



HAL
open science

APPROCHES EVOLUTIONNISTES POUR LA RESOLUTION DU 1-PDPTW STATIQUE ET DYNAMIQUE

Ryan Kammarti

► **To cite this version:**

Ryan Kammarti. APPROCHES EVOLUTIONNISTES POUR LA RESOLUTION DU 1-PDPTW STATIQUE ET DYNAMIQUE. Autre. Ecole Centrale de Lille, 2006. Français. NNT: . tel-00143782

HAL Id: tel-00143782

<https://theses.hal.science/tel-00143782>

Submitted on 25 May 2007

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

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

N° d'ordre : 34

ECOLE CENTRALE DE LILLE
UNIVERSITE DES SCIENCES ET TECHNOLOGIES DE LILLE

THESE

Présentée en vue
d'obtenir le grade de

DOCTEUR

en

Spécialité : Automatique et Informatique Industrielle

par

Ryan KAMMARTI

DOCTORAT DELIVRE CONJOINTEMENT PAR L'ECOLE CENTRALE DE LILLE
ET L'UNIVERSITE DES SCIENCES ET TECHNOLOGIES DE LILLE

Titre de la thèse :

**APPROCHES EVOLUTIONNISTES POUR LA RESOLUTION DU 1-PDPTW
STATIQUE ET DYNAMIQUE**

Soutenue le 13 décembre 2006 devant le jury d'examen :

<i>Président :</i>	Dimitru POPESCU	Professeur, Université Polytechnique de Bucarest
<i>Rapporteur :</i>	Khaled GHEDIRA	Professeur, ENSI Tunis
<i>Rapporteur :</i>	Abdellah EL MOUDNI	Professeur, UTBM
<i>Examineur :</i>	Pierre BORNE	Professeur, Ecole Centrale de Lille
<i>Examineur :</i>	Slim HAMMADI	Professeur, Ecole Centrale de Lille
<i>Examineur :</i>	Mekki KSOURI	Professeur, ENIT Tunis
<i>Examineur :</i>	Daniel JOLLY	Professeur, Université d'Artois

Thèse préparée dans le laboratoire LAGIS de l'Ecole Centrale de Lille et au LACS sous la direction des Professeurs P.Borne, S. Hammadi et M. Ksouri

A ma chère et bien aimée épouse Héla
A Lina, joie de ma vie
A mon père, ma mère, Lolo
A mon frère et ma sœur
A ma famille
A mes professeurs
A mes amis

Remerciements

Cette thèse est l'aboutissement de travaux réalisés dans le cadre d'une coopération et une cotutelle entre le laboratoire d'Automatique, Génie Informatique et Signal (LAGIS) de l'Ecole Centrale de Lille et le laboratoire d'Analyse et Commande des Systèmes (LACS) de l'Ecole Nationale d'Ingénieurs de Tunis. Elle a été effectuée sous la co-direction des Professeurs Pierre BORNE, Slim HAMADI et Mekki KSOURI.

Je tiens tout d'abord à exprimer ma profonde gratitude au Professeur Pierre BORNE, Professeur et directeur du département Automatique et Informatique Industrielle de l'Ecole Centrale de Lille, co-directeur de cette thèse, pour l'énorme soutien scientifique et moral qu'il a su m'accorder pendant ces dernières années. Je le remercie pour ses qualités humaines et scientifiques et pour m'avoir toujours encouragé à aller de l'avant.

Je tiens également à remercier chaleureusement Professeur Slim HAMADI, Professeur à l'Ecole Centrale de Lille et co-directeur de ma thèse, pour sa perspicacité et ses précieux conseils qui ont contribué à la réalisation de ce travail.

Je ne saurais oublier de remercier mon Professeur Mekki KSOURI, Professeur à l'Ecole Nationale d'Ingénieurs de Tunis, et co-directeur de ma thèse, pour sa rigueur scientifique et son aide si précieuse. Je tiens aussi à le remercier pour ses qualités humaines et son soutien permanent.

Que le Professeur Dimitru POPESCU, Doyen du département Automatique et Calculateur de l'Université de Bucharest, reçoive mes sincères remerciements pour l'honneur qu'il me fait en présidant ce jury.

Un témoignage de ma profonde reconnaissance s'adresse au Professeur Khaled GHEDIRA, Directeur de l'Ecole Nationale des Sciences de l'Informatique à Tunis (ENSI) et au Professeur Abdellah EL MOUDNI, Directeur du Laboratoire Systèmes et Transport de l'Université de Technologie de Belfort Montbéliard , pour avoir accepté d'examiner ma thèse et d'en être les rapporteurs et également pour leurs remarques et critiques constructives.

Pour m'avoir fait l'honneur d'accepter d'être examinateur de cette thèse et de participer à ce jury, un grand merci s'adresse à Professeur Daniel JOLLY, Directeur du Laboratoire de Génie Informatique et d'Automatique de l'Université d'Artois, dont les travaux figurent parmi nos grandes références.

Dans la crainte d'oublier quelqu'un, je remercie toutes les personnes qui ont contribué de près ou de loin à la réalisation de cette thèse, en particulier tout le personnel du LAGIS et les membres du LACS pour leur disponibilité et leur soutien.

Table des Matières

Index des Tableaux et des Figures	13
Introduction Générale.....	17
Chapitre 1:	
Les tournées de véhicules : état de l’art, problèmes et méthodes de résolution	21
1.1 Introduction	22
1.2 Modélisation mathématique	22
1.3 Le VRP et ses variantes.....	26
1.4 Le VRP et ses variantes : des problèmes NP-difficiles	29
1.5 Les méthodes de résolution du VRP et de ses variantes	29
1.5.1 Les méthodes exactes	30
1.5.1.1 La méthode de Branch and Bound	30
1.5.1.2 La méthode de Branch and cut.....	30
1.5.1.3 La programmation dynamique	30
1.5.2 Les heuristiques.....	31
1.5.2.1 L’heuristique « groupe en premier, route en second ».....	31
1.5.2.2 L’heuristique « route en premier, groupe en second»	31
1.5.2.3 L’algorithme de gain	31
1.5.2.4 L’algorithme en pétale	32
1.5.3 Les métaheuristiques	32
1.5.2.5 Les métaheuristiques de recherche locale	33
1.5.2.5.1 Méthode de descente	33
1.5.2.5.2 Méthode de Recuit Simulé	33
1.5.2.5.3 Recherche Tabou.....	35
1.5.2.6 Les métaheuristiques d’évolution.....	37
1.5.2.6.1 Les algorithmes génétiques	37
1.5.2.6.2 La recherche distribuée	38
1.5.2.6.3 Les colonies de fourmis.....	38
1.6 Le problème PDPTW	38
1.6.1 Formulation mathématique du problème	39
1.6.2 Etat de l’art des Pickup and Delivery Problem With Time windows	42

1.6.2.1	1-PDPTW	42
1.6.2.2	m-PDPTW	43
1.7	Conclusion.....	44

Chapitre 2 :

Les algorithmes évolutionnistes et la recherche Tabou : Principes et caractéristiques... 45

2.1	Introduction	46
2.2	Les algorithmes génétiques	46
2.2.1	Le principe des algorithmes génétiques	46
2.2.2	Le codage	47
2.2.2.1	Le codage binaire	48
2.2.2.2	Le codage symbolique.....	48
2.2.3	L'opérateur de croisement.....	48
2.2.4	L'opérateur de mutation	49
2.2.5	L'évaluation et la sélection des individus	51
2.2.6	Architecture d'un algorithme génétique.....	53
2.3	La recherche Tabou.....	54
2.3.1	Problème d'affectation quadratique	55
2.3.2	Recherche Tabou de base.....	55
2.3.2.1	Le voisinage d'une solution	55
2.3.2.2	Mouvements	56
2.3.2.3	Evaluation des solutions voisines.....	58
2.3.3	Les mouvements candidats.....	59
2.3.4	La mémoire à court terme	59
2.3.4.1	Table de hachage.....	60
2.3.4.2	Liste des attributs tabous	61
2.3.4.3	La durée des interdictions	62
2.3.4.4	Le critère d'aspiration	63
2.4	Conclusion.....	64

Chapitre 3 :

Approches évolutionnistes et approches évolutionnistes hybrides pour la résolution du 1-PDPTW 65

3.1	Intoduction	66
-----	-------------------	----

3.2	Approches évolutionnistes	66
3.2.1	Formulation mathématique	66
3.2.2	Evaluation multicritère	68
3.2.2.1	Stratégie globale d'évaluation	68
3.2.2.1.1	Approche d'évaluation multicritère	68
3.2.3	1 ^{ère} Approche évolutionniste [Kammarti et al 2004].....	69
3.2.3.1	Formulation mathématique adoptée	69
3.2.3.2	Méthode d'évaluation multicritère de la 1 ^{ère} approche évolutionniste.....	70
3.2.3.3	Le codage	71
3.2.3.4	Conception des solutions réalisables et correction des individus non viables.....	72
3.2.3.4.1	Procédure de vérification et correction des individus vis-à-vis des contraintes de précédence.....	73
3.2.3.4.2	Procédure de vérification et correction des individus vis-à-vis des contraintes de capacité	73
3.2.3.5	Génération des solutions initiales.....	76
3.2.3.5.1	Minimisation de la distance totale parcourue (<i>heuristique 1</i>).....	76
3.2.3.5.2	Minimisation de la somme des temps d'attente (<i>heuristique 2</i>).....	76
3.2.3.5.3	Minimisation de la somme des retards (<i>heuristique 3</i>)	77
3.2.3.6	Croisement	77
3.2.3.7	Mutation	79
3.2.3.7.1	1 ^{er} opérateur de mutation contrôlée.....	79
3.2.3.7.2	2 ^{ème} opérateur de mutation	80
3.2.3.7.3	3 ^{ème} opérateur de mutation contrôlée	80
3.2.3.7.4	4 ^{er} opérateur de mutation contrôlée.....	80
3.2.3.8	Sélection.....	80
3.2.3.9	Structure de l'algorithme évolutionniste de la 1 ^{ère} approche	81
3.2.4	2 ^{ème} Approche évolutionniste [Kammarti et al 2005b]	82
3.2.4.1	Formulation mathématique adoptée	82
3.2.4.2	La minimisation de la distance totale parcourue : borne inférieure	82
3.2.4.3	La minimisation du retard total : borne inférieure [Chu 1990].....	84
3.2.4.4	Méthode d'évaluation multicritère de la 2 ^{ème} approche évolutionniste.....	86
3.2.4.4.1	L'homogénéisation des critères à optimiser.....	87
3.2.4.4.2	Application de la Pareto optimalité.....	88

3.2.4.5	Les outils génétiques de la 2 ^{ème} approche évolutionniste.....	88
3.2.4.5.1	Croisement	89
3.2.4.6	Structure de l'algorithme évolutionniste de la 2 ^{ème} approche	90
3.3	Hybridation des approches évolutionnistes : étage d'amélioration par la recherche Tabou spécialisée	91
3.3.1	La construction du voisinage d'une solution [Kammarti et al 2005a]	92
3.3.1.1	Les heuristiques 2-opt et 3-opt	92
3.3.1.2	L'heuristique d'échange conçue pour le 1-PDPTW	92
3.3.2	La recherche Tabou dédiée au 1-PDPTW	93
3.3.2.1	Mise à jour de la liste Tabou : critère d'aspiration.....	93
3.4	Conclusion.....	94
Chapitre 4 :		
Adaptation des outils de résolution du 1-PDPTW pour le 1-DPDPTW		97
4.1	Introduction	98
4.2	Les problèmes de tournées de véhicules dynamiques et leurs caractéristiques	98
4.3	Les degrés de dynamisme et la classification du DVRP	101
4.3.1	Degré de dynamisme sans fenêtres de temps (<i>dod</i>)	101
4.3.2	Degré de dynamisme effectif (<i>edod</i>).....	102
4.3.3	Degré de Dynamisme effectif avec fenêtres de temps (<i>edod-tw</i>).....	103
4.3.4	Classification du DVRP	103
4.4	Formulation mathématique générale du DVRP	105
4.5	Les méthodes de résolution du DVRP et de ses variantes	105
4.5.1	Les méthodes séquentielles	105
4.5.1.1	Les heuristiques simples.....	106
4.5.1.2	Les heuristiques d'insertion	110
4.5.1.3	Les métaheuristiques	113
4.5.2	Les méthodes parallèles	114
4.6	Approche dynamique proposée pour le 1-DPDPTW [Kammarti et al. 2006]	115
4.6.1	Formulation mathématique	116
4.6.2	Structure générale des algorithmes évolutionnistes élaborés.....	117
4.6.3	Génération de la population initiale	118
4.6.3.1	Minimisation de la distance totale parcourue (<i>heuristique 1</i>).....	119
4.6.3.2	Minimisation de la somme des temps d'attente (<i>heuristique2</i>).....	120

4.6.3.3	Minimisation de la somme des retards (<i>heuristique3</i>).....	120
4.6.4	Conception des solutions réalisables et correction des individus non viables : les procédures de vérification et de correction.....	120
4.6.4.1	Vérification et correction des individus vis-à-vis des contraintes de précédence.....	120
4.6.4.2	Vérification et correction des individus vis-à-vis des contraintes de capacité.....	122
4.6.5	Croisement	124
4.6.6	Mutation	125
4.6.7	Hybridation de l'approche évolutionniste : étage d'amélioration par la recherche Tabou spécialisée.....	126
4.7	Conclusion.....	126
Chapitre 5 :		
Système informatisé de résolution du 1-PDPTW et du 1-DPDPTW, Simulation et Résultats		129
5.1	Introduction.....	130
5.2	Architecture du système informatisé de résolution du 1-PDPTW	130
5.3	Les benchmarks.....	132
5.3.1	Les benchmarks utilisés (Li et Lim).....	132
5.3.2	Les benchmarks élaborées [Kammarti et al 2006].....	134
5.4	Les paramètres de simulation adoptés.....	136
5.5	Le 1-PDPTW statique : Simulations et Résultats.....	139
5.5.1	Les benchmarks de Li et Lim : simulations et résultats	140
5.5.2	Les benchmarks P, PD, PT, PDT : simulations et résultats	142
5.5.2.1	Résultats	142
5.5.2.2	L'apport de l'étage d'hybridation	144
5.6	Le 1-PDPTW dynamique : Simulations et Résultats	146
5.6.1	Les benchmarks utilisés	146
5.6.2	L'acquisition des demandes dynamique	147
5.6.3	Résultats	147
5.7	Difficulté des benchmarks & Temps de calcul requis.....	152
5.7.1	Difficulté des benchmarks.....	152
5.7.2	Temps de calcul.....	153
5.8	Conclusion.....	155

Conclusion Générale	157
Annexes	161
ANNEXE 1 : Les benchmarks élaborés	162
ANNEXE 2 : Les paramètres de simulation	178
ANNEXE 3 : Résultats des simulations	179
Bibliographie Générale	183
Index	195

Index des Tableaux et des Figures

Index des Tableaux

Tableau 3.1 : Calcul de la borne inférieure minimisant la somme des retards	86
Tableau 5.1 : Solutions pour le 1-PDPTW basés sur des problèmes de Li et Lim (AEHs_1)	140
Tableau 5.2 : Solutions pour le 1-PDPTW basés sur des problèmes de Li et Lim (AEs_2)..	141
Tableau 5.3 : Solutions pour la famille des benchmarks P résolus par AEs_1.....	142
Tableau 5.4 : Solutions pour la famille des benchmarks P résolus par AEs_1.....	143
Tableau 5.5 : Solutions des problèmes P, PD, PT et PDT.	144
Tableau 5.6 : Nombre des sommets dynamiques par problème et degrés de dynamisme.	147
Tableau 5.7 : Nombre des sommets dynamiques par problème et degrés de dynamisme.	148
Tableau 5.8 : Solutions des problèmes Pd, Pdd, PTd et PDTd.	149
Tableau 5.8 : Tableau récapitulatif des résultats du 1-PDPTW et du 1-DPDPTW.	150
Tableau 5.9 : Calcul des degrés de difficulté.	153
Tableau 5.10 : Les temps moyens de calcul.....	154

Index des Figures

Figure 1.1 : Exemple de tournées avec 4 véhicules.	25
Figure 1.2 : Les principales méthodes de résolution du VRP.	29
Figure 1.3 : Pseudo-code de l'algorithme de Metropolis.	34
Figure 1.4 : Pseudo-code de l'algorithme du Recuit Simulé.....	35
Figure 1.5 : Pseudo-code de la recherche Tabou.	36
Figure 1.6 : Pseudo-code de l'algorithme des colonies de fourmis.	38
Figure 2.1 : Le principe des algorithmes génétiques.....	47
Figure 2.2 : Codage binaire	47
Figure 2.3 : Codage symbolique	47
Figure 2.4 : Le croisement à un point.....	49
Figure 2.5 : Le croisement à deux points	49
Figure 2.6 : Mutation Classique	50
Figure 2.7 : Transposition de deux allèles consécutifs.....	50
Figure 2.8 : Transposition de deux allèles quelconques.....	50

Figure 2.9 : Inversion d'allèles.....	50
Figure 2.10: La roulette de sélection.....	52
Figure 2.11 : Architecture d'un algorithme génétique.....	53
Figure II.12 : Voisinage d'une solution.....	55
Figure 2.13 : Illustration de l'ensemble des mouvements possibles d'une solution de 4 éléments.....	56
Figure 2.14 : Mouvement de la recherche Tabou : inversion de deux éléments successifs.....	57
Figure 2.15 : Mouvement de la recherche Tabou : permutation de deux éléments quelconques distincts.....	57
Figure 2.16 : Mouvement de la recherche Tabou : déplacement d'un élément.....	57
Figure 2.17 : Déconnexions et blocages dans la recherche Tabou.....	60
Figure 2.18 : Problème de minimisation d'altitude.....	62
Figure 3.1 : Approche Pareto optimalité.....	69
Figure 3.2 : Codage par liste de permutation.....	71
Figure 3.3 : Codage par liste de rang.....	72
Figure 3.4 : Codage par matrice de permutation.....	72
Figure 3.5.a : Individu non viable vis-à-vis des contraintes de précédences.....	73
Figure 3.5.b : Individu viable vis-à-vis des contraintes de précédences.....	73
Figure 3.5.c : Correction des individus vis-à-vis des contraintes de précédence.....	74
Figure 3.6.a : Individu non viable vis-à-vis des contraintes de capacité.....	74
Figure 3.6.b : Individu viable vis-à-vis des contraintes de capacité.....	74
Figure 3.6.c : Correction des individus vis-à-vis des contraintes de capacité.....	75
Figure 3.7.a : Algorithme de <i>l'heuristique 1</i>	76
Figure 3.7.b : Individu généré par <i>l'heuristique 1</i>	76
Figure 3.8 : Individu généré par <i>l'heuristique 2</i>	77
Figure 3.9 : Individu généré par <i>l'heuristique 3</i>	77
Figure 3.10.a : l'algorithme de croisement.....	78
Figure 3.10.b : Croisement.....	78
Figure 3.11 : Algorithme de la mutation contrôlée 1.....	79
Figure 3.12 : Illustration du principe de la mutation contrôlée.....	80
Figure 3.13.a : Individu avant la mutation.....	80
Figure 3.13.b : Individu après la mutation.....	80
Figure 3.14 : Principe d'évolution des solutions de la 1 ^{ère} approche de résolution du 1-PDPTW.....	81

Figure 3.15 : Structure de l'algorithme de la 1 ^{ère} approche évolutionniste pour le 1-PDPTW	81
Figure 3.16 : Illustration du théorème de Chu	86
Figure 3.17 : l'algorithme de croisement.	90
Figure 3.18 : Structure de l'algorithme de la 2 ^{ème} approche évolutionniste pour le 1-PDPTW	91
Figure 3.19 : L'heuristique d'échange dédiée au 1-PDPTW	92
Figure 3.20 : L'algorithme de la recherche Tabou spécialisée	93
Figure 4.1 : Problème de tournée dynamique	99
Figure 4.2 : Différents problèmes ayant les mêmes valeurs <i>dod</i> .	102
Figure 4.3 : Classification du DVRP	104
Figure 4.4 : Les méthodes de résolution du DVRP et de ses variantes.	106
Figure 4.5 : L'heuristique mod TSP	107
Figure 4.6 : L'heuristique GEN	108
Figure 4.7 : L'heuristique d'insertion classique	111
Figure 4.8 : Simulateur du système dynamique [Larsen 2001].	111
Figure 4.9 : Le Pseudo-code de la recherche Tabou adaptative parallèle [Gendreau et al1998b] [Gendreau et al 1999].	114
Figure 4.10 : La tournée du véhicule avant l'apparition de la demande dynamique urgente.	116
Figure 4.11 : La tournée du véhicule après l'apparition de la demande dynamique urgente.	116
Figure 4.12 : Stratégie de résolution du 1-DPDPTW	117
Figure 4.13 : Structure de l'algorithme de la 1 ^{ère} approche évolutionniste pour le 1-DPDPTW	118
Figure 4.14 : Principe des heuristiques de génération de la population initiale pour le 1-DPDPTW	119
Figure 4.15 : Algorithme de l'heuristique 1.	119
Figure 4.16.a : Correction des individus vis-à-vis des contraintes de précédence dans un 1-DPDPTW.	121
Figure 4.16.b : Individu non viable vis-à-vis des contraintes de précédences dans un 1-DPDPTW.	121
Figure 4.16.c : Individu viable vis-à-vis des contraintes de précédence dans un 1-DPDPTW.	121
Figure 4.17.a : Individu non viable vis-à-vis des contraintes de capacité dans un 1-DPDPTW.	122
Figure 4.17.b : Individu viable vis-à-vis des contraintes de capacité dans un 1-DPDPTW.	122

Figure 4.17.c : Correction des individus vis-à-vis des contraintes de capacité dans un 1-DPDPTW.	123
Figure 4.18.a : Croisement pour le 1_DPDPTW.	125
Figure 4.18.b : l’algorithme de croisement pour le 1-DPDPTW.	125
Figure 4.19 : L’algorithme de la recherche Tabou spécialisée pour le 1-DPDPTW.	126
Figure 5.1 : Le gestionnaire des demandes dynamiques.	132
Figure 5.2 : Jeux de tests de Li et Lim pour le PDPTW.	134
Figure 5.3 : Jeux de tests de Li et Lim pour le PDPTW.	134
Figure 5.4 : Jeux de tests élaborés pour le 1-PDPTW.	135
Figure 5.5 : Jeux de tests élaborés pour le 1-PDPTW.	135
Figure 5.6 : L’instance PDT10_1 élaborée.	136
Figure 5.7 : Convergence du critère global pour le problème pour le problème LRC105 résolu par AEs_1.	138
Figure 5.8 : Convergence du critère global pour le problème pour le problème PDT30 résolu par AEs_1.	138
Figure 5.9 : Evolution de la valeur de la solution avec la recherche Tabou spéciale pour le problème LRC102.	139
Figure 5.10 : Convergence du critère global pour le problème LRC205 résolu par AEs_2.	140
Figure 5.11 : Efficacité des approches pour la minimisation de F_1	145
Figure 5.12 : Efficacité des approches AEs_2 et AEHs_2 pour la minimisation de F_2	145
Figure 5.13 : L’instance PDT10d_1 pour le 1-DPDPTW.	146
Figure 5.14 : Efficacité des approches AEd_1 et AEHd_1 pour la minimisation de F_1	150
Figure 5.15 : Evolution de l’écart statique/dynamique.	151
Figure 5.16 : Evolution de l’écart statique/dynamique en pourcentage.	151
Figure 5.17 : Evolution du temps de calcul en fonction du degré de difficulté.	155

Introduction Générale

De nos jours, le transport occupe une place importante dans la vie des sociétés modernes. Compte tenu de leurs objectifs économiques et environnementaux, ces sociétés ont consacré des efforts importants à la recherche scientifique, la mise à niveau et l'amélioration des systèmes de transport et de leur sécurité. Le transport terrestre est un parmi les nombreux problèmes de transport qui les préoccupent le plus. Avec les contraintes spatiales, temporelles et économiques qu'impliquent ces problèmes, leur résolution devient très vite extrêmement difficile. Pour cette raison plusieurs scientifiques et plusieurs laboratoires de recherche se sont spécialisés dans la résolution des problèmes de transport.

Ces recherches se sont principalement orientées vers le transport de personnes, de biens, de marchandises, de produits dangereux et vers les transports d'urgence. Elles consistent essentiellement en l'élaboration de tournées de véhicules (**VRP** : Vehicule Routing Problem). Le VRP est un problème classique qui consiste à construire des routes visitant tous les clients en minimisant le coût du transport, en satisfaisant les demandes de ces derniers et en respectant les différentes capacités des véhicules.

Dans nos travaux, nous nous intéressons à une importante variante du VRP qui est le problème de collecte et distribution à fenêtres de temps avec un seul véhicule (1-PDPTW : Pickup and Delivery Problem with Time Windows). Le 1-PDPTW est traité dans ses deux versions :

- Le 1-PDPTW statique : dans lequel le véhicule doit transporter, sans dépasser sa capacité, des biens de fournisseurs à leurs clients respectifs en respectant leurs fenêtres de temps.
- Le 1-PDPTW dynamique : qui renferme toutes les propriétés du 1-PDPTW statique auxquelles est ajoutée une caractéristique dynamique qui concerne l'apparition de nouveaux clients à satisfaire après que le véhicule ait commencé sa tournée.

Notre but est donc de concevoir et de construire un outil de résolution du 1-PDPTW qui permettra d'offrir à l'utilisateur un ensemble de bonnes solutions. Pour ce faire, nous utilisons les algorithmes évolutionnistes et la recherche Tabou comme outils d'optimisation. A notre connaissance, aucun outil combinant ces deux méthodes de résolution n'a été utilisé pour traiter le 1-PDPTW statique et dynamique. Nous avons ainsi développé plusieurs méthodes de résolution utilisant ces deux approches. Les performances de ces outils ont été testées sur des benchmarks issus de la littérature du PDPTW et sur d'autres que nous avons spécialement conçues pour le PDPTW à un seul véhicule.

Ce mémoire, constitué de cinq chapitres, débute par une présentation générale des problèmes de tournées de véhicules. Nous y présentons, en première partie, le VRP et son état de l'art. Nous exposons alors sa formulation mathématique, ses différentes variantes et les différents outils et approches utilisés pour sa résolution. En deuxième partie, nous détaillons le PDPTW en montrant sa formulation mathématique ainsi que son état de l'art.

