélectionnée en raison de la grandeur intéressante (100 points) du problème.

À l'Annexe II, on trouve également l'analyse complète de deux (2) autres instances (40PE et PR2392) situées aux extrémités des tailles de problèmes traités.

### 5.4.1 Résultats obtenus pour l'algorithme de recuit simulé avec une amélioration de type 2-opt, 3-opt et 4-opt

Dans cette section, nous analysons les simulations sur le problème KROA100 en utilisant un algorithme de recuit simulé avec une procédure d'amélioration de type 2-opt, 3-opt et 4-opt.

Le Tableau 5.1 présente les résultats de cette méta-heuristique à chaque simulation pour un choix de température égale à 100 unités (problème KROA100 - voir Chapitre 4). Le taux de refroidissement a été fixé à 0.978 (après plusieurs essais) pour les deux (2) situations (température égale à 100 unités et 801 unités). La Figure 5.2 montre le comportement des algorithmes mentionnés pour une température donnée égale à 100 unités.

#### 5.4.1.1 Discussions

Dans ce contexte (voir sous-section 5.4.1), les trois (3) méthodes donnent des résultats qui sont situés assez loin de la valeur optimale. Le Tableau 5.2 montre les écarts-types entre les différentes méthodes d'optimisation et la valeur optimale du problème. Nous expliquons ces résultats par le fait que dans ce cas, l'algorithme de recuit simulé converge très vite vers une solution sous-optimale. On constate que cette solution ne s'améliore pas sous l'effet d'une procédure type  $k$ -opt ( $k$  égale 2,3 et 4). En effet, lorsque l'on applique une de ces procédures, la solution trouvée n'est pas assez bonne et reste piégée dans un minimum local situé loin de solution optimale (entre 53 % et 58 % - voir le tableau 5.2). La Figure 5.1 et 5.2 montrent le résultat obtenu pour le problème de voyageur de commerce KROA100 lorsque le choix de la température est fixé à 100 unités. D'après les résultats obtenus (tableau 5.1 et 5.2), on peut conclure que soit la température de recuit simulé n'était pas bien adaptée, soit les autres paramètres n'étaient pas bien choisis. Pour répondre à ces défis, on a décidé de trouver la bonne température de départ tout en gardent le même taux de refroidissement (0.978) et le même nombre d'itérations (2 000). Ce défi (trouver la bonne température pour le recuit simulé) a été

relevé grâce au programme présenté dans la sous-section 4.3.2. Suite à cette application la température de départ a été fixée à 801 unités. Les résultats obtenus pour les trois (3) algorithmes (température fixée à 801 unités) sont rassemblés au Tableau 5.3. On trouve également au tableau 5-4 les écarts-types obtenus pour les données du Tableau 5.3. On peut conclure, en terme d'efficacité, que pour un problème donné (problème du voyageur de commerce KROA100) l'algorithme de recuit simulé avec une procédure d'amélioration 3-opt est plus efficace que les autres deux (2) algorithmes. Même si l'amélioration apportée n'est pas assez grande (en valeur moyenne, entre 0.2 et 0.4 %) la procédure 3-opt, appliquée au méta-heuristique de recuit simulé, reste le meilleur choix pour ce problème (KROA100). La Figure 5.3 donne la synthèse du comportement des algorithmes mentionnés pour une température donnée égale à 801 unités. En ce qui concerne le résultat obtenu pour le problème du voyageur de commerce KROA100, quand le choix de la température a été fixé à 801 unités, il est donné à la Figure 5.4.

Tableau 5.1

Résultats obtenus pour l'algorithme de recuit simulé avec une amélioration 2-opt, 3-opt et 4-opt (choix de température égale à 100 unités)

Nombre simulations	Instance	RS 2-opt	RS 3-opt	RS 4-opt	Optimal	T	TR	Itérations
1	KROA100	33285.6428	33067.7737	32729.9335	21282	100	0.978	2000
2	KROA100	31884.8832	33772.7261	33084.6620	21282	100	0.978	2000
3	KROA100	33302.7732	32886.4001	32628.2217	21282	100	0.978	2000
4	KROA100	32991.9972	34852.6629	33302.7732	21282	100	0.978	2000
5	KROA100	33052.5104	31921.0020	35442.6629	21282	100	0.978	2000
6	KROA100	34903.6628	36926.7729	30639.4426	21282	100	0.978	2000
7	KROA100	33618.4002	32810.3320	37993.3290	21282	100	0.978	2000
8	KROA100	31922.0017	35662.0160	32991.9972	21282	100	0.978	2000
9	KROA100	30887.9926	31924.4428	32853.5538	21282	100	0.978	2000
10	KROA100	29552.3327	33285.6428	33429.3326	21282	100	0.978	2000
11	KROA100	33009.3299	32001.7228	31002.4437	21282	100	0.978	2000
12	KROA100	36441.8826	36291.7739	36882.3207	21282	100	0.978	2000
13	KROA100	32001.5266	35442.6629	33221.6399	21282	100	0.978	2000
14	KROA100	33884.5228	33005.4375	31053.3543	21282	100	0.978	2000
15	KROA100	29599.4418	31994.3006	32886.4001	21282	100	0.978	2000
16	KROA100	31423.7716	33302.7732	30994.3206	21282	100	0.978	2000
17	KROA100	32022.8812	30492.5529	34280.4426	21282	100	0.978	2000
18	KROA100	32991.9972	33739.3329	33285.6428	21282	100	0.978	2000
19	KROA100	33903.6639	36882.5183	33001.4238	21282	100	0.978	2000
20	KROA100	33285.6428	32886.4001	36441.8826	21282	100	0.978	2000
MV	KROA100	29552.3327	30492.5529	30639.4426	21282	100	0.978	2000
PV	KROA100	36441.8826	36926.7729	37993.3290	21282	100	0.978	2000
VM	KROA100	32698.34	33657.46	33407.29	21282	100	0.978	2000

**Notation :**

RS 2-opt algorithme de recuit simulé avec la méthode d'amélioration 2-opt

RS 3-opt algorithme de recuit simulé avec la méthode d'amélioration 3-opt

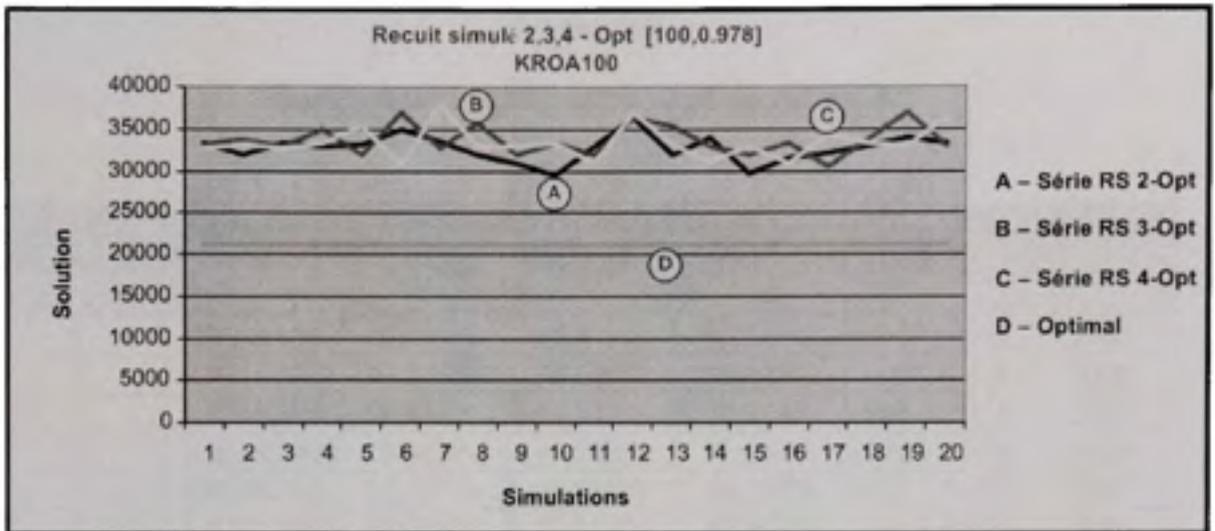
RS 4-opt algorithme de recuit simulé avec la méthode d'amélioration 4-opt

T température du recuit simulé

TR taux de refroidissement du recuit simulé

MV meilleure valeur

PV pire valeur ; VM valeur moyenne;



**Figure 5.1** Recuit simulé avec une amélioration 2-opt, 3-opt et 4-opt (choix de la température égale à 100 unités).



**Figure 5.2** Problème du voyageur de commerce KROA100 (algorithme de recuit simulé avec 2-opt, choix de la température égale à 100 unités).

Tableau 5.2

Écart-type obtenu pour les résultats du Tableau 5.1

Nombre simulations	Instance	RS 2- Opt Écart-type [%]	RS 3- Opt Écart-type [%]	RS 4- Opt Écart-type [%]	T	TR	Itérations
1	KROA100	56.403%	55.379%	53.792%	100	0.978	2000
2	KROA100	49.821%	58.692%	55.458%	100	0.978	2000
3	KROA100	56.483%	54.527%	53.314%	100	0.978	2000
4	KROA100	55.023%	63.766%	56.483%	100	0.978	2000
5	KROA100	55.307%	49.991%	66.538%	100	0.978	2000
6	KROA100	64.006%	73.512%	43.969%	100	0.978	2000
7	KROA100	57.966%	54.169%	78.523%	100	0.978	2000
8	KROA100	49.995%	67.569%	55.023%	100	0.978	2000
9	KROA100	45.137%	50.007%	54.372%	100	0.978	2000
10	KROA100	38.861%	56.403%	57.078%	100	0.978	2000
11	KROA100	55.104%	50.370%	45.674%	100	0.978	2000
12	KROA100	71.233%	70.528%	73.303%	100	0.978	2000
13	KROA100	50.369%	66.538%	56.102%	100	0.978	2000
14	KROA100	59.217%	55.086%	45.914%	100	0.978	2000
15	KROA100	39.082%	50.335%	54.527%	100	0.978	2000
16	KROA100	47.654%	56.483%	45.636%	100	0.978	2000
17	KROA100	50.469%	43.279%	61.077%	100	0.978	2000
18	KROA100	55.023%	58.535%	56.403%	100	0.978	2000
19	KROA100	59.307%	73.304%	55.067%	100	0.978	2000
20	KROA100	56.403%	54.527%	71.233%	100	0.978	2000
MEt	KROA100	38.861%	43.279%	43.969%	100	0.978	2000
PEt	KROA100	71.233%	73.304%	73.303%	100	0.978	2000
VMEt	KROA100	53.643%	58.150%	56.974%	100	0.978	2000

**Notation :**

RS 2-opt    algorithme de recuit simulé avec la méthode d'amélioration 2-opt

RS 3-opt    algorithme de recuit simulé avec la méthode d'amélioration 3-opt

RS 4-opt    algorithme de recuit simulé avec la méthode d'amélioration 4-opt

T            température du recuit simulé

TR          taux de refroidissement du recuit simulé

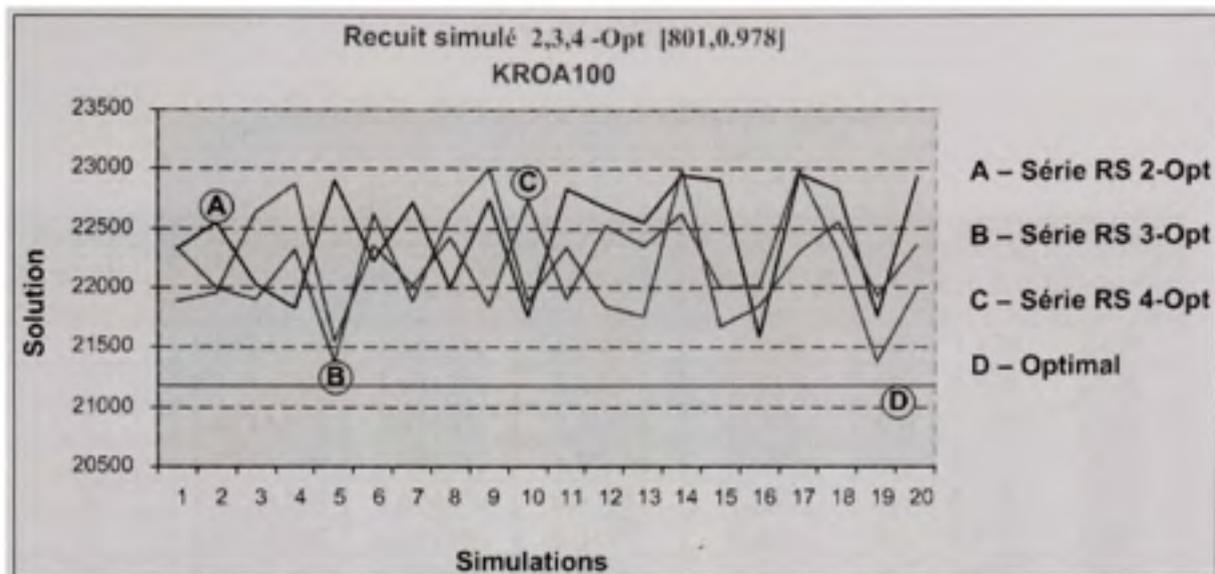
MEt        meilleure valeur écart-type

PEt        pire valeur écart-type;    VMEt        valeur moyenne écart-type;

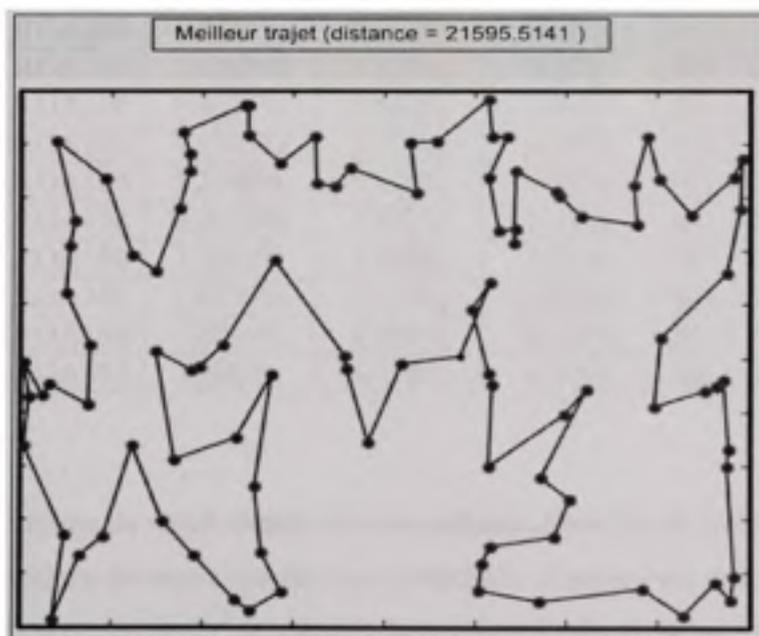
Tableau 5.3

Résultats obtenus pour l'algorithme de recuit simulé avec une amélioration 2-opt, 3-opt et 4-opt (choix de température égale à 801 unités)

Nombre simulations	Instance	RS 2-opt	RS 3-opt	RS 4-opt	Optimal	T	TR	Itérations
1	KROA100	22321.045	22337.226	21884.337	21282	801	0.978	2000
2	KROA100	22546.416	22005.554	21953.492	21282	801	0.978	2000
3	KROA100	22021.941	21902.553	22642.493	21282	801	0.978	2000
4	KROA100	21836.557	22319.443	22883.93	21282	801	0.978	2000
5	KROA100	22904.664	21383.383	21546.416	21282	801	0.978	2000
6	KROA100	22221.663	22626.3	22352.033	21282	801	0.978	2000
7	KROA100	22726.933	21884.337	22007.664	21282	801	0.978	2000
8	KROA100	22005.554	22625.217	22420.444	21282	801	0.978	2000
9	KROA100	22732.008	23005.554	21843.884	21282	801	0.978	2000
10	KROA100	21763.195	21889.003	22730.44	21282	801	0.978	2000
11	KROA100	22832.443	22332.884	21903.553	21282	801	0.978	2000
12	KROA100	22683.339	21843.884	22528.523	21282	801	0.978	2000
13	KROA100	22546.416	21763.195	22352.033	21282	801	0.978	2000
14	KROA100	22953.553	23003.43	22629.12	21282	801	0.978	2000
15	KROA100	22903.448	21668.338	22005.554	21282	801	0.978	2000
16	KROA100	21595.514	21843.884	22015.043	21282	801	0.978	2000
17	KROA100	22958.329	22302.663	22995.33	21282	801	0.978	2000
18	KROA100	22815.33	22546.416	22307.622	21282	801	0.978	2000
19	KROA100	21763.195	21931.602	21383.383	21282	801	0.978	2000
20	KROA100	22935.226	22352.033	21982.432	21282	801	0.978	2000
MV	KROA100	21595.514	21383.383	21383.383	21282	801	0.978	2000
PV	KROA100	22904.664	23005.554	22995.335	21282	801	0.978	2000
VM	KROA100	22453.34	22178.34	22218.39	21282	801	0.978	2000



**Figure 5.3** Recuit simulé avec une amélioration 2-opt, 3-opt et 4-opt (choix de la température égale à 801 unités).



**Figure 5.4** Problème du voyageur de commerce KROA100 (algorithme de recuit simulé avec 2-opt, choix de la température égale à 801 unités).

Tableau 5.4

Écart-type obtenu pour les résultats du Tableau 5.3

Nombre simulations	Instance	RS 2- Opt Écart-type [%]	RS 3- Opt Écart-type [%]	RS 4- Opt Écart-type [%]	T	TR	Itérations
1	KROA100	4.8823%	4.958%	2.83%	801	0.978	2000
2	KROA100	5.9412%	3.400%	3.16%	801	0.978	2000
3	KROA100	3.4768%	2.916%	6.39%	801	0.978	2000
4	KROA100	2.6058%	4.875%	7.53%	801	0.978	2000
5	KROA100	7.6246%	0.476%	1.24%	801	0.978	2000
6	KROA100	4.4153%	6.317%	5.03%	801	0.978	2000
7	KROA100	6.7895%	2.830%	3.41%	801	0.978	2000
8	KROA100	3.3998%	6.312%	5.35%	801	0.978	2000
9	KROA100	6.8133%	8.099%	2.64%	801	0.978	2000
10	KROA100	2.2610%	2.852%	6.81%	801	0.978	2000
11	KROA100	7.2852%	4.938%	2.92%	801	0.978	2000
12	KROA100	6.5846%	2.640%	5.86%	801	0.978	2000
13	KROA100	5.9412%	2.261%	5.03%	801	0.978	2000
14	KROA100	7.8543%	8.089%	6.33%	801	0.978	2000
15	KROA100	7.6189%	1.815%	3.40%	801	0.978	2000
16	KROA100	1.4731%	2.640%	3.44%	801	0.978	2000
17	KROA100	7.8767%	4.796%	8.05%	801	0.978	2000
18	KROA100	7.2048%	5.941%	4.82%	801	0.978	2000
19	KROA100	2.2610%	3.052%	0.48%	801	0.978	2000
20	KROA100	7.7682%	5.028%	3.29%	801	0.978	2000
MEt	KROA100	1.4731%	0.476%	0.476%	801	0.978	2000
PEt	KROA100	7.6246%	8.099%	8.051%	801	0.978	2000
VMEt	KROA100	4.642%	4.212%	4.40%	801	0.978	2000

**Notation :**

RS 2-opt    algorithme de recuit simulé avec la méthode d'amélioration 2-opt

RS 3-opt    algorithme de recuit simulé avec la méthode d'amélioration 3-opt

RS 4-opt    algorithme de recuit simulé avec la méthode d'amélioration 4-opt

T            température du recuit simulé

TR          taux de refroidissement du recuit simulé

MEt        meilleure valeur écart-type

PEt        pire valeur écart-type; VMEt        valeur moyenne écart-type;

Tableau 5.5

Temps de simulation pour l'algorithme de recuit simulé avec une amélioration 2-opt, 3-opt et 4-opt (choix de la température égale à 100 unités)