Le deuxième chapitre présente, les principes des outils et des approches adoptés pour la résolution du 1-PDPTW. Nous y détaillons en première partie le principe des algorithmes évolutionnistes, leurs caractéristiques, les opérateurs génétiques utilisés, les codages possibles et leurs domaines d'application. Ensuite, en deuxième partie, nous exposons la méthode de la recherche Tabou en présentant son principe général, la notion de voisinage, le concept des mouvements et la liste Tabou et sa gestion.

Dans le chapitre suivant, nous détaillons le problème 1-PDPTW adopté en présentant sa formulation mathématique. Nous y présentons aussi :

- la stratégie d'évaluation basée sur la Pareto Optimalité,
- la première approche évolutionniste conçue en détaillant les nouveaux opérateurs génétiques conçus destinés à l'amélioration des solutions et des temps de calcul,
- la deuxième approche évolutionniste basée sur l'utilisation de bornes inférieures pour homogénéiser les critères à optimiser,
- l'étape d'hybridation qui consiste en une recherche Tabou utilisant une heuristique d'échange spécialement conçue pour le 1-PDPTW et appliquée pour améliorer les résultats des précédentes approches évolutionnistes.

Le quatrième chapitre détaille l'adaptation de la première approche évolutionniste et de la recherche Tabou pour qu'elle puisse résoudre le cas dynamique du 1-PDPTW. Nous présentons alors les modifications apportées aux outils utilisés par ces méthodes comme l'opérateur de croisement, les opérateurs de mutation et l'heuristique d'échange de la recherche Tabou.

Le dernier chapitre de ce mémoire présente les résultats des simulations des différentes approches de résolution du 1-PDPTW statique et dynamique. Afin de mettre en valeur l'apport et l'efficacité de ces approches, nous donnons plusieurs résultats obtenus en utilisant des benchmarks issus de la littérature ainsi que certains benchmarks spécialement conçus pour le 1-PDPTW.

Nous présentons pour clôturer ce mémoire quelques conclusions qui émanent de nos travaux et les perspectives à envisager pour améliorer nos résultats.

Chapitre 1

Les tournées de véhicules : état de l'art, problèmes et méthodes de résolution

1.1 Introduction

Après presque cinquante années de recherche, le problème d'élaboration de tournées de véhicules introduit par [Dantzig et al 1959] demeure un des nombreux défis les plus liés à la vie courante, ce qui justifie que de nombreuses équipes de recherches s'y investissent. Communément appelé VRP (Vehicle Routing Problem), le principe général de ce problème consiste, en optimisant un ou plusieurs critères, à construire un ensemble de tournées, pour un nombre fini de véhicules, commençant et finissant à un dépôt. Dans ces tournées, un client doit être desservi une seule fois par un seul véhicule et la capacité de transport d'un véhicule pour une tournée ne doit pas être dépassée.

Durant ces années de recherche d'autres problèmes, dérivés du VRP, ont fait leurs apparitions comme le CVRP (Capacited VRP), le VRPTW (VRP with Time Windows), le PDPTW (Pickup and Delivery Problem with Time Windows) qui fera l'objet de nos travaux et bien d'autres que nous présenterons tout au long de ce chapitre.

Dans ce qui suit, nous présenterons, en première partie la modélisation mathématique générale du VRP et de ses principaux problèmes dérivés. Ensuite, nous présenterons les méthodes de résolution du VRP les plus connues. Enfin, nous détaillerons le PDPTW en présentant sa modélisation mathématique ainsi que l'état de l'art de ce problème.

1.2 Modélisation mathématique

Le VRP peut être un problème de collecte et ou de distribution. Dans tous ces cas, chaque client est affecté à une tournée assurée par un seul véhicule. Une solution à ce problème consiste à confectionner l'ensemble des trajets assurant la visite des clients. Les trajets proposés doivent respecter les différentes capacités des véhicules et satisfaire les quantités demandées par les clients ou à livrer à ces derniers.

Le critère le plus utilisé dans ce genre de problème est la minimisation de la distance totale parcourue par un nombre minimum de véhicules.

Dans la suite, nous présentons une modélisation mathématique pour le CVRP (Capacited Vehicle Routing Problem) proposée par Rego dans [Rego et al 1994].

Soit $G = (N, A)$ un graphe où $N = \{1, \dots, n\}$ est un ensemble de sommets avec le sommet 1 fixé comme dépôt et $A = \{(i, j) \mid i, j \in N \text{ et } i \neq j\}$ est l'ensemble des arcs.

$N' = N \setminus \{1\}$ est l'ensemble des clients à satisfaire obtenu par le fait d'enlever le dépôt de l'ensemble des sommets.

A chaque arc est associé un coût non négatif c_{ij} qui peut représenter le coût du voyage ou la durée du voyage entre i et j .

Nous envisageons le cas où une flotte de m véhicules ayant la même capacité de transport D (véhicules homogènes) est disponible.

Les contraintes prises en compte dans cette formulation :

- Une contrainte de capacité : A chaque sommet i de N' est associé un poids d_i non négatif représentant la demande, la somme des poids d'une tournée ne doit pas dépasser la capacité D du véhicule.
- Une contrainte de temps total : Le temps total d'une tournée ne doit pas dépasser une borne T . Ce temps est la somme des temps des voyages entre les sommets et les temps d'arrêt à ces derniers.

Une classification des problèmes de tournées de véhicules a été proposée par [Bodin et al. 1981] et [Desrochers et al. 1990]

La formulation de Rego et Roucairol dans [Rego et al 1994] adopte les notations suivantes :

Les constantes de données :

n = nombre de sommets,

m = nombre de véhicules,

D = capacité d'un véhicule,

T_k = temps maximal de la tournée du véhicule k ,

d_i = demande du sommet i , ($d_1 = 0$)

t_i^k = temps nécessaire au véhicule k pour charger ou décharger au sommet i ,

t_{ij}^k = temps nécessaire au véhicule k pour voyager du sommet i au sommet j ,

c_{ij} = coût ou distance du voyage du sommet i au sommet j .

Les variables de décision :

$$x_{ij}^k = \begin{cases} 1 & \text{Si le véhicule } k \text{ voyage du sommet } i \text{ au sommet } j \\ 0 & \text{Sinon} \end{cases}$$

Avec $X^k = (x_{ij}^k)$

La fonction à optimiser :

$$\text{Minimiser } \sum_{i=1}^n \sum_{j=1}^n c_{ij} \sum_{k=1}^m x_{ij}^k \quad (1.1)$$

sous

$$\sum_{i=1}^n \sum_{k=1}^m x_{ij}^k = 1 \quad j = 2, \dots, n \quad (1.2)$$

$$\sum_{j=1}^n \sum_{k=1}^m x_{ij}^k = 1 \quad i = 2, \dots, n \quad (1.3)$$

$$\sum_{i=1}^n x_{ip} - \sum_{j=1}^n x_{pj} = 0 \quad k = 1, \dots, m ; p = 1, \dots, n \quad (1.4)$$

$$\sum_{i=1}^n d_i \left(\sum_{j=1}^n x_{ij}^k \right) \leq D \quad k = 1, \dots, m \quad (1.5)$$

$$\sum_{i=1}^n t_i^k \sum_{j=1}^n x_{ij}^k + \sum_{i=1}^n \sum_{j=1}^n t_{ij}^k x_{ij}^k \leq T_k \quad k = 1, \dots, m \quad (1.6)$$

$$\sum_{j=2}^n x_{1j}^k \leq 1 \quad k = 1, \dots, m \quad (1.7)$$

$$\sum_{i=2}^n x_{i1}^k \leq 1 \quad k = 1, \dots, m \quad (1.8)$$

$$X^k \in S \quad (1.9)$$

Où S est proposée dans [Laporte 1992] de la façon suivante :

Sachant que Q est un ensemble de sommets visités par un seul véhicule.

$$S = \left\{ (x_{ij}^k) \mid \sum_{i \in Q} \sum_{j \notin Q} x_{ij}^k \geq 1, \forall Q \subset N, |Q| \geq 2; k = 1, \dots, m \right\} \quad (1.10)$$

$$S = \left\{ (x_{ij}^k) \mid \sum_{i \in Q} \sum_{j \in Q} x_{ij}^k \leq |Q| - 1, \forall Q \subset N, |Q| \geq 2; k = 1, \dots, m \right\} \quad (1.11)$$

$$S = \left\{ (x_{ij}^k) \mid \begin{array}{l} y_i^k - y_j^k + (n-1)x_{ij}^k \leq n-2, i, j = 1, \dots, n; i \neq j; k = 1, \dots, m \\ 1 \leq y_i^k \leq n-1, i = 2, \dots, n \end{array} \right\} \quad (1.12)$$

La fonction objectif (1.1) consiste à minimiser le coût total de transport.

Les équations (1.2) et (1.3) assurent que chaque sommet ne soit servi qu'une seule fois par un et un seul véhicule.

L'équation (1.4) assure la continuité d'une tournée par un véhicule : le sommet visité doit impérativement être quitté.

L'équation (1.5) assure le respect de la contrainte de capacité du véhicule.

L'équation (1.6) assure le respect de la contrainte de la durée totale d'une tournée.

Les équations (1.7) et (1.8) assurent le non dépassement de la disponibilité d'un véhicule. Un véhicule ne sort du dépôt et n'y revient qu'une seule fois.

Finalement les équations (1.9), (1.10), (1.11) et (1.12) assurent le respect des contraintes d'élimination des sous tours (tours revenant au client et non pas au dépôt).

La figure 1.1 suivante montre un exemple de tournées avec quatre véhicules. Ces derniers assurent la visite de tous les clients tout en partant et revenant au dépôt.

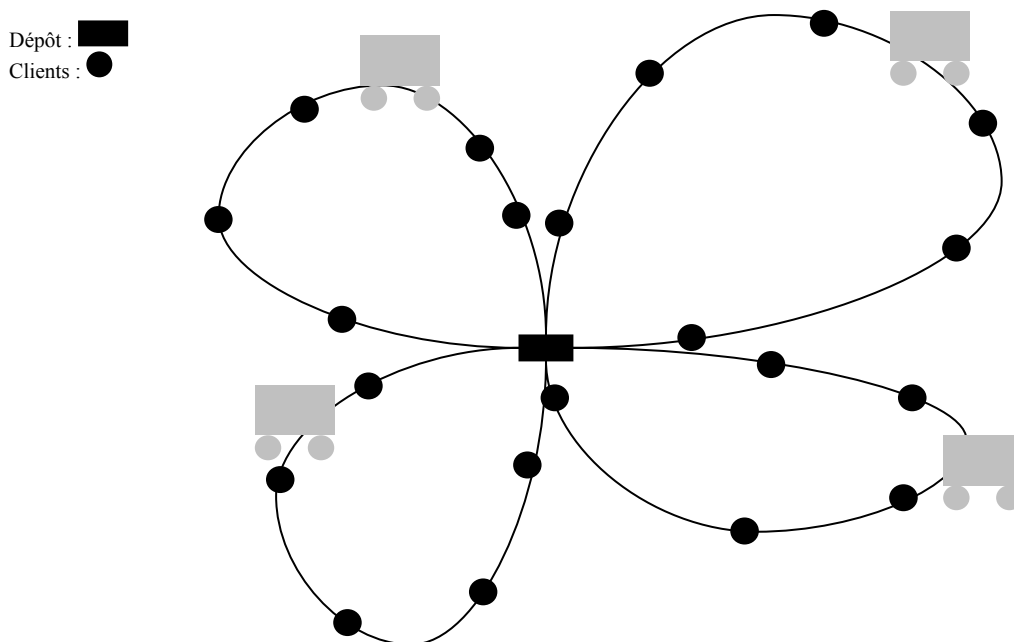


Figure 1.1 : Exemple de tournées avec 4 véhicules.

Outre, la conservation de l'intégrité du problème, le fait d'ajouter, de modifier ou de supprimer des contraintes peut nous faire passer d'un problème à un autre problème dérivé du VRP. Par exemple, la modélisation mathématique précédente est celle d'un CVRP ; en enlevant les contraintes (1.5) et (1.6) nous obtenons une modélisation d'un VRP.

1.3 Le VRP et ses variantes

Durant des années de recherches sur le VRP d'autres dérivées de ce problème sont apparues. Ces apparitions sont dues essentiellement aux activités des chercheurs qui travaillent de plus en plus sur les problèmes de transport et de distribution que rencontrent les sociétés. Dans ce qui suit nous allons présenter les principaux problèmes dérivés du VRP.

- **CVRP** (Capacitated Vehicle Routing Problem)

Comme défini précédemment, un problème CVRP consiste à affecter chaque client à une tournée effectuée par un seul véhicule de capacité finie. Ce véhicule commence et termine sa tournée au dépôt [Ralphs et al. 2001].

- **DVRP** (Dynamic Vehicle Routing Problem)

Contrairement au VRP classique où toutes les demandes des clients sont connues et planifiées à l'avance, le DVRP illustre mieux la réalité. Dans les problèmes de transport réels dans la société d'aujourd'hui la composante dynamique est très importante, elle s'exprime surtout par les apparitions de nouveaux clients, de nouvelles demandes, ou de pannes des véhicules de transport. Dans ce genre de situation, le décideur doit réorganiser, en un temps acceptable, la planification des itinéraires de sa flotte de véhicules pour répondre aux nouvelles demandes urgentes [Kilby et al. 1998] [Gendreau et al 1998a].

- **VRPTW** (Vehicle Routing Problem with Time Windows)

Le VRPTW est un des problèmes les plus étudiés. Dans un VRPTW, de nouvelles contraintes temporelles sont ajoutées : chaque client doit être servi dans un intervalle de temps durant lequel il est disponible pour être visité [Cordeau et al. 2000].

- **DVRPTW** (Dynamic Vehicle Routing Problem with Time Windows)

Ce problème est une fusion des deux derniers. Outre les nouvelles contraintes temporelles de fenêtres de temps, le DVRPTW assume aussi la composante dynamique d'arrivées d'événements urgents (nouvelle demande, nouveau client, panne d'un véhicule,...) [Gendreau et al. 1999].

- **VRPB** (Vehicle Routing Problem with backhauls)

Ce problème est une prolongation du VRP classique. Il comporte des clients auxquels des produits doivent être livrés et des fournisseurs dont les marchandises doivent être amenées au dépôt. Donc, pour éviter de réarranger les produits transportés

dans le véhicule sur une tournée, les clients doivent être visités avant les fournisseurs [Jacobs-Blecha et al. 1998].

- **PVRP** (Periodic Vehicle Routing Problem)

Le problème de tournées de véhicules multi périodique (PTVMP) consiste à livrer pour un ensemble de clients, la quantité demandée d'un ou de plusieurs produits sur un horizon de temps donné. Dans ce problème, la quantité de produits livrée à un client, permet à ce dernier de subvenir à ses besoins en attendant la prochaine visite du véhicule. Le but principal de ce problème se divise en deux parties : la première consiste à planifier les horaires de livraison de chaque client sur un horizon de temps prédéterminé ; la deuxième consiste à organiser les tournées des véhicules afin d'effectuer les livraisons nécessaires tout en optimisant le coût total de transport.

Outre la connaissance et l'utilisation des contraintes connues d'un problème de transport, le PVRP nécessite l'intégration des éléments de gestion de stock afin de respecter les besoins en quantité et en temps des clients [Witucki et al. 2001].

- **SVRP**

Un problème VRP est dit stochastique si au moins un de ses éléments est aléatoire, c'est-à-dire avec une certaine incertitude. Ces éléments peuvent être une ou plusieurs demandes des clients, les temps ou les coûts de transport,... Le problème avec les demandes stochastiques est celui le plus étudié dans la littérature. La majorité des recherches suppose que les demandes aléatoires suivent une loi de distribution « normale » [Groth 2002].

- **TSP** (Travelling Salesman Problem) (Problème du voyageur de commerce)

Le problème du voyageur de commerce consiste à trouver le chemin le plus court pour passer par l'ensemble des villes données une fois et une seule et revenir à la ville de départ. C'est donc un cas particulier du VRP sans contrainte de capacité et avec un seul véhicule [Rego et al. 1994].

- **VRPPD** (Vehicle Routing Problem with Pickup and Delivery)

Le problème de collecte et de livraison a les mêmes propriétés que le VRP. Il ajoute à celles-ci le fait que chaque client introduit deux positions géographiquement différentes : la première pour le ramassage du produit et la deuxième pour la livraison de ce dernier. Ceci va directement induire une contrainte de précédence à ajouter au

problème du VRP classique sachant que, dans une tournée, chaque opération de livraison doit être précédée par l'opération de ramassage respective [Mechti 1995].

- **MDVRP** (Multi-Depot Vehicle Routing Problem)

Dans ce type de problème, plusieurs dépôts géographiquement distribués existent. Une tournée, dans ce genre de problème, est assurée par un véhicule qui part et revient au même dépôt initial [Fischetti et al. 1994].

- **TRP** (Travelling Repairman Problem)

Le problème de tournée du dépanneur est une variante du VRP. Généralement l'objectif des méthodes qui traitent ce problème est de minimiser le temps de latence, qui est la somme des temps d'attente des clients avant qu'ils ne soient servis, qui n'est autre que la somme pondérée des temps de service des clients [Krumke et al 2001].

- **VRPHF** (Vehicle Routing Problem with Heterogeneous Fleet)

La seule différence entre un VRPHF et un VRP est que la flotte de véhicule est hétérogène. Une flotte de véhicules hétérogènes est composée de véhicules qui sont de types différents. Les véhicules peuvent être différenciés par leurs coûts de transport, leurs capacités de transport, leurs vitesses, leurs tailles, ... [Taillard 1999] [Prins 2002].

- **OVRP** (Open Vehicle Routing Problem)

La différence entre un OVRP et un VRP est que dans le premier les véhicules ne sont pas tenus de retourner au dépôt. Dans le cas où il le sont, il rebroussement chemin en revisitant les clients qui leurs sont affectés dans l'ordre inverse. C'est pourquoi les parcours des véhicules sont tous des chemins ouverts [FU et al 2003].

- **SDVRP** (Split Delivery Vehicle Routing Problem)

Dans ce genre de problème, la demande du client peut être satisfaite sur plusieurs tournées ; pour cela un client peut être visité plusieurs fois si cela est nécessaire. Dans ce type de problème, et contrairement aux autres, la demande d'un ou plusieurs clients peut excéder la capacité du véhicule [Archetti et al. 2002].

- **PDPTW** (Pickup and Delivery Problem with Time Windows)

Ce problème est une variante du VRPTW. Outre l'existence des contraintes de fenêtres de temps, ce problème possède un ensemble de clients et un ensemble de fournisseurs. A chacun de ces clients correspond un et un seul fournisseur. Les

véhicules ne doivent alors passer par un client qu'après avoir visité son fournisseur [Psaraftis 1983a]. Nous allons présenter ce problème plus en détails dans la suite.

1.4 Le VRP et ses variantes : des problèmes NP-difficiles

Un problème NP-difficile est un problème dont l'existence d'un algorithme déterministe exact qui puisse le résoudre en un temps polynomial est peu probable. Vu que le VRP est NP-difficile [Savlesbergh 1995b], ses extensions caractérisées par des contraintes additionnelles sont aussi des problèmes NP-difficiles.

1.5 Les méthodes de résolution du VRP et de ses variantes

Les méthodes de résolution consacrées aux VRP et à ses variantes sont principalement classées en deux catégories : les méthodes exactes et les méthodes approchées. La figure 1.2 suivante résume la plupart de ces méthodes.

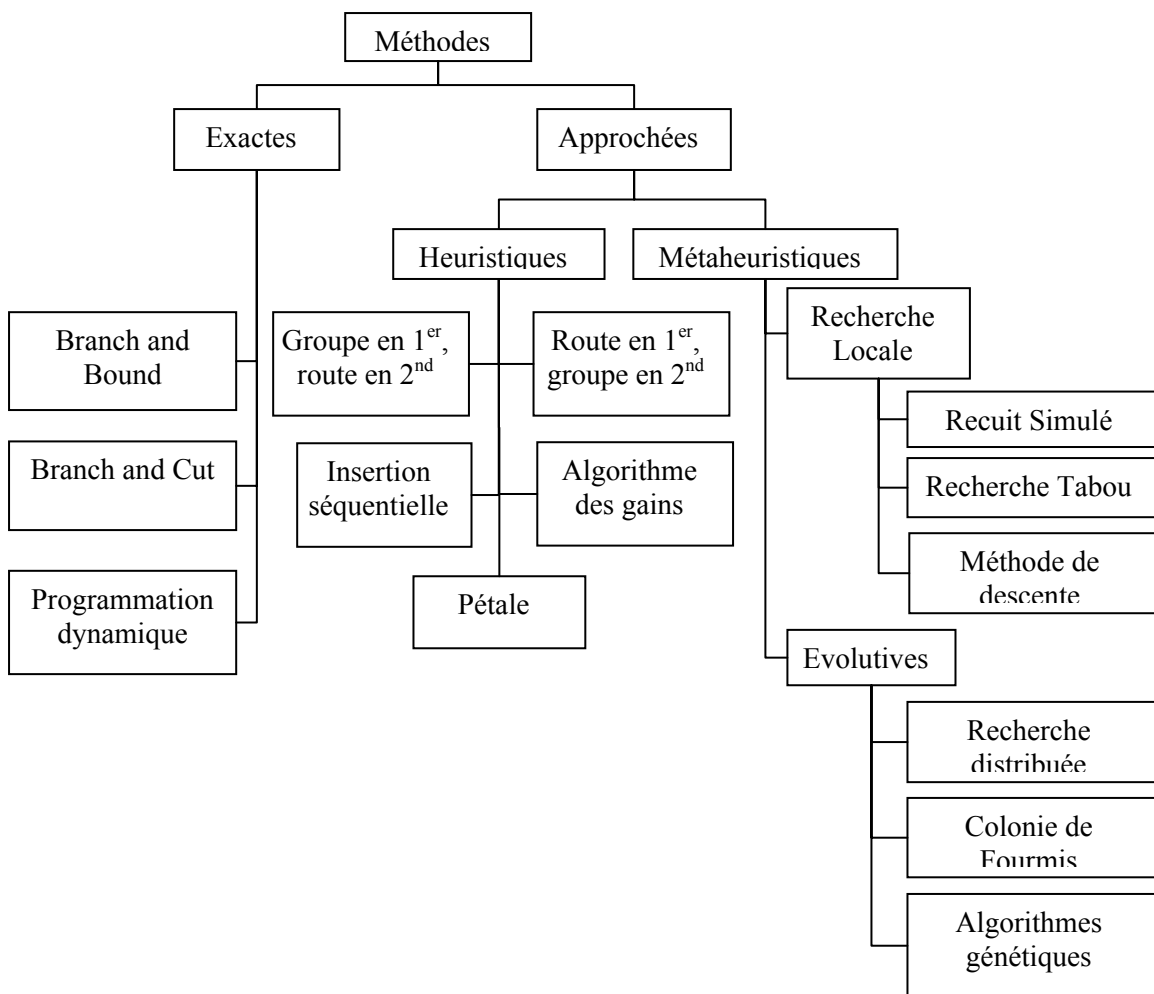


Figure 1.2 : Les principales méthodes de résolution du VRP.

1.5.1 Les méthodes exactes

Les méthodes exactes reposent sur l'utilisation d'algorithmes qui mènent de façon sûre vers la solution optimale. Le principe essentiel de ces méthodes est d'énumérer de manière implicite l'ensemble des solutions de l'espace de recherche. Malgré l'important temps de calcul que nécessitent, généralement, ces approches, plusieurs méthodes ont été développées. Elles permettent de résoudre efficacement des problèmes allant jusqu'à 50 clients. En 2001 une méthode résolvant un problème contenant 100 clients a été proposée dans [Ralphs 2002]

Parmi ces méthodes on peut citer :

1.5.1.1 La méthode de Branch and Bound

Connue aussi sous le nom de méthode de la recherche arborescente ou de la procédure de séparation et d'évaluation, cette méthode consiste en la construction d'un arbre de recherche qui sera exploré de manière à éviter les branches inutiles qui sont des branches contenant des solutions non intéressantes ou carrément non réalisables. L'exploration se fait avec des évaluations des branches et des comparaisons avec une valeur seuil du critère à optimiser. Cette technique donne de bons résultats pour les problèmes d'ordonnancement de petites tailles, mais dans le cas contraire, elle risque de générer des branches très étendues.

1.5.1.2 La méthode de Branch and cut

Elle est aussi appelée méthode de programmation en nombres entiers. Comme toute méthode énumérative implicite, l'algorithme construit une arborescence nommée *l'arbre du "Branch and Cut"*, les sous-problèmes qui forment l'arbre sont appelés *des nœuds*. Il existe trois types de nœuds dans l'arbre du "Branch and Cut", le nœud courant qui est en train d'être traité, les nœuds actifs qui sont dans la liste d'attente des problèmes et les nœuds inactifs qui ont été élagués au cours du déroulement de l'algorithme. Le principe est de partir d'une solution admissible entière du problème, et à l'aide du simplexe par exemple, d'aller vers une autre solution admissible entière jusqu'à l'optimum.

1.5.1.3 La programmation dynamique

Cette méthode se base sur le principe de Bellman : «Si C est un point qui appartient au chemin optimal entre A et B, alors la portion de ce même chemin allant de A à C est le chemin optimal entre A et C . » [Borne et al 1990]. Pour obtenir le chemin optimal du problème, il suffit donc de construire les différents sous chemins optimaux. Cette méthode a été utilisée dans [Rego et al. 1994] pour la résolution de problèmes allant de 10 à 25 clients.

1.5.2 Les heuristiques

Vu que les méthodes exactes restreignent le nombre des clients envisageables dans les problèmes et impliquent, dans la plupart des cas, un temps de calcul important, l'élaboration et l'utilisation des heuristiques se sont avérées d'une grande utilité. Ces méthodes permettent de gérer des problèmes de grandes tailles avec des temps de résolution et des résultats acceptables et admissibles.

Parmi les heuristiques qui traitent le VRP, nous citons :

- le groupe en premier, la route en second,
- la route en premier, le groupe en second,
- l'insertion séquentielle,
- l'algorithme des gains,
- l'algorithme en Pétale.

1.5.2.1 L'heuristique « groupe en premier, route en second »

C'est une des heuristiques les plus connues. Elle se base sur l'aspect géométrique du problème. Elle consiste à créer des groupes de clients qui sont géographiquement voisins puis à confectionner les tournées à l'intérieur de ces derniers [LeBouthillier 2000].

1.5.2.2 L'heuristique « route en premier, groupe en second »

Le principe de cette heuristique est de construire des tournées comportant un grand nombre de clients, qui sont réellement non réalisables, puis de les subdiviser en de petites tournées pour obtenir des solutions acceptables pour le VRP [Bodin et al. 1979] [Golden et al. 1982].

1.5.2.3 L'algorithme de gain

L'algorithme des gains de Clarke et Wright (1964) [Clarke et al. 1964] est une des méthodes les plus connues pour le VRP. Dans une version parallèle de cet algorithme, où le nombre de véhicules est une variable de décision, ce dernier nécessite le calcul du gain possible sur l'arc (i,j) et la fusion de route comme suit :

Créer n routes $(0,i,0)$ pour $i = 1, \dots, n$. (Ces routes partent du dépôt 0 et y retournent en passant par les clients i). Evaluer le gain de chaque fusion de routes possibles. Pour évaluer

le gain d'une fusion de route on calcul $S_{ij} = c_{i0} + c_{0j} - c_{ij}$ pour $i, j = 1, \dots, n$ et $i \neq j$, sachant que c_{ij} est le coût de l'arc (i, j) .

Trier les gains en ordre décroissant et choisir la fusion qui permet d'avoir le gain le plus important.

Pour étendre les routes par fusion, considérer tour à tour chaque route $(0, i, \dots, j, 0)$ pour déterminer le premier gain S_{ki} ou S_{j1} qui peut être utilisé pour former une route réalisable, par la fusion de la route courante et d'une autre se terminant par $(k, 0)$ ou commençant par $(0, l)$. Ainsi l'algorithme fonctionne comme suit :

Pour un gain S_{ij} donné, trouver s'il existe deux routes, une partant de $(0, j)$ et l'autre se terminant à $(i, 0)$ qui puissent être fusionnées pour former une route réalisable. Dans ce cas, combiner ces deux routes en effaçant $(0, j)$ et $(i, 0)$ et en introduisant (i, j) . Lorsqu'il n'y a plus de fusion qui améliore le gain, l'algorithme s'arrête.

1.5.2.4 L'algorithme en pétale

Cet algorithme de balayage génère plusieurs routes, appelées pétales, pour ensuite faire une sélection en résolvant un problème de partitionnement. Notons que si les routes correspondent à des secteurs continus des arcs, alors le problème peut être résolu en temps polynomial [Ryan et al. 1993].

1.5.3 Les métaheuristiques

Le but des métaheuristiques est similaire à celui des heuristiques : obtenir des solutions de bonne qualité en un temps raisonnable. Cependant, contrairement à une heuristique, l'algorithme et l'emploi d'une métaheuristique sont totalement indépendants du problème à traiter.

Dans ce qui suit, nous allons présenter les métaheuristiques les plus prisées par les chercheurs. Leurs différentes utilisations et les améliorations qu'elles ont connues ont fait que les résultats établis ne cessent de s'améliorer.

Les métaheuristiques sont subdivisées en deux grandes familles :

- les métaheuristiques de recherche locale :
 - la méthode de la descente ;
 - la recherche Tabou.

- les métaheuristiques d'évolution :
 - les algorithmes génétiques ;
 - la recherche distribuée ;
 - les colonies de fourmis.

1.5.2.5 Les métaheuristiques de recherche locale

Le principe de ces méthodes n'est plus de construire un ordonnancement à partir des données initiales du problème, mais de modifier les résultats d'un ordonnancement admissible en vue d'améliorer la valeur de la fonction objectif. Ces méthodes utilisent la notion de voisinage de solutions, on parle alors de méthodes par voisinage. Pour pouvoir construire une méthode par voisinage, il faut disposer d'une solution initiale complète, d'une fonction $f(x)$ qui définit la valeur du critère pour une solution x et d'une fonction voisinage $v(x)$ qui permet de calculer les fonctions voisines de x [Chérif 2002]. Parmi ces méthodes, nous pouvons citer la *Méthode de descente*, la *Méthode de Recuit Simulé* et la *Méthode Tabou*.

1.5.2.5.1 Méthode de descente

Appelée aussi méthode de recherche locale, cette méthode se base sur le choix d'une solution voisine meilleure à chaque itération. Si le voisin n'est pas meilleur que la solution courante, un autre ensemble de voisins est créé. Ce processus de voisinage sera répété jusqu'à l'obtention d'une solution satisfaisant au(x) critère(s) imposé(s) ou une stagnation prolongée de la valeur de la solution [Widme 2001]. Pour ce faire, cette méthode utilise des heuristiques pour effectuer des permutations entre arcs appartenant à une même tournée ou à des tournées différentes.

Les heuristiques λ -opt sont les heuristiques de recherche locale les plus connues et les plus utilisées. Elles ont été utilisées dans [Lin 1965] pour la résolution du VRP.

1.5.2.5.2 Méthode de Recuit Simulé

Le recuit simulé [Kirkpatrick 1983] est une métaheuristique inspirée d'un processus utilisé en métallurgie, qui alterne des cycles de refroidissement lent et de réchauffage (recuit) qui tendent à minimiser l'énergie du matériau. Cette méthode s'appuie sur l'algorithme de Metropolis [Metropolis et al 1953] qui permet de décrire l'évolution d'un système thermodynamique (voir Figure 1.3). Par analogie avec le processus physique, la fonction $f(x)$ à minimiser deviendra l'énergie E du système où x est un état donné de la matière. Un paramètre fictif est également introduit : la température T du système.

Partant d'une solution donnée nous générons une solution voisine en utilisant une transformation qui change x en $s(x)$. Si celle-ci améliore le critère que l'on cherche à optimiser, c'est-à-dire $\Delta f = f(s(x)) - f(x) < 0$, on dit alors qu'on a fait baisser l'énergie du système, sinon elle la dégrade. En acceptant une solution améliorant le critère, nous tendons ainsi à chercher l'optimum dans le voisinage de la solution de départ, alors que l'acceptation d'une solution moins bonne, c'est-à-dire $\Delta f = f(s(x)) - f(x) \geq 0$, permet alors d'explorer une plus grande partie de l'espace de solutions et tend à éviter de s'enfermer trop vite dans la recherche d'un optimum local. Dans ce dernier cas, la solution voisine $s(x)$ est acceptée avec

un probabilité p qui vaut $e^{\frac{-\Delta f}{T}}$.

```

SI ( $f(s(x)) \leq f(x)$ ) ALORS
     $f(x) = f(s(x))$ 
     $x = s(x)$ 
SINON
     $p = e^{\frac{-\Delta f}{T}}$ 
     $r =$  valeur aléatoire dans  $[0,1]$ 
    SI ( $r \leq p$ ) ALORS
         $f(x) = f(s(x))$ 
         $x = s(x)$ 
    FINSI
FINSI
    
```

Figure 1.3 : Pseudo-code de l'algorithme de Metropolis.

Dans cet algorithme, le paramètre de contrôle est la température T . Si la température est élevée la probabilité p tend vers 1 et presque tous les changements sont acceptés. Cette température diminue lentement au fur et à mesure du déroulement de l'algorithme pour simuler le processus de refroidissement des matériaux. Sa diminution est suffisamment lente pour que l'équilibre thermodynamique soit maintenu. Nous présentons dans la figure 1.4 le pseudo-code de l'algorithme du recuit simulé.

Le recuit simulé a été utilisé dans [Rego et al. 1994] pour traiter le problème VRP.

```

 $x$  = solution aléatoire
 $f_{\min} = f(x)$ 
 $x_{\min} = x$ 
Initialiser la température  $T$  à une valeur très élevée
REPETER
    Générer un voisin  $s(x) \in$  au voisinage de  $x$   $S(x)$ 
    SI  $f(x) < f_{\min}$  ALORS
         $f_{\min} = f(x)$  et  $x_{\min} = x$ 
    SINON
         $p = e^{\frac{-\Delta f}{T}}$ 
         $r$  = valeur aléatoire dans  $[0,1]$ 
        SI  $(r \leq p)$  ALORS
             $f(x) = f(s(x))$  et  $x = s(x)$ 
        Décroître la température  $T$ 
    FINSI
JUSQU'A (équilibre thermodynamique soit atteint)
    
```

Figure 1.4 : Pseudo-code de l'algorithme du Recuit Simulé.

1.5.2.5.3 Recherche Tabou

La recherche Tabou est une métaheuristique originalement développée par Glover [Glover 1989] [Glover 1990]. Elle est basée sur des idées simples, mais reste néanmoins efficace. Cette méthode combine une procédure de recherche locale avec un certain nombre de règles et de mécanismes lui permettant de surmonter l'obstacle des extremums locaux, tout en évitant les problèmes de cycles. Elle a été appliquée avec succès pour résoudre de nombreux problèmes difficiles d'optimisation combinatoire : problèmes de routage de véhicules, problèmes d'affectation quadratique, problèmes d'ordonnancement, problèmes de coloration de graphes, ...

Dans une première phase, la méthode de recherche Tabou peut être vue comme une généralisation des méthodes d'amélioration locale. En effet, en partant d'une solution quelconque x appartenant à l'ensemble de solutions X , nous nous dirigeons vers une solution $s(x)$ appartenant au voisinage $S(x)$ de x . L'algorithme explore donc itérativement l'espace de solutions X . Afin de choisir le meilleur voisin $s(x)$, l'algorithme évalue la fonction objectif f en chaque point $s(x)$, et retient le voisin qui améliore la valeur de f , ou celui qui la dégrade le moins.

L'originalité de la méthode de recherche Tabou, par rapport aux autres méthodes locales, réside dans le fait que l'on retient le meilleur voisin, même si celui-ci est plus mauvais que la solution dont il est le voisin direct. Pour cela, en autorisant les dégradations de la fonction objectif f l'algorithme évite, au mieux, d'être piégé dans un minimum local, mais il induit un risque de répétitions cycliques. En effet, lorsque l'algorithme a quitté un minimum quelconque par acceptation de la dégradation de la fonction objectif, il peut revenir sur ses pas aux itérations suivantes.

Pour pallier à ce problème, l'algorithme utilise une mémoire pour conserver pendant un moment la trace des dernières meilleures solutions déjà inspectées. Ces solutions sont déclarées *taboues*, d'où le nom de la méthode. Elles sont stockées dans une liste de longueur L donnée, appelée *liste Tabou*. Une nouvelle solution n'est acceptée que si elle n'appartient pas à cette liste Tabou. Ce critère d'acceptation d'une nouvelle solution évite le rebouclage de l'algorithme, durant la visite d'un nombre de solutions au moins égal à la longueur de la liste Tabou, et il dirige l'exploration de la méthode vers des régions du domaine de solutions non encore visitées.

Pour la résolution du VRP, cette méthode a été utilisée dans [Gendreau et al. 1994], [Taillard 1993], [Xu et al. 1996], [Rego et al. 1996] et finalement Taillard introduit un concept de mémoire adaptative [Taillard 1998] [Taillard et al 1998]. Dans la figure 1.5 nous présentons le pseudo-code de la recherche Tabou.

```

x = solution aléatoire
fmin = f(x)
xmin = x
TABOUE est vide
REPETER
    Générer N solutions / un voisin  $s_i(x) \in S(x)$  et  $\{x, s_i(x)\} \notin \text{TABOUE}$ 
     $f(s(x)) = \min_i [f(s_i(x))]$ ,  $1 \leq i \leq N$ 
    Ajouter  $(\{x, s_i(x)\}, \text{TABOUE})$ 
    SI  $f(x) < f_{\min}$  ALORS
         $f_{\min} = f(x)$  et  $x_{\min} = x$ 
    FINSI
JUSQU'A (satisfaction de la condition d'arrêt)
    
```

Figure 1.5 : Pseudo-code de la recherche Tabou.

Cette méthode sera plus détaillée dans le chapitre suivant.

1.5.2.6 Les métaheuristiques d'évolution

Le principe des métaheuristiques évolutives est de faire évoluer un ensemble de solutions vers l'optimum cherché. Cette évolution se fait à partir de transformations et de coopérations entre les individus qui représentent individuellement une solution de l'espace total du problème. Parmi ces méthodes, nous distinguons essentiellement les algorithmes génétiques, la recherche distribuée et les colonies de fourmis.

1.5.2.6.1 Les algorithmes génétiques

Les principes fondamentaux de ces algorithmes ont été exposés par [Holland 1975]. Ces algorithmes s'inspirent du fonctionnement de l'évolution naturelle des espèces, notamment la sélection de Darwin et la procréation selon les règles de Mendel. Ils ont été efficacement utilisés pour résoudre plusieurs problèmes d'optimisation multicritère [Coello 2001].

La sélection naturelle, que Darwin appelle l'élément "propulseur" de l'évolution, favorise les individus d'une population qui sont les mieux adaptés à un *environnement*. La sélection est suivie de la procréation, réalisée à l'aide de croisements et de mutations au niveau du patrimoine génétique des individus (ou "génotype") constitué d'un ensemble de gènes. Ainsi, deux individus "parents", qui se croisent, transmettent une partie de leur patrimoine génétique à leurs descendants. Le génotype de l'enfant fait que celui-ci est plus ou moins adapté à l'environnement. S'il est bien adapté, il a une plus grande chance de procréer dans la génération future. Au fur et à mesure des générations, on sélectionne les individus les mieux adaptés, et l'augmentation du nombre de ces individus fait évoluer la population entière.

Dans les algorithmes génétiques, nous simulons le processus d'évolution d'une population. On part d'une population initiale de N solutions du problème représentées par des individus judicieusement choisis. Le degré d'*adaptation* d'un individu à l'environnement est exprimé par la valeur de la fonction coût $f(x)$, où x est la solution que l'individu représente. On dit qu'un individu est d'autant mieux adapté à son environnement, que le coût de la solution qu'il représente est plus faible ou plus important selon le(s) critère(s) d'optimisation choisi(s). Au sein de cette population, intervient alors la *sélection* au hasard d'un ou deux parents, qui produisent une nouvelle solution, à travers les *opérateurs génétiques*, tels que le *croisement* et la *mutation*. La nouvelle population, obtenue par le choix

de N individus parmi les populations (parents et enfants), est appelée génération suivante. En itérant ce processus, on produit une population plus riche en individus mieux adaptés.

Le chapitre 2 fera l'objet d'une présentation plus approfondie des algorithmes génétiques.

1.5.2.6.2 La recherche distribuée

Partant du principe des algorithmes génétiques, cette méthode assure l'évolution des solutions à l'aide d'un opérateur de combinaison sans imposer de règles de codage. Enfin pour améliorer le résultat final, il est permis d'utiliser l'une des méthodes de la recherche locale [Esqui 2001].

1.5.2.6.3 Les colonies de fourmis

C'est une nouvelle méthode de résolution des problèmes d'ordonnement. Les colonies de fourmis sont basées sur le comportement réel de communication chez les fourmis qui consiste en « la trace » et « l'attrait ». Cette métaheuristique a été introduite pour la première fois dans [Colorni et al. 1992] et a été appliquée au problème du voyageur de commerce. [Gambardella et al. 2003] ont appliqué cette métaheuristique pour le VRP.

La figure 1.6 illustre le pseudo-code de l'algorithme des colonies de fourmis.

- **INITIALISER** les traces,
- **TANT QU'**un critère d'arrêt n'est pas satisfait
 - **REPETER** en parallèle pour chacune des p fourmis :
 - Construire une nouvelle solution à l'aide des informations contenues dans les traces et une fonction d'évaluation partielle
 - Evaluer la qualité de la solution
 - Mettre à jour les traces

Figure 1.6 : Pseudo-code de l'algorithme des colonies de fourmis.

1.6 Le problème PDPTW

Le problème que nous traitons dans ce travail est une extension du VRP qui est le PDPTW (Pickup and Delivery Problem with Time Windows). Ce problème sera étudié sous ses deux formes statique et dynamique. Nous pouvons rencontrer ce problème dans notre quotidien : les camions de transfert de fonds, le ramassage scolaire, la poste, le ramassage des déchets, ...

Ayant un ensemble de véhicules de même capacité (ou non), il s'agit de trouver une tournée (un ordonnancement) réalisable et de qualité qui permet de satisfaire toutes les demandes des couples fournisseur-client en respectant les différentes contraintes de capacité, de temps et de précédences qui peuvent être les suivantes :

- Les contraintes de capacité sont dues aux capacités limitées des véhicules. Ces capacités peuvent être impossibles à dépasser, et la contrainte de capacité sera une contrainte rigide, comme elles peuvent être dépassées tout en ayant, dans ce cas, une pénalité à assumer.
- Les contraintes de temps sont dues à la disponibilité limitée des clients. Ces contraintes sont appelées fenêtres de temps. Elles peuvent être impossibles à violer, et la contrainte de temps sera une contrainte rigide, comme elles peuvent être violées tout en ayant une pénalité à assumer.
- Les contraintes de précédences sont présentes pour garantir qu'un client ne soit pas visité avant son fournisseur.

Vu que le VRP est un problème NP-difficile [Savlesbergh 1995b], le PDPTW étant une extension de ce dernier à laquelle sont additionnées les différentes contraintes est aussi un problème NP-Difficile.

1.6.1 Formulation mathématique du problème

Nous présentons ici une formulation mathématique du PDPTW, nous supposons alors que :

- Un nœud n'est servi que par un seul véhicule et en une seule fois,
- Il y a un seul dépôt,
- Les contraintes de capacité doivent être respectées.
- Les contraintes de temps sont rigides concernant l'heure d'arrivée.
- Chaque véhicule commence le trajet du dépôt et y retourne à la fin.
- Un véhicule reste à l'arrêt à un nœud le temps nécessaire pour le traitement de la demande.
- Si un véhicule arrive au nœud i avant la date e_i de début de sa fenêtre, il attend.

Une formulation générale du PDP est donnée dans [Savlesbergh 1995a].

Nous considérons les variables suivantes :

Les variables de données :

N : Ensemble des nœuds clients, fournisseurs et dépôt,

N' : Ensemble des demandes clients, fournisseurs,

N^+ : Ensemble des nœuds fournisseurs,

N^- : Ensemble des nœuds clients,

K : Nombre de véhicules,

d_{ij} : distance euclidienne entre le nœud i et le nœud j , si $d_{ij} = \infty$ alors le chemin entre i et j n'existe pas (impasse, rue piétonne,...)

t_{ijk} : temps mis par le véhicule k pour aller du nœud i au nœud j ,

$[e_i, l_i]$: fenêtre de temps du nœud i ,

s_i : temps d'arrêt au nœud i ,

q_i : quantité à traiter au nœud i . Si $q_i > 0$, le nœud est fournisseur ; Si $q_i < 0$, le nœud est un client et si $q_i = 0$ alors le nœud a été servi.

Q_k : capacité du véhicule k ,

$i = 0..N$: indice des nœuds prédécesseurs,

$j = 0..N$: indice des nœuds successeurs,

$k = 0..K$: indice des véhicules.

Les variables de décisions :

$$X_{ijk} = \begin{cases} 1 & \text{Si le véhicule } k \text{ voyage du nœud } i \text{ au nœud } j \\ 0 & \text{Sinon} \end{cases}$$

A_i : temps d'arrivée au nœud i ,

D_i : temps de départ du nœud i , $D_i = a_i + s_i \forall i \in N$;

y_{ik} : quantité présente dans le véhicule k visitant le nœud i ,

La fonction à optimiser :

Les fonctions à optimiser diffèrent d'un problème à un autre. Le critère le plus utilisé dans ce genre de problème est la minimisation de la distance totale parcourue par un nombre minimum de véhicules.

$$\text{MINIMISER} \left(\alpha_1 K + \alpha_2 \sum_{k \in K} \sum_{i \in N} \sum_{j \in N} d_{ij} X_{ijk} \right) \quad (1.13)$$

Où α_1 et α_2 sont des coefficients de pondération et de mise à l'échelle.

Sous contraintes :

$$\sum_{i=1}^n \sum_{k=1}^m x_{ijk} = 1 \quad j = 2, \dots, n \quad (1.14)$$

$$\sum_{j=1}^n \sum_{k=1}^m x_{ijk} = 1 \quad i = 2, \dots, n \quad (1.15)$$

$$\sum_{i \in N} X_{i0k} = 1 \quad \forall k \in K; \quad (1.16)$$

$$\sum_{j \in N} X_{0jk} = 1 \quad \forall k \in K; \quad (1.17)$$

$$\sum_{i \in N} X_{iuk} - \sum_{j \in N} X_{ujk} = 0 \quad \forall k \in K; \forall u \in N \quad (1.18)$$

$$X_{ijk} = 1 \Rightarrow y_{ik} = y_{ik} + q_i \quad \forall i, j \in N; \forall k \in K \quad (1.19)$$

$$y_{0k} = 0 \quad \forall k \in K; \quad (1.20)$$

$$Q_k \geq y_{ik} \geq 0 \quad \forall i \in N; \forall k \in K \quad (1.21)$$

$$X_{ijk} = 1 \Rightarrow D_i + t_{ijk} \leq D_j \quad \forall i, j \in N; \forall k \in K \quad (1.22)$$

$$D_w \leq D_v \quad \forall i \in N; w = N_i^+, v = N_i^- \quad (1.23)$$

$$D_0 = 0 \quad (1.24)$$

$$X_{ijk} = 1 \Rightarrow e_i \leq A_i \leq l_i \quad \forall i, j \in N; \forall k \in K \quad (1.25)$$

$$X_{ijk} = 1 \Rightarrow e_i \leq A_i + s_i \leq l_i \quad \forall i, j \in N; \forall k \in K \quad (1.26)$$

Les équations (1.14) et (1.15) assurent que chaque sommet ne soit servi qu'une seule fois par un et un seul véhicule.

Les équations (1.16) et (1.17) assurent le non dépassement de la disponibilité d'un véhicule. Un véhicule ne sort du dépôt et n'y revient qu'une seule fois.

L'équation (1.18) assure la continuité d'une tournée par un véhicule : le sommet visité doit impérativement être quitté.

Les équations (1.19), (1.20), (1.21) et (1.22) assurent le non dépassement de la capacité de transport d'un véhicule.

Les équations (1.23) et (1.24) assurent le respect des précédences.

Les équations (1.25) et (1.26) assurent le respect des fenêtres de temps.

1.6.2 Etat de l'art des Pickup and Delivery Problem With Time windows

Vu son large domaine d'application dans la vie courante, le PDPTW a fait l'objet de plusieurs recherches. Une grande partie des recherches s'est intéressée à la résolution du problème «*Dial-A-Ride Problem*» qui concerne le transport des personnes [Cordeau 2003]. Une autre grande partie s'est intéressée au PDPTW avec des contraintes de capacité sur les véhicules. Ce dernier concerne les cas des transports de fonds, récolte des déchets, récolte et livraison de produits alimentaires...

Les problèmes PDPTW se divisent en deux : 1-PDPTW et m-PDPTW.

1.6.2.1 1-PDPTW

Le problème 1-PDPTW concerne le cas traitant d'un seul véhicule. Comme tout problème d'ordonnancement, plusieurs travaux basés sur des méthodes exactes et sur des méthodes approchées s'y sont intéressés. Parmi les recherches appliquant des méthodes exactes, nous citons les travaux de Psaraftis [Psaraftis 1983a] ainsi que ceux de Desrosiers, Dumas et Soumis [DESRO1986] qui ont utilisé des méthodes de programmation dynamique.

Vu que l'application de ces méthodes s'est limitée, pour l'obtention de solutions optimales, à un nombre restreint de clients, le recours aux méthodes approchées a été le choix de plusieurs chercheurs. Parmi ces chercheurs, nous pouvons citer Psaraftis [Psaraftis 1983b] qui a élaboré une heuristique basée sur la méthode d'arbre à recouvrement minimal, utilisant des inter échanges locaux et traitant jusqu'à 100 clients. En 1983, Sexton et Bodin [Sexton et al. 1983] ont proposé une heuristique d'insertion pour le 1-PDPTW, puis en 1985, ils ont traité un problème de *Dial-A-Ride* dans lequel chaque client spécifie un horaire de ramassage « pickup » et de livraison « delivery ». Pour cela, ils ont appliqué la décomposition de Bender à une formulation binaire mixte et non linéaire d'un 1-PDPTW [Sexton et al. 1985a][Sexton et al. 1985b]. En 1993, Van der Bruggen, Lenstra, et Schuur ont développé une méthode de résolution à deux phases basée sur une recherche en profondeur des variables avec un échange

d'arcs et améliorée par un recuit simulé [Bruggen et al 1993]. Leur méthode procure, pour une quarantaine de nœuds, des solutions de très bonne qualité avec un temps de calcul raisonnable. Finalement, Landrieu, Mati et Binder ont proposé deux méthodes de résolution : une recherche Tabou classique avec deux procédures classiques d'échange de nœuds et quelques règles d'évaluation des paramètres, et une recherche Tabou probabiliste basée sur la méthode classique et où les probabilités sont utilisées dans le choix du meilleur candidat pour la procédure d'échange à chaque itération [Landrieu et al. 2001].

1.6.2.2 m-PDPTW

Le problème m-PDPTW, qui a récemment attiré l'attention des chercheurs, concerne quant à lui le cas traitant de plusieurs véhicules. Comme dans le problème précédent, de nombreux travaux basés sur des méthodes exactes et sur des méthodes approchées ont été réalisés.

A notre connaissance, le seul algorithme optimal a été conçu en 1991 par Dumas, Desrosiers et Soumis [Dumas et al. 1991]. Ce trio a utilisé la méthode de décomposition de Dantzig-Wolfe, appelée aussi méthode de génération de colonnes, pour résoudre de façon optimale le cas 1-PDPTW avec des tailles allant jusqu'à 55 demandes et m-PDPTW avec de petites tailles.

En 1995, Savelsbergh et Solomon ont présenté un état de l'art du problème de routage de véhicules [Savelsbergh et al. 1995a]. La même année, Savelsbergh [Savelsbergh 1995b] subdivise le PDPTW général en quatre catégories qui sont : le 1-PDPTW statique, le m-PDPTW statique, le 1-PDPTW dynamique et m-PDPTW dynamique. Il présente un modèle général qui prend en compte les contraintes raisonnables et réalisables.