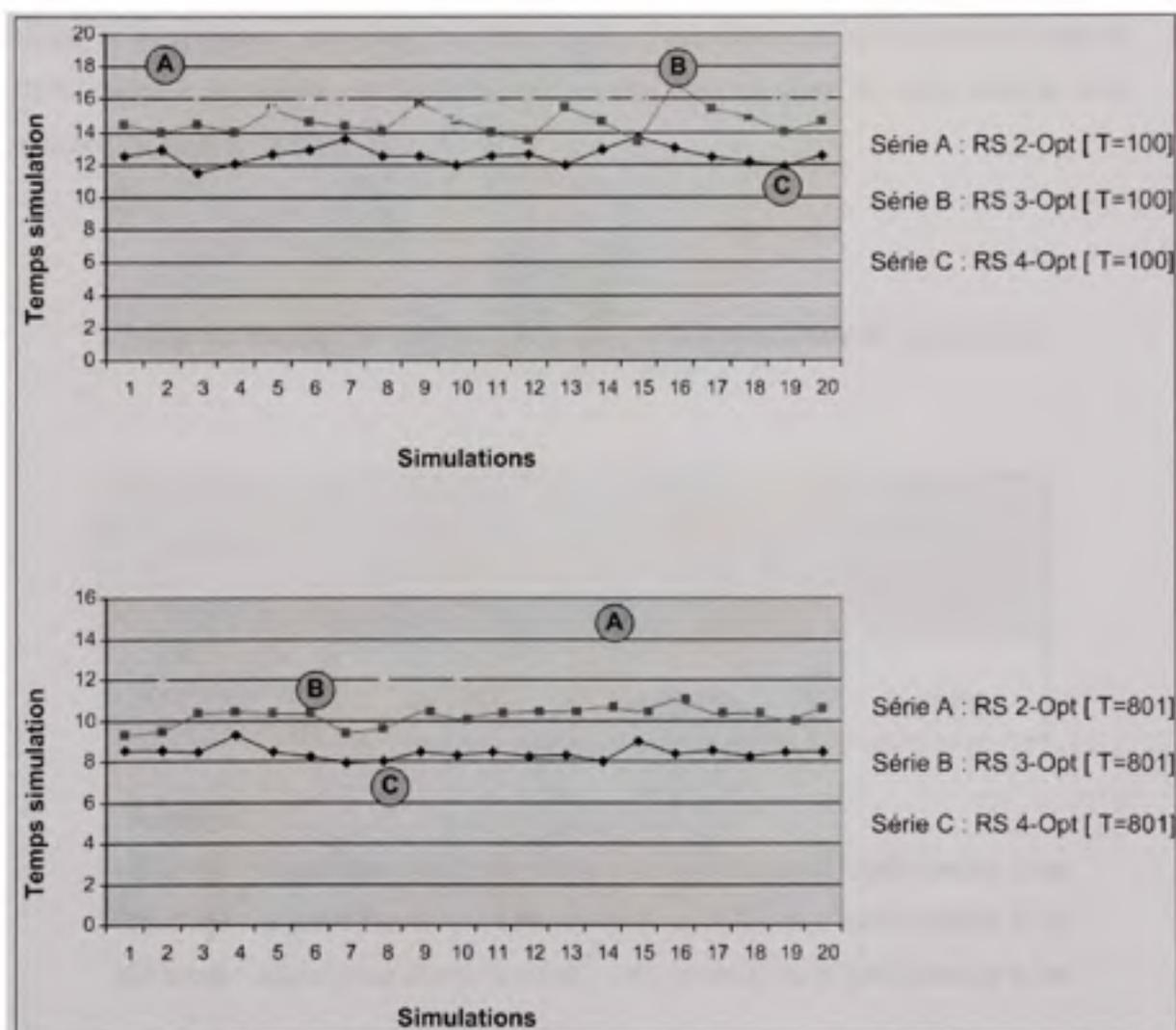
Nombre simulations	Instance	Temps simulations	Temps simulations	Temps simulations	T	TR	Itérations
		[s] RS 2- Opt	[s] RS 3- Opt	[s] RS 4- Opt			
1	KROA100	12.5183	14.4463	15.2252	100	0.978	2000
2	KROA100	12.8836	13.9943	17.2392	100	0.978	2000
3	KROA100	11.5537	14.4452	15.9033	100	0.978	2000
4	KROA100	12.0527	14.0057	14.8993	100	0.978	2000
5	KROA100	12.6333	15.3654	15.4439	100	0.978	2000
6	KROA100	12.9537	14.6637	16.2272	100	0.978	2000
7	KROA100	13.6300	14.2944	15.9400	100	0.978	2000
8	KROA100	12.5433	14.0546	15.3296	100	0.978	2000
9	KROA100	12.5437	15.7341	15.3327	100	0.978	2000
10	KROA100	12.0023	14.7443	15.0332	100	0.978	2000
11	KROA100	12.5538	13.9334	16.3200	100	0.978	2000
12	KROA100	12.6220	13.5332	16.3209	100	0.978	2000
13	KROA100	11.9926	15.4335	17.2199	100	0.978	2000
14	KROA100	12.9332	14.6630	16.3025	100	0.978	2000
15	KROA100	13.6634	13.4003	16.3299	100	0.978	2000
16	KROA100	12.9935	16.9533	15.2210	100	0.978	2000
17	KROA100	12.4377	15.3326	14.9228	100	0.978	2000
18	KROA100	12.2044	14.9445	15.3326	100	0.978	2000
19	KROA100	11.9332	14.0043	16.3224	100	0.978	2000
20	KROA100	12.5347	14.6638	16.3323	100	0.978	2000
MV	KROA100	11.5537	14.0043	14.8993	100	0.978	2000
PV	KROA100	13.6634	16.9533	17.2392	100	0.978	2000
VM	KROA100	12.5591	14.6305	15.8599	100	0.978	2000

Tableau 5.6

Temps de simulation pour l'algorithme de recuit simulé avec une amélioration 2-opt, 3-opt et 4-opt (choix de la température égale à 801 unités)

Nombre simulations	Instance	Temps simulations	Temps simulations	Temps simulations	T	TR	Itérations
		[s] RS 2- Opt	[s] RS 3- Opt	[s] RS 4- Opt			
1	KROA100	8.5300	9.3193	12.9285	801	0.978	2000
2	KROA100	8.5299	9.4299	12.0439	801	0.978	2000
3	KROA100	8.4430	10.3206	12.3430	801	0.978	2000
4	KROA100	9.3207	10.4300	12.3993	801	0.978	2000
5	KROA100	8.4388	10.3255	13.5529	801	0.978	2000
6	KROA100	8.2200	10.3250	13.5521	801	0.978	2000
7	KROA100	7.9300	9.3284	12.4348	801	0.978	2000
8	KROA100	8.0334	9.5538	12.2209	801	0.978	2000
9	KROA100	8.4377	10.4383	13.4337	801	0.978	2000
10	KROA100	8.3204	10.0327	12.0043	801	0.978	2000
11	KROA100	8.4362	10.3305	12.6300	801	0.978	2000
12	KROA100	8.2106	10.4338	12.4388	801	0.978	2000
13	KROA100	8.3155	10.4337	12.4377	801	0.978	2000
14	KROA100	8.0288	10.6599	13.7222	801	0.978	2000
15	KROA100	9.0026	10.4391	12.6118	801	0.978	2000
16	KROA100	8.3922	10.9945	12.7388	801	0.978	2000
17	KROA100	8.4933	10.3399	12.3320	801	0.978	2000
18	KROA100	8.2219	10.3299	13.0021	801	0.978	2000
19	KROA100	8.4391	9.9338	12.5447	801	0.978	2000
20	KROA100	8.4377	10.5488	12.6477	801	0.978	2000
MV	KROA100	7.9300	9.3193	12.0043	801	0.978	2000
PV	KROA100	9.3207	10.4383	13.7222	801	0.978	2000
VM	KROA100	8.4090	10.1973	12.7009	801	0.978	2000

Suite aux résultats présentés dans les tableaux précédents (5-1, 5-2, 5-3 et 5-4), les temps de simulations pour le problème de voyageur de commerce KROA100 sont regroupés dans les tableaux suivants. Le Tableau 5.5 et 5.6 montrent les temps obtenus quand la température de départ pour l'algorithme de recuit simulé est fixée à 100 et 801 unités.



**Figure 5.5 Temps de simulation pour l'algorithme de recuit simulé avec une amélioration 2-opt, 3-opt et 4-opt (choix de la température égale à 100 unités et 801 unités).**

Ces résultats montrent que les temps de simulations pour la série du Tableau 5.5 sont plus grands que celui du Tableau 5.6. Ceci veut dire que pour une température fixée à 100 unités, les temps de simulations varient entre 12 et 15 secondes (en moyenne) tandis que pour une température fixée à 801 unités, ils varient entre 8 et 12 secondes (en moyenne). La Figure 5.5 présente les temps de simulation donnés aux Tableau 5.5 et 5.6. On peut voir qu'il y a une amélioration assez importante (en terme du temps), entre les deux (2) choix de température, qui varie entre 49 % (pour l'algorithme de recuit simulé 2-opt) et 25 % (pour l'algorithme de recuit simulé 4-opt). Les résultats de cette analyse sont synthétisés dans le Tableau 5.7.

Tableau 5.7

Temps de simulation - amélioration calculée entre les choix de température à 100 et 801 unités

Instance	Temps simulations Écart-type [%] RS 2- Opt	Temps simulations Écart-type [%] RS 3- Opt	Temps simulations Écart-type [%] RS 4- Opt
KROA100 Amélioration Valeur moyenne	49.352%	43.473%	24.872%

**Notation :**

RS 2-opt algorithme de recuit simulé avec la méthode d'amélioration 2-opt

RS 3-opt algorithme de recuit simulé avec la méthode d'amélioration 3-opt

RS 4-opt algorithme de recuit simulé avec la méthode d'amélioration 4-opt

## 5.4.2 Résultats obtenus pour l'algorithme génétique avec une amélioration type 2-opt, 3-opt et 4-opt

Dans cette section, nous analysons les simulations sur le problème KROA100 en utilisant un algorithme génétique avec une procédure d'amélioration de type 2-opt, 3-opt et 4-opt. Les conditions d'expérimentations ont été présentées dans le Tableau 4-3 (voir Chapitre 4). Le Tableau 5.8 synthétise les résultats obtenus pour un nombre de 20 simulations et 2 000 itérations.

### 5.4.2.1 Discussions

Suite aux simulations, les résultats du Tableau 5.8 montrent qu'on obtient des valeurs qui sont assez proche de la solution optimale. Par contre, les résultats obtenus varient beaucoup, ce qui signifie que l'algorithme génétique n'est pas stable. On peut conclure que même si on utilise plusieurs types d'approches d'améliorations, l'algorithme n'est pas capable de se stabiliser en restant piégé dans différents optima locaux. La Figure 5.6 montre la variation des solutions pour les trois (3) types d'algorithmes d'amélioration utilisés. En ce qui concerne l'écart-type entre les différentes solutions obtenues pour les algorithmes proposés, il est présenté au Tableau 5.9. On peut noter, en terme d'efficacité, que pour un problème donné (problème du voyageur de commerce KROA100), l'algorithme génétique avec une procédure d'amélioration de type 3-opt est plus puissant que les autres deux algorithmes (voir les résultats du Tableau 5.8). Cette efficacité varie entre 0.2 % et 0.5 % (voir Tableau 5.9). En ce qui concerne les résultats obtenus pour les temps de simulation, ils sont présentés dans le Tableau 5.10. La Figure 5.7 montre les temps de simulation pour les données du Tableau 5-8. On peut constater qu'il y a une amélioration du temps de simulation qui varie entre 13 % et 38 % (amélioration constatée entre les différents types d'algorithmes génétiques proposés - voir Tableau 5.10).

Tableau 5.8

Résultats obtenus pour l'algorithme génétique avec une amélioration 2-opt, 3-opt et 4-opt

Nombre simulations	Instance	Algorithme Génétique 2-opt	Algorithme Génétique 3-opt	Algorithme Génétique 4-opt	Optimal	Taux mutation	Itérations
1	KROA100	21523.921	21942.428	22253.23	21282	0.1	2000
2	KROA100	21790.616	21735.648	22145.451	21282	0.1	2000
3	KROA100	21550.663	21889.375	21538.802	21282	0.1	2000
4	KROA100	22027.869	21538.802	22399.225	21282	0.1	2000
5	KROA100	21539.831	21550.66	21557.23	21282	0.1	2000
6	KROA100	21383.383	22145.451	22145.451	21282	0.1	2000
7	KROA100	22145.451	21883.948	22622.478	21282	0.1	2000
8	KROA100	21530.513	21966.934	21383.383	21282	0.1	2000
9	KROA100	22138.802	21523.921	21538.802	21282	0.1	2000
10	KROA100	21550.66	21538.802	22399.225	21282	0.1	2000
11	KROA100	22145.451	21538.802	22002.478	21282	0.1	2000
12	KROA100	21418.949	21889.375	21557.23	21282	0.1	2000
13	KROA100	22027.869	22145.451	22603.047	21282	0.1	2000
14	KROA100	22072.045	21383.383	21523.921	21282	0.1	2000
15	KROA100	22138.802	22027.869	21538.802	21282	0.1	2000
16	KROA100	22145.451	21966.934	22005.735	21282	0.1	2000
17	KROA100	21539.835	21538.802	22399.225	21282	0.1	2000
18	KROA100	22027.869	21550.66	21889.375	21282	0.1	2000
19	KROA100	22138.802	21538.802	21630.66	21282	0.1	2000
20	KROA100	22145.451	22002.478	21949.064	21282	0.1	2000
MV	KROA100	21383.383	21383.383	21363.195	21282	0.1	2000
PV	KROA100	22145.451	22145.451	22622.478	21282	0.1	2000
VM	KROA100	21849.11	21764.93	21954.14	21282	0.1	2000

**Notation :**

MV        meilleure valeur

PV        pire valeur

VM        valeur moyenne

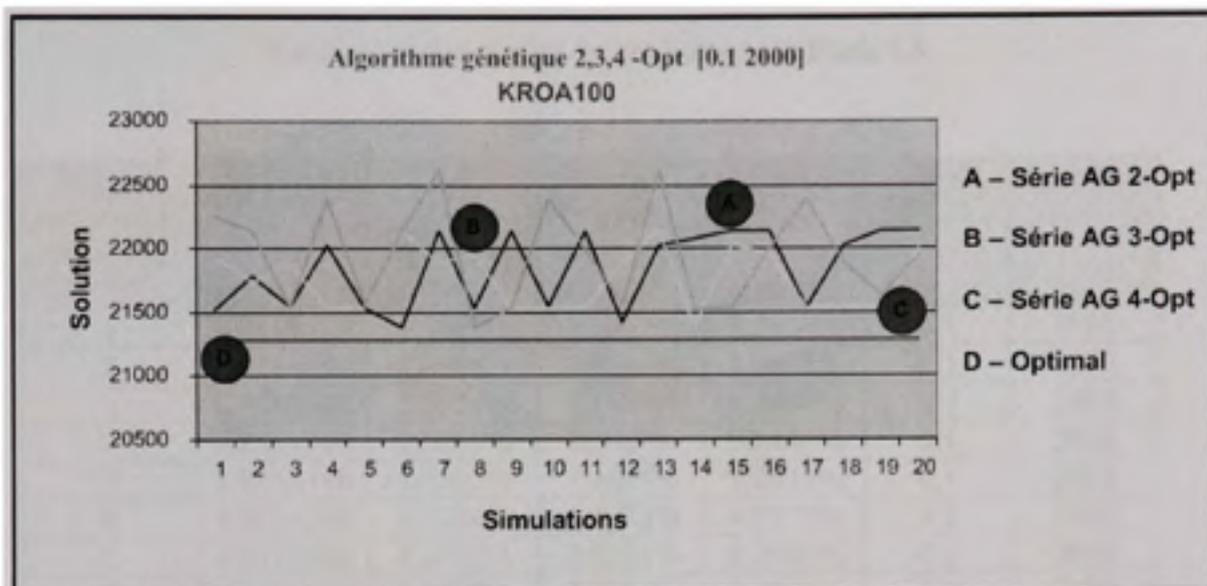


Figure 5.6 Algorithme génétique avec une amélioration 2-opt, 3-opt et 4-opt.

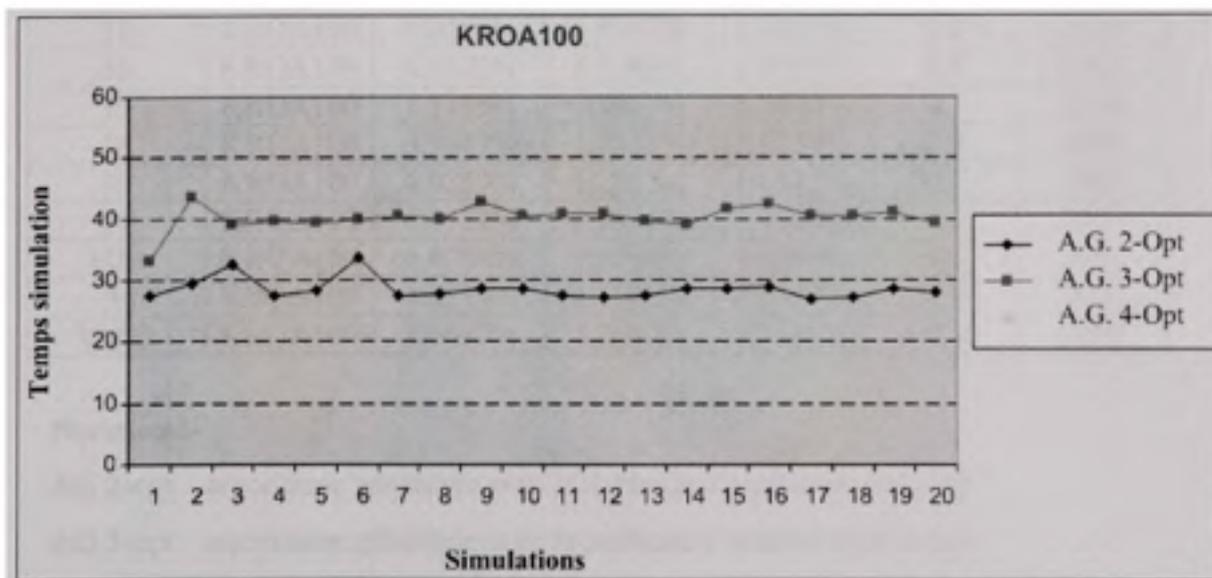


Figure 5.7 Temps de simulation algorithme génétique avec 2-opt, 3-opt et 4-opt pour Kroa100.

Tableau 5.9

Écart-type obtenu pour les résultats du Tableau 5.8

Nombre simulations	Instance	AG 2-opt Écart-type [%]	AG 3-opt Écart-type [%]	AG 4-opt Écart-type [%]	TM	Itérations
1	KROA100	1.1367%	3.1032%	4.5636%	0.1	2000
2	KROA100	2.3898%	2.1316%	4.0572%	0.1	2000
3	KROA100	1.2624%	2.8539%	1.2067%	0.1	2000
4	KROA100	3.5047%	1.2067%	5.2496%	0.1	2000
5	KROA100	1.2115%	1.2624%	1.2933%	0.1	2000
6	KROA100	0.4764%	4.0572%	4.0572%	0.1	2000
7	KROA100	4.0572%	2.8284%	6.2986%	0.1	2000
8	KROA100	1.1677%	3.2184%	0.4764%	0.1	2000
9	KROA100	4.0259%	1.1367%	1.2067%	0.1	2000
10	KROA100	1.2624%	1.2067%	5.2496%	0.1	2000
11	KROA100	4.0572%	1.2067%	3.3854%	0.1	2000
12	KROA100	0.6435%	2.8539%	1.2933%	0.1	2000
13	KROA100	3.5047%	4.0572%	6.2073%	0.1	2000
14	KROA100	3.7123%	0.4764%	1.1367%	0.1	2000
15	KROA100	4.0259%	3.5047%	1.2067%	0.1	2000
16	KROA100	4.0572%	3.2184%	3.4007%	0.1	2000
17	KROA100	1.2115%	1.2067%	5.2496%	0.1	2000
18	KROA100	3.5047%	1.2624%	2.8539%	0.1	2000
19	KROA100	4.0259%	1.2067%	1.6383%	0.1	2000
20	KROA100	4.0572%	3.3854%	3.1344%	0.1	2000
MEt	KROA100	0.4764%	0.4764%	0.4764%	0.1	2000
PEt	KROA100	4.0572%	4.0572%	6.2986%	0.1	2000
VMEt	KROA100	2.6647%	2.2692%	3.1583%	0.1	2000

**Notation :**

AG 2-opt    algorithme génétique avec la méthode d'amélioration 2-opt

AG 3-opt    algorithme génétique avec la méthode d'amélioration 3-opt

AG 4-opt    algorithme génétique avec la méthode d'amélioration 4-opt

TM            taux de mutation

MEt          meilleure valeur écart-type

PEt          pire valeur écart-type ; VMEt      valeur moyenne écart-type;

Tableau 5.10

Temps de simulation pour l'algorithme génétique avec une amélioration 2-opt, 3-opt et 4-opt

Nombre simulations	Instance	Temps simulations	Temps simulations	Temps simulations	Taux mutation	Itérations
		[s] AG 2-opt	[s] AG 3-opt	[s] AG 4-opt		
1	KROA100	27.4287	32.9942	46.4430	0.1	2000
2	KROA100	29.3002	43.6399	46.3202	0.1	2000
3	KROA100	32.6343	39.0329	47.8328	0.1	2000
4	KROA100	27.4993	39.6310	48.4015	0.1	2000
5	KROA100	28.4399	39.4492	46.7730	0.1	2000
6	KROA100	33.6329	39.9553	46.3041	0.1	2000
7	KROA100	27.4386	40.5539	46.0042	0.1	2000
8	KROA100	27.8002	39.8837	46.8399	0.1	2000
9	KROA100	28.5399	42.6639	46.3022	0.1	2000
10	KROA100	28.6320	40.5103	46.2219	0.1	2000
11	KROA100	27.5200	40.7745	47.8332	0.1	2000
12	KROA100	27.2210	40.6302	46.5529	0.1	2000
13	KROA100	27.5339	39.5520	46.0327	0.1	2000
14	KROA100	28.5300	39.1043	46.0022	0.1	2000
15	KROA100	28.5231	41.5520	46.8830	0.1	2000
16	KROA100	28.9937	42.5943	47.2284	0.1	2000
17	KROA100	27.0032	40.5539	46.3100	0.1	2000
18	KROA100	27.3053	40.6021	46.8830	0.1	2000
19	KROA100	28.5439	40.9320	46.8327	0.1	2000
20	KROA100	28.0036	39.4392	46.2219	0.1	2000
MV	KROA100	27.0032	39.0329	46.0022	0.1	2000
PV	KROA100	33.6329	42.6639	48.4015	0.1	2000
VM	KROA100	28.5269	40.2024	46.7114	0.1	2000

**Notation :**

AG 2-opt algorithme génétique avec la méthode d'amélioration 2-opt

AG 3-opt algorithme génétique avec la méthode d'amélioration 3-opt

AG 4-opt algorithme génétique avec la méthode d'amélioration 4-opt

TM taux de mutation ; PV pire valeur; MV meilleure valeur; VM valeur moyenne;

Tableau 5.11

Temps de simulation – écart de la valeur moyenne calculé entre les trois types d'algorithmes

Instance	Temps simulation (AG) Écart-type entre les méthodes 2-opt et 3-opt Valeur moyenne	Temps simulation (AG) Écart-type entre les méthodes 2-opt et 4-opt Valeur moyenne	Temps simulation (AG) Écart-type entre les méthodes 3-opt et 4-opt Valeur moyenne
Kroa100	29.044%	38.931%	13.934%

**Notation :**

AG 2-opt algorithme génétique avec la méthode d'amélioration 2-opt

AG 3-opt algorithme génétique avec la méthode d'amélioration 3-opt

AG 4-opt algorithme génétique avec la méthode d'amélioration 4-opt

### 5.4.3 Résultats obtenus pour l'algorithme de fourmis Max Min avec une amélioration 2-opt, 3-opt et 4-opt

Cette section synthétise les résultats obtenus pour un algorithme de fourmis avec une procédure d'amélioration 2-opt, 3-opt et 4-opt. Les conditions d'expérimentations ont été présentées dans le Tableau 4-4 (voir Chapitre 4). Le Tableau 5.12 présente les résultats des algorithmes à chaque simulation pour le problème du voyageur de commerce KROA100 (20 simulations et 2 000 itérations).

#### 5.4.3.1 Discussions

Les résultats présentés au Tableau 5.12 montrent que même si on applique différents types d'améliorations, la solution trouvée est pratiquement identique. On peut conclure que l'algorithme de fourmi Max Min n'est pas influencé par une de ces procédures d'amélioration proposées. En ce qui concerne la solution obtenue, elle se situe à 0.47 %

de la solution optimale. La Figure 5.10 montre les résultats (en valeur moyenne) pour les trois (3) algorithmes utilisés. Le Tableau 5.13 présente l'écart-type pour les résultats du Tableau 5.12.