Dans [Nanry et al. 2000], les auteurs proposent une approche par recherche Tabou réactive qu'ils nomment RTS-PDPTW. Leur approche sert à la minimisation du coût du voyage en utilisant une fonction objectif avec deux pénalités en fonction du temps. Ces pénalités sont applicables en cas de violation d'une ou plusieurs contraintes de fenêtres de temps ou de capacité. Ils utilisent trois différents déplacement de voisinage basés sur la dominance des précédences et sur les contraintes de couplage et nommés SPI (Single Paired Insertion), SBR (Swapping pairs Between Routes) et WRI (Within Route Insertion). Cette approche a été testée sur un ensemble de benchmarks de 25, 50 et 100 demandes basées sur le benchmark « C1 » de Solomon [Solomon 1987] pour le VRPTW et a conduit à des résultats de qualité.

En 2000, Sigurd, Pisinger et Sig [Sigurd et al. 2000] ont présenté une formulation adaptée à la méthode Dantzig-Wolfe et proposent des heuristiques et une application de l'algorithme Branch and Bound pour la résolution du problème.

Righini présente dans [Righini 2000], deux nouveaux algorithmes de construction et quelques algorithmes de recherche locale avec un voisinage qui peut être simple, complexe ou variable. Liang et Lau [Lau et al. 2001] proposent une méthode à 2 phases. La première consiste en une heuristique de construction qui génère la solution initiale, puis la deuxième consiste à appliquer la recherche Tabou pour l'amélioration de cette dernière. En se basant sur l'étude de cas du benchmark de Solomon pour le VRPTW, ils proposent une stratégie de génération de nouveaux benchmarks.

Li et Lim [Li et al. 2001] présentent une nouvelle métaheuristique consistant en un algorithme tabou intégré dans un recuit simulé. Cet algorithme redémarre la procédure de recherche à partir de la meilleure solution courante s'il n'y a pas eu d'amélioration pendant plusieurs itérations. En 2002, Lim, Lim et Rodrigues [Lim et al. 2002] proposent une méthode de résolution du PDPTW qui utilise la SWO (Squeaky Wheel Optimization) avec une recherche locale. Mitrovic-Minic, Krishnamurti et Laporte traitent dans [Mitrovic et al. 2003] un PDPTW dynamique à l'aide d'une heuristique à double horizons pour un objectif à court terme ou à long terme.

1.7 Conclusion

Dans ce chapitre nous avons présenté le problème du VRP. Passant de sa formulation mathématique et ses variantes à son état de l'art, nous avons énuméré les différentes méthodes de résolution qui ont été utilisées pour le résoudre. Ensuite, nous avons présenté le problème PDPTW, sa formulation et son état de l'art. Le chapitre suivant va détailler les algorithmes évolutionnistes et la recherche Tabou : leurs principes, leurs caractéristiques, leurs outils et leurs utilités...

Chapitre 2

Les algorithmes évolutionnistes et la recherche Tabou : Principes et caractéristiques

2.1 Introduction

Les algorithmes évolutionnistes (AEs) sont une extension des algorithmes génétiques classiques (AGs). Ils permettent de modéliser le processus d'apprentissage d'une population d'individus pour s'adapter aux différents types de problèmes d'ordonnancement. Malgré leur souplesse, et leur capacité à converger assez rapidement vers l'optimum global, les AGs sont inadéquats pour la résolution d'une grande partie des problèmes réels, en particulier les problèmes d'ordonnancement. Ceci a donné naissance aux algorithmes évolutionnistes, appelés aussi algorithmes à stratégie d'évolution.

La recherche Tabou est une méthode d'optimisation qui utilise la notion de voisinage d'une solution. Cette méthode permet, à partir d'une solution initiale, de visiter un ensemble de voisins. Ces voisins seront évalués et permettent l'évolution, à travers des règles spécifiques, vers une solution finale.

Dans ce chapitre nous présentons les principes de fonctionnement de ces deux méthodes d'optimisation ainsi que leurs particularités.

2.2 Les algorithmes génétiques

2.2.1 Le principe des algorithmes génétiques

Les algorithmes génétiques ont été développés par Holland en 1975 [Holland 1975]. Basés sur la théorie de l'évolution naturelle des espèces énoncée par Darwin, ils présentent des qualités intéressantes pour la résolution des problèmes d'ordonnancement. Les individus ou chromosomes d'un algorithme génétique sont des codages des solutions possibles du problème. Comme dans la nature, ces individus forment une population qui va évoluer dans le temps selon des lois de sélection qui vont favoriser les mieux adaptés à se croiser entre eux en produisant des populations meilleures. L'évolution des individus d'une population à une autre se fait à l'aide de la reproduction. Les individus parents vont se reproduire pour donner des individus enfants qui seront, après avoir subis des opérations génétiques de croisement et de mutation, plus performants.

Comme dans la nature, ces reproductions se font avec une part de hasard. En effet, les parents candidats à la reproduction sont choisis d'une manière probabiliste proportionnelle à leurs aptitudes et l'étape de reproduction est choisie d'une façon totalement aléatoire.

Finalement, passant d'une génération à une autre, les individus forment une progéniture plus performante qui s'approche au mieux de la solution optimale.

La figure 2.1 représente le principe des algorithmes génétiques.

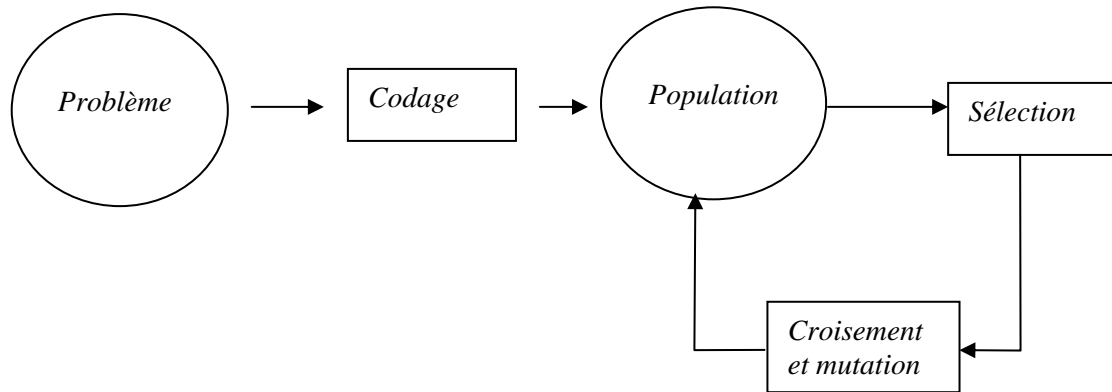


Figure 2.1 : Le principe des algorithmes génétiques

2.2.2 Le codage

La première étape de la résolution d'un problème d'ordonnancement à l'aide d'un algorithme génétique est la modélisation appropriée des solutions. Cette modélisation est appelée « codage » et permet de représenter les solutions sous forme de chromosomes.

Le codage se base sur deux notions importantes : le génotype et le phénotype. Un génotype représente l'ensemble des valeurs des gènes d'un chromosome. Quant au phénotype, c'est la représentation de la solution du problème qui traduit les données contenues dans le génotype. S'il y a un passage immédiat du phénotype au génotype, le codage est dit direct, sinon il est dit indirect et une procédure de passage est indispensable. Le codage d'une solution doit décrire toutes les données du problème et respecter toutes ses contraintes.

Un codage peut être donné par la concaténation des éléments d'un alphabet. Nous présentons respectivement dans les figures 2.2 et 2.3 un codage des plus classiques qui est le codage binaire et un codage symbolique appelé aussi codage réel.

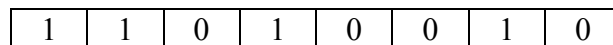


Figure 2.2 : Codage binaire

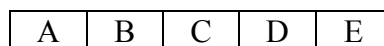


Figure 2.3 : Codage symbolique

2.2.2.1 Le codage binaire

Le principe de ce codage est de représenter la solution selon une chaîne de bits égaux à 0 ou 1. Ce codage a été utilisé lors des premiers travaux de Holland pour l'élaboration de ses théories. D'après Holland, ce codage est plus efficace que les autres et il le démontre de manière assez simple. En effet, coder une solution en chaîne de bits permet d'avoir des individus plus longs que ceux codés autrement et avoir ainsi beaucoup plus de possibilité de croisements. Le brassage génétique sur lequel repose l'efficacité des algorithmes génétiques sera donc plus important dans le premier cas.

Néanmoins, ce codage est peu « naturel » et ne permet pas une représentation facile et directe des problèmes.

2.2.2.2 Le codage symbolique

Ce codage est le plus employé dans les travaux et les outils utilisant les algorithmes évolutionnistes. Ce dernier permet de représenter d'une manière directe, plus simple et mieux intelligible les solutions des problèmes.

Ce type de codage a permis l'élaboration de nouveaux outils génétiques comme les opérateurs de croisements et de mutations contrôlés.

2.2.3 L'opérateur de croisement

L'opérateur de croisement est un opérateur génétique qui permet à deux chromosomes parents de donner deux chromosomes enfants. Cet opérateur se produit selon une probabilité P_c fixée par l'utilisateur selon le problème à optimiser. A travers cette reproduction les chromosomes enfants héritent de parties du patrimoine génétique de leurs parents. L'opérateur de croisement permet de créer de nouvelles séquences de gènes pour les chromosomes enfants à partir d'une base de configuration des séquences héritées des chromosomes parents.

Dans la littérature, il existe plusieurs opérateurs de croisement qui dépendent essentiellement du type du codage et de la nature du problème à traiter [Mesghouni 1999]. Pour le codage binaire, nous distinguons plusieurs opérateurs de croisement tels que le croisement à un point, le croisement multipoints et le croisement uniforme. Pour le codage symbolique, nous distinguons plusieurs opérateurs de croisement. Nous pouvons citer à titre d'exemple, le croisement OX (Order Crossover), le croisement PMX (Partially Mapped Crossover), le croisement MPX (Maximal Preservative Crossover), ...

▪ **Croisement à un point**

Une position est choisie au hasard dans les chromosomes parents. Les chromosomes enfants sont le fruit d'un échange de deux parties des parents (Figure 2.4). Soit k la position de croisement.

Exemple 2.1 :

Parent 1	1010 1111
Parent 2	1110 0000
$k = 4$	
Enfant1	1010 0000
Enfant2	1110 1111

Figure 2.4 : Le croisement à un point

▪ **Croisement multi-points**

Ce croisement est similaire au précédent sauf que plusieurs points de croisement y sont impliqués. Les enfants sont issus à partir d'un assortiment de parties appartenantes aux parents. L'exemple de la figure 2.5 suivante présente un croisement à deux points k_1 et k_2 .

Exemple 2.2 :

Parent 1	1010 11 1100
Parent 2	1110 00 0011
$k_1 = 4, k_2 = 6$	
Enfant1	1010 00 1100
Enfant2	1110 11 0011

Figure 2.5 : Le croisement à deux points

2.2.4 L'opérateur de mutation

La mutation est un changement aléatoire occasionnel avec une faible probabilité P_m (fixée par l'utilisateur) de la valeur d'un ou plusieurs allèles d'un chromosome. En général, la mutation ne permet pas l'obtention de meilleures solutions, mais elle permet de garder une diversité dans l'évolution des individus et d'éviter les optimums locaux. Une des mutations classiques consiste à transformer dans un chromosome binaire un 1 en un 0 ou un 0 en un 1 (Figure 2.6).

Exemple 2.3 :

Soit $k = 3$ la position choisie au hasard de l'allèle à muter.

Avant	10 1 011100
Après	10 0 011100

Figure 2.6 : Mutation Classique

Cependant, il existe dans la littérature plusieurs opérateurs de mutation :

- **Transposition de deux allèles consécutifs**

Cette mutation consiste à choisir deux allèles consécutifs au hasard et d'échanger leurs valeurs respectives (Figure 2.7).

Exemple 2.4 :

Soient $k_1 = 3$ et $k_2 = 4$ les positions choisies au hasard.

Avant	10 10 11100
Après	10 01 11100

Figure 2.7 : Transposition de deux allèles consécutifs

- **Transposition de deux allèles quelconques**

Cette mutation consiste à choisir deux allèles au hasard et d'échanger leurs valeurs respectives (Figure 2.8).

Exemple 2.5 :

Soient $k_1 = 4$ et $k_2 = 7$ les positions choisies au hasard.

Avant	101 0 11 1 00
Après	101 1 11 0 00

Figure 2.8 : Transposition de deux allèles quelconques

- **Inversion d'allèles**

Cette mutation consiste à choisir deux allèles au hasard et d'inverser l'ordre des allèles contenus dans la zone sélectionnée (Figure 2.9).

Exemple 2.6 :

Soient $k_1 = 4$ et $k_2 = 7$ les positions choisies au hasard.

Avant	101 0111 00
Après	101 1110 00

Figure 2.9 : Inversion d'allèles

Remarque : les opérateurs de mutation présentés peuvent être appliqués lors de l'utilisation d'un codage symbolique.

2.2.5 L'évaluation et la sélection des individus

Pour évoluer d'une génération à une autre les individus se reproduisent. Sachant que le but de cette évolution est d'améliorer le patrimoine génétique, la reproduction doit alors se faire entre les individus les mieux adaptés. Une opération de sélection est donc nécessaire pour pouvoir choisir les chromosomes qui garantiront une amélioration de la qualité des solutions.

L'opération de sélection se base essentiellement sur l'évaluation des solutions. L'évaluation d'une solution consiste à donner une valeur à un chromosome en fonction de sa qualité. Après avoir évalué une population, ses individus passent par un processus de sélection qui permettra d'élire les parents de la population suivante.

Il existe plusieurs techniques de sélection. Nous citons ci-dessous les plus connues d'entre elles.

- **La sélection par la roulette (RWS : Roulette Wheel Selection)**

La sélection par la roulette consiste à affecter une probabilité de sélection à chaque individu. Cette probabilité est proportionnelle à la valeur d'évaluation.

Soient :

F_p : l'évaluation de l'individu p ,

P_{Sp} : Probabilité de sélection de l'individu p ,

Pour maximiser la fonction d'évaluation, la probabilité de sélection d'un individu est la suivante :

$$P_{Sp} = \frac{F_p}{\sum_p F_p} \quad (2.1)$$

Pour minimiser la fonction d'évaluation, la probabilité de sélection d'un individu est la suivante :

$$P_{Sp} = \frac{\frac{1}{F_p}}{\sum_p \frac{1}{F_p}} \quad (2.2)$$

Se voyant affectés ces probabilités, tous les individus de la population peuvent participer à la reproduction. En effet, une roulette divisée en un nombre de secteur égal au nombre de chromosomes est lancée. L'aire de chaque secteur est proportionnelle à l'évaluation de l'individu correspondant, ainsi, la position d'arrêt de la roulette indique l'individu sélectionné.

Exemple 2.7 :

$$F_1 = 10, F_2 = 5, F_3 = 2, F_4 = 2 \text{ et } F_5 = 1$$

$$P_{s1} = \frac{F}{\sum_p F_p} = \frac{10}{20} = \frac{1}{2} = 0.5, P_{s2} = 0.25, P_{s3} = 0.1, P_{s4} = 0.1 \text{ et } P_{s5} = 0.05$$

La roulette décrite dans la figure 2.10 représente les secteurs correspondants à chaque individu.

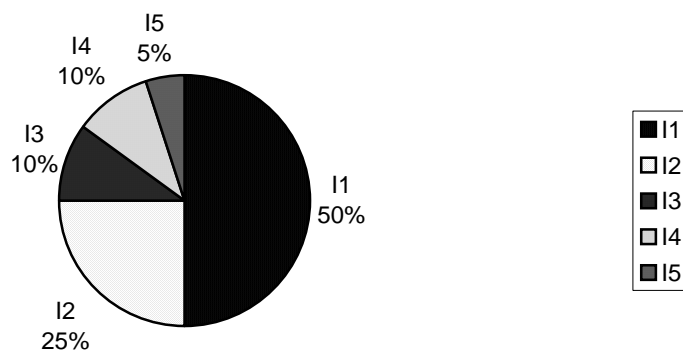


Figure 2.10: La roulette de sélection

L'inconvénient de cette méthode réside dans la présence probable d'un *Super Héros*. Le *Super Héros* est un individu dont la probabilité de sélection est très supérieure à celle des autres individus. Il risque alors d'être toujours choisi ce qui peut limiter le champ d'exploration.

▪ **La sélection par tournoi**

La sélection par tournoi consiste à choisir deux ou plusieurs chromosomes au hasard et permettre à celui qui a une meilleure évaluation d'être sélectionné. L'avantage de cette méthode est d'éviter d'avoir un *Super Héros* qu'on peut rencontrer dans la sélection par la roulette. Par contre, en utilisant cette méthode, le

meilleur individu peut ne pas être sélectionné et ainsi le champ d'exploration est réduit.

- **La sélection par classement**

La sélection par classement consiste à trier les individus en fonction de leurs valeurs d'évaluation. Ainsi classés, un nombre fixe de chromosomes est choisi pour participer à la reproduction. L'inconvénient de cette méthode est que des individus plus ou moins bons, contenant de bonnes configurations, peuvent ne pas être sélectionnés, ainsi le champ d'exploration est réduit.

2.2.6 Architecture d'un algorithme génétique

L'architecture d'un algorithme génétique est basée sur la succession de générations à travers des opérations d'évaluation, de sélection, de croisement et de mutation. La figure 2.11 montre le schéma principal d'un algorithme génétique [Goldberg 1989] [Portmann 2001].

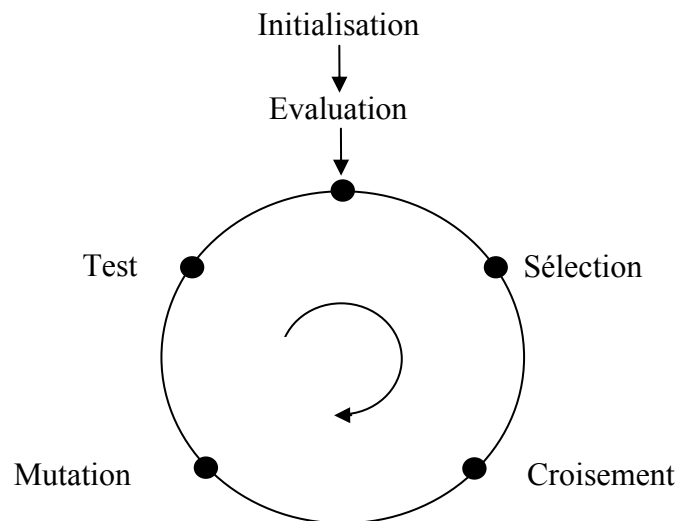


Figure 2.11 : Architecture d'un algorithme génétique

- **Initialisation** : Générer une population initiale P_0 de N individus.
- **Evaluation** : Evaluer la « performance » de chaque individu de la population P_{k-1} .
- **Sélection** : Sélectionner $N/2$ couples d'individus dans la population P_{k-1} .
- **Croisement** : Avec une probabilité de croisement p_c , tout couple d'individus est remplacé par un nouveau couple d'individus obtenu en lui appliquant un opérateur

génétique de croisement. Avec une probabilité de $1 - p_c$ le couple d'individus est conservé. (exemple : $p_c = 0.75$).

- **Mutation** : Avec la probabilité p_m tout individu de P_k subit une mutation. Avec la probabilité $(1 - p_m)$ l'individu est conservé. (exemple : $p_m = 0.05$).
- **Test** : La population P_{k-1} est remplacée par la population P_k . On reprend l'évaluation tant que le test d'arrêt n'est pas vérifié. Ce test peut concerner le nombre de générations créées ainsi que le temps de calcul. Il est toutefois nécessaire de conserver la meilleure solution obtenue dans P_{k-1} si elle ne figure pas dans P_k .

2.3 La recherche Tabou

Appelée aussi recherche avec tabous, cette méthode, introduite par Fred Glover [Glover 1986], a attiré l'attention de nombreuses équipes de recherche dans plusieurs domaines. Partant de l'informatique et des réseaux numériques d'information et allant à l'industrie, la logistique et le transport, les bons résultats obtenus par la recherche Tabou appliquée à des problèmes de tailles et de natures différentes ont montré son efficacité.

Le principe de la recherche Tabou est similaire à celui des méthodes de recherche locale itératives. Basée sur la création et l'évaluation d'un voisinage de solutions, cette recherche possède une mémoire qui garde les mouvements déjà effectués pour éviter d'y revenir et être piégé dans un cycle qui impliquera l'obtention d'un optimum local. Toutefois, une fonction d'aspiration pourra autoriser le retour vers une solution déjà visitée permettant l'exploration d'autres zones et obtenir ainsi de meilleures solutions.

Dans ce qui suit nous présenterons la recherche Tabou de base avec la notion de voisinage, les mouvements à faire pour créer un voisinage et l'évaluation d'un voisin. Nous détaillerons ensuite comment gérer une liste de mouvements candidats pour la création du voisinage en présentant l'effet de la mémoire de la recherche Tabou, le principe d'une table de hachage, la liste des attributs tabous, la durée des interdictions et finalement les critères d'aspiration.

Pour mieux illustrer les principes de la recherche Tabou nous adopterons le problème d'affectation quadratique sachant que ce dernier trouve des applications dans nombreux domaines comme l'informatique, l'automatique, le transport, et bien d'autres...

2.3.1 Problème d'affectation quadratique

Etant donnés n objets et des flots f_{ij} entre l'objet i et l'objet j ($i, j = 1, \dots, n$), et n emplacements avec des distances d_{rs} entre les emplacements r et s connus ($r, s = 1, \dots, n$), il s'agit de placer les n objets sur les n emplacements de manière à minimiser la somme des produits flots \times distances. Mathématiquement, cela revient à chercher une permutation \mathbf{p} , dont la $i^{\text{ème}}$ composante p_i donne la place de l'objet i , qui minimise $\sum_{i=1}^n \sum_{j=1}^n f_{ij} \cdot d_{p_i p_j}$. [Siarry et al 2003].

Sachant que le problème du voyageur de commerce, qui est un problème NP-Difficile, peut être formulé sous forme quadratique, il en résulte que le problème de l'affectation quadratique est un problème NP-Difficile.

2.3.2 Recherche Tabou de base

Nous considérons dans ce qui suit le problème à résoudre :

$$\min_{s \in S} f(s) \quad (2.3)$$

Où :

- s : une solution admissible du problème,
- S : l'ensemble des solutions admissibles du problème,
- f : la fonction objectif du problème

2.3.2.1 Le voisinage d'une solution

Par définition, dans la recherche Tabou, pour toute solution admissible s du problème, il existe un ensemble $N(s) \subset S$ appelé ensemble des solutions voisines de s .

Exemple 2.7 :

Pour un problème d'affectation quadratique, une solution s est une permutation de n objets et l'ensemble $N(s)$ pourra être constitué des solutions qu'il est possible d'obtenir en échangeant deux objets dans une permutation comme le montre la figure 2.12 suivante.

1	2	3	4	5	6	7
---	---	---	---	---	---	---

Solution avant permutation

1	6	3	4	5	2	7
---	---	---	---	---	---	---

Solution voisine (après permutation)

Figure II.12 : Voisinage d'une solution

La figure précédente montre une solution de départ et sa solution voisine. Cette dernière a été obtenue par une permutation élémentaire des objets « 2 » et « 6 ». Dans la recherche Tabou cette permutation est appelée mouvement.

2.3.2.2 Mouvements

Vue d'un côté pratique l'ensemble $N(s)$ des solutions voisines de s n'est autre que l'ensemble des modifications que l'on peut apporter à cette dernière. On appelle *mouvement* une modification apportée à une solution. Dans un problème d'affectation quadratique, un mouvement peut être caractérisé par les deux éléments à permuter. La figure 2.13 illustre l'ensemble des mouvements possibles dans le cas d'une solution composée de quatre éléments.

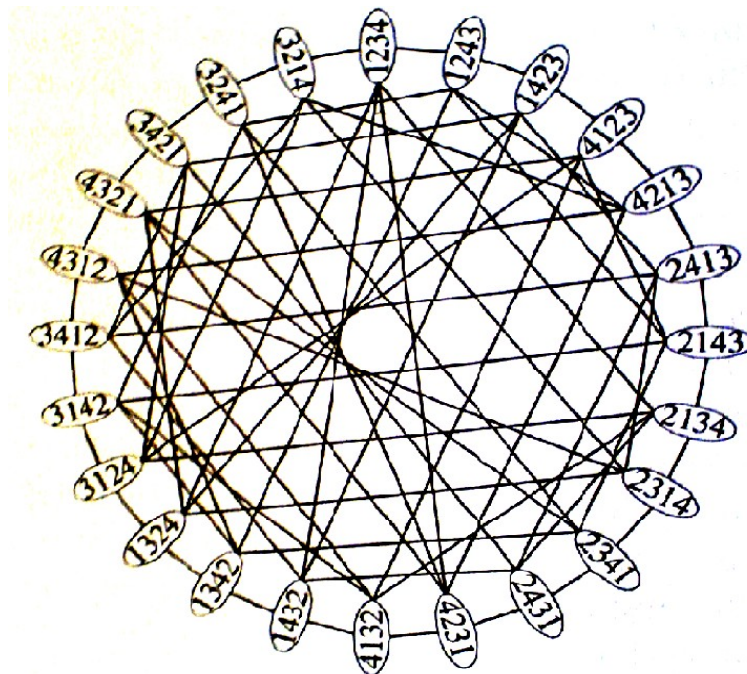


Figure 2.13 : Illustration de l'ensemble des mouvements possibles d'une solution de 4 éléments

L'ensemble $N(s)$ des solutions voisines de s est l'ensemble de solutions valables susceptibles d'être obtenues en appliquant à cette dernière un mouvement m appartenant à l'ensemble M des mouvements possibles. Dans la littérature concernant la recherche Tabou, l'application d'un mouvement m à une solution s est notée $s \oplus m$ d'où l'expression suivante du voisinage $N(s) = \{s' | s' = s \oplus m, m \in M\}$. Dans le cas des mouvements avec transposition d'éléments nous pouvons noter que $|S| = n!$ et que $|M| = \frac{n \cdot (n-1)}{2}$. Ceci nous

mène à constater que l'ensemble des voisins est beaucoup plus important que l'ensemble des mouvements.

Pour un problème avec peu de contraintes, nous avons typiquement $|N(s)| = |M|$. Contrairement à ce dernier cas, dans les problèmes fortement contraints, nous pouvons avoir des mouvements qui mèneraient à des solutions non admissibles. Dans ce cas nous aurons $|N(s)| \leq |M|$.

Dans les problèmes d'ordonnancement et d'affectation existent plusieurs types de mouvements possibles susceptibles d'engendrer des solutions voisines admissibles. Parmi ces mouvements, nous présentons ici, dans les figures 2.14, 2.15 et 2.16 :

- l'inversion de deux éléments successifs dans la solution initiale,
- la permutation de deux éléments quelconques distincts,
- le déplacement d'un élément de sa place d'origine à une nouvelle.



Figure 2.14 : Mouvement de la recherche Tabou : inversion de deux éléments successifs



Figure 2.15 : Mouvement de la recherche Tabou : permutation de deux éléments quelconques distincts

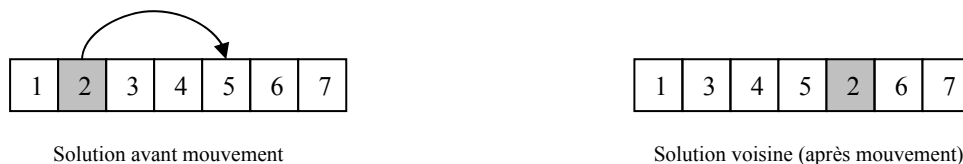


Figure 2.16 : Mouvement de la recherche Tabou : déplacement d'un élément

Le premier mouvement permet d'avoir un voisinage de taille $n - 1$, le deuxième un voisinage de taille $\frac{n \cdot (n - 1)}{2}$ et le troisième un voisinage de $n(n - 2) + 1$ [Dréo et al 2003].

D'après [Taillard 1990], le premier mouvement est mauvais pour une grande partie des

problèmes, le second permet d'avoir de meilleurs résultats que le troisième surtout dans la famille des problèmes d'affectation quadratique alors que le troisième est meilleur pour les problèmes d'ordonnement.

2.3.2.3 Evaluation des solutions voisines

La valeur du rapport entre la qualité de mouvements et le temps de calcul nécessaire à leur évaluation permet de juger de l'efficacité de l'algorithme de recherche ; plus il est élevé plus l'algorithme est efficace. Contrairement aux méthodes expérimentales et empiriques adoptées pour le jugement de la qualité d'un mouvement, les solutions voisines sont dans la plupart des cas évaluées algébriquement ce qui permet de justifier mathématiquement le choix d'une solution par rapport à une autre.

Généralement pour évaluer une solution on calcule $\Delta(s, m) = f(s \oplus m) - f(s)$. Si on veut maximiser f , plus la valeur $\Delta(s, m)$ est élevée, meilleure est la solution voisine. De même, si on veut minimiser f , plus la valeur $\Delta(s, m)$ est petite, meilleure est la solution voisine.

Dans certains cas, on parvient à simplifier l'expression $f(s \oplus m) - f(s)$ et à évaluer rapidement $\Delta(s, m)$. Par analogie avec l'optimisation continue, l'évaluation numérique de $f(s \oplus m) - f(s)$ est équivalente à l'évaluation numérique du gradient, tandis que le calcul et la simplification de la fonction $\Delta(s, m)$ est l'équivalent de l'évaluation du gradient à travers une fonction de dérivées partielles exprimées algébriquement.

En considérant, m' un mouvement appliqué sur s à l'itération précédente, il est possible dans certains cas d'exprimer l'itération courante $\Delta(s \oplus m', m)$ en fonction de $\Delta(s, m)$ et d'accélérer ainsi, en mémorisant les $\Delta(s, m)$, l'examen de la totalité du voisinage.

Des études ont montré qu'il se peut que l'évaluation de $\Delta(s, m)$ soit très difficile et coûteuse à effectuer [Dréo et al 2003]. En traitant le problème de distribution de biens, [Dréo et al 2003] ont montré qu'une solution s (tournée) peut être partitionnée en plusieurs parties (trajet : un ou plusieurs arcs) dont les poids ne sont pas supérieurs à la capacité des véhicules. Ceci les ramène à considérer que le calcul de $f(s)$ suppose, pour chacun des sous-ensembles, de trouver un ordre optimal pour la distribution des biens, ce qui est un problème

difficile en soi. Il en découle que le calcul de $f(s)$ et aussi $\Delta(s,m)$ ne peut être raisonnablement appliqué pour tous les mouvements possibles appartenant à M , mais simplement pour les mouvements réalisés. En pratique, ils se contentent de calculer la vraie valeur de $f(s)$ pour un nombre très restreint de solutions et d'évaluer $\Delta(s,m)$ approximativement sans investir un effort de calcul très important.

2.3.3 Les mouvements candidats

Dans la pratique, il est rare de considérer à chaque itération de la recherche Tabou tout l'ensemble $N(s)$ des voisins de la solution s . Dans la plupart des cas, la recherche Tabou confectionne intelligemment un sous-ensemble de solutions voisines de manière à réduire la taille du voisinage et ainsi accélérer son examen.

Il existe plusieurs façons de concevoir un ensemble réduit $N'(s)$ de solutions voisines. Le créateur de la recherche Tabou F. Glover a proposé le concept de *Liste de mouvements candidats*, en se basant sur l'hypothèse qu'un mouvement de bonne qualité pour une solution restera bon pour des solutions pas trop différentes. Il suffit donc, pour concrétiser cette hypothèse, de trier à chaque itération l'ensemble des mouvements possibles dans l'ordre décroissant de leurs qualités. Après, et au cours des quelques itérations futures, seuls les meilleurs mouvements seront considérés pour la construction du voisinage. Il est évident dans ce cas que la qualité des solutions données par ces mouvements va se dégrader au fil des itérations, c'est pour cela qu'il est nécessaire d'évaluer périodiquement le voisinage et de retenir une nouvelle liste de candidats.

Une des méthodes appliquées par certains chercheurs est de prendre au hasard du voisinage $N(s)$ un nombre de solutions plus petit que $|N(s)|$. Une autre méthode consiste à considérer, à chaque itération de la recherche Tabou, un voisinage donné par un seul sous-ensemble statique de mouvements M' appartenant à M . Cette méthode permet de garantir un examen partiel et cyclique du voisinage qui permettra de dégager les meilleurs mouvements plus rapidement.

2.3.4 La mémoire à court terme

La recherche Tabou est une méthode itérative qui explore un espace de solutions. Comme toute méthode itérative, il est intéressant d'éviter de revisiter des solutions pour essayer de converger vers la solution optimale le plus rapidement possible et ceci n'est

possible qu'à l'aide d'une mémoire. Cependant l'utilisation d'une mémoire peut s'avérer dans bien des cas peu efficace, voire mauvaise.

Outre le fait de mettre cette idée en pratique est difficile, cela suppose que l'on doit mémoriser chaque solution visitée et de tester à chaque itération et pour chaque solution admissible si cette dernière a été déjà examinée. Les tables de hachage permettent de faire cela de manière efficace, mais ceci n'empêche pas la croissance linéaire de la taille de la mémoire avec le nombre des itérations effectuées. De plus, dans beaucoup de problèmes, l'interdiction de revisiter des solutions mènerait à des incohérences comme la déconnexion de la solution courante de la solution optimale ou bien le blocage de la recherche itérative par cause d'absence de solutions voisines non visitées. La figure 2.17 suivante illustre ces derniers cas incohérents.

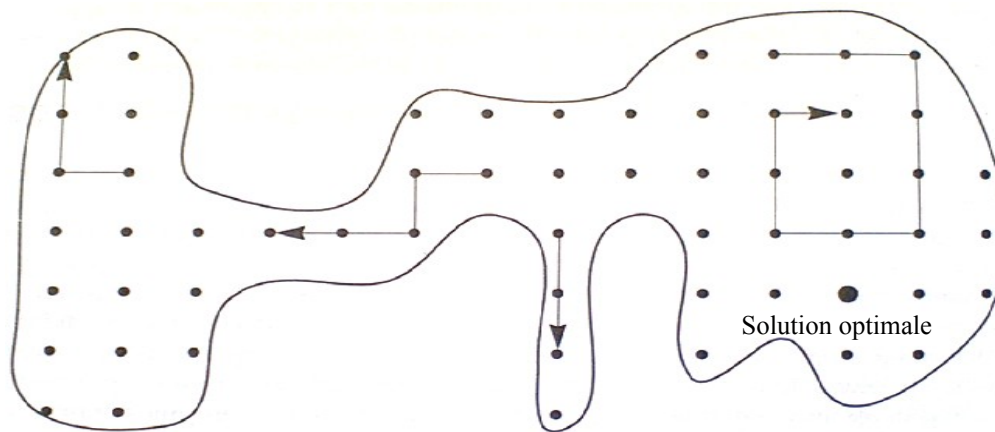


Figure 2.17 : Déconnexions et blocages dans la recherche Tabou

2.3.4.1 Table de hachage

Afin d'éviter de revisiter des solutions déjà énumérées, on utilise une table de hachage. Une table de hachage est définie par un tableau T de J entiers, avec J choisi relativement grand et assez voisin de la capacité de la machine utilisée tout en prenant en considération la taille du problème. Le but de la table de hachage est d'interdire le retour aux solutions obtenues pendant les t dernières opérations et ainsi éliminer les cycles de longueurs égales ou inférieures à t .

Pour remplir le tableau T de taille k on utilise une fonction de hachage. Parmi les fonctions les plus utilisées nous citons :

- $E(s_k) \bmod J$: Si $E(s_k)$ est la valeur entière de s_k où s_k est la solution s à l'itération k , on stockera $T[E(s_k) \bmod J]$ la valeur de $k + t$. Si $s_{k'}$ est une solution

du voisinage potentiel de s_k à l'itération k' telle que $T[E(s_{k'}) \bmod J] > k'$, elle ne sera plus considérée comme une solution éligible.

Cette méthode permet de :

- mémoriser efficacement les solutions interdites
- d'interdire, pendant t itérations, des solutions selon leurs valeurs entières et aussi selon leurs valeurs modulo J .

Il faut néanmoins signaler que cette fonction ne peut être utilisée que si la fonction objectif a une grande étendue de valeurs possibles.

- $\sum_{i=1}^n i^2 \cdot p_i$: Cette fonction est utilisée dans le cas du problème de permutation, elle prend un nombre potentiel de valeurs différentes proportionnel à $O(n^4)$.
- $\sum_{i=1}^n z_i \cdot x_i$: Avec z_i une suite de n nombres triés aléatoirement en début de recherche [Woodrouff et al 1993]. Cette fonction est utilisée dans le cas où une solution du problème s'écrit sous la forme d'un vecteur \mathbf{x} de variables binaires.

Pour réaliser un outil d'interdiction adéquat basé sur les fonctions de hachage, il faut tenir compte de trois conditions importantes. La première consiste à ce que la fonction objectif utilisée possède une grande étendue de valeurs possibles. La seconde consiste à ce que le calcul de la fonction de hachage d'une solution voisine ne nécessite pas un effort de calcul plus important que celui dédié au calcul de la fonction objectif. La troisième, proposée par [Taillard 1995], consiste à utiliser plusieurs fonctions de hachage et plusieurs tables simultanément afin de diminuer les risques de collisions.

2.3.4.2 Liste des attributs tabous

Comme cela a été montré précédemment, il serait intéressant et plus efficace de permettre l'incorporation des solutions déjà énumérées dans le voisinage $N(s)$ et ne pas se contenter de solutions non encore visitées. Cela permettrait en premier temps d'augmenter les chances d'éviter les minimas locaux, et en deuxième temps d'échapper aux problèmes de déconnection. Généralement le voisinage $N(s)$ est de taille modeste ($O(n)$ ou $O(n^2)$ avec n la taille du problème) et doit posséder d'une part la caractéristique de connexité, c'est-à-dire que la solution optimale peut être atteinte en partant de n'importe quelle solution

admissible, et d'autre part la caractéristique de réversibilité c'est-à-dire que pour tout mouvement m applicable à la solution s doit exister un mouvement m^{-1} tel que $(s \oplus m) \oplus m^{-1} = s$. Il serait donc bien utile d'interdire l'application de m^{-1} juste après m et de limiter ainsi l'ensemble des mouvements applicables à $s \oplus m$ à ceux différents de m^{-1} . De plus, cette condition permet d'éviter de se bloquer dans des cycles quand la solution s est un minimum local. Dans ce cas m^{-1} est dit un mouvement tabou.

Cette technique se résume donc à l'interdiction de tout mouvement inverse pendant un nombre d'itérations t et permet donc d'éviter d'autres cycles relatifs à un nombre plus important de solutions intermédiaires. Après les t itérations, les chances de pouvoir utiliser ces mouvements inverses sans pour autant revenir aux solutions initiales sont bonnes. Le nombre de mouvements interdits doit cependant rester restreint, parce que, si on suppose que M ne dépend pas de la solution courante, le fait de n'interdire qu'une seule partie de M est plus raisonnable. En conclusion, cette technique est une mémoire à court terme allant généralement jusqu'à quelques dizaines d'itérations.

2.3.4.3 La durée des interdictions

Le but d'une mémoire à court terme est d'interdire d'effectuer certains mouvements qui impliqueraient des cycles bloquant ainsi la recherche dans des minima locaux. Ces mouvements seront donc mémorisés directement ou indirectement en mémorisant leurs attributs voire même les attributs des solutions.

Supposons le problème d'optimisation représenté par la figure 2.18 suivante.

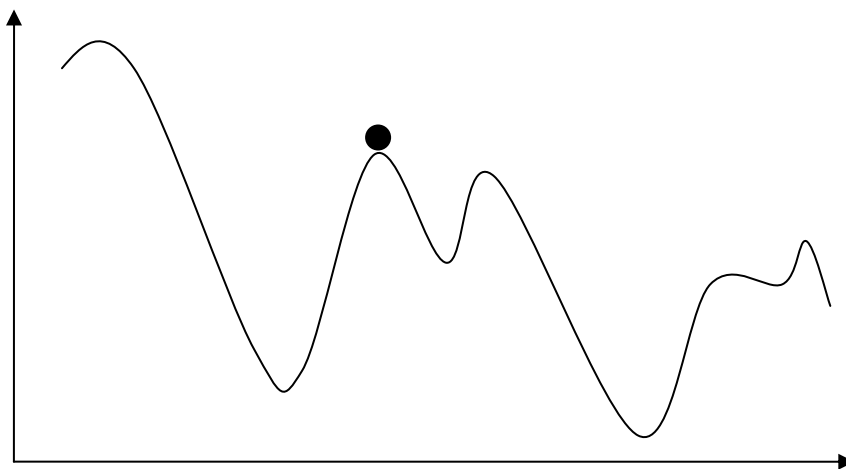


Figure 2.18 : Problème de minimisation d'altitude

Le problème consiste à minimiser l'altitude à laquelle se trouve le randonneur. La courbe représente l'ensemble des solutions admissibles. Le problème revient donc à minimiser une fonction objectif qui n'est autre que la fonction d'altitude à laquelle se trouve le randonneur. Dans ce cas l'effet de la mémoire est de visiter des vallées, sans y être toujours bloqué, et aussi de visiter des cols pour pouvoir passer à d'autres vallées.

Plus le nombre des mouvements interdits est important plus la chance de passer à d'autres vallées en franchissant des cols est grande, mais les vallées ne seront pas bien visitées en profondeur. Réciproquement, si le nombre des mouvements interdits est petit, le randonneur aura peu de chance de pouvoir franchir les cols et visiter d'autres vallées et risque par ce fait de rester bloqué dans un minimum local, mais il aura tout de même de fortes chances d'arriver au fond de la vallée dans laquelle il se trouve.

Formellement, le fait d'interdire un grand nombre de mouvements aura pour conséquence de rendre l'obtention de bons résultats très délicate faute de mouvements disponibles. Si ce nombre diminue, cela augmentera les chances d'une meilleure exploration aux alentours des optimums locaux et d'obtenir ainsi de meilleures solutions. Il ne faut cependant pas trop diminuer ce nombre, car, dans ce cas, il devient très probable de se trouver prisonnier d'un ensemble très restreint de solutions et de les visiter tout le temps de la recherche.

Pour pallier à ce compromis du choix de nombre des mouvements à interdire et de bénéficier simultanément des avantages d'un petit nombre, qui permet une visite approfondie d'une vallée, et d'un grand nombre qui permet de franchir les cols et s'échapper des vallées, ce nombre doit varier au cours du processus itératif. Pour ce faire, plusieurs méthodes existent. Ce nombre peut être tiré au hasard à partir d'un intervalle donné à chaque itération ou après un nombre d'itérations, comme il peut aussi croître ou décroître en fonction des résultats obtenus au cours de la recherche [Taillard 1990] [Taillard 1991] [Taillard 1995].

2.3.4.4 Le critère d'aspiration

Il se peut, dans la recherche Tabou, qu'un mouvement qui donne ou qui mène à une solution meilleure que toutes celles déjà visitées soit un mouvement interdit. Pour pallier à ce problème et pouvoir utiliser ce mouvement, on modifie donc son statut tabou et on dit que ce mouvement est aspiré. Il est cependant possible d'élaborer d'autres critères d'aspiration plus complexes. L'inconvénient de revenir fréquemment à l'aspiration est qu'elle peut éliminer, dans certains cas, la protection offerte par la liste Tabou vis-à-vis du cyclage.

2.4 Conclusion

Dans ce chapitre, nous avons présenté le principe des algorithmes évolutionnistes et leurs particularités. Nous avons aussi montré qu'une telle approche est basée sur des concepts très souples et très modulaires qui sont le croisement, la mutation et la sélection. Nous avons aussi présenté la recherche Tabou en détaillant ses principes et ses caractéristiques. Ces deux outils seront utilisés dans ce mémoire comme outils principaux d'optimisation pour la résolution du PDPTW statique et le PDPTW dynamique.

Chapitre 3

Approches évolutionnistes et approches évolutionnistes hybrides pour la résolution du 1-PDPTW

3.1 Introduction

Dans ce chapitre nous exposons les différentes approches élaborées pour la résolution du 1-PDPTW. Nous commençons par présenter la première approche évolutionniste qui est basée sur une nouvelle fonction objectif et des opérateurs génétiques contrôlés. Ensuite nous présentons la deuxième approche évolutionniste qui est basée sur l'utilisation de bornes inférieures pour l'évaluation des individus et de leurs qualités à travers une autre nouvelle fonction objectif. En dernier lieu nous présentons l'étape d'hybridation qui permettra d'améliorer les résultats obtenus par les deux approches évolutionnistes.

3.2 Approches évolutionnistes

Vu la complexité algorithmique exponentielle des problèmes d'ordonnancement et malgré la multitude des méthodes de résolution disponibles, il est souvent nécessaire de modifier ces dernières pour mieux les adapter au problème et pour améliorer ainsi les solutions obtenues. Dans la suite, nous allons présenter deux approches évolutionnistes pour la résolution du 1-PDPTW en détaillant la formulation mathématique adoptée, le type de codage, l'initialisation ainsi que les différents opérateurs génétiques élaborés pour chacune d'elles.

3.2.1 Formulation mathématique

Nous présentons ici une formulation mathématique du PDPTW, nous supposons que :

- il y a un seul véhicule,
- il y a un seul fournisseur pour un seul client et un seul client pour un seul fournisseur,
- un nœud n'est servi que par le véhicule une et une seule fois,
- il y a un seul dépôt,
- les contraintes de capacité doivent être respectées,
- les contraintes de temps sont rigides concernant l'heure d'arrivée,
- le véhicule commence le trajet du dépôt et y retourne à la fin,
- le véhicule reste à l'arrêt à un nœud le temps nécessaire pour le traitement de la demande.

- si le véhicule arrive au nœud i avant la date e_i de début de sa fenêtre, il le sert accusant une pénalité égale au temps d'attente.

Une formulation du PDP général est donnée dans [Savlesbergh 1995a].

Nous considérons les variables suivantes :

Les variables de données :

N : Ensemble des nœuds clients, fournisseurs et dépôt,

N' : Ensemble des demandes clients, fournisseurs,

N^+ : Ensemble des nœuds fournisseurs,

N_i^+ : Fournisseur du nœud i ,

N^- : Ensemble des nœuds clients,

N_i^- : Client du nœud i ,

d_{ij} : distance euclidienne entre le nœud i et le nœud j , si $d_{ij} = \infty$ alors le chemin entre i et j n'existe pas (impasse, rue piétonne,...)

t_{ij} : temps mis par le véhicule pour aller du nœud i au nœud j ,

$[e_i, l_i]$: fenêtre de temps du nœud i ,

s_i : temps d'arrêt au nœud i ,

q_i : quantité à traiter au nœud i . Si $q_i > 0$, le nœud est fournisseur ; Si $q_i < 0$, le nœud est un client et si $q_i = 0$ alors le nœud a été servi.

Q : capacité du véhicule,

$i = 0..N$: indice des nœuds prédécesseurs,

$j = 0..N$: indice des nœuds successeurs,

Les variables de décisions :

$$X_{ij} = \begin{cases} 1 & \text{Si le véhicule voyage du nœud } i \text{ au nœud } j \\ 0 & \text{Sinon} \end{cases}$$

A_i : temps d'arrivée au nœud i ,

D_i : temps de départ du nœud i , $D_i = A_i + s_i \forall i \in N$;

y_i : quantité présente dans le véhicule visitant le nœud i ,

La fonction à optimiser et les contraintes à respecter :

Les fonctions à optimiser et les contraintes à respecter diffèrent d'un problème à un autre. Dans la suite, nous présentons les fonctions à optimiser et les contraintes adoptées pour chacune des approches de résolution du 1-PDPTW.

3.2.2 Evaluation multicritère

Vu que la plupart des problèmes d'ordonnancement sont très difficiles à résoudre, la majorité des méthodes de résolution se contente de traiter un seul critère. Le cas que nous traitons a pour objet d'optimiser une fonction d'évaluation à trois critères fortement antagonistes. Dans la suite nous proposons une méthode d'évaluation basée sur l'utilisation de la Pareto dominance.

3.2.2.1 Stratégie globale d'évaluation

Ayant une fonction d'évaluation contenant deux critères dont les valeurs appartiennent à deux intervalles de magnitudes et d'unités différentes, une approche d'évaluation multicritère basée sur la notion de dominance est nécessaire pour que ces objectifs ne soient pas systématiquement dominés les uns par les autres.

Exemple 3.1 : Soit une fonction d'évaluation $F(x) = f_1(x) + f_2(x) + \dots + f_n(x)$. Si $f_1(x) \in [1000, 2000]$ et $f_2(x), f_3(x), \dots, f_n(x) \in [10, 20]$, le premier objectif risque de dominer les autres.

3.2.2.1.1 Approche d'évaluation multicritère

L'approche *Pareto dominance* (*Pareto optimalité*) est une des approches d'évaluation multicritère les plus utilisées [Sarker 2001]. L'objectif de cette approche est de fournir au décideur un ensemble de solutions afin de garantir une flexibilité face au choix. Cette approche peut être formulée comme suit :

- l'ensemble optimal au sens de Pareto est constitué de solutions non dominées,
- la solution est dominante, si elle n'est dominée par aucune autre solution,
- x domine y ($x \in \Omega$ et $y \in \Omega$) si $\forall 1 \leq q \leq 2, f_q(x) \leq f_q(y)$. Ceci est valable si nous minimisons f_q . Dans le cas contraire, nous aurons $\forall 1 \leq q \leq 2, f_q(x) \geq f_q(y)$

La figure 3.1 suivante montre le principe de la *Pareto optimalité* avec deux fonctions à minimiser f_1 et f_2 .

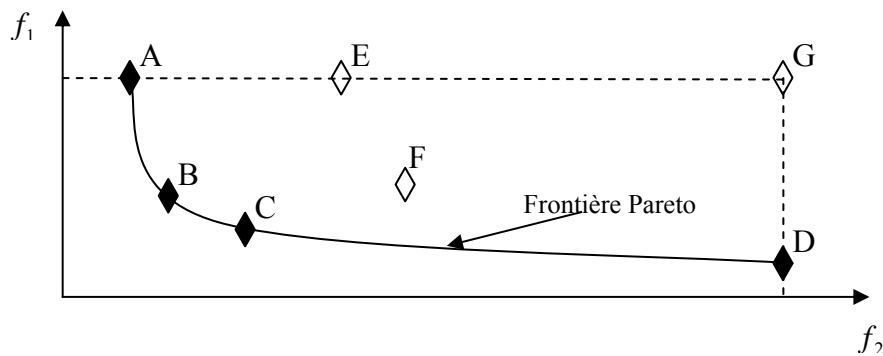


Figure 3.1 : Approche Pareto optimalité

Avec $A(60,10)$, $B(30,15)$, $C(20,30)$, $D(10,70)$, $E(60,35)$, $F(50,40)$ et $G(60,70)$, nous pouvons conclure que les solutions E, F et G sont dominées et que l'ensemble $\{A,B,C,D\}$ regroupe les solutions optimales au sens de Pareto (non dominées). Cet ensemble de solutions forme la frontière Pareto à partir de laquelle le décideur prendra sa décision.

3.2.3 1^{ère} Approche évolutionniste [Kammarti et al 2004]

3.2.3.1 Formulation mathématique adoptée

La fonction à optimiser :

La fonction à optimiser adoptée dans cette 1^{ère} approche évolutionniste pour la résolution du 1-PDPTW est la suivante

$$\text{MINIMISER } F_1 = \left(\begin{array}{l} \alpha_1 \sum_{i \in N} \sum_{j \in N} d_{ij} X_{ij} + \\ \alpha_2 \sum_{i \in N} \sum_{j \in N} \max(0, e_i - A_i) + \\ \alpha_3 \sum_{i \in N} \sum_{j \in N} \max(0, l_i - D_i) \end{array} \right) \quad (3.1)$$

Où α_1, α_2 et α_3 sont des coefficients de pondération et de mise à l'échelle.