Tableau 5-12

Résultats obtenus pour l'algorithme de fourmis Max Min avec une amélioration 2-opt, 3-opt et 4-opt

Nombre simulations	Instance	Algorithme Fourmi 2-opt	Algorithme Fourmi 3-opt	Algorithme Fourmi 4-opt	Optimal	Itérations
1	KROA100	21383.383	21383.383	21383.383	21282	2000
2	KROA100	21383.383	21383.383	21383.383	21282	2000
3	KROA100	21383.383	21383.383	21383.383	21282	2000
4	KROA100	21383.383	21383.383	21383.383	21282	2000
5	KROA100	21383.383	21383.383	21383.383	21282	2000
6	KROA100	21383.383	21383.383	21383.383	21282	2000
7	KROA100	21383.383	21383.383	21383.383	21282	2000
8	KROA100	21383.383	21383.383	21383.383	21282	2000
9	KROA100	21383.383	21383.383	21383.383	21282	2000
10	KROA100	21383.383	21383.383	21383.383	21282	2000
11	KROA100	21383.383	21383.383	21383.383	21282	2000
12	KROA100	21383.383	21383.383	21383.383	21282	2000
13	KROA100	21383.383	21383.383	21383.383	21282	2000
14	KROA100	21383.383	21383.383	21383.383	21282	2000
15	KROA100	21383.383	21383.383	21383.383	21282	2000
16	KROA100	21383.383	21383.383	21383.383	21282	2000
17	KROA100	21383.383	21383.383	21383.383	21282	2000
18	KROA100	21383.383	21383.383	21383.383	21282	2000
19	KROA100	21383.383	21383.383	21383.383	21282	2000
20	KROA100	21383.383	21383.383	21383.383	21282	2000
MV	KROA100	21383.383	21383.383	21383.383	21282	2000
PV	KROA100	21383.383	21383.383	21383.383	21282	2000
VM	KROA100	21383.383	21383.383	21383.383	21282	2000

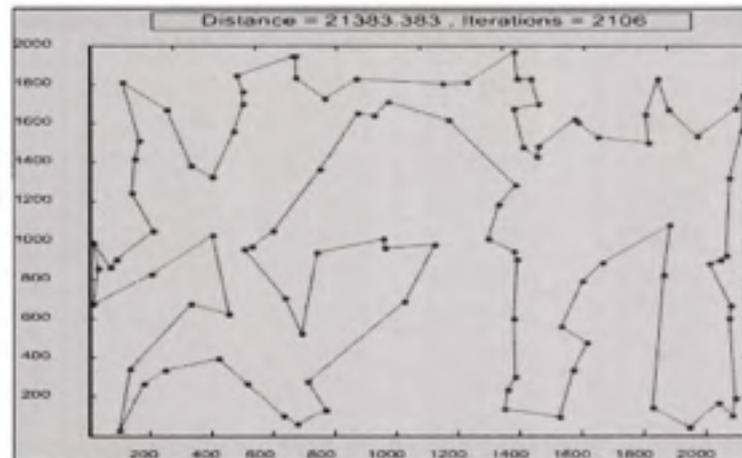
**Notation :**

MV meilleure valeur;

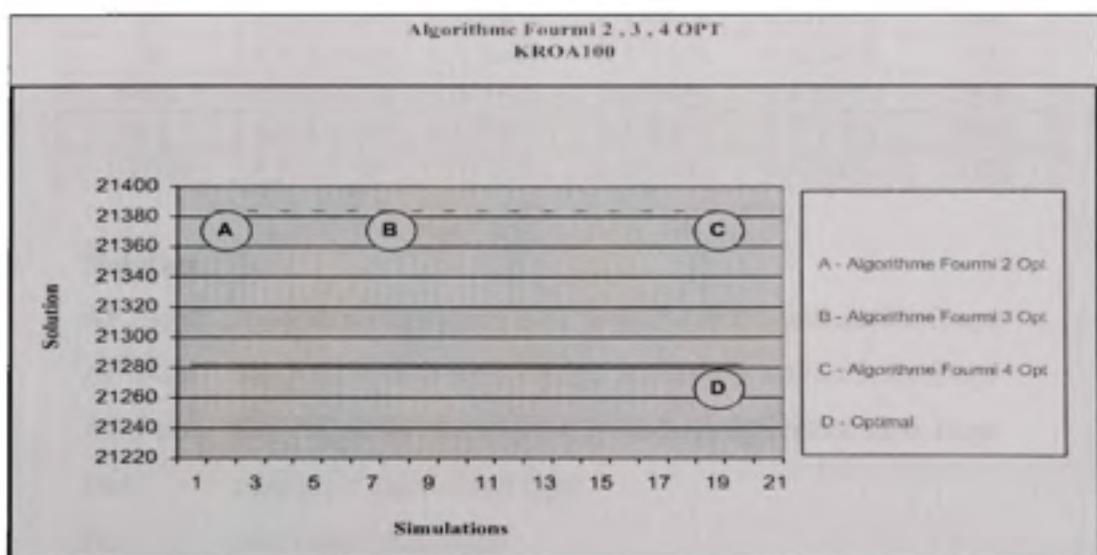
PV pire valeur;

VM valeur moyenne;

La Figure 5.8 montre le résultat obtenu grâce à un algorithme de fourmis avec une méthode d'amélioration 3-opt, pour le problème du voyageur de commerce KROA100. La Figure 5.9 montre le diagramme des résultats du Tableau 5.14 pour le problème du voyageur de commerce KROA100.



**Figure 5.8** Problème du voyageur de commerce Kroa100 (algorithme de fourmis avec 3-opt).



**Figure 5.9** Algorithme de fourmis avec une amélioration 2-opt, 3-opt et 4-opt

Tableau 5.13

Écart-type obtenu pour les résultats du Tableau 5.12

Nombre simulations	Instance	AF 2-opt Écart-type [%]	AF 3-opt Écart-type [%]	AF 4-opt Écart-type [%]	Itérations
1	KROA100	0.4764%	0.4764%	0.4764%	2000
2	KROA100	0.4764%	0.4764%	0.4764%	2000
3	KROA100	0.4764%	0.4764%	0.4764%	2000
4	KROA100	0.4764%	0.4764%	0.4764%	2000
5	KROA100	0.4764%	0.4764%	0.4764%	2000
6	KROA100	0.4764%	0.4764%	0.4764%	2000
7	KROA100	0.4764%	0.4764%	0.4764%	2000
8	KROA100	0.4764%	0.4764%	0.4764%	2000
9	KROA100	0.4764%	0.4764%	0.4764%	2000
10	KROA100	0.4764%	0.4764%	0.4764%	2000
11	KROA100	0.4764%	0.4764%	0.4764%	2000
12	KROA100	0.4764%	0.4764%	0.4764%	2000
13	KROA100	0.4764%	0.4764%	0.4764%	2000
14	KROA100	0.4764%	0.4764%	0.4764%	2000
15	KROA100	0.4764%	0.4764%	0.4764%	2000
16	KROA100	0.4764%	0.4764%	0.4764%	2000
17	KROA100	0.4764%	0.4764%	0.4764%	2000
18	KROA100	0.4764%	0.4764%	0.4764%	2000
19	KROA100	0.4764%	0.4764%	0.4764%	2000
20	KROA100	0.4764%	0.4764%	0.4764%	2000
MEt	KROA100	0.4764%	0.4764%	0.4764%	2000
PEt	KROA100	0.4764%	0.4764%	0.4764%	2000
VMEt	KROA100	0.4764%	0.4764%	0.4764%	2000

**Notation :**

AF 2-opt    algorithme de fourmi avec la méthode d'amélioration 2-opt

AF 3-opt    algorithme de fourmi avec la méthode d'amélioration 3-opt

AF 4-opt    algorithme de fourmi avec la méthode d'amélioration 4-opt

Met        meilleure valeur écart-type

Pet        pire valeur écart-type

VMEt      valeur moyenne écart-type

Tableau 5.14

Temps de simulation pour l'algorithme de fourmis Max Min  
avec une amélioration 2-opt, 3-opt et 4-opt

Nombre simulations	Instance	Temps simulations	Temps simulations	Temps simulations	Itérations
		[s] AF 2-opt	[s] AF 3-opt	[s] AF 4-opt	
1	KROA100	310.7332	375.6030	440.3329	2000
2	KROA100	312.5033	378.3329	445.2955	2000
3	KROA100	317.3288	375.2001	443.8820	2000
4	KROA100	310.9328	375.8818	448.3310	2000
5	KROA100	315.3329	380.2290	442.7731	2000
6	KROA100	310.5202	375.2199	440.9944	2000
7	KROA100	310.4239	375.2942	438.3492	2000
8	KROA100	312.6209	378.3288	442.7299	2000
9	KROA100	312.4377	375.2218	448.2085	2000
10	KROA100	310.5205	378.4424	446.2210	2000
11	KROA100	310.7111	375.0054	440.3855	2000
12	KROA100	315.4299	378.3329	445.7399	2000
13	KROA100	310.5529	377.2291	440.5226	2000
14	KROA100	310.6639	370.6633	446.3296	2000
15	KROA100	312.6302	374.6652	440.7748	2000
16	KROA100	314.6620	375.9926	446.7389	2000
17	KROA100	314.2853	374.7334	440.5299	2000
18	KROA100	310.5505	378.5538	444.0042	2000
19	KROA100	312.5299	376.2266	442.5200	2000
20	KROA100	324.0042	375.0053	445.6390	2000
MV	KROA100	310.4239	370.6633	438.3492	2000
PV	KROA100	324.0042	380.2290	448.3310	2000
VM	KROA100	312.9687	376.2081	443.5151	2000

**Notation :**

AF 2-opt algorithme de fourmi avec la méthode d'amélioration 2-opt

AF 3-opt algorithme de fourmi avec la méthode d'amélioration 3-opt

AF 4-opt algorithme de fourmi avec la méthode d'amélioration 4-opt

MV meilleure valeur

PV pire valeur; VM valeur moyenne;

Tableau 5.15

Temps de simulation – écart de la valeur moyenne calculé entre les trois types d’algorithmes

Instance	Temps simulation (AF) Écart-type entre les méthodes 2-opt et 3-opt Valeur moyenne	Temps simulation (AF) Écart-type entre les méthodes 2-opt et 4-opt Valeur moyenne	Temps simulation (AF) Écart-type entre les méthodes 3-opt et 4-opt Valeur moyenne
Kroa100	16.81%	29.43%	15.18%

En ce qui concerne les temps de simulation, ils sont donnés dans le Tableau 5.14. Pour ce type de problème (KROA100) les temps de simulation se situent entre 313 et 444 secondes (en moyenne), ce qui donne un écart-type situé entre 17 % et 29 % (écart-type calculé entre les différentes méthodes présentées au Tableau 5.14). La Figure 5.10 montre le diagramme de temps de simulations pour l’algorithme de fourmis avec une amélioration 2,3 et 4-opt (problème Kroa100).

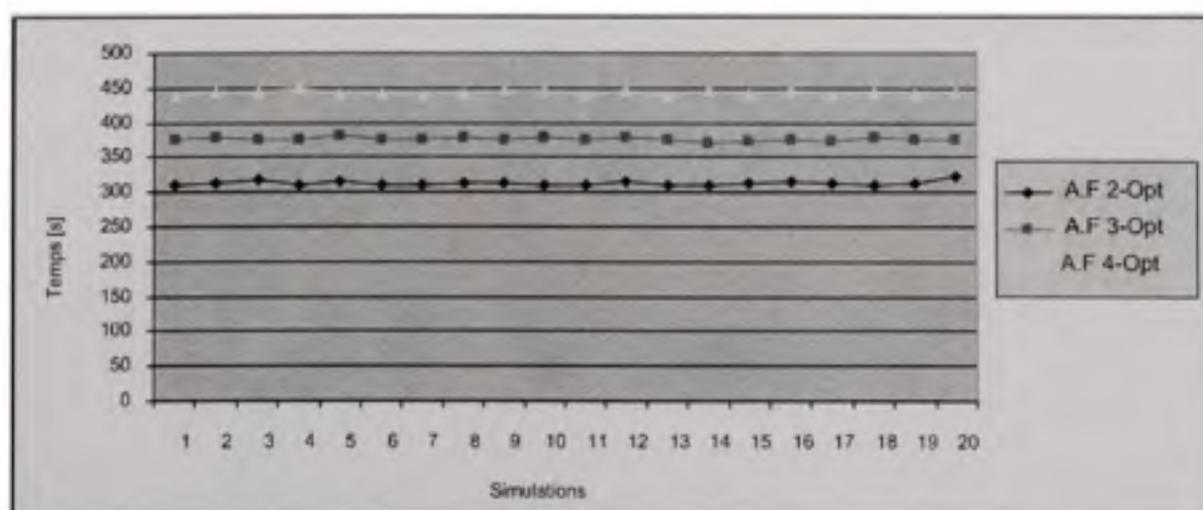


Figure 5.10 Temps de simulation algorithme de fourmis avec 2-opt, 3-opt et 4-opt pour Kroa100.

## 5.5 Conclusions

Les Tableaux 5.16, 5.17 et 5.18 résument les résultats du problème du voyageur de commerce KROA100 présentée dans ce chapitre. L'utilisation de ces algorithmes (voir les tableaux précédents) montre qu'ils sont capables de donner comme meilleure solution une valeur située à 0.47 % de la solution optimale (sauf l'algorithme de recuit simulé avec une amélioration type 2-opt). En analysant la totalité de simulations on tire la conclusion que pour l'algorithme de recuit simulé et pour l'algorithme génétique, l'amélioration type 3-opt donne le meilleur résultat. La comparaison, entre les trois (3) algorithmes utilisés (voir Tableaux 5.16 et 5.17), montre que pour ce type de problème (KROA100) l'algorithme de fourmis Max Min est le plus puissant. Par contre, si on regarde le résumé des temps de simulations (voir Tableau 5.18) on peut conclure que les meilleurs résultats sont obtenus par l'algorithme de recuit simulé.

Tableau 5.16

Comportement des algorithmes – problème du voyageur de commerce KROA100

NS	Instance	RS 2 - Opt	RS 3 - Opt	RS 4 - Opt	AG 2-opt	AG 3-opt	AG 4-opt	AF 2-opt	AF 3-opt	AF 4-opt	Optimal
1	Kroa100	22321.045	22337.226	21884.337	21523.921	21942.428	22253.23	21383.383	21383.38	21383.383	21282
2	Kroa100	22546.416	22005.554	21953.492	21790.6	21735.648	22145.451	21383.383	21383.38	21383.38	21282
3	Kroa100	22021.941	21902.553	22642.493	21550.66	21889.375	21538.802	21383.383	21383.38	21383.38	21282
4	Kroa100	21836.557	22319.443	22883.93	22027.869	21538.802	22399.225	21383.383	21383.38	21383.38	21282
5	Kroa100	22904.664	21383.383	21546.416	21539.83	21550.66	21557.23	21383.383	21383.38	21383.38	21282
6	Kroa100	22221.663	22626.3	22352.033	21383.383	22145.451	22145.451	21383.383	21383.38	21383.38	21282
7	Kroa100	22726.933	21884.337	22007.664	22145.451	21883.948	22622.478	21383.383	21383.38	21383.38	21282
8	Kroa100	22005.554	22625.217	22420.444	21530.513	21966.934	21383.383	21383.383	21383.38	21383.38	21282
9	Kroa100	22732.008	23005.554	21843.884	22138.802	21523.921	21538.802	21383.383	21383.38	21383.38	21282
10	Kroa100	21763.195	21889.003	22730.44	21550.66	21538.802	22399.225	21383.383	21383.38	21383.38	21282
11	Kroa100	22832.443	22332.884	21903.553	22145.451	21538.802	22002.478	21383.383	21383.38	21383.38	21282
12	Kroa100	22683.339	21843.884	22528.523	21418.949	21889.375	21557.23	21383.383	21383.38	21383.38	21282
13	Kroa100	22546.416	21763.195	22352.033	22027.869	22145.451	22603.047	21383.383	21383.38	21383.38	21282
14	Kroa100	22953.553	23003.43	22629.12	22072.045	21383.383	21523.921	21383.383	21383.38	21383.38	21282
15	Kroa100	22903.448	21668.338	22005.554	22138.802	22027.869	21538.802	21383.383	21383.38	21383.38	21282
16	Kroa100	21595.514	21843.884	22015.043	22145.451	21966.934	22005.735	21383.383	21383.38	21383.38	21282
17	Kroa100	22958.329	22302.663	22995.33	21539.83	21538.802	22399.225	21383.383	21383.38	21383.38	21282
18	Kroa100	22815.33	22546.416	22307.622	22027.869	21550.66	21889.375	21383.383	21383.38	21383.38	21282
19	Kroa100	21763.195	21931.602	21383.383	22138.802	21538.802	21630.66	21383.383	21383.38	21383.38	21282
20	Kroa100	22935.226	22352.033	21982.432	22145.451	22002.478	21949.064	21383.383	21383.38	21383.38	21282
MS	Kroa100	21595.514	21383.383	21383.383	21383.383	21383.383	21383.383	21383.383	21383.38	21383.38	21282
PS	Kroa100	23153.553	23005.554	22995.335	22145.451	22145.451	22622.478	21383.383	21383.38	21383.38	21282
SM	Kroa100	22453.34	22178.34	22218.39	21849.11	21764.93	21954.14	21383.383	21383.38	21383.38	21282

Tableau 5.17

Écart-type obtenu pour les résultats du Tableau 5-15

Nombre simulations	Instance	RS 2-opt Écart-type [%]	RS 3-opt Écart-type [%]	RS 4-opt Écart-type [%]	AG 2-opt Écart-type [%]	AG 3-opt Écart-type [%]	AG 4-opt Écart-type [%]	AF 2-opt Écart-type [%]	AF 3-opt Écart-type [%]	AF 4-opt Écart-type [%]
1	Kroa100	4.88%	4.95%	2.83%	1.13%	3.10%	4.56%	0.47%	0.47%	0.47%
2	Kroa100	5.94%	3.40%	3.16%	2.38%	2.13%	4.05%	0.47%	0.47%	0.47%
3	Kroa100	3.47%	2.91%	6.39%	1.26%	2.85%	1.20%	0.47%	0.47%	0.47%
4	Kroa100	2.60%	4.87%	7.53%	3.50%	1.20%	5.24%	0.47%	0.47%	0.47%
5	Kroa100	7.62%	0.47%	1.24%	1.21%	1.26%	1.29%	0.47%	0.47%	0.47%
6	Kroa100	4.41%	6.31%	5.03%	0.47%	4.05%	4.05%	0.47%	0.47%	0.47%
7	Kroa100	6.78%	2.83%	3.41%	4.05%	2.82%	6.29%	0.47%	0.47%	0.47%
8	Kroa100	3.39%	6.31%	5.35%	1.16%	3.21%	0.47%	0.47%	0.47%	0.47%
9	Kroa100	6.81%	8.09%	2.64%	4.02%	1.13%	1.20%	0.47%	0.47%	0.47%
10	Kroa100	2.26%	2.85%	6.81%	1.26%	1.20%	5.24%	0.47%	0.47%	0.47%
11	Kroa100	7.28%	4.93%	2.92%	4.05%	1.20%	3.38%	0.47%	0.47%	0.47%
12	Kroa100	6.58%	2.64%	5.86%	0.64%	2.85%	1.29%	0.47%	0.47%	0.47%
13	Kroa100	5.94%	2.26%	5.03%	3.50%	4.05%	6.20%	0.47%	0.47%	0.47%
14	Kroa100	7.85%	8.08%	6.33%	3.71%	0.47%	1.13%	0.47%	0.47%	0.47%
15	Kroa100	7.61%	1.81%	3.40%	4.02%	3.50%	1.20%	0.47%	0.47%	0.47%
16	Kroa100	1.47%	2.64%	3.44%	4.05%	3.21%	3.40%	0.47%	0.47%	0.47%
17	Kroa100	7.87%	4.79%	8.05%	1.21%	1.20%	5.24%	0.47%	0.47%	0.47%
18	Kroa100	7.20%	5.94%	4.82%	3.50%	1.26%	2.85%	0.47%	0.47%	0.47%
19	Kroa100	2.26%	3.05%	0.48%	4.02%	1.20%	1.63%	0.47%	0.47%	0.47%
20	Kroa100	7.76%	5.02%	3.29%	4.05%	3.38%	3.13%	0.47%	0.47%	0.47%
Écart-type moyenne	Kroa100	4.64%	4.21%	4.40%	2.66%	2.26%	3.15%	0.47%	0.47%	0.47%

Tableau 5.18

Temps de simulation- problème du voyageur de commerce KROA100

Nombre simulations	Instance	RS 2 – Opt TS [s]	RS 3 – Opt TS [s]	RS 4 – Opt TS [s]	AG 2-opt TS [s]	AG 3-opt TS [s]	AG 4-opt TS [s]	AF 2-opt TS [s]	AF 3-opt TS [s]	AF 4-opt TS [s]
1	Kroa100	8.5300	9.3193	12.9285	27.4287	32.9942	46.4430	310.7332	375.6030	440.3329
2	Kroa100	8.5299	9.4299	12.0439	29.3002	43.6399	46.3202	312.5033	378.3329	445.2955
3	Kroa100	8.4430	10.3206	12.3430	32.6343	39.0329	47.8328	317.3288	375.2001	443.8820
4	Kroa100	9.3207	10.4300	12.3993	27.4993	39.6310	48.4015	310.9328	375.8818	448.3310
5	Kroa100	8.4388	10.3255	13.5529	28.4399	39.4492	46.7730	315.3329	380.2290	442.7731
6	Kroa100	8.2200	10.3250	13.5521	33.6329	39.9553	46.3041	310.5202	375.2199	440.9944
7	Kroa100	7.9300	9.3284	12.4348	27.4386	40.5539	46.0042	310.4239	375.2942	438.3492
8	Kroa100	8.0334	9.5538	12.2209	27.8002	39.8837	46.8399	312.6209	378.3288	442.7299
9	Kroa100	8.4377	10.4383	13.4337	28.5399	42.6639	46.3022	312.4377	375.2218	448.2085
10	Kroa100	8.3204	10.0327	12.0043	28.6320	40.5103	46.2219	310.5205	378.4424	446.2210
11	Kroa100	8.4362	10.3305	12.6300	27.5200	40.7745	47.8332	310.7111	375.0054	440.3855
12	Kroa100	8.2106	10.4338	12.4388	27.2210	40.6302	46.5529	315.4299	378.3329	445.7399
13	Kroa100	8.3155	10.4337	12.4377	27.5339	39.5520	46.0327	310.5529	377.2291	440.5226
14	Kroa100	8.0288	10.6599	13.7222	28.5300	39.1043	46.0022	310.6639	370.6633	446.3296
15	Kroa100	9.0026	10.4391	12.6118	28.5231	41.5520	46.8830	312.6302	374.6652	440.7748
16	Kroa100	8.3922	10.9945	12.7388	28.9937	42.5943	47.2284	314.6620	375.9926	446.7389
17	Kroa100	8.4933	10.3399	12.3320	27.0032	40.5539	46.3100	314.2853	374.7334	440.5299
18	Kroa100	8.2219	10.3299	13.0021	27.3053	40.6021	46.8830	310.5505	378.5538	444.0042
19	Kroa100	8.4391	9.9338	12.5447	28.5439	40.9320	46.8327	312.5299	376.2266	442.5200
20	Kroa100	8.4377	10.5488	12.6477	28.0036	39.4392	46.2219	324.0042	375.0053	445.6390
Valeur moyenne	Kroa100	8.40909	10.19737	12.70096	28.52619	40.20244	46.71114	312.96887	376.2081	443.5151

## CHAPITRE 6

### CONCLUSIONS ET PERSPECTIVES

#### 6.1 Conclusions

L'objectif principal du travail présenté dans ce mémoire est l'étude du potentiel de différents types d'algorithmes pour résoudre un problème d'optimisation non-déterministe polynomiale difficile, soit le problème du voyageur de commerce. Pour atteindre ce but, plusieurs aspects ont été abordés pendant cette étude : l'étude des algorithmes d'optimisation, l'application des méthodes d'amélioration, la programmation et la simulation avec le logiciel MATLAB, la paramétrisation des algorithmes, ainsi que l'étude des résultats. Ce mémoire servira comme base pour le développement d'autres problèmes (problèmes théoriques et réels - voir la section 6.2).