Sous les contraintes :

$$\sum_{j \in N} X_{ij} = 1 \quad \forall i \in N; \quad (3.2)$$

$$\sum_{i \in N} X_{i0} = 1 \quad (3.3)$$

$$\sum_{j \in N} X_{0j} = 1 \quad (3.4)$$

$$\sum_{i \in N} X_{iu} - \sum_{j \in N} X_{uj} = 0 \quad \forall u \in N; \quad (3.5)$$

$$X_{ij} = 1 \Rightarrow y_j = y_i + q_i \quad \forall i, j \in N; \quad (3.6)$$

$$y_0 = 0 \quad (3.7)$$

$$Q \geq y_i \geq 0 \quad \forall i \in N; \quad (3.8)$$

$$X_{ij} = 1 \Rightarrow D_i + t_{ij} \leq D_j \quad \forall i, j \in N; \quad (3.9)$$

$$D_w \leq D_v \quad \forall i \in N; w = N_i^+, v = N_i^- \quad (3.10)$$

$$D_0 = 0 \quad (3.11)$$

L'équation (3.2) assure que chaque sommet n'est visité qu'une seule fois par le véhicule.

Les équations (3.3) et (3.4) assurent que le véhicule ne sort du dépôt et n'y revient qu'une seule fois.

L'équation (3.5) assure la continuité d'une tournée par un véhicule : le sommet visité doit impérativement être quitté.

Les équations (3.6), (3.7) et (3.8) assurent le non dépassement de la capacité de transport du véhicule.

Les équations (3.9), (3.10) et (3.11) assurent le respect des contraintes de précédences.

Remarque:

Les contraintes de fenêtres de temps ont été relaxées et sont introduites dans la fonction à optimiser F_1 .

3.2.3.2 Méthode d'évaluation multicritère de la 1^{ère} approche évolutionniste

La stratégie d'évaluation de la première approche est basée sur l'utilisation de la méthode de la Pareto Optimalité.

Dans cette application, puisque nous minimisons F_1 , on dit qu'une solution sol_1 domine une autre sol_2 si et seulement si $f_{11} \leq f_{12}$, $f_{21} \leq f_{22}$ et $f_{31} \leq f_{32}$. Et on dit aussi que deux solutions sont non dominées l'une par rapport à l'autre si est seulement si elles vérifient une des conditions du systèmes suivant :

$$\left\{ \begin{array}{l} \blacksquare f_{11} > f_{12}, f_{21} \leq f_{22} \text{ et } f_{31} \leq f_{32} \\ \blacksquare f_{11} \leq f_{12}, f_{21} > f_{22} \text{ et } f_{31} \leq f_{32} \\ \blacksquare f_{11} \leq f_{12}, f_{21} \leq f_{22} \text{ et } f_{31} > f_{32} \\ \blacksquare f_{11} > f_{12}, f_{21} > f_{22} \text{ et } f_{31} \leq f_{32} \\ \blacksquare f_{11} > f_{12}, f_{21} \leq f_{22} \text{ et } f_{31} > f_{32} \\ \blacksquare f_{11} \leq f_{12}, f_{21} > f_{22} \text{ et } f_{31} > f_{32} \end{array} \right.$$

Avec :

$$f_1 = \alpha_1 \sum_{i \in N} \sum_{j \in N} d_{ij} X_{ij}, f_2 = \alpha_2 \sum_{i \in N} \sum_{j \in N} \max(0, e_i - A_i) \text{ et } f_3 = \alpha_3 \sum_{i \in N} \sum_{j \in N} \max(0, l_i - D_i)$$

f_{11}, f_{21} et f_{31} les valeurs respectives de f_1, f_2 et f_3 de la solution sol_1 ;

f_{12}, f_{22} et f_{32} les valeurs respectives de f_1, f_2 et f_3 de la solution sol_2 ;

3.2.3.3 Le codage

Notre problème consiste à ordonnancer le passage d'un véhicule de service de capacité limitée Q chez les clients et leurs fournisseurs. Ces derniers que nous appelons « nœuds » ou « sommet » ont chacun une position géographique (X,Y) , une fenêtre de temps de service $[e_i, l_i]$, un temps de service s_i et une quantité à traiter q_i .

Parmi les approches de codage des solutions pour les problèmes d'ordonnancement des systèmes de transport, nous distinguons le codage par liste de permutation, le codage par liste de rang et le codage par matrice de permutation.

Afin d'illustrer ces différentes méthodes nous allons considérer un ordonnancement qui consiste à séquencer les différents nœuds selon l'ordre suivant : nœud 2, nœud 4, nœud 1 et nœud 3.

- **Codage par liste de permutation**

Ce type de codage consiste à classer les nœuds dans l'ordre de passage du véhicule.

Exemple 3.2:

Position	1 ^{ère}	2 ^{ème}	3 ^{ème}	4 ^{ème}
Nœud (i)	2	4	1	3

Figure 3.2 : Codage par liste de permutation.

▪ **Codage par liste de rang**

Ce type de codage consiste à présenter le rang de passage du véhicule sur les nœuds.

Exemple 3.3 :

Nœud (i)	1	2	3	4
Rang	3	1	4	2

Figure 3.3 : Codage par liste de rang.

▪ **Codage par matrice de permutation [Portmann 1996]**

Cette méthode consiste à représenter l'ordre de passage du véhicule par les nœuds par une matrice antisymétrique MT .

$MT_{i,j} = 1$ si le véhicule traite i avant j .

$MT_{i,j} = -1$ si le véhicule traite j avant i .

$MT_{i,i} = 0$ pour tout i .

Exemple 3.4 :

Nœud	1	2	3	4
1	0	-1	1	-1
2	1	0	1	1
3	-1	-1	0	-1
4	1	-1	1	0

Figure 3.4 : Codage par matrice de permutation.

Notre choix s'est porté sur le codage par liste de permutation. Ceci trouve sa légitimité dans la facilité d'utilisation et de programmation de ce type de codage et dans les besoins de notre problème.

3.2.3.4 Conception des solutions réalisables et correction des individus non viables

Pour satisfaire les contraintes (3.6), (3.7), (3.8) de capacité du véhicule et (3.9) et (3.10) de précédences entre les nœuds, nous avons élaboré un moyen de correction des solutions non viables. Ce moyen consiste à repérer les gènes mal placés et à rectifier leur position. Ceci nous permet d'avoir une population initiale totalement composée d'individus réalisables, et de corriger les individus non réalisables après l'étape de mutation. Ce moyen est composé de deux procédures de correction : la première concernant les contraintes de précédences et la deuxième les contraintes de capacité. En appliquant la première suivie de la deuxième nous obtenons des individus viables.

3.2.3.4.1 Procédure de vérification et correction des individus vis-à-vis des contraintes de précedence

Pour assurer le passage du véhicule par un fournisseur avant son client, nous avons élaboré une procédure de correction. L'algorithme de la procédure de correction des individus vis-à-vis des contraintes de précedence est donné par la figure 3.5.c de la page suivante.

Le principe de cette procédure est de parcourir le chromosome gène par gène et de vérifier quand le noeud est un client, si son fournisseur respectif a été visité. Si ce n'est pas le cas, nous insérons ce fournisseur juste devant le client. Quand la correction est faite, la procédure se poursuit jusqu'à la fin du chromosome.

Exemple 3.5 :

Soient les couples de (fournisseur, client) suivants : (3,1) ; (10,2) ; (5,7) ; (8,4) et (9,6).

Le dépôt est le noeud 0. Considérons l'individu :

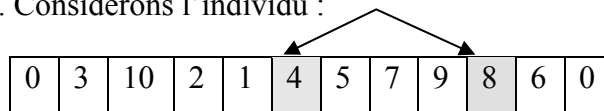


Figure 3.5.a : Individu non viable vis-à-vis des contraintes de précédences.

L'algorithme permet de localiser le gène (8) mal placé et de l'insérer avant son client (4) pour donner l'individu viable suivant :

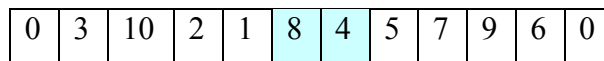


Figure 3.5.b : Individu viable vis-à-vis des contraintes de précédences.

3.2.3.4.2 Procédure de vérification et correction des individus vis-à-vis des contraintes de capacité

Pour assurer le respect de la capacité maximale du véhicule nous avons élaboré une procédure de correction. L'algorithme de la procédure de correction des individus vis-à-vis des contraintes de capacité est illustré dans la figure 3.6.c.

Le but de cette procédure est d'assurer, après avoir appliqué la procédure de correction de précedence, la vérification et la correction si cela est nécessaire vis à vis des contraintes de capacité. Son principe est de parcourir le chromosome gène par gène et de calculer la quantité transportée. Si celle-ci excède la capacité du véhicule, la procédure revient au fournisseur précédent qui n'est pas suivi par son client puis elle le fait immédiatement suivre par ce dernier. Quand la correction est faite, la procédure se poursuit jusqu'à la fin du chromosome.

```

Début
i,j,a,k : entiers
Pour i de 1 à n-2
    Si (le noeud I[i] est un client (successeur de I[i]=0))
        Pour j de i à n-2
            Si (le fournisseur de I[i] est trouvé (prédécesseur de I[i]=I[j]))
                a=I[j]
                Pour k de j à i
                    I[k]=I[k-1]
                    k=k-1
                    Nous insérons le nœud fournisseur juste avant son client
                Fin_Pour
                I[i]=a;
            Fin_Si
            j=j+1
        Fin_Pour
        i=i+1
    Fin_Si
Fin_Pour
Fin
    
```

Figure 3.5.c : Correction des individus vis-à-vis des contraintes de précédence.

Exemple 3.6 :

Soient les couples de (fournisseur, client) suivants : (3,1) ; (10,2) ; (5,7) ; (8,4) et (9,6).

Le dépôt est le nœud 0 ; $Q = 60$; $q[1]=-20$; $q[2]=-20$; $q[3]=20$; $q[4]=-20$; $q[5]=20$; $q[6]=-20$; $q[7]=-20$; $q[8]=20$; $q[9]=20$; $q[10]=20$;

Considérons l'individu :

0	3	10	8	9	6	5	7	1	2	4	0
$q=0$	$q=20$	$q=40$	$q=60$	$q=80$	$q=60$	$q=80$	$q=60$	$q=40$	$q=20$	$q=0$	

Figure 3.6.a : Individu non viable vis-à-vis des contraintes de capacité.

L'algorithme permet de localiser tout d'abord le gène (9) mal placé et de rapprocher au fournisseur précédent (8) son client respectif (4) pour donner l'individu viable suivant :

0	3	10	8	4	9	6	5	7	1	2	0
$q=0$	$q=20$	$q=40$	$q=60$	$q=40$	$q=60$	$q=40$	$q=60$	$q=40$	$q=20$	$q=0$	

Figure 3.6.b : Individu viable vis-à-vis des contraintes de capacité.

```

Début
quant, Q : réel
i,a,j,k,f,test : entiers
quant=0
Q = quantité maximale du véhicule
Pour i de 1 à n-2
    quant=quant+quantité du noeud I[i]
    Si (quant>Q)
        Pour f de i-1 à 1
            Si (le noeud I[f] est un fournisseur (prédécesseur de I[f]=0))
                test=0
                Pour k de f+1 à i
                    Si (le client de I[f] est trouvé (successeur de I[f]=I[k]))
                        test=1
                    Fin_Si
                    k=k+1
                Fin_Pour
            Si (test=0)
                Pour j de i+1 à n-1
                    Si (le client de I[f] est trouvé (successeur de
                                                                I[f]=I[j]))
                        a=I[j]
                        Pour k de j à f+1)
                            I[k]=I[k-1];
                        Fin_Pour
                        k=k-1
                        I[f+1]=a;
                    Fin_Si
                Fin_Pour
                j=j+1
            Fin_Si
        Fin_Si
        f=f-1
    Fin_Pour
    Fin_Si
    i=i+1
Fin_Pour
Fin

```

Figure 3.6.c : Correction des individus vis-à-vis des contraintes de capacité.

3.2.3.5 Génération des solutions initiales

Afin d'améliorer la qualité des solutions de la population initiale, nous avons opté pour la construction de celle-ci à l'aide d'heuristiques élaborées dans le but de minimiser la distance totale parcourue, la somme des temps d'attente et la somme des retards en tenant compte des différentes caractéristiques du problème.

3.2.3.5.1 Minimisation de la distance totale parcourue (*heuristique 1*)

Pour minimiser la distance totale parcourue par les individus de la population initiale, une heuristique a été élaborée. Le principe de cette heuristique est de fixer le nœud de départ (dépôt) puis de le faire suivre par celui qui lui est le plus proche et de finir par le retour au dépôt. Cette procédure est répétée jusqu'à l'obtention du nombre désiré d'individus à placer dans la population initiale. La figure 3.7.a illustre le principe de cette heuristique.

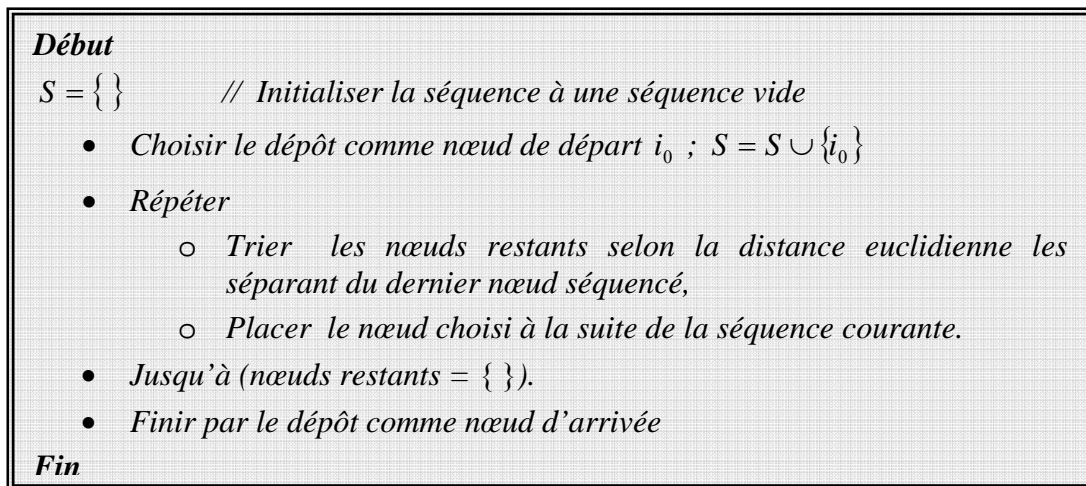


Figure 3.7.a : Algorithme de l'heuristique 1.

Exemple 3.7 :

Soient les 4 noeuds A (dépôt), B, C et D avec les coordonnées suivantes : A(5,5) ; B(7,7) ; C(10,10) ; D(4,4). En appliquant l'heuristique 1 l'individu construit est illustré par la figure 3.7.b.

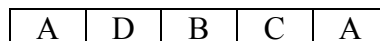


Figure 3.7.b : Individu généré par l'heuristique 1

3.2.3.5.2 Minimisation de la somme des temps d'attente (*heuristique 2*)

Afin de faire en sorte que la génération des individus de la population initiale soit guidée, nous avons élaboré une heuristique qui utilise les caractéristiques temporelles du problème en vue de minimiser la somme des temps d'attente. Le principe de la deuxième heuristique est de partir du dépôt et d'y retourner en passant par le reste des noeuds triés dans

l'ordre décroissant de la somme de la date de fin de fenêtre de temps et du temps de service de chacun simultanément $(l_i + s_i)$.

Exemple 3.8 :

Soient les 3 noeuds B, C et D avec leur dates de fin de fenêtre de temps respectives 30, 40, 50 et leurs temps de service respectifs 10, 20, 25. Soit le noeud A le dépôt.

En appliquant l'*heuristique 2* l'individu construit est présenté par la figure 3.8.

A	D	C	B	A
---	---	---	---	---

Figure 3.8 : Individu généré par l'*heuristique 2*

3.2.3.5.3 Minimisation de la somme des retards (*heuristique 3*)

Afin de minimiser la somme des retards et de faire en sorte que la génération des individus de la population initiale soit logique, nous avons élaboré une heuristique qui utilise les caractéristiques temporelles du problème. Le principe de la troisième heuristique est de partir du dépôt et d'y retourner en passant par le reste des noeuds triés dans l'ordre croissant de la date de fin de fenêtre de temps.

Exemple 3.9 :

Soient les 3 noeuds B, C et D avec leurs dates de fin de fenêtre de temps respectives 30, 40, 50. Soit le noeud A le dépôt.

En appliquant l'*heuristique 3* l'individu construit est présenté par la figure 3.9.

A	B	C	D	A
---	---	---	---	---

Figure 3.9 : Individu généré par l'*heuristique 3*

3.2.3.6 Croisement

Afin d'assurer la viabilité des solutions et de décroître le temps total de calcul, nous avons conçu un opérateur de croisement qui ne nécessite pas de procédures de correction. Considérons p la position de croisement, la figure 3.10.a de la page suivante illustre l'algorithme de l'opérateur de croisement.

Exemple 3.10 :

Soient les couples de (fournisseur, client) suivants : (3,1) ; (10,2) ; (5,7) ; (8,4) et (9,6). Le dépôt est le nœud 0. $Q = 60$; $q[1]=-20$; $q[2]=-20$; $q[3]=20$; $q[4]=-20$; $q[5]=20$; $q[6]=-20$; $q[7]=-20$; $q[8]=20$; $q[9]=20$; $q[10]=20$;

La figure 3.10.b montre un exemple d'une opération de croisement.

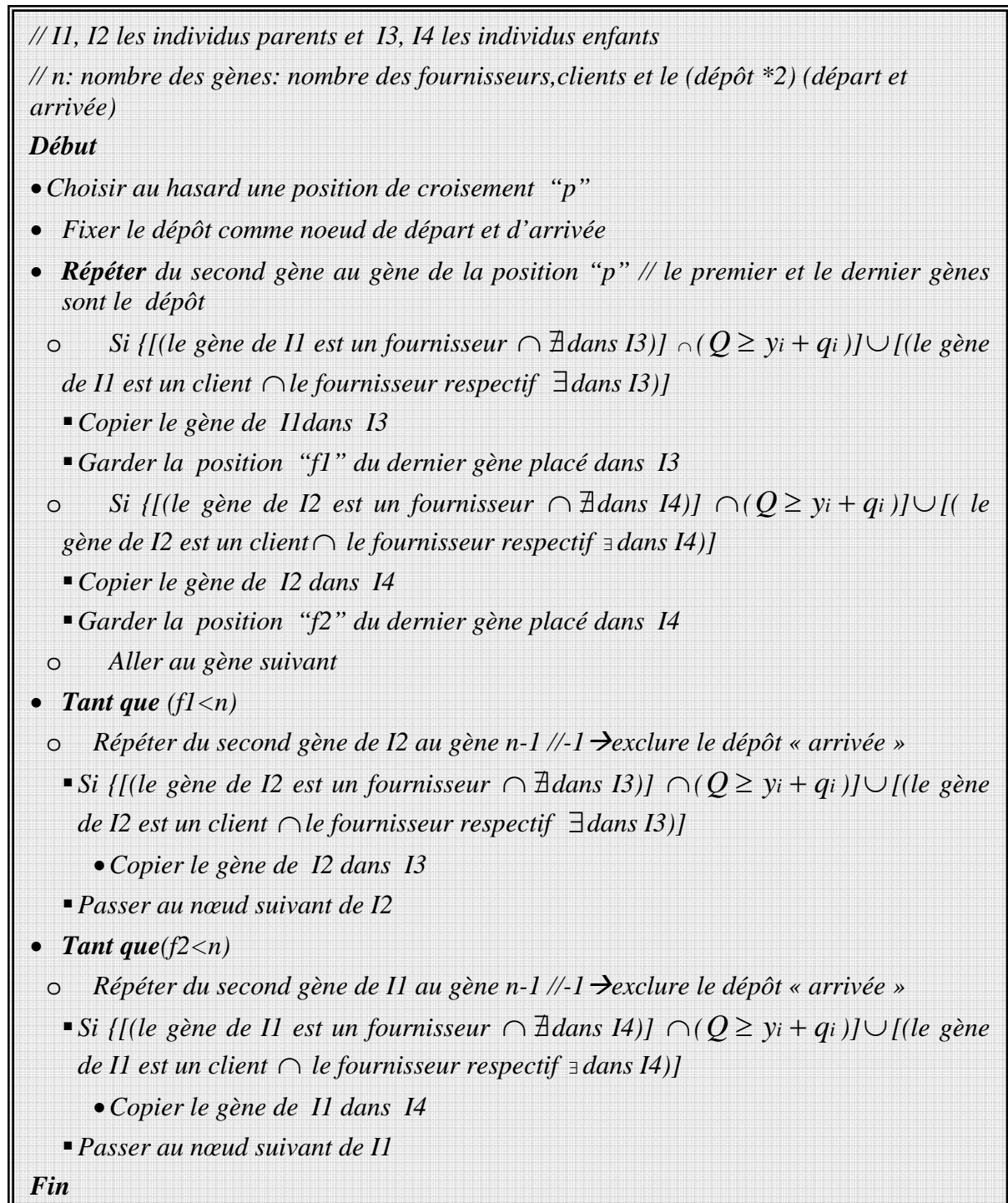


Figure 3.10.a : l’algorithme de croisement.

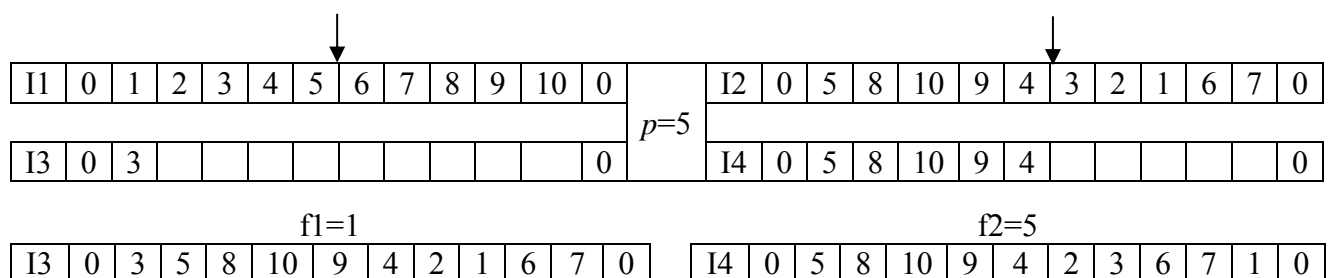


Figure 3.10.b : Croisement.

Le principe de cet opérateur est basé sur celui de l'opérateur 1.X. Nous choisissons au hasard une position de croisement p . Après nous fixons le dépôt comme noeud de départ et d'arrivée et nous parcourons la solution (chromosome) gène par gène du dépôt de départ vers la position de croisement p . Si le sommet est un fournisseur respectant les contraintes de capacité nous le copions dans l'enfant correspondant. Si le sommet est un client et son fournisseur a été servi nous le copions aussi dans l'enfant correspondant.

Pour compléter le premier enfant du croisement, nous parcourons le deuxième parent gène par gène. Si le sommet est un fournisseur pas encore servi dans le premier enfant et son insertion dans ce dernier permet de vérifier la contrainte de capacité, alors nous le copions dans cet enfant. Si le sommet est un client dont le fournisseur est visité dans le premier enfant alors nous le copions dans ce dernier. Sinon nous répétons cette procédure jusqu'à compléter le premier enfant. Puis nous appliquons cette procédure pour générer le deuxième enfant en utilisant ses deux parents.

3.2.3.7 Mutation

L'opération de mutation est assurée aléatoirement par un des quatre opérateurs spécialement conçus pour notre problème. Après chaque mutation, l'individu concerné est traité par les deux procédures de vérification et de correction. Dans ce qui suit nous présentons ces différents opérateurs.

3.2.3.7.1 1^{er} opérateur de mutation contrôlée

Le premier est un opérateur contrôlé basé sur une condition d'optimalité locale. Son but est de minimiser la distance totale parcourue. Son principe est de faire la somme des distances parcourues de trois nœuds successifs en parcourant l'individu, puis de permuter les gènes centraux des deux groupes ayant les plus grandes sommes de distances parcourues.

- *Pour i de 1 à $n-2$*
 - *Faire la somme des distances entre le gène i et ses deux voisins ;*
- *Permuter les gènes centraux des deux groupes ayant les plus grandes sommes de distances;*

Figure 3.11 : Algorithme de la mutation contrôlée 1

A l'aide de cet opérateur nous augmentons les chances d'amélioration de l'individu après une opération de mutation.

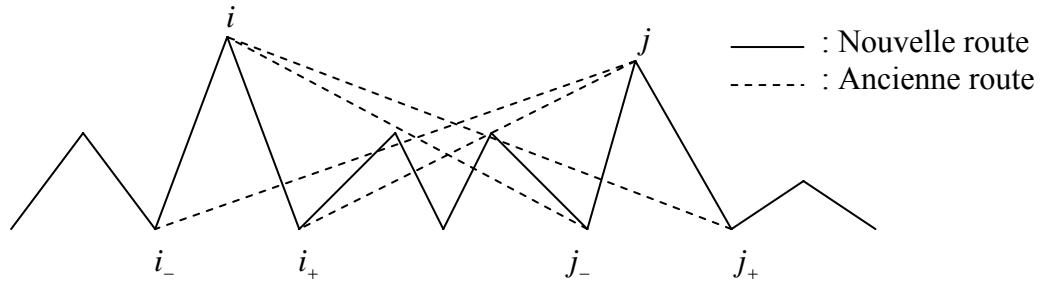


Figure 3.12 : Illustration du principe de la mutation contrôlée

3.2.3.7.2 2^{ème} opérateur de mutation

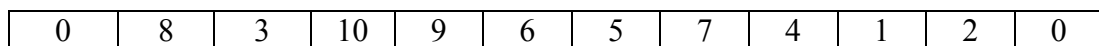
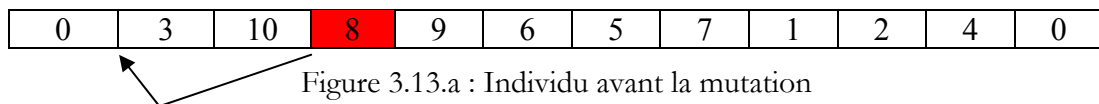
Le deuxième opérateur est basé sur le déplacement d'un gène quelconque par son insertion dans une position choisie au hasard.

3.2.3.7.3 3^{ème} opérateur de mutation contrôlée

Le troisième opérateur de mutation contrôlée consiste à choisir un nœud au hasard, de le translater vers une position quelconque puis à chercher son fournisseur ou son client respectif et à le translater avec la même ampleur si c'est possible.