Dans le premier chapitre, nous avons situé le problème dans son contexte. Par la suite, dans le deuxième chapitre, une revue de la littérature a été présentée. Une présentation de la méthodologie et une revue des algorithmes ont fait l'objet du troisième chapitre. Les expérimentations ont fait l'objet du quatrième chapitre où nous avons résolu plusieurs instances et où particulièrement, une méthode qui trouve la température de départ pour le recuit simulé a été proposée et vérifiée. Dans ce contexte, plusieurs algorithmes ont été choisis pour répondre au problème du voyageur de commerce. Les résultats obtenus ont été présentés et commentés où cinquième chapitre.

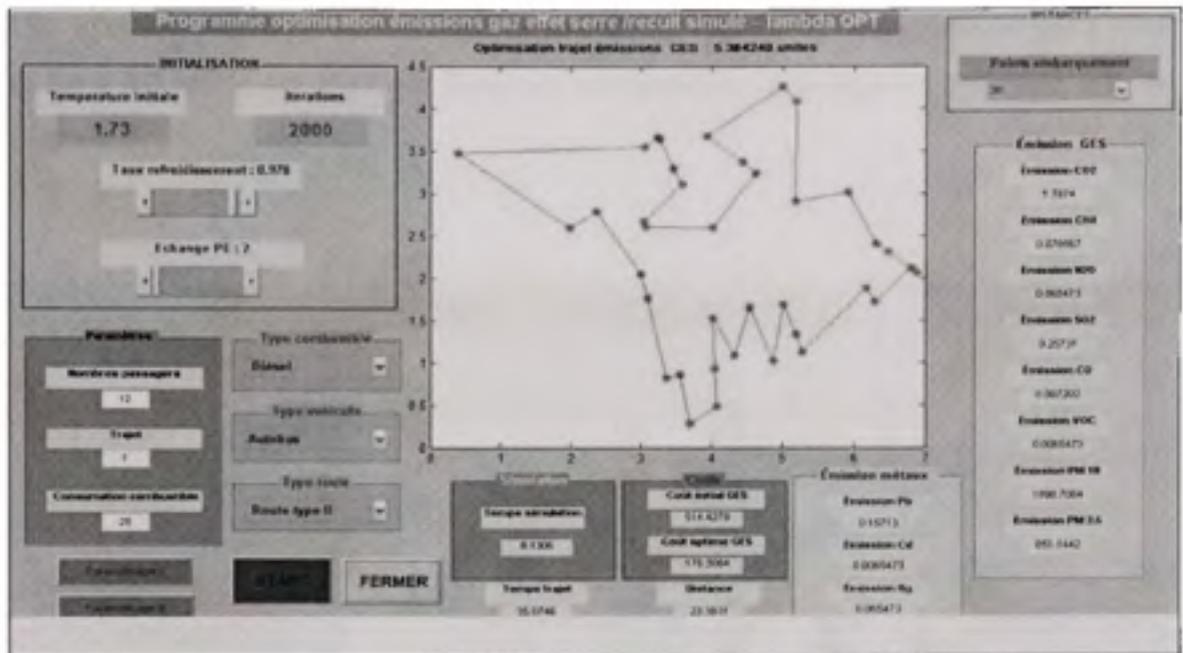
Notre travail de recherche dans le présent mémoire a permis d'approfondir les comportements des différentes méta-heuristiques pour un problème non-déterministe polynomiale difficile. Il a aussi montré la capacité d'amélioration des ces algorithmes pour différentes instances du problème.

## 6.2 Perspectives de recherche

Les perspectives de ce travail sont relativement nombreuses, tant du point de vue théorique que pratique.

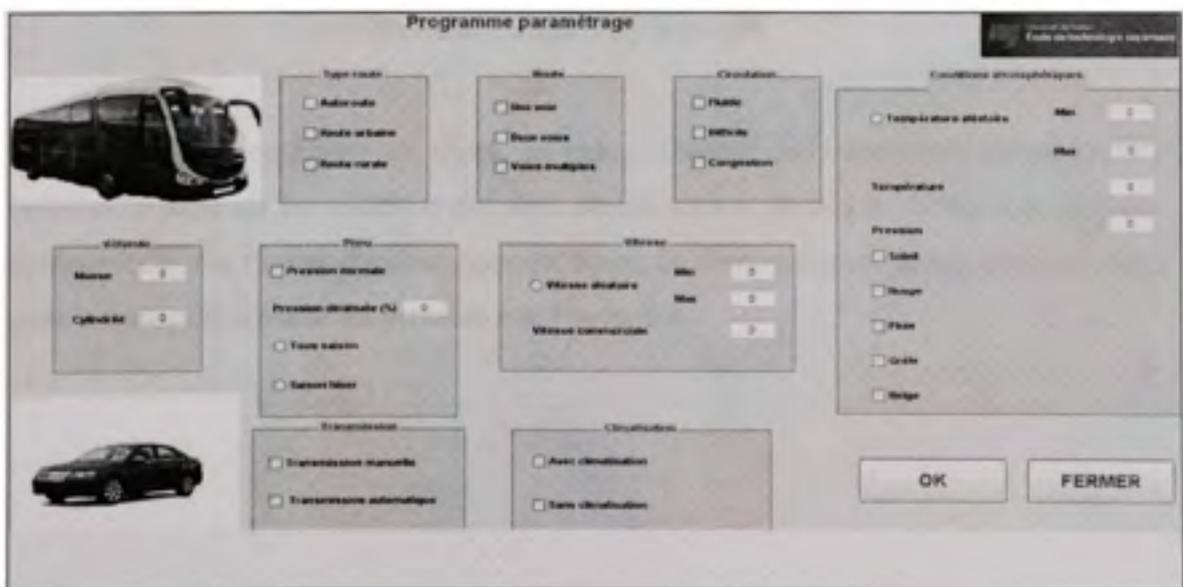
De point de vue théorique, il est possible d'approfondir les modalités de réglage des algorithmes, ce qui peut donner de bons résultats sur leur comportement. En ce qui concerne le côté pratique, plusieurs perspectives se présentent, non seulement sur les algorithmes utilisés mais aussi sur d'autres types d'algorithmes. Une perspective importante est l'étude approfondie et ciblée entre les différents types de méthodes d'amélioration présentées. Enfin une perspective importante reste les applications à des problèmes d'optimisation dans les domaines de transport (optimisation du transport scolaire) et environnemental (optimisation des émissions de gaz à effet de serre.).

Plusieurs de ces perspectives ont été déjà résolus et d'autres sont en train d'être étudiées. Parmi les problèmes pour lesquelles on a obtenu des résultats, on note le problème du transport scolaire (avec des contraintes de circulation), le problème d'optimisation des émissions de gaz à effet de serre pour un problème non-déterministe polynomiale difficile (problème du voyageur de commerce). D'autres perspectives, comme le problème du paramétrage des véhicules (module pour le programme qui détermine les émissions de gaz à effet de serre), le problème d'optimisation qui utilise une nouvelle méta-heuristique, le problème de planification des véhicules, sont en étude. La Figure 6.1 montre l'interface graphique pour le programme qui optimise les émissions de gaz à effet de serre.



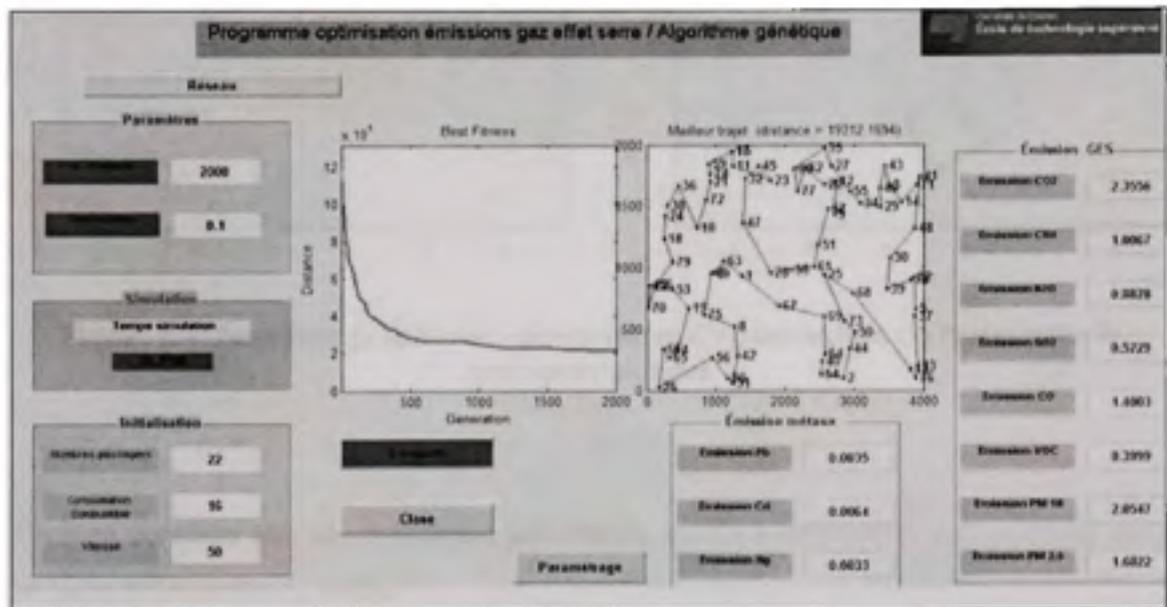
**Figure 6.1** Interface graphique - programme d'optimisation des émissions de gaz à effet de serre

La Figure 6.2 présente l'interface graphique du module de paramétrage pour les véhicules.



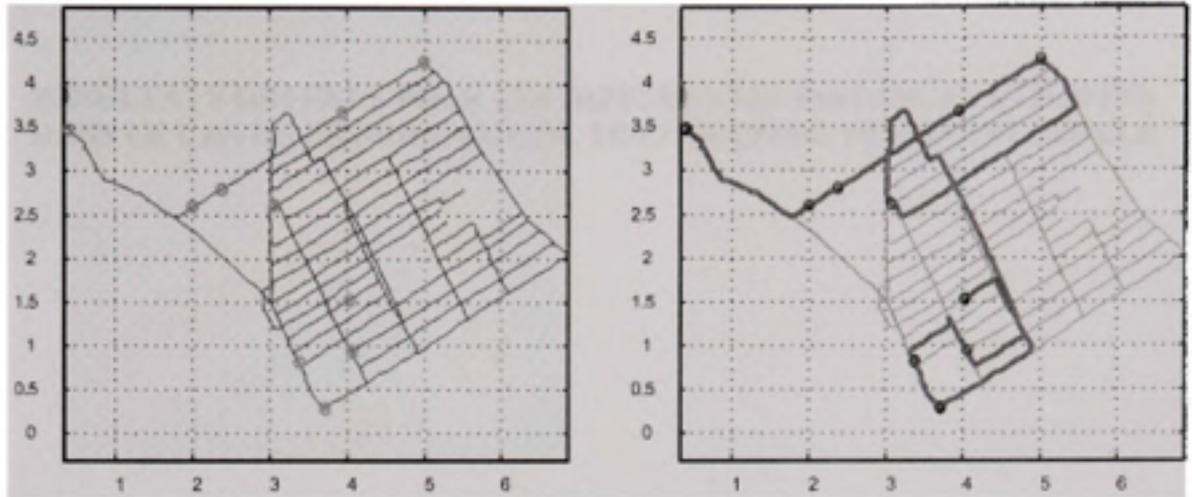
**Figure 6.2** Interface graphique - programme de paramétrage pour les véhicules

Une interface graphique pour un programme d'optimisation des émissions de gaz à effet de serre qui utilise une approche basée sur un algorithme génétique est présentée dans la Figure 6.3.



**Figure 6.3** Interface graphique - programme d'optimisation des émissions de gaz à effet de serre qui utilise une approche basée sur l'algorithme génétique

La perspective d'optimiser un trajet, en tenant compte des contraintes de circulation (vitesse variable sur les routes, arrêts, sens de circulation, goulot de circulation, *etc.*) des contraintes d'état (temps d'embarquement, temps de débarquement, temps d'attente, *etc.*) pour le transport scolaire est présente à la Figure 6.4.



**Figure 6.4 Interface graphique – programme d'optimisation du trajet pour le transport scolaire**

On peut conclure que ces différentes possibilités, qu'on a présentées, sont étudiées dans le cadre des travaux en cours.

## **ANNEXE I**

**RÉSULTATS OBTENUS POUR LES DIFFÉRENTES INSTANCES UTILISÉES  
DANS LE CAS DU PROGRAMME DE TEMPÉRATURE DU RECUIT SIMULÉ**

Cette annexe résume les simulations qui ont été réalisés, pour trouver la température de départ du recuit simulé. Les Tableaux 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 et 11 présentes les résultats trouvés pour les différentes instances utilisées. La Figure 1 montre le diagramme de distribution aléatoire de la température pour laquelle la valeur de la solution initiale est égale à 600 unités.

Tableau 1

La variation de la température médiane  $T_0$  en fonction de la solution initiale  $s_0$

Essais	Valeur de la solution initiale $s_0$	Nombre aléatoire d'itérations	Valeur de la température médiane $T_0$	Temps (s) de simulation
1	25000	5000	683.3948	2.297
2	25000	5000	687.6188	2.282
3	25000	5000	672.4482	2.244
4	25000	5000	670.2367	2.213
5	25000	5000	692.3793	2.330
6	25000	5000	665.0842	2.215
7	25000	5000	636.2781	2.374
8	25000	5000	665.1223	2.297
9	25000	5000	696.5739	2.437
10	25000	5000	673.0465	2.328
11	25000	10000	668.2489	9.116
12	25000	10000	667.9553	9.125
13	25000	10000	667.3431	9.347
14	25000	10000	668.6589	9.105
15	25000	10000	634.1423	9.452
16	25000	10000	669.2166	9.226
17	25000	10000	668.5773	9.603
18	25000	10000	667.6410	9.330
19	25000	10000	671.3949	9.120
20	25000	10000	670.1233	9.767
21	25000	15000	674.5809	21.255
22	25000	15000	670.2457	21.328
23	25000	15000	669.8817	21.173
24	25000	15000	669.7104	21.454
25	25000	15000	634.1365	21.338
26	25000	15000	671.2121	21.116
27	25000	15000	674.8891	21.883
28	25000	15000	669.2337	21.559
29	25000	15000	668.6441	21.552
30	25000	15000	669.7993	21.009
31	25000	20000	669.5527	38.297
32	25000	20000	669.8677	38.265
33	25000	20000	687.3863	38.406
34	25000	20000	669.2259	38.110
35	25000	20000	662.4463	38.224

Essais	Valeur de la solution initiale $s_0$	Nombre aléatoire d'itérations	Valeur de la température médiane $T_0$	Temps (s) de simulation
36	25000	20000	667.2238	38.110
37	25000	20000	672.3225	38.446
38	25000	20000	667.4476	38.523
39	25000	20000	666.2221	38.442
40	25000	20000	629.5587	38.221

Tableau 2

La variation de la température médiane  $T_0$  en fonction de la solution initiale  $s_0$

Essais	Valeur de la solution initiale $s_0$	Nombre aléatoire d'itérations	Valeur de la température médiane $T_0$	Temps (s) de simulation
1	15000	5000	417.4430	2.375
2	15000	5000	400.2664	2.328
3	15000	5000	412.1926	2.225
4	15000	5000	410.1425	2.167
5	15000	5000	392.0827	2.112
6	15000	5000	412.3378	2.370
7	15000	5000	400.9672	2.338
8	15000	5000	408.2210	2.104
9	15000	5000	415.0124	2.227
10	15000	5000	402.0460	2.114
11	15000	10000	409.7424	9.094
12	15000	10000	408.6691	9.224
13	15000	10000	409.4489	9.446
14	15000	10000	405.7241	9.116
15	15000	10000	407.5016	9.227
16	15000	10000	409.9021	9.128
17	15000	10000	394.7441	9.104
18	15000	10000	392.5592	9.536
19	15000	10000	398.1098	9.257
20	15000	10000	406.2033	9.142
21	15000	15000	415.9788	21.109
22	15000	15000	410.4932	21.244
23	15000	15000	412.0447	21.468

Essais	Valeur de la solution initiale $s_0$	Nombre aléatoire d'itérations	Valeur de la température médiane $T_0$	Temps (s) de simulation
24	15000	15000	417.2287	21.154
25	15000	15000	410.1922	21.683
26	15000	15000	401.2304	21.246
27	15000	15000	400.4593	21.116
28	15000	15000	407.4838	21.468
29	15000	15000	407.0880	21.223
30	15000	15000	402.3827	21.005
31	15000	20000	398.6264	38.562
32	15000	20000	404.7254	38.233
33	15000	20000	400.3682	38.257
34	15000	20000	399.2522	38.524
35	15000	20000	403.2266	38.355
36	15000	20000	398.3544	38.225
37	15000	20000	399.0441	38.367
38	15000	20000	402.3533	38.243
39	15000	20000	404.2536	38.152
40	15000	20000	402.3126	38.159

Tableau 3

La variation de la température médiane  $T_0$  en fonction de la solution initiale  $s_0$

Essais	Valeur de la solution initiale $s_0$	Nombre aléatoire d'itérations	Valeur de la température médiane $T_0$	Temps (s) de simulation
1	8000	5000	224.0917	2.105
2	8000	5000	216.7863	2.001
3	8000	5000	215.4112	2.446
4	8000	5000	215.4002	2.119
5	8000	5000	218.0447	2.034
6	8000	5000	220.0235	2.561
7	8000	5000	214.4960	2.102
8	8000	5000	217.0583	2.128
9	8000	5000	218.4222	2.114
10	8000	5000	218.3552	2.092
11	8000	10000	220.6697	8.160

Essais	Valeur de la solution initiale $s_0$	Nombre aléatoire d'itérations	Valeur de la température médiane $T_0$	Temps (s) de simulation
12	8000	10000	218.9603	8.261
13	8000	10000	215.8909	8.247
14	8000	10000	217.5148	8.269
15	8000	10000	218.3392	8.116
16	8000	10000	218.2201	8.355
17	8000	10000	218.6005	8.266
18	8000	10000	216.3646	8.008
19	8000	10000	218.3631	8.233
20	8000	10000	218.2777	8.211
21	8000	15000	215.9018	19.003
22	8000	15000	218.4009	19.013
23	8000	15000	217.9028	18.967
24	8000	15000	218.4421	19.112
25	8000	15000	218.6723	18.447
26	8000	15000	218.0309	18.860
27	8000	15000	217.1192	18.924
28	8000	15000	216.7621	18.667
29	8000	15000	218.4501	19.011
30	8000	15000	218.9972	19.002
31	8000	20000	215.5702	35.002
32	8000	20000	218.3344	34.487
33	8000	20000	218.2926	34.760
34	8000	20000	218.4401	34.559
35	8000	20000	216.3362	34.004
36	8000	20000	218.3342	34.910
37	8000	20000	218.0264	35.002
38	8000	20000	218.6682	34.995
39	8000	20000	218.3324	34.504
40	8000	20000	216.2216	34.661

Tableau 4

La variation de la température médiane  $T_\theta$  en fonction de la solution initiale  $s_\theta$ 

Essais	Valeur de la solution initiale $s_\theta$	Nombre aléatoire d'itérations	Valeur de la température médiane $T_\theta$	Temps (s) de simulation
1	6000	5000	160.7724	2.148
2	6000	5000	162.9460	2.105
3	6000	5000	159.8805	2.143
4	6000	5000	161.4338	2.132
5	6000	5000	162.1123	2.044
6	6000	5000	161.5509	2.212
7	6000	5000	162.3466	2.036
8	6000	5000	162.2241	2.167
9	6000	5000	162.7743	2.202
10	6000	5000	163.2240	2.004
11	6000	10000	160.5531	8.014
12	6000	10000	159.1123	8.441
13	6000	10000	162.6002	8.220
14	6000	10000	161.1824	8.331
15	6000	10000	161.7345	8.267
16	6000	10000	161.2001	8.183
17	6000	10000	159.9914	8.014
18	6000	10000	162.7017	8.316
19	6000	10000	162.1433	8.210
20	6000	10000	161.8802	8.311
21	6000	15000	162.3391	19.006
22	6000	15000	161.8294	18.926
23	6000	15000	159.4367	18.991
24	6000	15000	159.9102	19.114
25	6000	15000	163.5243	19.223
26	6000	15000	161.6535	18.770
27	6000	15000	162.0012	18.929
28	6000	15000	161.4336	19.202
29	6000	15000	162.3327	18.534
30	6000	15000	162.4356	18.881
31	6000	20000	161.3030	34.773
32	6000	20000	161.7903	35.110
33	6000	20000	159.2023	35.192
34	6000	20000	162.1127	34.733
35	6000	20000	162.3325	34.991