Exemple 3.11 :

Soient les couples de (fournisseur, client) suivants : (3,1) ; (10,2) ; (5,7) ; (8,4) et (9,6). Le dépôt est le nœud 0



3.2.3.7.4 4^{er} opérateur de mutation contrôlée

Le quatrième opérateur agit sur le même principe que le troisième sauf que le nœud choisi au hasard doit être un fournisseur.

3.2.3.8 Sélection

La méthode de sélection élaborée est basée sur le principe de la roulette exposé au chapitre précédent. Dans cette méthode, les individus non dominés sont gardés afin de permettre l'obtention de différentes solutions Pareto optimales possibles qui constitueront un support de décision et garantiront aussi la diversité de l'évolution des individus.

3.2.3.9 Structure de l'algorithme évolutionniste de la 1^{ère} approche

Nous présentons dans la figure 3.14 le principe de la procédure d'évolution des solutions et dans la figure 3.15 la structure de l'algorithme évolutionniste élaboré pour la première approche de résolution du 1-PDPTW où F_1 est la fonction objectif (3.1).

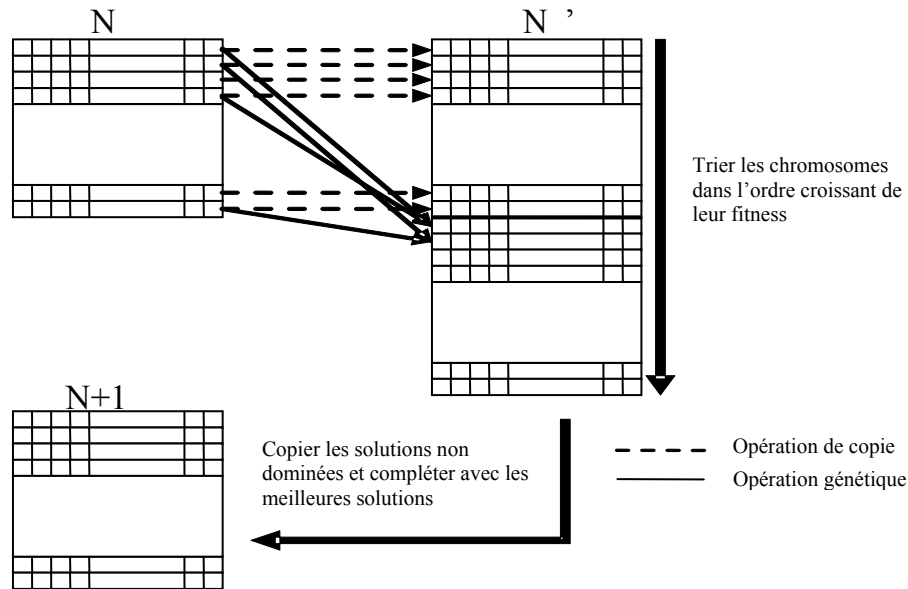


Figure 3.14 : Principe d'évolution des solutions de la 1^{ère} approche de résolution du 1-PDPTW

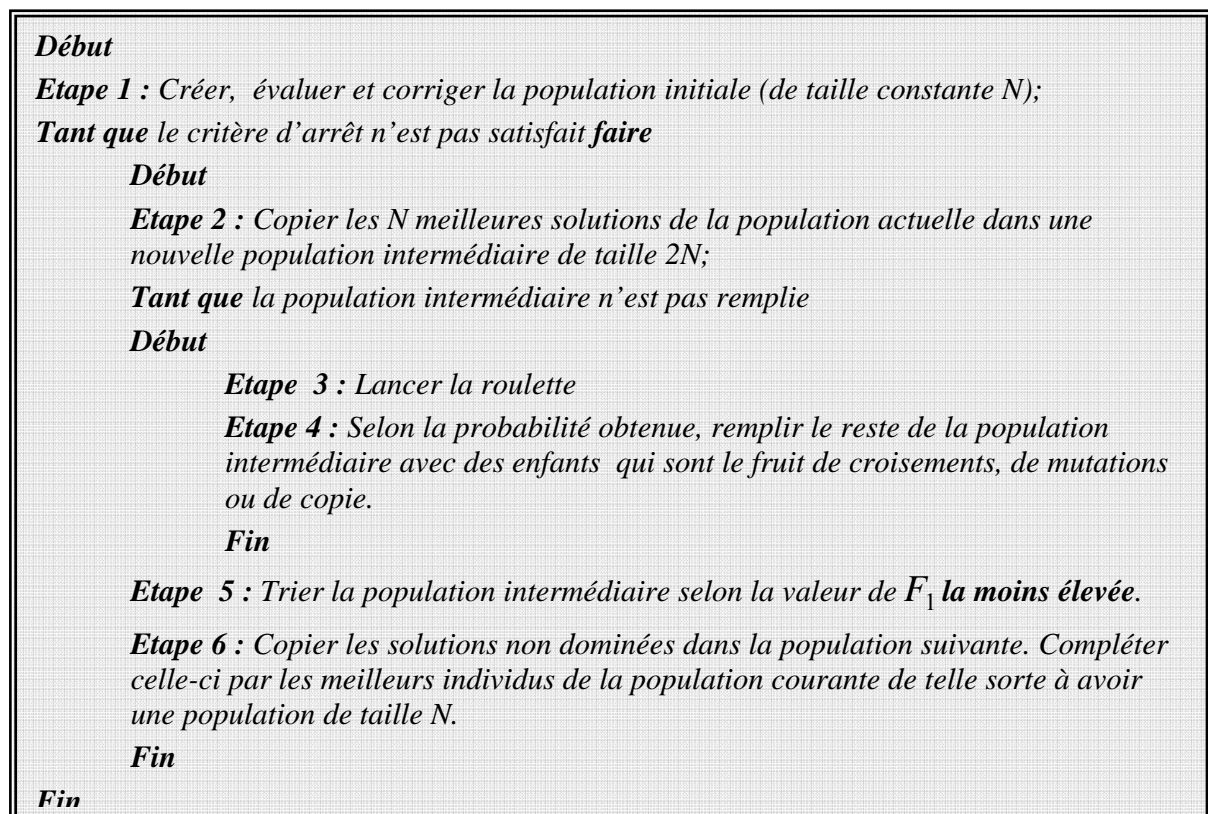


Figure 3.15 : Structure de l'algorithme de la 1^{ère} approche évolutionniste pour le 1-PDPTW

3.2.4 2^{ème} Approche évolutionniste [Kammarti et al 2005b]

Comme introduit précédemment, cette approche est basée sur l'utilisation de bornes inférieures pour l'évaluation des solutions et de leur qualité. Nous commencerons par introduire la nouvelle fonction à optimiser, puis les bornes inférieures élaborées en détaillant leur utilisation pour l'évaluation des solutions et nous présenterons finalement les outils génétiques élaborés et utilisés dans cette approche.

3.2.4.1 Formulation mathématique adoptée

La fonction à optimiser :

La fonction à optimiser adoptée dans cette 2^{ème} approche évolutionniste pour la résolution du 1-PDPTW est la suivante

$$\text{MINIMISER } F_2 = \left(\begin{array}{l} \alpha_1 \sum_{i \in N} \sum_{j \in N} d_{ij} X_{ij} + \\ \alpha_2 \sum_{i \in N} \sum_{j \in N} \max(0, l_i - D_i) \end{array} \right) \quad (3.12)$$

Où α_1 et α_2 sont des coefficients de pondération et de mise à l'échelle.

Sous les contraintes :

Cette fonction est soumise aux contraintes (3.2), (3.3), (3.4), (3.5), (3.6), (3.7), (3.8), (3.9), (3.10), (3.11) et

$$X_{ij} = 1 \Rightarrow e_i \leq A_i \quad \forall i, j \in N; \quad (3.12)$$

L'équation (3.12) assure que le véhicule ne visite un sommet qu'après l'ouverture de sa fenêtre de temps. En d'autres termes, cela permettra, moyennant une certaine flexibilité l'acceptation pour ne pas bloquer la recherche évolutionniste, d'annuler le temps d'attente dans les solutions.

3.2.4.2 La minimisation de la distance totale parcourue : borne inférieure

Afin de trouver une borne inférieure au critère de la distance totale parcourue $f_1 = \sum_{i,j} d_{ij} X_{ij}$, nous avons assimilé ce problème à un problème de recherche d'un chemin hamiltonien minimal. Pour la résolution de ce problème, nous avons eu recours à la programmation en nombres entiers après une relaxation des différentes contraintes.

Pour le problème considéré, le calcul de la borne inférieure est donné par la résolution du système linéaire suivant après sa relaxation.

$$\left\{ \begin{array}{l}
 \text{Minimiser} \left(\sum_{i,j} d_{ij} X_{ij} \right) \\
 \text{sous} \\
 \sum_i X_{ij} \leq 1, \forall i = 1, \dots, N \\
 \sum_j X_{ij} \leq 1, \forall j = 1, \dots, N \\
 \sum_{i,j} X_{ij} = N - 1 \\
 X_{ii} = 0, \forall i = 1, \dots, N \\
 X_{ij} + X_{ji} \leq 1, \forall i \neq j \\
 u_1 = 1^{(**)} \\
 2 \leq u_i \leq N \quad \forall i \neq 1, \forall j \neq 1^{(**)} \\
 u_i - u_j + (n-1)X_{ij} + (n-3)X_{ji} \leq (n-2)^{(**)} \quad (3.13)
 \end{array} \right.$$

Les lignes du système de référence ^(**) représente une condition d'élimination des sous tours introduite dans [Desrochers et al. 1991]. La variable u_i représente le rang du nœud visité. (ex : $u_5 = 64$ veut dire que le nœud 5 est visité au 64^{ème} arrêt du véhicule).

La résolution d'un tel système permet d'avoir une borne inférieure de distance totale.

Exemple 3.12 :

Nous proposons de visiter trois sommets. L'ordre de passage du véhicule par ces sommets influe directement sur la valeur de distance totale parcourue. Quelle est la plus petite valeur de distance totale parcourue qu'on peut atteindre ?

Soit la matrice suivante des distances entre les sommets :

$$\bullet \quad d[i][j] = \begin{pmatrix} 0 & \textcircled{1} & 4 \\ 2 & 0 & 5 \\ \textcircled{3} & 6 & 0 \end{pmatrix}$$

▪ i : numéro des lignes ; j : numéro des colonnes

Pour la séquence 3-1-2 et sachant que i est le prédécesseur de j nous avons une distance totale parcourue de $3 + 1 = 4$ où la distance séparant le nœud 3 du nœud 1 est égale à 3 et la distance séparant le nœud 1 du nœud 2 est égales à 1.

La fonction à minimiser est $\sum_{i,j} d_{ij} X_{ij}$ où :

d_{ij} : distance entre les sommets i et j si i précède j

et

$$X_{ij} = \begin{cases} 1 & \text{Si le véhicule voyage du sommet } i \text{ au sommet } j \\ 0 & \text{Sinon} \end{cases}$$

Cela revient à résoudre le problème linéaire suivant :

Soient les coefficients entiers $X_{11}, X_{12}, X_{13}, X_{21}, X_{22}, X_{23}, X_{31}, X_{32}$ et X_{33}

$$\text{Minimiser } \left(\sum_{i,j} d_{ij} X_{ij} \right) \quad (3.14)$$

tels que

$$\begin{cases} X_{11} + X_{12} + X_{13} \leq 1 \\ X_{21} + X_{22} + X_{23} \leq 1 \\ X_{31} + X_{32} + X_{33} \leq 1 \\ X_{11} + X_{21} + X_{31} \leq 1 \\ X_{12} + X_{22} + X_{32} \leq 1 \\ X_{13} + X_{23} + X_{33} \leq 1 \end{cases} \quad (3.15)$$

$$X_{11} + X_{12} + X_{13} + X_{21} + X_{22} + X_{23} + X_{31} + X_{32} + X_{33} = n - 1 = 2 \quad (3.16)$$

$$X_{11} = 0, X_{22} = 0 \text{ et } X_{33} = 0 \quad (3.17)$$

$$X_{12} + X_{21} \leq 1 \quad (3.18)$$

$$X_{13} + X_{31} \leq 1 \quad (3.19)$$

$$X_{23} + X_{32} \leq 1 \quad (3.20)$$

Le système d'expression (3.15) permet de vérifier qu'un sommet n'est directement succédé que par un et un seul sommet.

L'expression (3.16) permet d'exprimer la condition sur la longueur d'un chemin hamiltonien.

Les expressions (3.17) (3.18) (3.19) et (3.20) permettent d'imposer la condition de non cyclicité.

Pour ce problème, nous obtenons une valeur de distance totale parcourue de 4. Ce qui représente une borne inférieure du problème.

3.2.4.3 La minimisation du retard total : borne inférieure [Chu 1990]

En 1990 Chu a proposé une borne inférieure pour le problème $1|r_i|\sum T_i$ [CHU1990].

Cette borne est basée sur une relaxation de la contrainte de non préemption du problème et

l'utilisation de la règle SRPT (Short Remaining Processing Time). Cette relaxation consiste à considérer que les tâches sont interrompibles. Dans sa démonstration, Chu définit et utilise plusieurs nouvelles notions et théorèmes.

Chu propose plusieurs théorèmes et définitions pour introduire l'algorithme de calcul d'une borne inférieure minimisant la somme des retards [CHU1990].

Dans cet algorithme, on permet la préemption et on ordonnance à chaque itération une tâche disponible dont la durée opératoire restante est la plus petite. Son exécution peut être interrompue et remplacée par une autre lorsqu'une tâche, ayant une durée opératoire restante strictement plus petite que celle de la tâche en cours, devient disponible. Dans l'ordonnement obtenu la machine n'est jamais laissée inoccupée lorsqu'il existe une tâche disponible et non terminée.

Pour cet algorithme Chu propose le théorème suivant [CHU1990]:

Théorème de Chu [CHU1990] :

Etant donné un problème $P(r,p,d)^{(1)}$, soient :

- i) $(d'_1, d'_2, \dots, d'_n)$ la suite triée par valeurs non décroissantes à partir de la suite (d_1, d_2, \dots, d_n) ;
- ii) Ω le problème relaxé d'ordonnement obtenu à partir du problème P en autorisant la préemption ;
- iii) O_Ω un ordonnancement du problème Ω , non nécessairement réalisable pour P , obtenu en appliquant la règle SRPT.

Pour tout O , ordonnancement réalisable de P , on a :

$$\sum_{i=1}^n \max(C[i, O_\Omega] - d'_i, 0) \leq T(P, O) \quad (3.21)$$

où $C[i, \sigma]$ est la date de fin de la tâche se terminant à la $i^{\text{ème}}$ position dans l'ordonnement σ , $T(\Pi, \sigma)$ est la somme des retards de l'ordonnement σ (réalisable pour le problème Π) calculée à partir des données du problème Π .

⁽¹⁾ $P(r,p,d)$ désigne le problème dont les paramètres sont données par les vecteurs $r = (r_i)_{i=1, \dots, n}$, $p = (p_i)_{i=1, \dots, n}$ et $d = (d_i)_{i=1, \dots, n}$.

Exemple 3.13 :

Soient :

- $I = \{1,2,3,4\}$ l'ensemble des tâches i ,
- $r_i = \{0,1,3,5\}$ les dates de disponibilité respectives (exécution au plus tôt) de l'ensemble I des tâches,
- $p_i = \{5,2,1,2\}$ Les durées d'exécutions respectives de l'ensemble I des tâches,
- $d_i = \{6,6,5,7\}$ les dates dues respectives de l'ensemble I des tâches,
- $C[i]$ les dates de fin de la tâche se terminant à la $i^{\text{ème}}$ position.

i	1	2	3	4
r_i	0	1	3	5
p_i	5	2	1	2
d_i	6	6	5	7
d'_i	5	6	6	7

Tableau 3.1 : Calcul de la borne inférieure minimisant la somme des retards

Les d'_i sont obtenus par le tri dans l'ordre croissant des d_i .

Les données précédentes nous mènent à l'ordonnancement suivant :

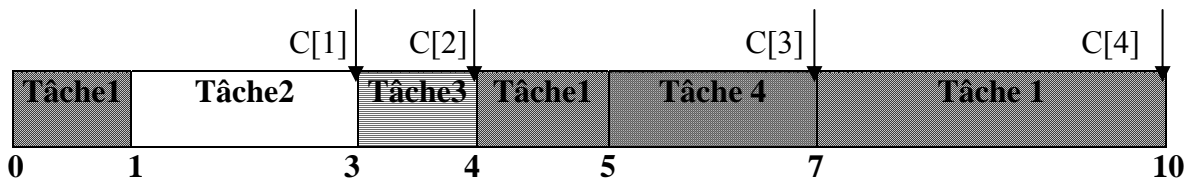


Figure 3.16 : Illustration du théorème de Chu

En appliquant, $\sum_{i=1}^n \max(C[i] - d'_i, 0)$ nous obtenons une borne inférieure $Cr = \max((3 - 5), 0) + \max((4 - 6), 0) + \max((7 - 6), 0) + \max((10 - 7), 0) = 4$.

Pour la démonstration de cette méthode et des théorèmes utilisés nous invitons les lecteurs à consulter la thèse de Chu [CHU1990].

3.2.4.4 Méthode d'évaluation multicritère de la 2^{ème} approche évolutionniste

La méthode d'évaluation multicritère de la deuxième approche évolutionniste pour la résolution du 1-PDPTW est basée sur la méthode de la Pareto optimalité et sur l'utilisation des bornes inférieures trouvées afin d'homogénéiser les valeurs des deux critères de F_2 . Dans

ce qui suit nous présentons la méthode d'homogénéisation des critères puis l'application de la Pareto optimalité.

3.2.4.4.1 L'homogénéisation des critères à optimiser

Rappelons les deux critères à optimiser qui sont des objectifs fortement antagonistes.

$$f_1 = \sum_{i,j} d_{ij} X_{ij} \quad (3.22)$$

$$f_2 = \sum_i \max(0, l_i - D_i) \quad (3.23)$$

Pour homogénéiser les objectifs d'une fonction agrégative, nous nous sommes inspirés du principe de base de la logique floue [Borne 1998] qui permet de générer pour chaque objectif une fonction et une valeur d'appartenance.

Cette application est basée sur les étapes suivantes :

- Calculer f_1^* et f_2^* les deux bornes inférieures respectives des deux objectifs f_1 et f_2 .
- Soit H_1 une heuristique qui permet d'améliorer l'objectif f_1 et $f_1^{H_1}$ la valeur donnée par cette heuristique selon la première fonction objectif.
- Soit H_2 une heuristique qui permet d'améliorer l'objectif f_2 et $f_2^{H_2}$ la valeur donnée par cette heuristique selon la deuxième fonction objectif.
- Considérons la fonction suivante :

$$\tilde{f}_G(x) = w_1 \tilde{f}_1(x) + w_2 \tilde{f}_2(x) \quad (3.24)$$

où

$$\tilde{f}_1(x) = \frac{f_1^{H_1} + \varepsilon_1 - f_1(x)}{f_1^{H_1} - f_1^* + \varepsilon_1} \quad \text{et} \quad \tilde{f}_2(x) = \frac{f_2^{H_2} + \varepsilon_2 - f_2(x)}{f_2^{H_2} - f_2^* + \varepsilon_2} \quad (3.25)$$

Sachant que les valeurs de $f_1(x)$ et $f_2(x)$ ne peuvent pas être inférieures à leurs bornes respectives f_1^* et f_2^* . Il suffit de poser $\tilde{f}_1(x) = 0$ si $f_1(x) \geq f_1^{H_1} + \varepsilon_1$ et $\tilde{f}_2(x) = 0$ si $f_2(x) \geq f_2^{H_2} + \varepsilon_2$ pour avoir $\tilde{f}_1(x)$ et $\tilde{f}_2(x)$ appartenant à l'intervalle $[0,1]$.

$\varepsilon_1 = cte1 \times f_1^*$ et $\varepsilon_2 = cte2 \times f_2^*$ où $cte1$ et $cte2$ sont des constantes de valeur très faible permettant d'avoir ε_1 et ε_2 de l'ordre de 0.001.

Les constantes ε_1 et ε_2 permettent d'éviter une division par zéro si $f_1^* = f_1^{H_1}$ et $f_2^* = f_2^{H_2}$.

Finalement nous fixons $w_1 = w_2 = 0.5$ pour donner la même importance aux deux critères, d'où $\tilde{f}_G(x) \in [0, 1]$.

Ainsi, la nouvelle fonction d'évaluation nous permet de mieux juger la qualité des solutions qui seront utilisées et fournies par l'algorithme évolutionniste. Plus $\tilde{f}_G(x)$ est proche de 1 plus notre solution s'approche des deux bornes inférieures donc proche de l'optimum.

Les deux critères de notre fonction d'évaluation influent directement sur la prise de décision, il faut alors un moyen d'évaluation multicritère de la qualité des solutions proposées. Ceci va être assuré par l'application de la méthode de la Pareto optimalité.

3.2.4.4.2 Application de la Pareto optimalité

Dans cette application, puisque nous maximisons $\tilde{f}_G(x)$, on dit qu'une solution sol_1 domine une autre sol_2 si et seulement si $\tilde{f}_1(sol_1) \geq \tilde{f}_1(sol_2)$ et $\tilde{f}_2(sol_1) \geq \tilde{f}_2(sol_2)$. Et on dit aussi que deux solutions sont non dominées l'une par rapport à l'autre si et seulement si elles vérifient une des conditions du même système introduit précédemment dans la 1^{ère} approche évolutionniste (voir paragraphe 3.2.3.2).

3.2.4.5 Les outils génétiques de la 2^{ème} approche évolutionniste

Le but de cette deuxième approche évolutionniste est de fournir des solutions du 1-PDPTW ayant un temps d'attente nul. Pour cela, nous avons conçu un opérateur de nouveau croisement qui permet, même à partir de deux solutions non admissibles, de générer deux enfants réalisables respectant toutes les contraintes du problème. Dans notre cas, les solutions non admissibles peuvent être le fruit d'une mutation quelconque.

Cette nouvelle approche se base donc sur le même codage de liste, utilise les mêmes procédures de vérification et de correction des solutions vis-à-vis des contraintes de précedence et de capacité, adopte les mêmes heuristiques pour la génération de la population initiale et opérateurs de mutation de la première.

Dans ce qui suit nous détaillons le principe de l'opérateur de croisement élaboré.

3.2.4.5.1 Croisement

Pour assurer la viabilité des solutions et diminuer le temps total de calcul, nous avons conçu un opérateur de croisement qui garantit, si cela est possible, que les solutions engendrées soient admissibles et réalisables. Si les données du problème et la position de croisement ne permettent pas à cet opérateur de générer des solutions alors nous appliquons l'opérateur de la première approche évolutionniste. Considérons p la position de croisement, la figure 3.17 illustre l'algorithme de l'opérateur de croisement.

Comme l'opérateur de croisement précédent, le principe de celui-ci est aussi basé sur celui de l'opérateur 1.X. Nous choisissons au hasard une position de croisement p . Après, nous fixons le dépôt comme noeud de départ et d'arrivée et nous parcourons la solution (chromosome) gène par gène du dépôt de départ vers la position de croisement p . Si le sommet est un fournisseur respectant les contraintes de capacité et la date d'arrivée du véhicule est supérieure ou égale à la date d'ouverture de la fenêtre de temps de ce sommet ($A_i \geq e_i$), nous le copions dans l'enfant correspondant. Si le sommet est un client dont le fournisseur a été servi et la date d'arrivée du véhicule est supérieure ou égale à la date d'ouverture de la fenêtre de temps de ce sommet, nous le copions aussi dans l'enfant correspondant.

Pour compléter le premier enfant du croisement, nous parcourons le deuxième parent gène par gène. Si le sommet est un fournisseur pas encore servi dans le premier enfant et son insertion dans ce dernier permet de vérifier la contrainte de capacité et la contrainte de la date d'arrivée du véhicule ($A_i \geq e_i$), alors nous le copions dans cet enfant. Si le sommet est un client dont le fournisseur est visité dans le premier enfant et la contrainte de la date d'arrivée du véhicule ($A_i \geq e_i$), alors nous le copions dans ce dernier. Sinon nous répétons cette procédure jusqu'à compléter le premier enfant. Puis nous appliquons cette procédure pour générer le deuxième enfant en utilisant ses deux parents.

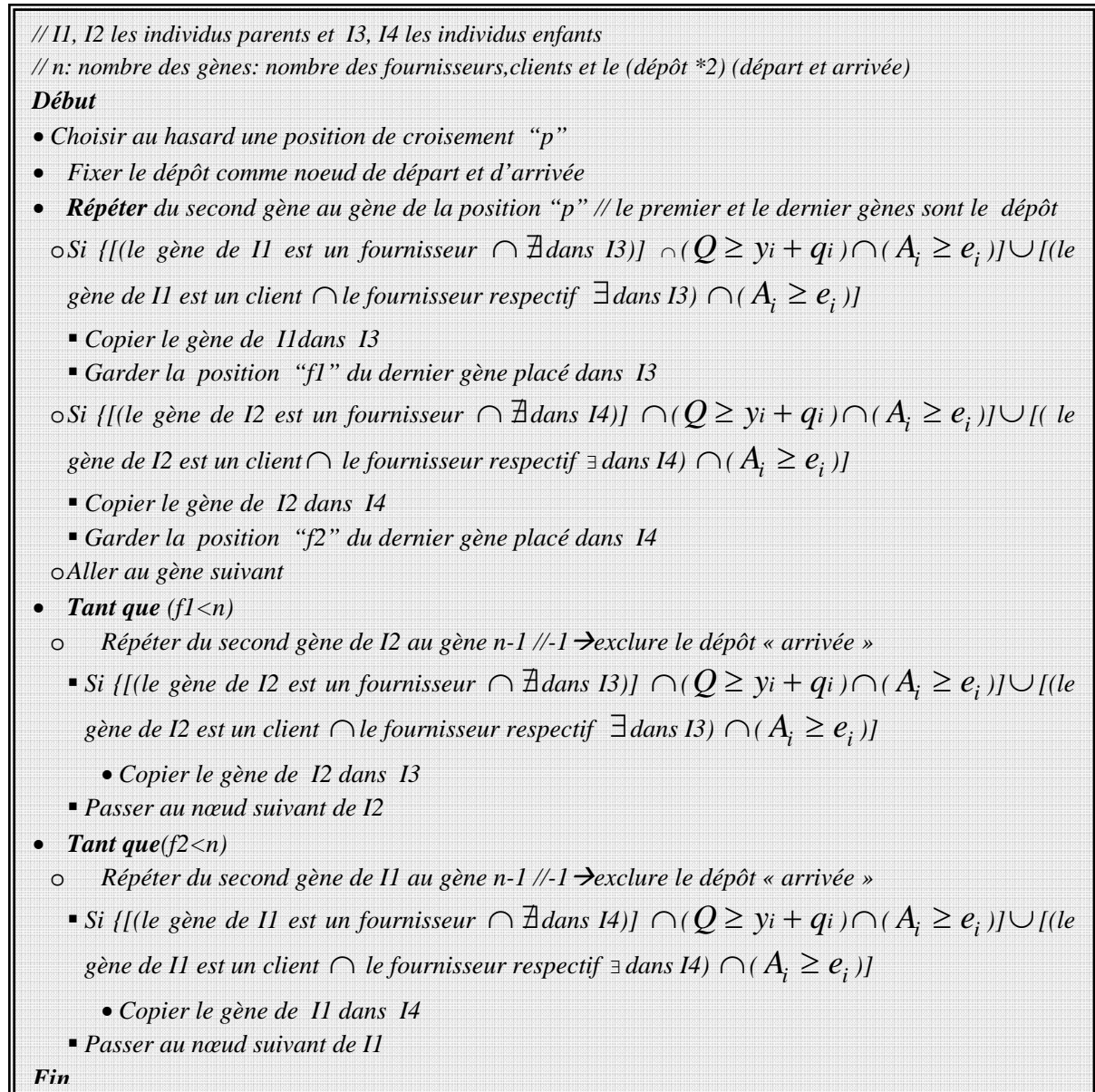


Figure 3.17 : l’algorithme de croisement.