Essais	Valeur de la solution initiale $s_0$	Nombre aléatoire d'itérations	Valeur de la température médiane $T_0$	Temps (s) de simulation
36	6000	20000	161.1947	34.012
37	6000	20000	160.4498	34.027
38	6000	20000	161.7772	34.661
39	6000	20000	159.2743	34.928
40	6000	20000	159.2450	35.332

Tableau 5

La variation de la température médiane  $T_0$  en fonction de la solution initiale  $s_0$

Essais	Valeur de la solution initiale $s_0$	Nombre aléatoire d'itérations	Valeur de la température médiane $T_0$	Temps (s) de simulation
1	3000	5000	83.1470	2.248
2	3000	5000	83.2244	2.353
3	3000	5000	81.2609	2.331
4	3000	5000	83.6264	2.410
5	3000	5000	83.8826	2.635
6	3000	5000	83.1028	2.113
7	3000	5000	82.2657	2.445
8	3000	5000	81.5367	2.330
9	3000	5000	81.0375	2.213
10	3000	5000	81.5394	2.007
11	3000	10000	80.9979	8.933
12	3000	10000	81.3835	8.624
13	3000	10000	81.5202	8.094
14	3000	10000	83.7333	8.466
15	3000	10000	82.5501	8.635
16	3000	10000	81.8824	8.654
17	3000	10000	83.9942	8.325
18	3000	10000	82.8345	8.099
19	3000	10000	82.5007	8.546
20	3000	10000	82.6306	8.442
21	3000	15000	81.7000	19.049
22	3000	15000	83.1144	19.326

Essais	Valeur de la solution initiale $s_0$	Nombre aléatoire d'itérations	Valeur de la température médiane $T_0$	Temps (s) de simulation
23	3000	15000	82.7605	19.058
24	3000	15000	82.1141	19.747
25	3000	15000	82.9888	19.535
26	3000	15000	83.0005	19.663
27	3000	15000	80.9646	19.044
28	3000	15000	81.3925	19.443
29	3000	15000	82.7348	19.856
30	3000	15000	81.3239	19.443
31	3000	20000	83.9771	35.097
32	3000	20000	80.6946	35.732
33	3000	20000	82.6601	34.190
34	3000	20000	83.0062	35.221
35	3000	20000	81.4502	35.443
36	3000	20000	82.7209	35.929
37	3000	20000	81.4402	34.007
38	3000	20000	82.1103	35.348
39	3000	20000	81.9907	35.146
40	3000	20000	83.0066	34.955

Tableau 6

La variation de la température médiane  $T_0$  en fonction de la solution initiale  $s_0$

Essais	Valeur de la solution initiale $s_0$	Nombre aléatoire d'itérations	Valeur de la température médiane $T_0$	Temps (s) de simulation
1	2000	5000	52.8195	2.401
2	2000	5000	55.5829	2.056
3	2000	5000	52.4583	2.116
4	2000	5000	53.5818	2.196
5	2000	5000	52.4472	2.225
6	2000	5000	53.2259	2.663
7	2000	5000	53.1147	2.306
8	2000	5000	52.4401	2.540
9	2000	5000	53.6306	2.326
10	2000	5000	53.1944	2.610

Essais	Valeur de la solution initiale $s_0$	Nombre aléatoire d'itérations	Valeur de la température médiane $T_0$	Temps (s) de simulation
11	2000	10000	53.5632	8.309
12	2000	10000	53.1140	8.255
13	2000	10000	53.7702	8.401
14	2000	10000	52.4319	8.358
15	2000	10000	52.9990	8.199
16	2000	10000	53.8104	8.035
17	2000	10000	52.1149	8.226
18	2000	10000	53.5771	8.453
19	2000	10000	52.9556	8.164
20	2000	10000	53.4402	8.357
21	2000	15000	53.5517	19.433
22	2000	15000	53.4488	19.068
23	2000	15000	52.3702	19.266
24	2000	15000	52.9390	19.803
25	2000	15000	53.3447	19.038
26	2000	15000	53.6978	19.547
27	2000	15000	52.4425	19.358
28	2000	15000	53.4564	19.669
29	2000	15000	53.9766	19.079
30	2000	15000	52.8845	19.353
31	2000	20000	52.4372	34.988
32	2000	20000	52.1984	35.422
33	2000	20000	53.4771	34.920
34	2000	20000	52.5655	35.001
35	2000	20000	52.5688	35.333
36	2000	20000	53.4646	34.602
37	2000	20000	52.9741	34.117
38	2000	20000	53.0026	35.833
39	2000	20000	53.4681	35.221
40	2000	20000	52.2520	34.883

Tableau 7

La variation de la température médiane  $T_0$  en fonction de la solution initiale  $s_0$

Essais	Valeur de la solution initiale $s_0$	Nombre aléatoire d'itérations	Valeur de la température médiane $T_0$	Temps (s) de simulation
1	1200	5000	31.2711	2.024
2	1200	5000	32.3231	2.036
3	1200	5000	32.2114	2.164
4	1200	5000	32.3003	2.229
5	1200	5000	32.0073	2.005
6	1200	5000	31.5525	2.212
7	1200	5000	31.9832	2.265
8	1200	5000	31.4772	2.552
9	1200	5000	32.4564	2.005
10	1200	5000	31.5505	2.110
11	1200	10000	32.0022	8.447
12	1200	10000	32.3546	8.233
13	1200	10000	31.4421	8.109
14	1200	10000	31.1173	8.580
15	1200	10000	31.0035	8.446
16	1200	10000	32.3314	8.356
17	1200	10000	32.2334	8.225
18	1200	10000	31.2145	8.441
19	1200	10000	31.0788	8.009
20	1200	10000	32.1127	8.116
21	1200	15000	31.8832	17.993
22	1200	15000	31.7105	18.401
23	1200	15000	31.6943	18.337
24	1200	15000	32.3771	18.442
25	1200	15000	32.1905	18.205
26	1200	15000	31.2718	18.003
27	1200	15000	31.4290	18.449
28	1200	15000	32.6606	18.277
29	1200	15000	32.4004	17.992
30	1200	15000	32.6638	18.336
31	1200	20000	31.9532	34.233
32	1200	20000	31.4368	34.576
33	1200	20000	32.6640	34.336
34	1200	20000	32.6631	34.449
35	1200	20000	32.5638	34.400

Essais	Valeur de la solution initiale $s_0$	Nombre aléatoire d'itérations	Valeur de la température médiane $T_0$	Temps (s) de simulation
36	1200	20000	32.5536	34.033
37	1200	20000	31.7736	34.221
38	1200	20000	31.5363	34.423
39	1200	20000	32.6336	34.443
40	1200	20000	32.0039	34.219

Tableau 8

La variation de la température médiane  $T_0$  en fonction de la solution initiale  $s_0$

Essais	Valeur de la solution initiale $s_0$	Nombre aléatoire d'itérations	Valeur de la température médiane $T_0$	Temps (s) de simulation
1	600	5000	16.5068	2.505
2	600	5000	16.3310	2.110
3	600	5000	16.2644	2.139
4	600	5000	16.2267	2.403
5	600	5000	16.1733	2.227
6	600	5000	16.3221	2.393
7	600	5000	16.1141	2.118
8	600	5000	16.8233	2.372
9	600	5000	16.1899	2.330
10	600	5000	16.1844	2.117
11	600	10000	16.1865	8.205
12	600	10000	16.3360	8.336
13	600	10000	16.2466	8.441
14	600	10000	16.1689	8.058
15	600	10000	16.3523	8.254
16	600	10000	16.2779	8.225
17	600	10000	16.1844	8.004
18	600	10000	16.2201	8.035
19	600	10000	16.2732	8.278
20	600	10000	16.1633	8.361
21	600	15000	16.5211	18.986
22	600	15000	16.4409	18.445

Essais	Valeur de la solution initiale $s_0$	Nombre aléatoire d'itérations	Valeur de la température médiane $T_0$	Temps (s) de simulation
23	600	15000	16.4003	18.336
24	600	15000	16.2314	18.211
25	600	15000	16.5709	18.403
26	600	15000	16.3342	18.335
27	600	15000	16.1677	18.032
28	600	15000	16.3353	18.477
29	600	15000	16.3545	18.662
30	600	15000	16.2218	18.474
31	600	20000	16.3534	34.546
32	600	20000	16.3435	34.003
33	600	20000	16.3886	34.221
34	600	20000	16.0944	34.440
35	600	20000	16.2623	34.231
36	600	20000	16.1947	34.337
37	600	20000	16.5123	34.045
38	600	20000	16.1933	34.376
39	600	20000	16.3001	34.422
40	600	20000	16.2258	34.065

Tableau 9

La variation de la température médiane  $T_0$  en fonction de la solution initiale  $s_0$ .

Essais	Valeur de la solution initiale $s_0$	Nombre aléatoire d'itérations	Valeur de la température médiane $T_0$	Temps (s) de simulation
1	150	5000	3.9433	2.051
2	150	5000	4.0308	2.134
3	150	5000	4.0192	2.220
4	150	5000	4.0224	2.034
5	150	5000	4.0357	2.056
6	150	5000	4.0024	2.052
7	150	5000	4.0532	2.098
8	150	5000	3.9963	2.116
9	150	5000	4.0455	2.041
10	150	5000	4.0038	2.009

Essais	Valeur de la solution initiale $s_0$	Nombre aléatoire d'itérations	Valeur de la température médiane $T_0$	Temps (s) de simulation
11	150	10000	4.1462	8.146
12	150	10000	4.1271	8.156
13	150	10000	4.0998	8.222
14	150	10000	4.1103	8.166
15	150	10000	4.0262	8.055
16	150	10000	4.1771	8.119
17	150	10000	4.0489	8.208
18	150	10000	4.1337	8.117
19	150	10000	4.0552	8.106
20	150	10000	4.0771	8.200
21	150	15000	4.1124	18.796
22	150	15000	3.9711	18.546
23	150	15000	4.1726	18.335
24	150	15000	4.1132	18.756
25	150	15000	4.2445	18.214
26	150	15000	4.1757	18.338
27	150	15000	4.0352	18.233
28	150	15000	4.0461	18.684
29	150	15000	3.9922	18.246
30	150	15000	4.1444	18.309
31	150	20000	4.0006	34.571
32	150	20000	4.1182	34.745
33	150	20000	4.2868	34.356
34	150	20000	4.0034	34.343
35	150	20000	3.9981	34.757
36	150	20000	4.0422	34.355
37	150	20000	4.2666	34.245
38	150	20000	4.1773	34.606
39	150	20000	4.0552	34.523
40	150	20000	4.0565	34.225

Tableau 10

La variation de la température médiane  $T_0$  en fonction de la solution initiale  $s_0$ 

Essais	Valeur de la solution initiale $s_0$	Nombre aléatoire d'itérations	Valeur de la température médiane $T_0$	Temps (s) de simulation
1	75	5000	2.0020	2.147
2	75	5000	1.9817	2.287
3	75	5000	2.0844	2.265
4	75	5000	2.0876	2.110
5	75	5000	2.0210	2.004
6	75	5000	1.9932	2.244
7	75	5000	2.0568	2.321
8	75	5000	2.0269	2.001
9	75	5000	2.0458	2.056
10	75	5000	2.0192	2.147
11	75	10000	1.9863	8.154
12	75	10000	2.0283	8.222
13	75	10000	2.1074	8.177
14	75	10000	2.0661	8.225
15	75	10000	2.0390	8.113
16	75	10000	1.9938	8.018
17	75	10000	2.1102	8.229
18	75	10000	2.0044	8.117
19	75	10000	2.0885	8.024
20	75	10000	2.0972	8.019
21	75	15000	2.0321	19.033
22	75	15000	2.0603	19.144
23	75	15000	2.1124	19.037
24	75	15000	2.0590	19.068
25	75	15000	2.0357	19.244
26	75	15000	2.1873	19.100
27	75	15000	2.0444	19.044
28	75	15000	2.0201	19.552
29	75	15000	2.0602	19.411
30	75	15000	2.1910	19.089
31	75	20000	2.0464	34.853
32	75	20000	2.0503	34.980
33	75	20000	2.0324	34.979
34	75	20000	2.1256	34.235
35	75	20000	2.0632	34.679

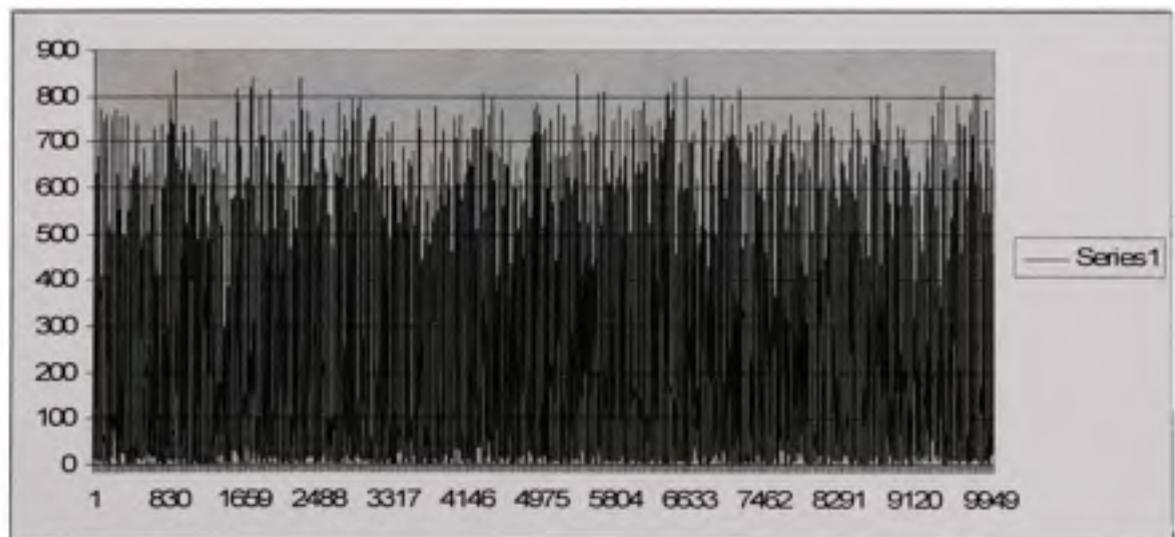
Essais	Valeur de la solution initiale $s_0$	Nombre aléatoire d'itérations	Valeur de la température médiane $T_0$	Temps (s) de simulation
36	75	20000	2.0122	34.906
37	75	20000	2.1100	34.558
38	75	20000	2.0441	34.671
39	75	20000	2.0354	34.883
40	75	20000	2.0155	34.457

Tableau 11

La variation de la température médiane  $T_0$  en fonction de la solution initiale  $s_0$

Essais	Valeur de la solution initiale $s_0$	Nombre aléatoire d'itérations	Valeur de la température médiane $T_0$	Temps (s) de simulation
1	30	5000	0.8015	2.179
2	30	5000	0.8154	2.092
3	30	5000	0.8290	2.003
4	30	5000	0.8115	2.110
5	30	5000	0.8041	2.022
6	30	5000	0.8220	2.015
7	30	5000	0.8321	2.225
8	30	5000	0.8016	2.098
9	30	5000	0.8023	2.004
10	30	5000	0.8157	2.056
11	30	10000	0.8209	8.225
12	30	10000	0.8268	8.210
13	30	10000	0.8175	8.341
14	30	10000	0.8243	8.116
15	30	10000	0.8046	8.003
16	30	10000	0.8021	8.115
17	30	10000	0.8009	8.336
18	30	10000	0.8290	8.207
19	30	10000	0.8292	8.322
20	30	10000	0.8173	8.003
21	30	15000	0.7983	18.827
22	30	15000	0.8014	18.661

Essais	Valeur de la solution initiale $s_0$	Nombre aléatoire d'itérations	Valeur de la température médiane $T_0$	Temps (s) de simulation
23	30	15000	0.8003	18.900
24	30	15000	0.7996	18.522
25	30	15000	0.8105	18.504
26	30	15000	0.8116	18.397
27	30	15000	0.7999	18.478
28	30	15000	0.8002	18.336
29	30	15000	0.8112	18.756
30	30	15000	0.8228	18.235
31	30	20000	0.7991	34.645
32	30	20000	0.7986	34.249
33	30	20000	0.7991	34.157
34	30	20000	0.7983	34.557
35	30	20000	0.7901	34.267
36	30	20000	0.8111	34.648
37	30	20000	0.8003	34.653
38	30	20000	0.7964	34.110
39	30	20000	0.8001	34.633
40	30	20000	0.7988	34.443



**Figure 1** La distribution aléatoire de la température pour la valeur de la solution égale à 600 unités.

**ANNEXE II****SYNTHÈSE DES RÉSULTATS**

Cette annexe résume les simulations qui ont été réalisés, pour les algorithmes étudiés. Les Tableaux 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18 et 19, présentes les résultats trouvés pour les instances 40PE (réseau Côte Saint-Luc) et PR1002 (Padberg/Rinaldi). La Figure 1 montre les résultats pour le problème PR1002. Le Tableau 20 montre les résultats finaux pour les problèmes étudiés.

Tableau 1

Résultats obtenus pour l'algorithme de recuit simulé avec une amélioration 2, 3 et 4-opt  
(choix de température égale à 100 unités)

Nombre simulations	Instance	R S 2-opt	R S 3-opt	R S 4-opt	Borne supérieure	T	TR	Itérations
1	40PE	70.4563	76.0348	77.9767	23.2203	100	0.978	2000
2	40PE	80.2748	70.0326	71.6262	23.2203	100	0.978	2000
3	40PE	85.8936	82.1644	78.0021	23.2203	100	0.978	2000
4	40PE	72.9836	78.9937	81.6355	23.2203	100	0.978	2000
5	40PE	76.7940	72.6645	77.9926	23.2203	100	0.978	2000
6	40PE	73.1423	81.6635	72.8835	23.2203	100	0.978	2000
7	40PE	76.0035	82.1402	71.5531	23.2203	100	0.978	2000
8	40PE	72.7357	73.0042	78.6634	23.2203	100	0.978	2000
9	40PE	77.9935	78.1901	72.1172	23.2203	100	0.978	2000
10	40PE	76.2318	79.2333	81.8825	23.2203	100	0.978	2000
11	40PE	72.1102	76.6620	73.2835	23.2203	100	0.978	2000
12	40PE	72.6291	78.0025	73.8823	23.2203	100	0.978	2000
13	40PE	71.8825	74.9901	74.0244	23.2203	100	0.978	2000
14	40PE	74.0034	72.9201	72.2254	23.2203	100	0.978	2000
15	40PE	84.9231	71.1644	77.7738	23.2203	100	0.978	2000
16	40PE	80.7790	75.0602	72.1128	23.2203	100	0.978	2000
17	40PE	78.1640	76.3556	79.0251	23.2203	100	0.978	2000
18	40PE	70.0648	73.6494	80.9923	23.2203	100	0.978	2000
19	40PE	74.0244	72.9903	76.1631	23.2203	100	0.978	2000
20	40PE	74.8393	70.9143	78.2394	23.2203	100	0.978	2000
MV	40PE	70.0648	70.0326	71.5531	23.2203	100	0.978	2000
PV	40PE	85.8936	82.1644	81.8825	23.2203	100	0.978	2000
VM	40PE	75.7964	75.4915	76.1027	23.2203	100	0.978	2000

**Notations :**

RS 2-opt algorithme de recuit simulé avec la méthode d'amélioration 2-opt

RS 3-opt algorithme de recuit simulé avec la méthode d'amélioration 3-opt

RS 4-opt algorithme de recuit simulé avec la méthode d'amélioration 4-opt

MV meilleur valeur

PV pire valeur

VM valeur moyenne

Tableau 2

Résultats obtenus pour l'algorithme de recuit simulé avec une amélioration 2, 3 et 4-opt  
(choix de température égale à 1.93 unités)

Nombre simulations	Instance	R S 2-opt	R S 3-opt	R S 4-opt	Borne supérieure	T	TR	Itérations
1	40PE	22.9773	23.1501	23.9401	23.2203	1.93	0.978	2000
2	40PE	23.1124	23.3631	22.9843	23.2203	1.93	0.978	2000
3	40PE	23.054	23.875	22.8119	23.2203	1.93	0.978	2000
4	40PE	23.3446	23.6088	23.747	23.2203	1.93	0.978	2000
5	40PE	23.4303	23.4176	23.9142	23.2203	1.93	0.978	2000
6	40PE	23.0765	23.0088	23.9446	23.2203	1.93	0.978	2000
7	40PE	22.9862	22.2119	23.0169	23.2203	1.93	0.978	2000
8	40PE	22.498	22.8864	23.9981	23.2203	1.93	0.978	2000
9	40PE	22.064	23.2516	23.054	23.2203	1.93	0.978	2000
10	40PE	24.0998	23.0998	23.1902	23.2203	1.93	0.978	2000
11	40PE	23.2343	23.3264	24.3342	23.2203	1.93	0.978	2000
12	40PE	22.0089	23.9142	23.9406	23.2203	1.93	0.978	2000
13	40PE	23.1624	23.2242	24.0106	23.2203	1.93	0.978	2000
14	40PE	23.2316	23.0688	23.054	23.2203	1.93	0.978	2000
15	40PE	23.0005	22.9149	23.4176	23.2203	1.93	0.978	2000
16	40PE	23.1409	23.0564	23.656	23.2203	1.93	0.978	2000
17	40PE	24.9004	23.2314	23.3522	23.2203	1.93	0.978	2000
18	40PE	23.3014	23.9849	23.0668	23.2203	1.93	0.978	2000
19	40PE	23.5573	22.634	23.5254	23.2203	1.93	0.978	2000
20	40PE	22.9957	22.2254	23.5682	23.2203	1.93	0.978	2000
MV	40PE	22.0089	22.2119	22.8119	23.2203	1.93	0.978	2000
PV	40PE	24.9004	23.9849	24.3342	23.2203	1.93	0.978	2000
VM	40PE	23.1588	23.1726	23.5263	23.2203	1.93	0.978	2000

**Notations :**

RS 2-opt algorithme de recuit simulé avec la méthode d'amélioration 2-opt

RS 3-opt algorithme de recuit simulé avec la méthode d'amélioration 3-opt

RS 4-opt algorithme de recuit simulé avec la méthode d'amélioration 4-opt

MV meilleur valeur

PV pire valeur

VM valeur moyenne

Tableau 3

Écart type obtenu pour les résultats du Tableau 2.2

Nombre simulations	Instance	RS 2-opt Écart-type [%]	RS 3-opt Écart-type [%]	RS 4-opt Écart-type [%]	T	TR	Itérations
1	40PE	-1.05%	-0.30%	3.10%	1.93	0.978	2000
2	40PE	-0.46%	0.61%	-1.02%	1.93	0.978	2000
3	40PE	-0.72%	2.82%	-1.76%	1.93	0.978	2000
4	40PE	0.54%	1.67%	2.27%	1.93	0.978	2000
5	40PE	0.90%	0.85%	2.99%	1.93	0.978	2000
6	40PE	-0.62%	-0.91%	3.12%	1.93	0.978	2000
7	40PE	-1.01%	-4.34%	-0.88%	1.93	0.978	2000
8	40PE	-3.11%	-1.44%	3.35%	1.93	0.978	2000
9	40PE	-4.98%	0.13%	-0.72%	1.93	0.978	2000
10	40PE	3.79%	-0.52%	-0.13%	1.93	0.978	2000
11	40PE	0.06%	0.46%	4.80%	1.93	0.978	2000
12	40PE	-5.22%	2.99%	3.10%	1.93	0.978	2000
13	40PE	-0.25%	0.02%	3.40%	1.93	0.978	2000
14	40PE	0.05%	-0.65%	-0.72%	1.93	0.978	2000
15	40PE	-0.95%	-1.32%	0.85%	1.93	0.978	2000
16	40PE	-0.34%	-0.71%	1.88%	1.93	0.978	2000
17	40PE	7.24%	0.05%	0.57%	1.93	0.978	2000
18	40PE	0.35%	3.29%	-0.66%	1.93	0.978	2000
19	40PE	1.45%	-2.52%	1.31%	1.93	0.978	2000
20	40PE	-0.97%	-4.28%	1.50%	1.93	0.978	2000
MEt	40PE	-5.22%	-4.34%	-1.76%	1.93	0.978	2000
PEt	40PE	7.24%	3.29%	4.80%	1.93	0.978	2000
VMEt	40PE	-0.26%	-0.21%	1.32%	1.93	0.978	2000

**Notations :**

RS 2-opt algorithme de recuit simulé avec la méthode d'amélioration 2-opt

RS 3-opt algorithme de recuit simulé avec la méthode d'amélioration 3-opt

RS 4-opt algorithme de recuit simulé avec la méthode d'amélioration 4-opt

MEt meilleure valeur écart-type

PEt pire valeur écart-type

VMEt valeur moyenne écart-type

Tableau 4

Temps de simulation pour l'algorithme de recuit simulé avec une amélioration 2, 3 et 4-opt

Nombre simulations	Instance	Temps simulations	Temps simulations	Temps simulations	T	TR	Itérations
		[s] RS 2-opt	[s] RS 3-opt	[s] RS 4-opt			
1	40PE	9.0027	8.6395	9.6394	1.93	0.978	2000
2	40PE	9.4299	8.4288	9.2186	1.93	0.978	2000
3	40PE	8.3945	9.6388	9.3775	1.93	0.978	2000
4	40PE	8.7736	9.0037	9.5493	1.93	0.978	2000
5	40PE	8.4922	9.5384	8.5399	1.93	0.978	2000
6	40PE	9.0036	8.5488	8.3384	1.93	0.978	2000
7	40PE	8.1865	8.0372	8.3299	1.93	0.978	2000
8	40PE	8.2888	8.3295	8.4549	1.93	0.978	2000
9	40PE	8.3898	8.0373	8.9364	1.93	0.978	2000
10	40PE	8.2286	8.0038	9.0438	1.93	0.978	2000
11	40PE	8.9905	9.3395	9.8395	1.93	0.978	2000
12	40PE	7.9937	9.2294	9.5483	1.93	0.978	2000
13	40PE	8.0025	9.4394	8.5434	1.93	0.978	2000
14	40PE	8.4429	8.0017	8.3394	1.93	0.978	2000
15	40PE	8.4210	8.3933	8.3200	1.93	0.978	2000
16	40PE	9.4032	8.0388	8.4003	1.93	0.978	2000
17	40PE	9.5337	8.5493	8.3302	1.93	0.978	2000
18	40PE	9.3299	9.0266	8.0294	1.93	0.978	2000
19	40PE	8.3208	8.4393	8.9937	1.93	0.978	2000
20	40PE	7.9326	9.0372	9.4438	1.93	0.978	2000
MV	40PE	7.9326	8.0017	8.0294	1.93	0.978	2000
PV	40PE	9.4299	9.6388	9.8395	1.93	0.978	2000
VM	40PE	8.6280	8.6850	8.8608	1.93	0.978	2000

**Notations :**

RS 2-opt algorithme de recuit simulé avec la méthode d'amélioration 2-opt

RS 3-opt algorithme de recuit simulé avec la méthode d'amélioration 3-opt

RS 4-opt algorithme de recuit simulé avec la méthode d'amélioration 4-opt

MV meilleur valeur

PV pire valeur ; VM valeur moyenne;

Tableau 5

Résultats obtenus pour l'algorithme génétique avec une amélioration 2, 3 et 4-opt

Nombre simulations	Instance	Algorithme Génétique 2-opt	Algorithme Génétique 3-opt	Algorithme Génétique 4-opt	Borne supérieure	Taux mutation	Itérations
1	40PE	23.3065	23.4471	23.1641	23.2203	0.1	2000
2	40PE	23.2115	23.0171	23.6529	23.2203	0.1	2000
3	40PE	22.9808	23.4538	22.9545	23.2203	0.1	2000
4	40PE	23.1641	23.0853	23.1992	23.2203	0.1	2000
5	40PE	23.4471	23.7104	23.0171	23.2203	0.1	2000
6	40PE	23.4121	23.1641	23.3065	23.2203	0.1	2000
7	40PE	22.9545	23.3065	23.3065	23.2203	0.1	2000
8	40PE	23.2508	22.9466	23.8288	23.2203	0.1	2000
9	40PE	23.3065	23.0033	23.1641	23.2203	0.1	2000
10	40PE	23.2508	23.0171	23.2757	23.2203	0.1	2000
11	40PE	23.4471	23.0533	23.8119	23.2203	0.1	2000
12	40PE	23.3065	23.3065	23.0171	23.2203	0.1	2000
13	40PE	23.2115	23.0438	23.3065	23.2203	0.1	2000
14	40PE	22.9545	23.3065	22.9466	23.2203	0.1	2000
15	40PE	22.9545	22.9545	23.2115	23.2203	0.1	2000
16	40PE	22.9466	23.0171	23.4471	23.2203	0.1	2000
17	40PE	23.4471	23.3065	23.1641	23.2203	0.1	2000
18	40PE	23.2508	23.2256	23.2115	23.2203	0.1	2000
19	40PE	23.3065	23.1641	23.0171	23.2203	0.1	2000
20	40PE	23.1641	22.9545	23.3065	23.2203	0.1	2000
MV	40PE	22.9466	22.9466	22.9466	23.2203	0.1	2000
PV	40PE	23.4471	23.7104	23.8288	23.2203	0.1	2000
VM	40PE	23.2137	23.1741	23.2654	23.2203	0.1	2000

**Notations :**

MV meilleur valeur

PV pire valeur

VM valeur moyenne

Tableau 6

Écart-type obtenu pour les résultats du Tableau 2.5

Nombre simulations	Instance	AG 2-Opt Écart-type [%]	AG 3-Opt Écart-type [%]	AG 4-Opt Écart-type [%]	TM	Itérations
1	40PE	0.371%	0.977%	-0.242%	0.1	2000
2	40PE	-0.038%	-0.875%	1.863%	0.1	2000
3	40PE	-1.031%	1.006%	-1.145%	0.1	2000
4	40PE	-0.242%	-0.581%	-0.091%	0.1	2000
5	40PE	0.977%	2.111%	-0.875%	0.1	2000
6	40PE	0.826%	-0.242%	0.371%	0.1	2000
7	40PE	-1.145%	0.371%	0.371%	0.1	2000
8	40PE	0.131%	-1.179%	2.621%	0.1	2000
9	40PE	0.371%	-0.935%	-0.242%	0.1	2000
10	40PE	0.131%	-0.875%	0.239%	0.1	2000
11	40PE	0.977%	-0.719%	2.548%	0.1	2000
12	40PE	0.371%	0.371%	-0.875%	0.1	2000
13	40PE	-0.038%	-0.760%	0.371%	0.1	2000
14	40PE	-1.145%	0.371%	-1.179%	0.1	2000
15	40PE	-1.145%	-1.145%	-0.038%	0.1	2000
16	40PE	-1.179%	-0.875%	0.977%	0.1	2000
17	40PE	0.977%	0.371%	-0.242%	0.1	2000
18	40PE	0.131%	0.023%	-0.038%	0.1	2000
19	40PE	0.371%	-0.242%	-0.875%	0.1	2000
20	40PE	-0.242%	-1.145%	0.371%	0.1	2000
MEt	40PE	-1.179%	-1.179%	-1.179%	0.1	2000
PEt	40PE	0.977%	2.864%	3.155%	0.1	2000
VMEt	40PE	-0.039%	-0.114%	0.293%	0.1	2000

**Notations :**

AG 2-opt    algorithme génétique avec la méthode d'amélioration 2-opt

AG 3-opt    algorithme génétique avec la méthode d'amélioration 3-opt

AG 4-opt    algorithme génétique avec la méthode d'amélioration 4-opt

TM            taux de mutation

MEt           meilleure valeur écart type

PEt           pire valeur écart type ; VMEt    valeur moyenne écart type;

Tableau 7

Temps de simulation pour l'algorithme génétique avec une amélioration 2, 3 et 4-opt

Nombre simulations	Instance	Temps simulations	Temps simulations	Temps simulations	Taux mutation	Itérations
		[s] AG 2-opt	[s] AG 3-opt	[s] AG 4-opt		
1	40PE	126.3382	128.2734	128.9438	0.1	2000
2	40PE	127.9360	128.9942	128.3373	0.1	2000
3	40PE	126.2774	128.3611	129.4733	0.1	2000
4	40PE	126.3758	129.4825	131.8354	0.1	2000
5	40PE	128.9337	129.6623	130.0037	0.1	2000
6	40PE	128.0367	129.6010	130.4722	0.1	2000
7	40PE	129.0015	128.8843	128.9554	0.1	2000
8	40PE	126.9388	128.9327	129.0554	0.1	2000
9	40PE	127.6283	128.3393	130.6465	0.1	2000
10	40PE	128.0033	128.4288	130.2214	0.1	2000
11	40PE	127.2734	129.3885	130.6433	0.1	2000
12	40PE	127.2282	130.5528	131.6534	0.1	2000
13	40PE	128.9022	128.3329	129.0043	0.1	2000
14	40PE	128.2747	127.9981	129.4366	0.1	2000
15	40PE	128.1141	128.5204	128.5535	0.1	2000
16	40PE	129.3002	128.0438	128.4426	0.1	2000
17	40PE	129.2733	128.5537	129.4436	0.1	2000
18	40PE	128.0437	129.3261	129.4437	0.1	2000
19	40PE	129.3362	128.4437	130.4425	0.1	2000
20	40PE	128.3827	128.9377	130.5994	0.1	2000
MV	40PE	126.3758	127.9981	128.3373	0.1	2000
PV	40PE	129.3362	130.5528	131.8354	0.1	2000
VM	40PE	127.9799	128.8529	129.7804	0.1	2000

**Notation :**

AG 2-opt    algorithme génétique avec la méthode d'amélioration 2-opt

AG 3-opt    algorithme génétique avec la méthode d'amélioration 3-opt

AG 4-opt    algorithme génétique avec la méthode d'amélioration 4-opt

TM            taux de mutation

MV            meilleure valeur

Tableau 8

Résultats obtenus pour l'algorithme de fourmis avec une amélioration 2, 3 et 4-opt

Nombre simulations	Instance	Algorithme Fourmi 2 - Opt	Algorithme Fourmi 3 - Opt	Algorithme Fourmi 4 - Opt	Borne supérieure	Itérations
1	40PE	22.9903	22.9089	22.9903	23.2203	2000
2	40PE	22.9903	22.9089	22.9903	23.2203	2000
3	40PE	22.9903	22.9089	22.9903	23.2203	2000
4	40PE	22.9089	22.9089	22.9903	23.2203	2000
5	40PE	22.9903	22.9089	22.9903	23.2203	2000
6	40PE	22.9089	22.9089	22.9903	23.2203	2000
7	40PE	22.9089	22.9089	22.9089	23.2203	2000
8	40PE	22.9903	22.9903	22.9903	23.2203	2000
9	40PE	22.9089	22.9089	22.9903	23.2203	2000
10	40PE	22.9089	22.9089	22.9903	23.2203	2000
11	40PE	22.9089	22.9089	22.9089	23.2203	2000
12	40PE	22.9089	22.9089	22.9903	23.2203	2000
13	40PE	22.9089	22.9089	22.9903	23.2203	2000
14	40PE	22.9089	22.9903	22.9903	23.2203	2000
15	40PE	22.9903	22.9903	22.9089	23.2203	2000
16	40PE	22.9089	22.9089	22.9903	23.2203	2000
17	40PE	22.9089	22.9089	22.9903	23.2203	2000
18	40PE	22.9903	22.9089	22.9089	23.2203	2000
19	40PE	22.9903	22.9089	22.9903	23.2203	2000
20	40PE	22.9903	22.9089	22.9903	23.2203	2000
MV	40PE	22.9089	22.9089	22.9089	23.2203	2000
PV	40PE	22.9903	22.9903	22.9903	23.2203	2000
VM	40PE	22.9455	22.9211	22.9740	23.2203	2000

**Notation :**

TM       taux de mutation

MV       meilleure valeur

PV       pire valeur

VM       valeur moyenne

Tableau 9

Écart type obtenu pour les résultats du Tableau 2.8

Nombre simulations	Instance	AF 2-Opt Écart-type [%]	AF 3-Opt Écart-type [%]	AF 4-Opt Écart-type [%]	Itérations
1	40PE	-0.991%	-1.341%	-0.991%	2000
2	40PE	-0.991%	-1.341%	-0.991%	2000
3	40PE	-0.991%	-1.341%	-0.991%	2000
4	40PE	-1.341%	-1.341%	-0.991%	2000
5	40PE	-0.991%	-1.341%	-0.991%	2000
6	40PE	-1.341%	-1.341%	-0.991%	2000
7	40PE	-1.341%	-1.341%	-1.341%	2000
8	40PE	-0.991%	-0.991%	-0.991%	2000
9	40PE	-1.341%	-1.341%	-0.991%	2000
10	40PE	-1.341%	-1.341%	-0.991%	2000
11	40PE	-1.341%	-1.341%	-1.341%	2000
12	40PE	-1.341%	-1.341%	-0.991%	2000
13	40PE	-1.341%	-1.341%	-0.991%	2000
14	40PE	-1.341%	-0.991%	-0.991%	2000
15	40PE	-0.991%	-0.991%	-1.341%	2000
16	40PE	-1.341%	-1.341%	-0.991%	2000
17	40PE	-1.341%	-1.341%	-0.991%	2000
18	40PE	-0.991%	-1.341%	-1.341%	2000
19	40PE	-0.991%	-1.341%	-0.991%	2000
20	40PE	-0.991%	-1.341%	-0.991%	2000
MEt	40PE	-1.341%	-1.341%	-1.341%	2000
PEt	40PE	-0.991%	-0.991%	-0.991%	2000
VMEt	40PE	-1.183%	-1.288%	-1.061%	2000

**Notation :**

AF 2-opt    algorithme de fourmis avec la méthode d'amélioration 2-opt

AF 3-opt    algorithme de fourmis avec la méthode d'amélioration 3-opt

AF 4-opt    algorithme de fourmis avec la méthode d'amélioration 4-opt

MEt        meilleure valeur écart type

PEt        pire valeur écart type

VMEt      valeur moyenne écart type

Tableau 10

Temps de simulation pour l'algorithme de fourmi Max Min  
avec une amélioration 2, 3 et 4-opt

Nombre simulations	Instance	Temps simulations	Temps simulations	Temps simulations	Itérations
		[s] AF 2-opt	[s] AF 3-opt	[s] AF 4-opt	
1	40PE	227.9327	227.4388	228.9433	2000
2	40PE	225.2884	227.4889	229.0377	2000
3	40PE	226.6377	227.9266	229.4822	2000
4	40PE	227.7277	227.0266	229.4199	2000
5	40PE	227.4830	228.3721	229.4933	2000
6	40PE	226.3394	228.4866	229.6932	2000
7	40PE	227.3844	229.0277	228.7428	2000
8	40PE	226.5374	227.8557	229.6399	2000
9	40PE	228.2251	227.8955	229.3866	2000
10	40PE	227.0363	228.6433	229.9447	2000
11	40PE	226.7284	228.9668	230.3885	2000
12	40PE	227.0011	227.3755	229.4933	2000
13	40PE	225.9433	227.0944	230.4993	2000
14	40PE	227.9137	228.9656	230.5994	2000
15	40PE	227.0443	227.9337	229.5994	2000
16	40PE	226.8266	228.4485	229.5844	2000
17	40PE	226.3295	228.5044	228.6845	2000
18	40PE	226.0437	228.3833	229.0336	2000
19	40PE	227.7329	228.1044	230.4711	2000
20	40PE	227.3295	227.4833	229.4482	2000
MV	40PE	225.2884	227.0266	228.6845	2000
PV	40PE	228.2251	228.9668	230.5994	2000
VM	40PE	226.9743	228.0711	229.5793	2000

**Notation :**

AF 2-opt algorithme de fourmis avec la méthode d'amélioration 2-opt

AF 3-opt algorithme de fourmis avec la méthode d'amélioration 3-opt

AF 4-opt algorithme de fourmis avec la méthode d'amélioration 4-opt

MV meilleure valeur

PV pire valeur

Tableau 11

Comportement des algorithmes – problème du voyageur de commerce 40PE

NS	Instance	RS 2-opt	RS 3-opt	RS 4-opt	AG 2-opt	AG 3-opt	AG 4-opt	AF 2-opt	AF 3-opt	AF 4-opt	Borne supérieure
1	40PE	22.9773	23.1501	23.9401	23.3065	23.4471	23.1641	22.9903	22.9089	22.9903	23.2203
2	40PE	23.1124	23.3631	22.9843	23.2115	23.0171	23.6529	22.9903	22.9089	22.9903	23.2203
3	40PE	23.054	23.875	22.8119	22.9808	23.4538	22.9545	22.9903	22.9089	22.9903	23.2203
4	40PE	23.3446	23.6088	23.747	23.1641	23.0853	23.1992	22.9089	22.9089	22.9903	23.2203
5	40PE	23.4303	23.4176	23.9142	23.4471	23.7104	23.0171	22.9903	22.9089	22.9903	23.2203
6	40PE	23.0765	23.0088	23.9446	23.4121	23.1641	23.3065	22.9089	22.9089	22.9903	23.2203
7	40PE	22.9862	22.2119	23.0169	22.9545	23.3065	23.3065	22.9089	22.9089	22.9089	23.2203
8	40PE	22.498	22.8864	23.9981	23.2508	22.9466	23.8288	22.9903	22.9903	22.9903	23.2203
9	40PE	22.064	23.2516	23.054	23.3065	23.0033	23.1641	22.9089	22.9089	22.9903	23.2203
10	40PE	24.0998	23.0998	23.1902	23.2508	23.0171	23.2757	22.9089	22.9089	22.9903	23.2203
11	40PE	23.2343	23.3264	24.3342	23.4471	23.0533	23.8119	22.9089	22.9089	22.9089	23.2203
12	40PE	22.0089	23.9142	23.9406	23.3065	23.3065	23.0171	22.9089	22.9089	22.9903	23.2203
13	40PE	23.1624	23.2242	24.0106	23.2115	23.0438	23.3065	22.9089	22.9089	22.9903	23.2203
14	40PE	23.2316	23.0688	23.054	22.9545	23.3065	22.9466	22.9089	22.9903	22.9903	23.2203
15	40PE	23.0005	22.9149	23.4176	22.9545	22.9545	23.2115	22.9903	22.9903	22.9089	23.2203
16	40PE	23.1409	23.0564	23.656	22.9466	23.0171	23.4471	22.9089	22.9089	22.9903	23.2203
17	40PE	24.9004	23.2314	23.3522	23.4471	23.3065	23.1641	22.9089	22.9089	22.9903	23.2203
18	40PE	23.3014	23.9849	23.0668	23.2508	23.2256	23.2115	22.9903	22.9089	22.9089	23.2203
19	40PE	23.5573	22.634	23.5254	23.3065	23.1641	23.0171	22.9903	22.9089	22.9903	23.2203
20	40PE	22.9957	22.2254	23.5682	23.1641	22.9545	23.3065	22.9903	22.9089	22.9903	23.2203
MS	40PE	22.0089	22.2119	22.8119	22.9466	22.9466	22.9466	22.9089	22.9089	22.9089	23.2203
PS	40PE	24.9004	23.9849	24.3342	23.4471	23.7104	23.8288	22.9903	22.9903	22.9903	23.2203
SM	40PE	23.1588	23.1726	23.5263	23.2137	23.1741	23.2654	22.9455	22.9211	22.9740	23.2203

Tableau 12

Écart type obtenu pour les résultats du Tableau 11

NS	Instanc e	RS 2-opt Écart type [%]	RS 3-opt Écart type [%]	RS 4-opt Écart type [%]	AG 2-opt Écart type [%]	AG 3-opt Écart type [%]	AG 4-opt Écart type [%]	AF 2-opt Écart type [%]	AF 3-opt Écart type [%]	AF 4-opt Écart type [%]
1	40PE	-1.05%	-0.30%	3.10%	0.371%	0.977%	-0.242%	-0.991%	-1.341%	-0.991%
2	40PE	-0.46%	0.61%	-1.02%	-0.038%	-0.875%	1.863%	-0.991%	-1.341%	-0.991%
3	40PE	-0.72%	2.82%	-1.76%	-1.031%	1.006%	-1.145%	-0.991%	-1.341%	-0.991%
4	40PE	0.54%	1.67%	2.27%	-0.242%	-0.581%	-0.091%	-1.341%	-1.341%	-0.991%
5	40PE	0.90%	0.85%	2.99%	0.977%	2.111%	-0.875%	-0.991%	-1.341%	-0.991%
6	40PE	-0.62%	-0.91%	3.12%	0.826%	-0.242%	0.371%	-1.341%	-1.341%	-0.991%
7	40PE	-1.01%	-4.34%	-0.88%	-1.145%	0.371%	0.371%	-1.341%	-1.341%	-1.341%
8	40PE	-3.11%	-1.44%	3.35%	0.131%	-1.179%	2.621%	-0.991%	-0.991%	-0.991%
9	40PE	-4.98%	0.13%	-0.72%	0.371%	-0.935%	-0.242%	-1.341%	-1.341%	-0.991%
10	40PE	3.79%	-0.52%	-0.13%	0.131%	-0.875%	0.239%	-1.341%	-1.341%	-0.991%
11	40PE	0.06%	0.46%	4.80%	0.977%	-0.719%	2.548%	-1.341%	-1.341%	-1.341%
12	40PE	-5.22%	2.99%	3.10%	0.371%	0.371%	-0.875%	-1.341%	-1.341%	-0.991%
13	40PE	-0.25%	0.02%	3.40%	-0.038%	-0.760%	0.371%	-1.341%	-1.341%	-0.991%
14	40PE	0.05%	-0.65%	-0.72%	-1.145%	0.371%	-1.179%	-1.341%	-0.991%	-0.991%
15	40PE	-0.95%	-1.32%	0.85%	-1.145%	-1.145%	-0.038%	-0.991%	-0.991%	-1.341%
16	40PE	-0.34%	-0.71%	1.88%	-1.179%	-0.875%	0.977%	-1.341%	-1.341%	-0.991%
17	40PE	7.24%	0.05%	0.57%	0.977%	0.371%	-0.242%	-1.341%	-1.341%	-0.991%
18	40PE	0.35%	3.29%	-0.66%	0.131%	0.023%	-0.038%	-0.991%	-1.341%	-1.341%
19	40PE	1.45%	-2.52%	1.31%	0.371%	-0.242%	-0.875%	-0.991%	-1.341%	-0.991%
20	40PE	-0.97%	-4.28%	1.50%	-0.242%	-1.145%	0.371%	-0.991%	-1.341%	-0.991%
VME1	40PE	-0.26%	-0.21%	1.32%	-0.039%	-0.114%	0.293%	-1.183%	-1.288%	-1.061%

Tableau 13

Temps de simulation- problème du voyageur de commerce 40PE

Nombre simulations	Instance	RS		RS		AG		AG		AF		AF	
		2-opt TS [s]	3-opt TS [s]	4-opt TS [s]	2-opt TS [s]	3-opt TS [s]	4-opt TS [s]	2-opt TS [s]	3-opt TS [s]	4-opt TS [s]	2-opt TS [s]	3-opt TS [s]	4-opt TS [s]
1	40PE	9.0027	8.6395	9.6394	126.3382	128.2734	128.9438	227.9327	227.4388	228.9433			
2	40PE	9.4299	8.4288	9.2186	127.9360	128.9942	128.3373	225.2884	227.4889	229.0377			
3	40PE	8.3945	9.6388	9.3775	126.2774	128.3611	129.4733	226.6377	227.9266	229.4822			
4	40PE	8.7736	9.0037	9.5493	126.3758	129.4825	131.8354	227.7277	227.0266	229.4199			
5	40PE	8.4922	9.5384	8.5399	128.9337	129.6623	130.0037	227.4830	228.3721	229.4933			
6	40PE	9.0036	8.5488	8.3384	128.0367	129.6010	130.4722	226.3394	228.4866	229.6932			
7	40PE	8.1865	8.0372	8.3299	129.0015	128.8843	128.9554	227.3844	229.0277	228.7428			
8	40PE	8.2888	8.3295	8.4549	126.9388	128.9327	129.0554	226.5374	227.8557	229.6399			
9	40PE	8.3898	8.0373	8.9364	127.6283	128.3393	130.6465	228.2251	227.8955	229.3866			
10	40PE	8.2286	8.0038	9.0438	128.0033	128.4288	130.2214	227.0363	228.6433	229.9447			
11	40PE	8.9905	9.3395	9.8395	127.2734	129.3885	130.6433	226.7284	228.9668	230.3885			
12	40PE	7.9937	9.2294	9.5483	127.2282	130.5528	131.6534	227.0011	227.3755	229.4933			
13	40PE	8.0025	9.4394	8.5434	128.9022	128.3329	129.0043	225.9433	227.0944	230.4993			
14	40PE	8.4429	8.0017	8.3394	128.2747	127.9981	129.4366	227.9137	228.9656	230.5994			
15	40PE	8.4210	8.3933	8.3200	128.1141	128.5204	128.5535	227.0443	227.9337	229.5994			
16	40PE	9.4032	8.0388	8.4003	129.3002	128.0438	128.4426	226.8266	228.4485	229.5844			
17	40PE	9.5337	8.5493	8.3302	129.2733	128.5537	129.4436	226.3295	228.5044	228.6845			
18	40PE	9.3299	9.0266	8.0294	128.0437	129.3261	129.4437	226.0437	228.3833	229.0336			
19	40PE	8.3208	8.4393	8.9937	129.3362	128.4437	130.4425	227.7329	228.1044	230.4711			
20	40PE	7.9326	9.0372	9.4438	128.3827	128.9377	130.5994	227.3295	227.4833	229.4482			
MS	40PE	8.6280	8.6850	8.8608	127.9799	128.8529	129.7804	226.9743	228.0711	229.5793			

Tableau 14

Résultats obtenus pour l'algorithme de recuit simulé avec une amélioration 2, 3 et 4-opt  
(choix de température égale à 7421 unités)

Nombre simulations	Instance	R S 2-opt	R S 3-opt	R S 4-opt	Optimal	T	TR	Itérations
1	PR1002	261004.3328	260228.5300	261118.8839	259045	7421	0.978	2000
2	PR1002	260882.3926	260442.8342	260545.4421	259045	7421	0.978	2000
3	PR1002	260529.5188	261332.4388	260771.6388	259045	7421	0.978	2000
4	PR1002	261337.4282	260599.2992	261004.2214	259045	7421	0.978	2000
5	PR1002	262220.3328	260221.2225	260112.4436	259045	7421	0.978	2000
6	PR1002	260052.3104	259987.3319	261907.1118	259045	7421	0.978	2000
7	PR1002	259996.7202	261003.2226	260366.0327	259045	7421	0.978	2000
8	PR1002	260727.4427	260553.9353	260559.4439	259045	7421	0.978	2000
9	PR1002	260442.8342	260773.2267	260001.4384	259045	7421	0.978	2000
10	PR1002	260221.7348	261025.2225	262633.3027	259045	7421	0.978	2000
11	PR1002	261035.4437	260977.3299	260553.9353	259045	7421	0.978	2000
12	PR1002	260545.4421	260332.2216	261593.4439	259045	7421	0.978	2000
13	PR1002	260554.5537	261004.2214	261332.4388	259045	7421	0.978	2000
14	PR1002	261990.5532	259987.3319	261221.0043	259045	7421	0.978	2000
15	PR1002	260663.5530	260335.3328	260442.8342	259045	7421	0.978	2000
16	PR1002	260624.9251	260160.3285	260117.6288	259045	7421	0.978	2000
17	PR1002	261882.4368	259987.3319	260886.4429	259045	7421	0.978	2000
18	PR1002	260052.3104	260442.3326	261004.3328	259045	7421	0.978	2000
19	PR1002	260221.3204	260092.1193	261004.2214	259045	7421	0.978	2000
20	PR1002	260553.9353	260829.2992	260337.8827	259045	7421	0.978	2000
MV	PR1002	259996.7202	259987.3319	260001.4384	259045	7421	0.978	2000
PV	PR1002	262220.3328	261332.4388	262633.3027	259045	7421	0.978	2000
VM	PR1002	260777	260515.8	260875.7	259045	7421	0.978	2000

**Notations :**

RS 2-opt algorithme de recuit simulé avec la méthode d'amélioration 2-opt

RS 3-opt algorithme de recuit simulé avec la méthode d'amélioration 3-opt

RS 4-opt algorithme de recuit simulé avec la méthode d'amélioration 4-opt

MV meilleur valeur

PV pire valeur

Tableau 15

Résultats obtenus pour l'algorithme génétique avec une amélioration 2, 3 et 4-opt

Nombre simulations	Instance	Algorithme Génétique 2-opt	Algorithme Génétique 3-opt	Algorithme Génétique 4-opt	Optimal	Taux mutation	Itérations
1	PR1002	261409.76	259379.367	262044.335	259045	0.1	2000
2	PR1002	259666.43	261820.11	259379.37	259045	0.1	2000
3	PR1002	261529.38	261038.024	260442.202	259045	0.1	2000
4	PR1002	260663.79	261901.835	260663.795	259045	0.1	2000
5	PR1002	259379.37	260392.552	261409.761	259045	0.1	2000
6	PR1002	261288.78	261288.784	261529.384	259045	0.1	2000
7	PR1002	260663.79	260392.552	259666.434	259045	0.1	2000
8	PR1002	262044.34	260442.202	261392.552	259045	0.1	2000
9	PR1002	259666.43	260663.795	261288.784	259045	0.1	2000
10	PR1002	261409.76	259666.434	261409.761	259045	0.1	2000
11	PR1002	259666.43	261409.761	259666.434	259045	0.1	2000
12	PR1002	259379.37	259379.37	260392.552	259045	0.1	2000
13	PR1002	259666.43	259379.367	261529.384	259045	0.1	2000
14	PR1002	261409.76	260392.552	260663.795	259045	0.1	2000
15	PR1002	260663.79	261409.761	260442.202	259045	0.1	2000
16	PR1002	261288.78	259666.434	261409.761	259045	0.1	2000
17	PR1002	259379.37	260663.795	261901.835	259045	0.1	2000
18	PR1002	261529.38	261529.384	261409.761	259045	0.1	2000
19	PR1002	259666.43	260392.552	261409.761	259045	0.1	2000
20	PR1002	261409.76	259379.367	261288.784	259045	0.1	2000
MV	PR1002	259379.37	259379.37	259379.37	259045	0.1	2000
PV	PR1002	262044.34	261529.384	262044.335	259045	0.1	2000
VM	PR1002	260589.1	260529.4	260967	259045	0.1	2000

**Notations :**

MV meilleur valeur

PV pire valeur

VM valeur moyenne

Tableau 16

Résultats obtenus pour l'algorithme de fourmis avec une amélioration 2, 3 et 4-opt

Nombre simulations	Instance	Algorithme Fourmi 2-opt	Algorithme Fourmi 3-opt	Algorithme Fourmi 4-opt	Optimal	Itérations
1	PR1002	259379.37	259379.37	259379.37	259045	2000
2	PR1002	259379.37	259379.37	259379.37	259045	2000
3	PR1002	259379.37	259379.37	259379.37	259045	2000
4	PR1002	259379.37	259379.37	259379.37	259045	2000
5	PR1002	259379.37	259379.37	259379.37	259045	2000
6	PR1002	259379.37	259379.37	259379.37	259045	2000
7	PR1002	259379.37	259379.37	259379.37	259045	2000
8	PR1002	259379.37	259379.37	259379.37	259045	2000
9	PR1002	259379.37	259379.37	259379.37	259045	2000
10	PR1002	259379.37	259379.37	259379.37	259045	2000
11	PR1002	259379.37	259379.37	259379.37	259045	2000
12	PR1002	259379.37	259379.37	259379.37	259045	2000
13	PR1002	259379.37	259379.37	259379.37	259045	2000
14	PR1002	259379.37	259379.37	259379.37	259045	2000
15	PR1002	259379.37	259379.37	259379.37	259045	2000
16	PR1002	259379.37	259379.37	259379.37	259045	2000
17	PR1002	259379.37	259379.37	259379.37	259045	2000
18	PR1002	259379.37	259379.37	259379.37	259045	2000
19	PR1002	259379.37	259379.37	259379.37	259045	2000
20	PR1002	259379.37	259379.37	259379.37	259045	2000
MV	PR1002	259379.37	259379.37	259379.37	259045	2000
PV	PR1002	259379.37	259379.37	259379.37	259045	2000
VM	PR1002	259379.37	259379.37	259379.37	259045	2000

**Notations :**

MV meilleur valeur

PV pire valeur

VM valeur moyenne

Tableau 17

Comportement des algorithmes – problème du voyageur de commerce PR1002

Nombre simulations	Instance	RS 2-opt	RS 3-opt	RS 4-opt	AG 2-opt	AG 3-opt	AG 4-opt	AF 2-opt	AF 3-opt	AF 4-opt	Optimal
1	PR1002	261004.33	260228.53	261118.88	261409.76	259379.36	262044.33	259379.37	259379.3	259379.3	259045
2	PR1002	260882.39	260442.83	260545.44	259666.43	261820.11	259379.37	259379.37	259379.3	259379.3	259045
3	PR1002	260529.51	261332.43	260771.63	261529.38	261038.02	260442.20	259379.37	259379.3	259379.3	259045
4	PR1002	261337.42	260599.29	261004.22	260663.79	261901.83	260663.79	259379.37	259379.3	259379.3	259045
5	PR1002	262220.33	260221.22	260112.44	259379.37	260392.55	261409.76	259379.37	259379.3	259379.3	259045
6	PR1002	260052.31	259987.33	261907.11	261288.78	261288.78	261529.38	259379.37	259379.3	259379.3	259045
7	PR1002	259996.72	261003.22	260366.03	260663.79	260392.55	259666.43	259379.37	259379.3	259379.3	259045
8	PR1002	260727.44	260553.93	260559.44	262044.34	260442.20	261392.55	259379.37	259379.3	259379.3	259045
9	PR1002	260442.83	260773.22	260001.43	259666.43	260663.79	261288.78	259379.37	259379.3	259379.3	259045
10	PR1002	260221.73	261025.22	262633.30	261409.76	259666.43	261409.76	259379.37	259379.3	259379.3	259045
11	PR1002	261035.44	260977.32	260553.93	259666.43	261409.76	259666.43	259379.37	259379.3	259379.3	259045
12	PR1002	260545.44	260332.22	261593.44	259379.37	259379.37	260392.55	259379.37	259379.3	259379.3	259045
13	PR1002	260554.55	261004.22	261332.43	259666.43	259379.36	261529.38	259379.37	259379.3	259379.3	259045
14	PR1002	261990.55	259987.33	261221.00	261409.76	260392.55	260663.79	259379.37	259379.3	259379.3	259045
15	PR1002	260663.55	260335.33	260442.83	260663.79	261409.76	260442.20	259379.37	259379.3	259379.3	259045
16	PR1002	260624.92	260160.32	260117.62	261288.78	259666.43	261409.76	259379.37	259379.3	259379.3	259045
17	PR1002	261882.43	259987.33	260886.44	259379.37	260663.79	261901.83	259379.37	259379.3	259379.3	259045
18	PR1002	260052.31	260442.33	261004.33	261529.38	261409.76	261409.76	259379.37	259379.3	259379.3	259045
19	PR1002	260221.32	260092.11	261004.22	259666.43	260392.55	261409.76	259379.37	259379.3	259379.3	259045
20	PR1002	260553.93	260829.29	260337.88	261409.76	259379.36	261288.78	259379.37	259379.3	259379.3	259045
MS	PR1002	259996.72	259987.33	260001.43	259379.37	259379.37	259379.37	259379.37	259379.3	259379.3	259045
PS	PR1002	262220.33	261332.43	262633.30	262044.34	261529.38	262044.33	259379.37	259379.3	259379.3	259045
SM	PR1002	260777	260515.8	260875.7	260589.1	260529.4	260967	259379.37	259379.3	259379.3	259045

Tableau 18

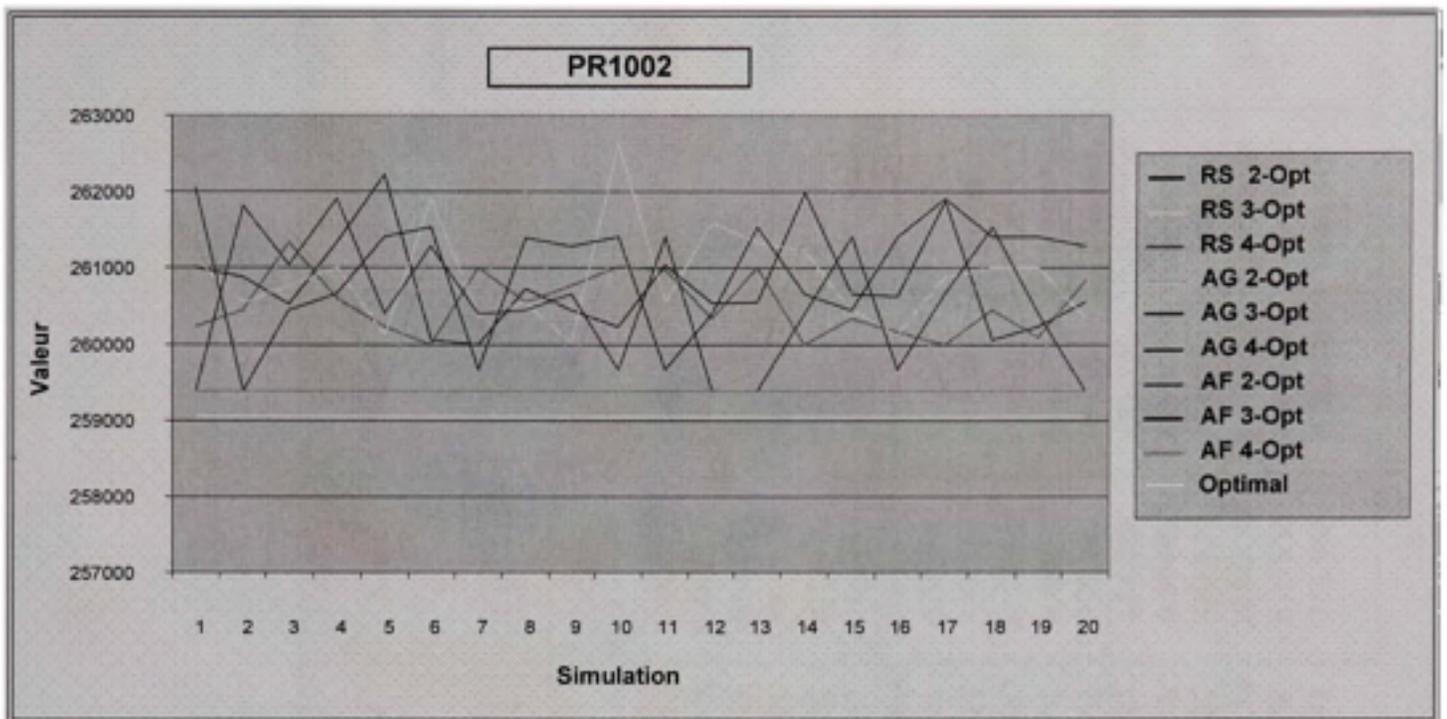
Écart-type obtenu pour les résultats du Tableau 17

Nombre simulations	Instance	RS 2-opt Écart type [%]	RS 3-opt Écart type [%]	RS 4-opt Écart type [%]	AG 2-opt Écart type [%]	AG 3-opt Écart type [%]	AG 4-opt Écart type [%]	AF 2-opt Écart type [%]	AF 3-opt Écart type [%]	AF 4-opt Écart type [%]
1	PR1002	0.756%	0.457%	0.801%	0.913%	0.129%	1.158%	0.129%	0.129%	0.129%
2	PR1002	0.709%	0.540%	0.579%	0.240%	1.071%	0.129%	0.129%	0.129%	0.129%
3	PR1002	0.573%	0.883%	0.667%	0.959%	0.769%	0.539%	0.129%	0.129%	0.129%
4	PR1002	0.885%	0.600%	0.756%	0.625%	1.103%	0.625%	0.129%	0.129%	0.129%
5	PR1002	1.226%	0.454%	0.412%	0.129%	0.520%	0.913%	0.129%	0.129%	0.129%
6	PR1002	0.389%	0.364%	1.105%	0.866%	0.866%	0.959%	0.129%	0.129%	0.129%
7	PR1002	0.367%	0.756%	0.510%	0.625%	0.520%	0.240%	0.129%	0.129%	0.129%
8	PR1002	0.649%	0.582%	0.585%	1.158%	0.539%	0.906%	0.129%	0.129%	0.129%
9	PR1002	0.540%	0.667%	0.369%	0.240%	0.625%	0.866%	0.129%	0.129%	0.129%
10	PR1002	0.454%	0.764%	1.385%	0.913%	0.240%	0.913%	0.129%	0.129%	0.129%
11	PR1002	0.768%	0.746%	0.582%	0.240%	0.913%	0.240%	0.129%	0.129%	0.129%
12	PR1002	0.579%	0.497%	0.984%	0.129%	0.129%	0.520%	0.129%	0.129%	0.129%
13	PR1002	0.583%	0.756%	0.883%	0.240%	0.129%	0.959%	0.129%	0.129%	0.129%
14	PR1002	1.137%	0.364%	0.840%	0.913%	0.520%	0.625%	0.129%	0.129%	0.129%
15	PR1002	0.625%	0.498%	0.540%	0.625%	0.913%	0.539%	0.129%	0.129%	0.129%
16	PR1002	0.610%	0.431%	0.414%	0.866%	0.240%	0.913%	0.129%	0.129%	0.129%
17	PR1002	1.095%	0.364%	0.711%	0.129%	0.625%	1.103%	0.129%	0.129%	0.129%
18	PR1002	0.389%	0.539%	0.756%	0.959%	0.959%	0.913%	0.129%	0.129%	0.129%
19	PR1002	0.454%	0.404%	0.756%	0.240%	0.520%	0.913%	0.129%	0.129%	0.129%
20	PR1002	0.582%	0.689%	0.499%	0.913%	0.129%	0.866%	0.129%	0.129%	0.129%
VMEt	PR1002	0.669%	0.568%	0.607%	0.596%	0.573%	0.742%	0.129%	0.129%	0.129%

Tableau 19

Temps de simulation- problème du voyageur de commerce PR1002

Nombre simulations	Instance	RS 2-opt TS [s]	RS 3-opt TS [s]	RS 4-opt TS [s]	AG 2-opt TS [s]	AG 3-opt TS [s]	AG 4-opt TS [s]	AF 2-opt TS [s]	AF 3-opt TS [s]	AF 4-opt TS [s]
1	PR1002	78.9376	82.4993	84.9277	269.0046	276.9027	278.8836	741.6390	756.8211	778.7377
2	PR1002	79.3825	82.8332	86.4300	269.5937	277.5528	282.9338	748.5396	759.0363	774.3329
3	PR1002	79.5835	86.9433	88.4099	266.6307	272.5109	282.8338	748.4438	759.9277	778.4438
4	PR1002	82.5395	83.7328	88.5488	268.4100	276.8327	279.0336	748.5309	771.3728	775.3209
5	PR1002	78.5300	79.4299	87.4387	271.4297	272.9277	285.9488	749.4301	758.5539	776.2100
6	PR1002	78.9332	85.2191	88.1993	269.5539	273.8395	282.8840	748.3089	758.9011	776.2296
7	PR1002	79.4388	86.4429	91.4294	268.4900	272.1002	285.9034	747.6699	758.4429	773.9332
8	PR1002	78.0443	83.7029	92.5499	269.1993	272.9337	285.4388	748.3209	758.0337	778.2221
9	PR1002	80.5388	81.4293	94.3439	268.0274	276.5538	283.5538	747.0037	758.2774	778.2218
10	PR1002	81.3633	81.4288	94.7201	270.5510	279.4438	283.9443	751.4391	771.0932	774.0733
11	PR1002	78.3054	85.9281	88.5399	269.0337	276.4399	282.6354	753.0538	758.5539	773.3110
12	PR1002	78.9388	82.5339	91.4390	268.0443	276.4439	284.8339	748.3388	756.9733	773.9118
13	PR1002	78.4884	86.4430	91.4386	272.6339	282.5385	288.5382	748.4443	758.0092	777.3200
14	PR1002	79.3993	81.2991	92.5488	272.9553	271.5538	282.6638	748.9438	758.4439	776.3296
15	PR1002	80.4993	82.4392	88.5377	269.5532	276.4399	281.7203	747.6110	766.3244	778.2208
16	PR1002	78.0337	85.9044	84.4394	268.0955	278.5395	284.9937	748.5539	756.5334	778.3295
17	PR1002	78.6388	85.0037	84.6388	267.8377	283.5356	283.6448	748.0933	759.6638	777.0327
18	PR1002	78.4993	86.4012	86.5499	268.7382	284.0032	284.8336	748.4309	762.8377	779.2228
19	PR1002	82.4493	86.4436	88.4295	269.5539	285.6395	284.7339	749.5349	758.0228	781.3028
20	PR1002	81.3043	85.4308	89.4088	272.4964	286.9303	282.9937	749.5538	758.4399	778.2184
MS	PR1002	79.5924	84.0744	89.1484	269.4916	277.6831	283.6475	748.4942	760.2131	776.8462



**Figure 1 Résultats pour le problème PR1002.**



Instance	RS			RS			RS			AG			AG			AG			AF			AF			AF			Optimal / Borne supérieure
	2-opt	3-opt	4-opt	2-opt	3-opt	4-opt	2-opt	3-opt	4-opt	2-opt	3-opt	4-opt	2-opt	3-opt	4-opt	2-opt	3-opt	4-opt	2-opt	3-opt	4-opt	2-opt	3-opt	4-opt				
	$v_m$	$\varepsilon_t$ [%]	$t_{opt}$ [s]	$v_m$	$\varepsilon_t$ [%]	$t_{opt}$ [s]	$v_m$	$\varepsilon_t$ [%]	$t_{opt}$ [s]	$v_m$	$\varepsilon_t$ [%]	$t_{opt}$ [s]	$v_m$	$\varepsilon_t$ [%]	$t_{opt}$ [s]	$v_m$	$\varepsilon_t$ [%]	$t_{opt}$ [s]	$v_m$	$\varepsilon_t$ [%]	$t_{opt}$ [s]	$v_m$	$\varepsilon_t$ [%]	$t_{opt}$ [s]				
Thompson70	685.4416 1.547% 8.652	683.0015 1.185% 8.995	685.9301 1.6193% 11.300	681.42 0.9511% 131.417	681.12 0.9067% 134.166	685.9301 1.6193% 11.300	681.42 0.9511% 131.417	681.12 0.9067% 134.166	681.12 0.9067% 134.166	677.0022 0.297% 247.008	676.9668 0.291% 263.481	681.6634 0.988% 138.003	677.0022 0.297% 247.008	676.9668 0.291% 263.481	681.6634 0.988% 138.003	677.0022 0.297% 247.008	676.9668 0.291% 263.481	681.6634 0.988% 138.003	677.0022 0.297% 247.008	676.9668 0.291% 263.481	677.0022 0.297% 247.008	676.9668 0.291% 263.481	677.0022 0.297% 247.008	676.9668 0.291% 263.481	677.0022 0.297% 247.008	675		
Berlin52	7674.338 1.7547% 8.006	7632.267 1.196% 8.442	7648.369 1.410% 9.659	7642.946 1.338% 128.109	7615.303 0.971% 133.523	7648.369 1.410% 9.659	7642.946 1.338% 128.109	7615.303 0.971% 133.523	7615.303 0.971% 133.523	7551.061 0.120% 241.448	7549.387 0.098% 253.902	7678.477 1.809% 136.922	7551.061 0.120% 241.448	7549.387 0.098% 253.902	7678.477 1.809% 136.922	7551.061 0.120% 241.448	7549.387 0.098% 253.902	7678.477 1.809% 136.922	7551.061 0.120% 241.448	7549.387 0.098% 253.902	7551.061 0.120% 241.448	7549.387 0.098% 253.902	7551.061 0.120% 241.448	7551.061 0.120% 256.371	7542			
Bayg29	1668.822 3.654% 7.119	1662.033 3.232% 7.336	1671.933 3.847% 7.411	1638.723 1.784% 114.903	1636.55 1.649% 116.382	1671.933 3.847% 7.411	1638.723 1.784% 114.903	1636.55 1.649% 116.382	1636.55 1.649% 116.382	1622.633 0.785% 198.476	1622.339 0.766% 201.007	1640.299 1.882% 117.003	1622.633 0.785% 198.476	1622.339 0.766% 201.007	1640.299 1.882% 117.003	1622.633 0.785% 198.476	1622.339 0.766% 201.007	1640.299 1.882% 117.003	1622.633 0.785% 198.476	1622.339 0.766% 201.007	1622.633 0.785% 205.047	1622.339 0.766% 205.047	1622.633 0.785% 205.047	1622.633 0.785% 205.047	1610			
Ulysses22	7158.327 2.072% 7.006	7116.501 1.476% 7.119	7164.033 2.154% 7.204	7112.947 1.425% 104.492	7091.003 1.112% 108.337	7164.033 2.154% 7.204	7112.947 1.425% 104.492	7091.003 1.112% 108.337	7091.003 1.112% 108.337	7079.110 0.973% 184.202	7079.110 0.973% 186.991	7116.730 1.479% 109.926	7079.110 0.973% 184.202	7079.110 0.973% 186.991	7116.730 1.479% 109.926	7079.110 0.973% 184.202	7079.110 0.973% 186.991	7116.730 1.479% 109.926	7079.110 0.973% 184.202	7079.110 0.973% 186.991	7079.110 0.973% 186.991	7079.110 0.973% 188.337	7079.110 0.973% 188.337	7079.110 0.973% 188.337	7013			
11PE	14.4139 -1.549% 6.277	14.3562 -1.943% 6.318	14.3624 -1.901% 6.399	14.3399 -2.055% 86.011	14.3397 -2.057% 87.472	14.3624 -1.901% 6.399	14.3399 -2.055% 86.011	14.3397 -2.057% 87.472	14.3397 -2.057% 87.472	14.3397 -2.055% 133.392	14.3397 -2.055% 138.003	14.3397 -2.057% 89.335	14.3397 -2.055% 133.392	14.3397 -2.055% 138.003	14.3397 -2.057% 89.335	14.3397 -2.055% 133.392	14.3397 -2.055% 138.003	14.3397 -2.057% 89.335	14.3397 -2.055% 133.392	14.3397 -2.055% 138.003	14.3397 -2.055% 142.558	14.3397 -2.055% 142.558	14.3397 -2.055% 142.558	14.3397 -2.055% 142.558	14.6408			
15PE	14.9709 -0.013% 6.846	14.9441 -0.191% 6.982	14.9598 -0.086% 6.966	14.8208 -1.015% 87.388	14.7968 -1.175% 88.936	14.9598 -0.086% 6.966	14.8208 -1.015% 87.388	14.7968 -1.175% 88.936	14.7968 -1.175% 88.936	14.7968 -1.175% 141.778	14.7968 -1.175% 143.043	14.84494 -0.854% 91.063	14.7968 -1.175% 141.778	14.7968 -1.175% 143.043	14.84494 -0.854% 91.063	14.7968 -1.175% 141.778	14.7968 -1.175% 143.043	14.84494 -0.854% 91.063	14.7968 -1.175% 143.043	14.7968 -1.175% 146.473	14.7968 -1.175% 146.473	14.7968 -1.175% 146.473	14.7968 -1.175% 146.473	14.7968 -1.175% 146.473	14.9728			
18PE	16.7753 -0.281% 6.851	16.7296 -0.553% 6.986	16.7832 -0.234% 6.958	16.7214 -0.602% 87.882	16.717 -0.628% 89.913	16.7832 -0.234% 6.958	16.7214 -0.602% 87.882	16.717 -0.628% 89.913	16.717 -0.628% 89.913	16.5988 -1.330% 143.966	16.4116 -2.443% 147.227	16.7996 -0.137% 92.642	16.5988 -1.330% 143.966	16.4116 -2.443% 147.227	16.7996 -0.137% 92.642	16.5988 -1.330% 143.966	16.4116 -2.443% 147.227	16.7996 -0.137% 92.642	16.5988 -1.330% 143.966	16.4116 -2.443% 148.322	16.4116 -2.443% 148.322	16.4116 -2.443% 148.322	16.4116 -2.443% 148.322	16.4116 -2.443% 148.322	16.8226			
25PE	18.8645 -0.083% 7.102	18.8329 -0.250% 7.114	18.8422 -0.201% 7.193	18.6166 -1.396% 105.442	18.5403 -1.800% 109.837	18.8422 -0.201% 7.193	18.6166 -1.396% 105.442	18.5403 -1.800% 109.837	18.5403 -1.800% 109.837	18.5991 -1.488% 135.402	18.5723 -1.630% 139.118	18.6614 -1.158% 111.228	18.5991 -1.488% 135.402	18.5723 -1.630% 139.118	18.6614 -1.158% 111.228	18.5991 -1.488% 135.402	18.5723 -1.630% 139.118	18.6614 -1.158% 111.228	18.5991 -1.488% 135.402	18.5723 -1.630% 139.118	18.6114 -1.423% 146.991	18.6114 -1.423% 146.991	18.6114 -1.423% 146.991	18.6114 -1.423% 146.991	18.8802			

Instance	R S		R S		R S		A G		A F		A F		A F		Optimal / Borne supérieure	
	2-opt	3-opt	4-opt	2-opt	3-opt		4-opt									
40PE	$v_m$															
	$\varepsilon_t$ [%]															
	$t_{ms}$ [s]															
	23.1588	23.1726	23.5263	23.2137	23.1741	23.2654	22.9455	22.9211	22.9740	22.9455	22.9211	22.9740	22.9455	22.9211	22.9740	23.2203
	-0.26%	-0.21%	1.32%	-0.039%	-0.114%	0.293%	-1.183%	-1.288%	-1.061%	-1.183%	-1.288%	-1.061%	-1.183%	-1.288%	-1.061%	
	8.6280	8.6850	8.8608	127.9799	128.8529	129.7804	226.9743	228.0711	229.5793	226.9743	228.0711	229.5793	226.9743	228.0711	229.5793	

**Notations :** $v_m$  valeur moyenne; $\varepsilon_t$  [%] écart-type moyenne; $t_{ms}$  [s] temps simulations moyen

## BIBLIOGRAPHIE

- Aarts A. & Lenstra P. (1997). "Local search in combinatorial optimization", Transportation Research Board, Washington, D.C, 51-57.
- Achuthan, N. R., L. Caccetta, & al. (1998). "Capacitated vehicle routing problem: some new cutting planes." *Asia-Pacific Journal of Operational Research* 15(1): 109-123.
- Acton, S. T. & A. C. Bovik (1996). "Generalized deterministic annealing." *IEEE Transactions on Neural Networks* 7(3): 686-99.
- Golden, B.L, DeArmon J.S., & al. (1983). "Computational experiments with algorithms for a class of routing problem." *Computers and Operations Research* 10(1): 47-59.
- Baker, D., Edward K., & al. (1986). "Solution Improvement Heuristics for the Vehicle Routing Problem with Time Window Constraints." *American Journal of Mathematical and Management Sciences* 6(Special Issues): 261-300.
- Bennett, B. & Gazis D. (1972). "School bus routing by computer." *Transportation Research Board* 6: 317-326.
- Berger, J. & Barkaoui M. (2004). "A parallel hybrid genetic algorithm for the vehicle routing problem with time windows." *Computers and Operations Research* 31(12): 2037-2053.
- Berger, J., Barkaoui M, & al. (2003). "A route-directed hybrid genetic approach for the vehicle routing problem with time windows." *INFOR Journal* 41(2): 179-194.
- Bodin L., (1979). "Routing and scheduling of school buses by computer." *Transportation Science* 13: 113-129.
- Bourgeois, M., G. Laporte, & al. (2003). "Heuristics for the black and white traveling salesman problem." *Computers and Operations Research* 30(1): 75-85.

- Braca, J., Bramel J., & al. (1997). "A computerized approach to the New York City school bus routing problem." *IIE Transactions* 29(8): 693-702.
- Blum, C. (2005). "Ant colony optimization: Introduction and recent trends." *Physics of Life Reviews* 2: 353-373.
- Chao, I. M., Golden B. L., & al. (1995). "New heuristic for the period traveling salesman problem." *Computers & Operations Research* 22(5): 553-565.
- Chen, Z. (2007). An improved ant algorithm for VRP with the shortest delivery time in distribution center, Haikou, Hainan, China, Institute of Electrical and Electronics Engineers Computer Society, Piscataway, NJ 08855-1331, United States.
- Chiung, M., K. Jongsoo, & al. (2002). "Integrated process planning and scheduling with minimizing total tardiness in multi-plants supply chain." *Computers & Industrial Engineering* 43(1-2): 331-49.
- Christofides, N. & Eilon S. (1989). "An algorithm for the vehicle dispatching problem." *Operations Research*: 309-318.
- Clarke, G. & Wright J. (1964). "Scheduling of vehicle from a central depot to a number of delivery points." *Operations Research* 12(4): 568-581.
- Cohen-Tannoudji, C. (1997). "Mécanique quantique", Harvard Business School Press
- Cordeau, J. F., Gendreau M., & al. (1997). "A taboo search heuristic for periodic and multi-depot vehicle routing problems." *Networks* 30(2): 105-19.
- Danzig, G.B. (1959). "The truck dispatching problem." *Management Science* 6: 81-91.
- Doerner, K. F., Hartl R. F., & al. (2004). Applying ant colony optimization to the capacitated arc routing problem, Brussels, Belgium, Springer-Verlag.
- Dorigo M. (2003). *Handbook of Metaheuristics*. Boston, Kluwer Academic Publishers.

- Dorigo M., (2004). "Ant Colony Optimization." MIT Press.
- Dumas, J.-G. (2007). *Théorie des codes : compression, cryptage, correction*. Paris, Dunod.
- Fogel, L. J. (1966). "Artificial intelligence through simulated evolution." *Artificial Intelligence through Simulated Evolution*: 170.
- Ulusoy, G. (1985). "The fleet size and mix problem for capacited arc routing." *European Journal of Operational Research* 22: 329-337.
- Gavish B., (1979). "An approach for solving a class transportation scheduling problems." *EJOR* 3 2: 122-134.
- Geman, S. & Geman D. (1984). "Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images." *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-6(6)*: 721-741.
- Gendreau, M., Hertz A., & al. (1996). "The traveling salesman problem with backhauls." *Computers and Operations Research* 23(5): 501-508.
- Ghaziri, H. & Osman I. (2003). "A neural network algorithm for the traveling salesman problem with backhauls." *Computers and Industrial Engineering* 44(2): 267-281.
- Gianpaolo, G., Luca L.G. (2005). "A heuristic for the periodic rural postman problem." *Computers and Operations Research* 32(2): 219-228.
- Glover, F., Gutin G., & al. (2001). "Construction heuristics for the asymmetric TSP." *European Operational Research* 129: 555-568.
- Goldberg, D. E. (1989). *Genetic Algorithm in Search, Optimization and Machine Learning*. New York.
- Goldberg, D. E. (1991). "Real-coded genetic algorithms, virtual alphabets, and blocking." *Complex Systems* 5(2): 139-67.

- Groetschel, M., Juenger M., & al. (1984). "Cutting plane algorithm for the linear ordering problem." *Operations Research* 32(6): 1195-1220.
- Helsgaun, K. (2000). "An effective implementation of the Lin-Kernighan traveling salesman heuristic." *European Journal of Operational Research* 126(1): 106.
- Heung-Suk, H. (2002). "An improved model for vehicle routing problem with time constraint based on genetic algorithm." *Computers & Industrial Engineering* 42(2-4): 361-9.
- Holland, J. H. (1975). "Adaptation in Natural and Artificial Systems." University of Michigan Press.
- Ingber, L. & Rosen B. (1992). "Genetic algorithms and very fast simulated reannealing a comparison." *Mathematical and Computer Modeling* 16(11): 87-100.
- Ioannou, G., Kritikos M., & al. (2003). "A problem generator-solver heuristic for vehicle routing with soft time windows." *Omega* 31(1): 41-53.
- Deneubourg J., Goss L., & al. (1989). "Self-organized short cuts in the argentine ant." *Naturwissenschaften* 76: 579-581.
- Kernighan, L. S. (1973). "An Effective Heuristic Algorithm for the Traveling Salesman Problem." *Operations Research* 21: 2245-2269.
- Kirkpatrick, S., Gelett. C.D. & al. (1983). "Optimization by Simulated Annealing." *Science* 220: 671-680.
- Kirkpatrick S. (1983). "Optimization by Simulated Annealing." *Science* 220: 671-680.
- Kube, C. R. & Bonabeau E. (1998). "Cooperative Transport by Ants and Robots" *Physics of Life Reviews* 45: 256-163.
- Hölldobler L. & Wilson C. (1990). "Competitive mechanisms underlying the displacement of native ants by the invasive Argentine ant." *Life Science* 11(4): 396-404.

- Laporte, G. (1992). "The Vehicle Routing Problem: An Overview of Exact and Approximate Algorithms." *European Journal of Operational Research* 59: 345-358.
- Li, Z., Guo S., & al. (2004). *Improved GRASP with taboo search for vehicle routing with both time window and limited number of vehicles*, Ottawa, Ont., Canada, Springer Verlag, Heidelberg, D-69121, Germany.
- Mladenovic, N. & Hansen P. (1997). "Variable neighborhood search." *Computers & Operations Research* 24(11): 1097-1100.
- Osman, I. H. (1993). *Heuristics for the Generalized Assignment Problem: Simulated Annealing and Taboo Search*. UKC/IMS/OR93/10b. Canterbury, University of Kent.
- Perdalos, P. & Resende M.G .C. Eds. (2002). *Handbook of metaheuristics*. Kluwer.
- Paletta, G. (2002). "The period traveling salesman problem: A new heuristic algorithm." *Computers & Operations Research* 29(10): 1343-52.
- Patterson, R. (2003). "The cardinality constrained covering traveling salesman problem." *Computers and Operations Research* 30(1): 97-116.
- Wong, R.T. (1981). "Capacited arc routing problems." *Networks* 11: 305-315.
- Renaud, J., & Ouenniche J. (2000). "A routing for the pickup and delivery traveling salesman problem." *Computers and Operations Research* 21(9): 905-916.
- Rochat, Y. & Taillard E. D. (1995). "Probabilistic diversification and intensification in local search for vehicle routing." *Journal of Heuristics* 1(1): 147-167.
- Salazar, R. & Toral R. (1997). *Hybrid simulated annealing*, Madrid, Spain, World Scientific.
- Schwefel, H. (1981). *Numerical Optimization of Computers Models.* World Scientific.

- Shutler, P. M. (2001). "An improved branching rule for the symmetric traveling salesman problem." *The Journal of the Operational Research Society* 52: 2.
- Solomon, M. M. (1987). "Algorithm for the vehicle routing problem with time windows." *Operations Research* 35: 254-265.
- Stützle, T. (1999). *Local Search Algorithms for Combinatorial Problems – Analysis, Algorithms and New Applications*. DISKI-Dissertationen zur Künstlichen Intelligenz. Infix. S. Augustin.
- Swersey A.J., (1984). "Scheduling school buses." *Management Science* 30(7): 844-853.
- Szu, H. H. & Hartley R. (1987). "Noneconomic optimization by fast simulated annealing." *Proceedings of the IEEE* 75: 1538-1540.
- Tai-Hsi, W., Chinyao L., & al. (2002). "Heuristic solutions to multi-depot location-routing problems." *Computers & Operations Research* 29(10): 1393-415.
- Tarantilis, C. D., Ioannou G., *et al.* (2004). "A threshold accepting approach to the open vehicle routing problem." *RAIRO-Operations Research* 38(4): 345-60.
- Thangiah, S. R. & Nygard K. E. (1992). *School bus routing using genetic algorithms*, Orlando, FL, USA, Publication of Int Soc for Optical Engineering, Bellingham, WA, USA.
- Toth, P. & Vigo D. (2002). "Models, relaxations and exact approaches for the capacitated vehicle routing problem." *Discrete Applied Mathematics* 123(1-3): 487-512.
- Tsallis, C. & Stariolo D. A. (1996). "Generalized simulated annealing." *Physica A* 233(1-2): 395-406.
- Utgoff, P.E., Precup, D. (1998). "Constructive function approximation." In Motoda & Liu (Eds.), *Feature extraction, construction, and selection: A data-mining perspective*. Kluwer. An earlier version appeared as Technical Report UM-CS-1997-004, Department of Computer Science, University of Massachusetts, Amherst, MA 01003-4610.

- Voß, Martello, & al. (1999). *Meta-Heuristics – Advances and Trends in Local Search Paradigms for Optimisation*. Dordrecht, Kluwer Academic Publisher.
- Gutjahr, W.J. (2002). "ACO algorithms with guaranteed convergence to the optimal solution." *Inf.Process* 82(3): 145-153.
- Wang, C., Cui D.W., & al. (2006). "An efficient genetic algorithm for TSP." *Journal of Xian University of Technology* 22(1): 37-41.
- Xi-Huai, W. & Jun-Jun L. (2004). *Hybrid particle swarm optimization with simulated annealing*, Shanghai, China, IEEE.