3.2.4.6 Structure de l’algorithme évolutionniste de la 2^{ème} approche

Le principe de la procédure d’évolution des solutions est présenté dans la figure 3.14 de la première approche. Nous présentons dans la figure 3.18 la structure de l’algorithme évolutionniste élaboré pour la deuxième approche de résolution du 1-PDPTW.

Il est à noter que lors d’une opération de mutation, l’individu engendré, même après avoir appliqué les procédures de vérification et de correction, risque d’être non admissible vis-à-vis de la contrainte (3.12). Mais vu la diversification des solutions qu’engendre la mutation d’une population par rapport à la suivante, nous avons décidé de garder l’individu muté vu qu’il aura de très forte chance d’être sélectionné pour un croisement s’il est performant ou bien de disparaître s’il est de qualité médiocre.

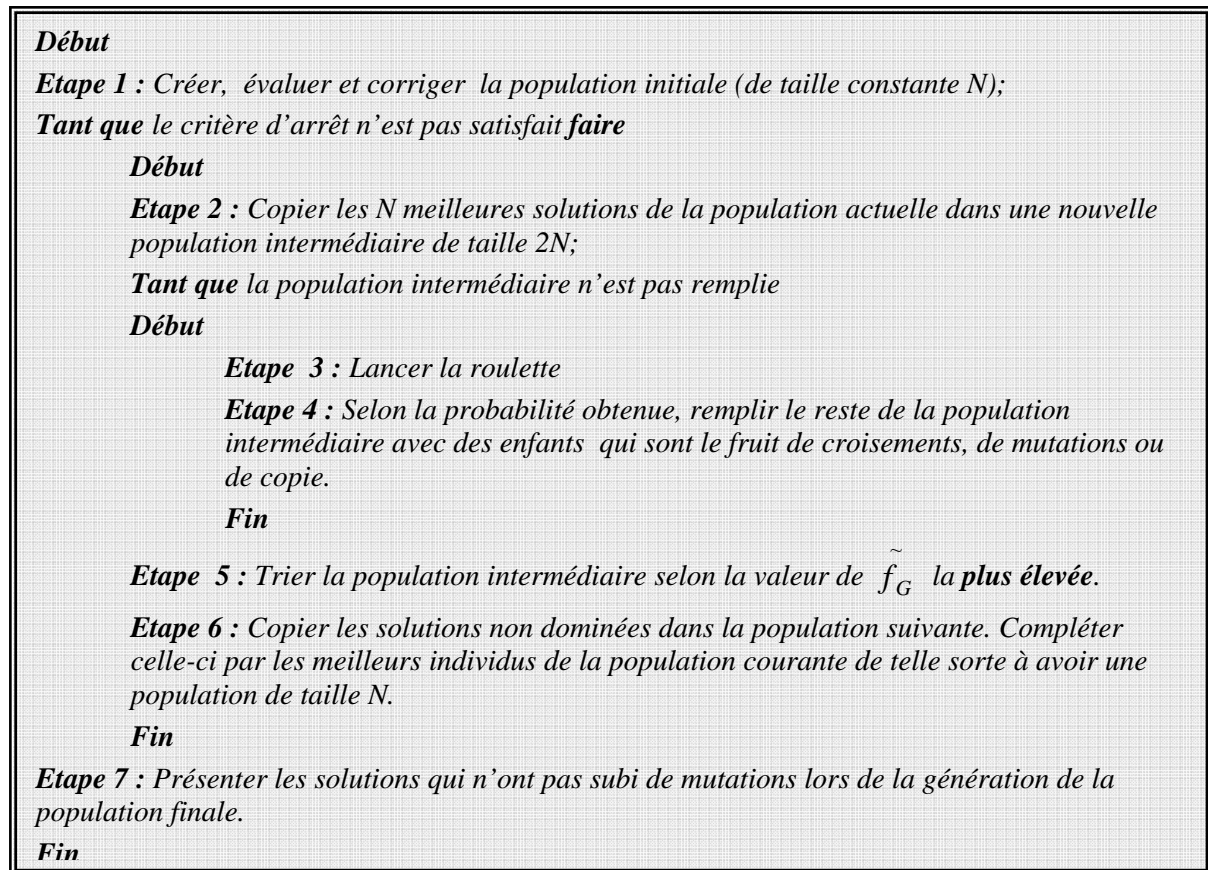


Figure 3.18 : Structure de l'algorithme de la 2^{ème} approche évolutionniste pour le 1-PDPTW

3.3 Hybridation des approches évolutionnistes : étage d'amélioration par la recherche Tabou spécialisée

La recherche Tabou explore l'espace des solutions en passant à chaque itération de calcul d'une solution s à la meilleure solution du voisinage $V(s)$. Contrairement à la méthode de descente classique, la solution courante peut se détériorer de l'itération (i) à l'itération suivante ($i+1$). Pour cela, afin d'éviter une boucle fermée de recherche, les solutions récemment explorées sont temporairement déclarées taboues et ne peuvent plus être considérées. Le statut « tabou » d'une solution peut être annulé si certaines conditions sont vérifiées. Ceci est appelé le critère d'aspiration et cela se produit par exemple lorsqu'une solution taboue est meilleure que toutes les solutions précédemment visitées. D'autres techniques sont aussi souvent employées dans le but de diversifier et d'intensifier le processus de recherche.

Dans notre cas, la solution initiale du calcul va être la première solution non dominée résultant d'une des approches évolutionnistes détaillées précédemment.

3.3.1 La construction du voisinage d'une solution [Kammarti et al 2005a]

Dans ce qui suit, nous présentons quelques heuristiques d'échange qui vont permettre la construction du voisinage d'une solution.

3.3.1.1 Les heuristiques 2-opt et 3-opt

Les heuristiques d'échange d'arcs comme 2-opt et 3-opt sont largement utilisées pour l'amélioration des solutions dans les problèmes de routage de véhicules. Généralement, ces genres d'heuristiques d'échange sont incorporés dans des structures de recherche locale de la manière suivante [Taillard 1997] :

- **Etape 1** : Commencer avec une solution initiale définie aussi comme solution courante,
- **Etape 2** : Générer le voisinage total de la solution courante en appliquant l'heuristique d'échange considérée.
- **Etape 3** : Sélectionner la meilleure solution du voisinage et la définir comme solution courante.
- **Etape 4** : Revenir à l'étape 2.

3.3.1.2 L'heuristique d'échange conçue pour le 1-PDPTW

Le principe de notre heuristique d'échange est de choisir au hasard deux couples (fournisseur i , client i) et (fournisseur j , client j) et échanger respectivement le fournisseur i avec le fournisseur j et le client i avec le client j . La figure 3.19 illustre notre heuristique d'échange.

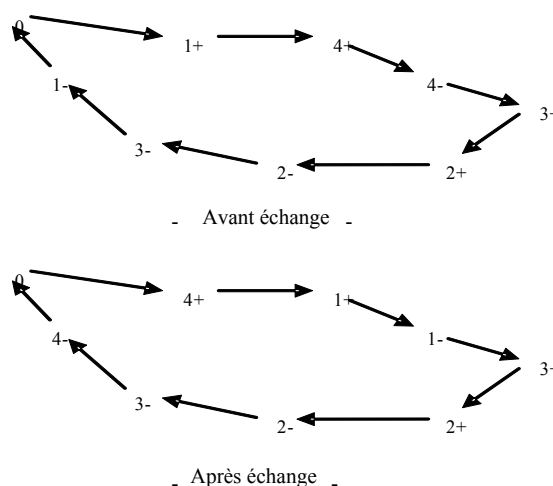


Figure 3.19 : L'heuristique d'échange dédiée au 1-PDPTW

La solution initiale de la recherche Tabou est réalisable puisqu'elle est la première solution d'une des approches évolutionnistes élaborées. De ce fait, l'heuristique d'échange

conçue pour le 1-PDPTW va nous permettre de générer des individus viables vis-à-vis de la contrainte de précédence qui garde l'ordre de précédence entre les fournisseurs et leurs clients respectifs. Néanmoins, l'application d'une procédure de correction vis-à-vis des contraintes de capacité demeure nécessaire.

3.3.2 La recherche Tabou dédiée au 1-PDPTW

Dans le but d'améliorer les solutions fournies par les approches évolutionnistes nous appliquons une recherche Tabou spécialement conçue pour notre problème. L'algorithme de celle-ci est détaillé dans la figure 3.20.

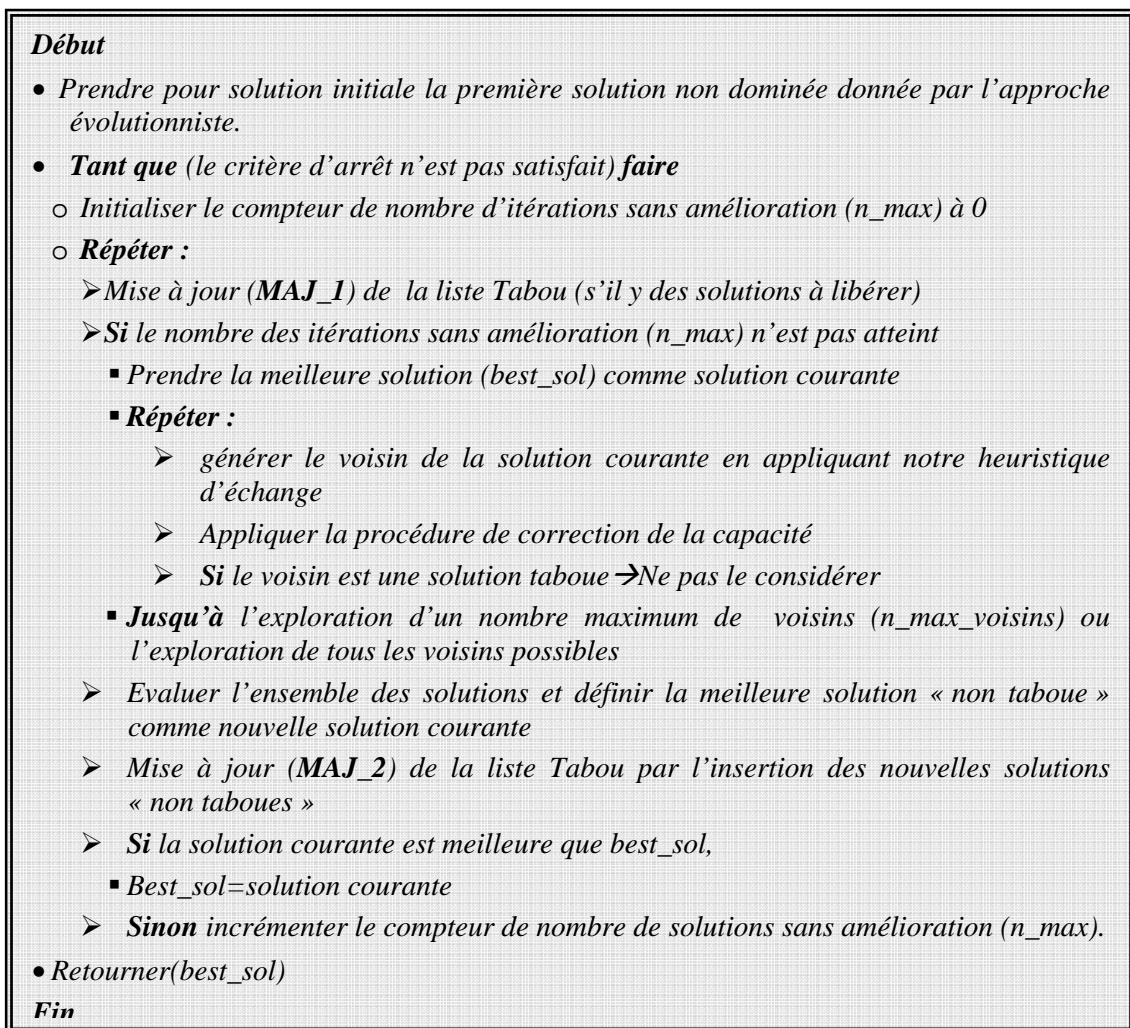


Figure 3.20 : L'algorithme de la recherche Tabou spécialisée

3.3.2.1 Mise à jour de la liste Tabou : critère d'aspiration

La liste Tabou permet de mémoriser les solutions déjà explorées par la recherche Tabou et d'interdire leur réutilisation. Cependant, dans certain cas, le fait de repartir d'une solution taboue permet de se diriger vers des solutions meilleures et de franchir les sommets

pour sortir des optimums locaux. Pour cela, une mise à jour de cette liste s'avère utile pour permettre de lever le statut tabou sur quelques solutions. Cette mise à jour se fait à l'aide de l'utilisation d'un critère d'aspiration.

Notre critère d'aspiration est basé sur l'utilisation de compteurs n_tabou_i . Un compteur n_tabou_i permet de calculer le nombre de générations que la solution i a passé bloquée à l'état tabou. Si ce compteur dépasse une limite n_tabou_limite , le statut tabou sera levé de la solution i concernée. Dans notre cas, la fixation de la limite n_tabou_limite ne dépend pas de la valeur de la fonction objectif de la solution, par contre elle dépend directement du nombre d'itération de la recherche Tabou. C'est-à-dire que nous levons le statut tabou d'une façon équitable pour toutes les solutions. Dans notre cas le statut tabou d'une solution ne dépasse pas :

$$n_tabou_limite = \frac{\text{nombre d'itérations de la recherche Tabou}}{40}$$

De ce fait la taille de notre liste Tabou $liste_Tabou_taille$, dépend directement du nombre maximum de voisin est égale à :

$$liste_Tabou_taille = n_tabou_limite * n_max_voisins$$

La procédure de mise à jour **MAJ1** permet de vérifier s'il y a des solutions à aspirer (enlever le statut tabou). Cependant, la procédure de mise à jour **MAJ2** permet d'insérer les nouvelles solutions non taboues dans la liste pour qu'elles ne soient plus visitées dans les itérations antérieures jusqu'au levé de leur statut tabou.

3.4 Conclusion

Dans ce chapitre, nous avons présenté les deux méthodes évolutionnistes élaborées pour la résolution du 1-PDPTW statique. Pour cela nous avons détaillé les nouveaux outils et algorithmes élaborés pour la diminution du temps de calcul et l'amélioration de la qualité des solutions obtenues. Ces outils sont essentiellement :

- des heuristiques pour une génération guidée de la population initiale,
- des procédures de correction vis-à-vis des différentes contraintes du problème permettant d'engendrer des solutions réalisables,

- des opérateurs de croisement contrôlés qui permettent d'assurer directement une descendance de solutions réalisables et qui ne nécessite pas de vérification ce qui permet un gain considérable en terme de temps de calcul,
- des opérateurs de mutations contrôlés qui augmentent considérablement la chance d'amélioration des solutions.

Nous avons aussi présenté l'étape d'hybridation conçu et qui consiste en une recherche Tabou spéciale dédiée pour le 1-PDPTW statique. Celle-ci renferme une nouvelle heuristique d'échange permettant de respecter les contraintes de précédence du problème et permet ainsi d'éviter l'étape de vérification vis-à-vis de ces conditions et assurer une diminution du temps de calcul. Cette hybridation est utilisée après l'application d'une des approches évolutionnistes pour l'amélioration des résultats.

Dans le chapitre suivant, nous présenterons les problèmes de tournées de véhicules dynamiques et le 1-PDPTW dynamique ainsi que les approches évolutionnistes conçues pour sa résolution.

Chapitre 4

Adaptation des outils de résolution du 1-PDPTW pour le 1-DPDPTW¹

¹ DPDPTW : Dynamic Pickup and Delivery Problem with Time Windows

4.1 Introduction

L'avancée technologique en matière de communications et d'informations, a rendu possible le traitement de données en temps réel. Par conséquent, le problème de résolution et de gestion de tournées des véhicules est devenu plus intéressant aux yeux des chercheurs vu l'apport direct qu'il peut engendrer en augmentant les performances des transports en général.

En effet, les gestionnaires des tournées de véhicules sont souvent confrontés à des situations d'urgence comme l'apparition d'un nouveau client à servir, un accident sur la route qui nécessite un détour,... Dans le cas d'un problème VRP, ceci peut être considéré comme un manque d'information lors de l'élaboration des tournées et que des clients peuvent apparaître en cours de journée quand la planification ait été déjà effectuée et que les véhicules sont en train de servir des clients déjà planifiés. Les nouveaux clients doivent alors être insérés dans les routes confectionnées de manière à satisfaire leurs besoins et ceux du gestionnaire. Le problème du VRP statique devient alors un problème dynamique DVRP. Il en découle que les variantes du VRP statique peuvent devenir des variantes dynamiques comme le cas du PDPTW et sa version dynamique DPDPTW.

Dans ce chapitre nous commencerons par présenter les problèmes de tournées de véhicules dynamiques et leurs caractéristiques, le degré de dynamisme d'un problème de tournée de véhicules et sa classification. Puis, nous introduirons la formulation mathématique générale d'un problème de tournée dynamique et les méthodes de résolution du DVRP et de ses variantes. Enfin, nous détaillerons les approches de résolution du DPDPTW élaborées.

4.2 Les problèmes de tournées de véhicules dynamiques et leurs caractéristiques

Dans le premier chapitre nous avons présenté le problème de tournées statique VRP et ses variantes. Ce problème est caractérisé par :

- La connaissance totale et complète des données du problème nécessaire à la confection des routes : coordonnées géographiques, fenêtres de temps, temps de service,...
- Ces données n'évoluent pas après la confection des routes.

Contrairement au cas statique du VRP et de ses variantes, le cas dynamique de ces derniers est caractérisé par [Larsen 2001] :

- Les données du problème ne sont pas entièrement connues avant la confection des routes. Elles peuvent augmenter ou diminuer en nombre (apparition de nouveaux clients ou suppression de clients non visités,...).
- Ces données peuvent varier après que les routes aient été confectionnées (un trajet peut prendre plus ou moins de temps pour une raison quelconque,...).

La figure 4.1 suivante illustre le cas d'un problème dynamique dans lequel un véhicule doit servir un ensemble de clients. Les clients statiques connus avant la planification des routes sont représentés par des noeuds blancs alors que les clients dynamiques qui sont apparus après la confection des routes sont représentés par des noeuds noirs. Le trajet du véhicule avant l'apparition des clients dynamiques est représenté par des flèches pleines alors que les déviations proposées pour insérer les clients dynamiques afin de les servir sont représentés par des flèches en pointillés. La position actuelle du véhicule est représentée par un véhicule.

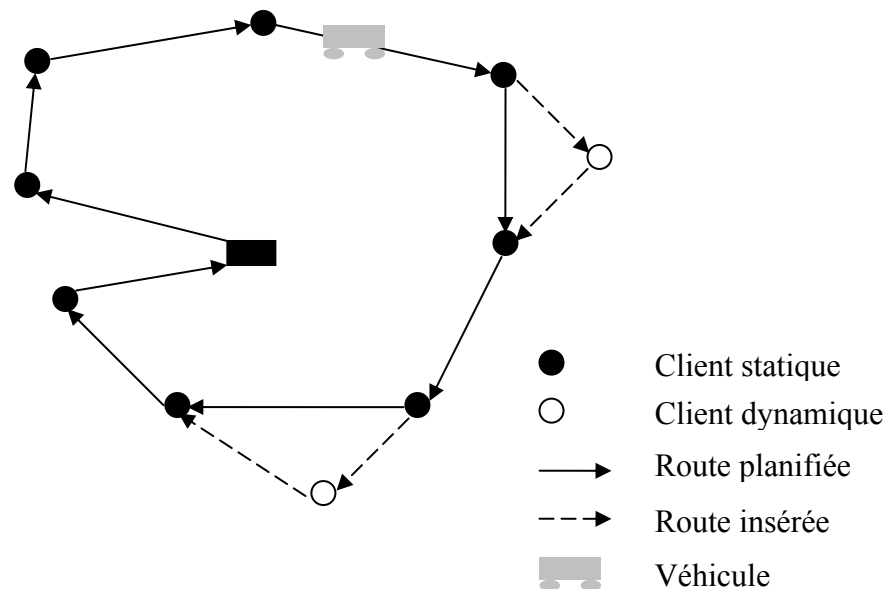


Figure 4.1 : Problème de tournée dynamique

Outre les données directes concernant le problème à résoudre telles que les positions géographiques des clients, les temps de service, etc., un cas dynamique se distingue du cas statique par plusieurs autres caractéristiques. Dans la suite nous présentons quelques une de ces particularités tirées des travaux de Larsen [2001].

- **Le temps est primordial** : Contrairement aux problèmes statiques où le temps peut ne pas avoir de l'importance, cas où l'on minimise la distance totale parcourue, le rôle du temps dans un problème dynamique est en permanence essentiel : le décideur doit

connaître en permanence la position de tous les véhicules, la liste des clients servis et la liste des clients non encore servis lors de l'arrivée d'une nouvelle requête de service.

- **Un temps de calcul court est capital** : Contrairement aux problèmes statiques où le temps de calcul pris par la planification peut atteindre des heures pour l'obtention d'une solution de haute qualité, les problèmes dynamiques ne bénéficient pas de cet avantage. Dans la plupart des problèmes dynamiques, le planificateur doit proposer une solution le plus rapidement possible pour ne pas être contraint d'arrêter un ou plusieurs véhicules.
- **La mise à jour permanente des informations est essentielle** : Vu que dans le cas d'un problème dynamique les données peuvent évoluer d'un instant à un autre, il est nécessaire d'élaborer des outils pour la mise à jour de ces changements.
- **Le réordonnement est inévitable** : Dans plusieurs cas, les décisions prises par le planificateur, lors de l'apparition d'une ou de plusieurs nouvelles demandes, aboutissent à des solutions non optimales voire mauvaises. Dans ce cas, le planificateur est dans l'obligation de réordonner les clients et les véhicules et il peut même être contraint à utiliser en renfort un ou plusieurs véhicules.
- **Les demandes les plus proches sont plus importantes** : Dans un problème dynamique, le planificateur doit, de préférence, s'occuper des demandes les plus proches en temps.
- **Les données futures peuvent être imprécises ou inconnues** : Dans les problèmes statiques, toutes les données sont connues avant la planification et de même qualité. Dans les problèmes dynamiques, la connaissance des événements futurs avec certitude n'est pas possible, et à défaut, des données avec des probabilités peuvent être employées.
- **Le choix de retarder ou d'annuler une nouvelle demande est possible** : Pour une raison justifiée comme l'éloignement géographique important où l'impossibilité de satisfaire à temps une nouvelle demande, le planificateur peut la retarder indéfiniment ou l'annuler.
- **Une file d'attente pour la gestion des arrivées des tâches est importante** : Dans le cas d'un problème dynamique, le système de planification doit posséder une file

d'attente pour garder les demandes survenues lorsque ce dernier est occupé par le réordonnement.

4.3 Les degrés de dynamisme et la classification du DVRP

Dans un problème statique d'élaboration de tournées de véhicules, la performance de la méthode de résolution dépend essentiellement du nombre des clients et de leur distribution spatiale. Alors que dans un problème dynamique, certes, le nombre des clients et leur distribution sont nécessaires à l'évaluation de la performance de la méthode de résolution, mais le nombre des événements dynamiques, et leurs instants d'apparition y jouent aussi un grand rôle. Pour cela, un critère de dynamisme s'avère utile afin d'aider à l'évaluation de la performance d'une méthode de résolution. Dans ces travaux [Larsen 2001] a utilisé la notion de degré de dynamisme pour le problème DVRP avec ou sans fenêtre de temps. Nous présentons ici les degrés de dynamisme et la classification du DVRP donnés par Larsen.

4.3.1 Degré de dynamisme sans fenêtres de temps (dod^2)

Pour le calcul du degré de dynamisme d'un problème sans fenêtres de temps, deux données sont nécessaires :

- Le nombre de clients statiques (demandes statiques).
- Le nombre de clients dynamiques (demandes dynamiques).

Le degré de dynamisme (dod : Degree Of Dynamism) est le suivant [Lund et al 1996] :

$$dod = \frac{\text{le nombre de clients dynamiques}}{\text{le nombre total des clients}} \quad (4.1)$$

Toutefois, le dod ne tient pas compte des temps d'arrivée des demandes dynamiques et ceci pourrait induire en erreur lors de l'évaluation de la performance d'une méthode de résolution. En effet, plusieurs problèmes différents peuvent avoir la même valeur dod tout en ayant des distributions de l'apparition des demandes dynamiques différentes. La Figure 4.2 montre quatre problèmes ayant la même valeur dod .

Notation :

- T : instant de fermeture du dépôt.

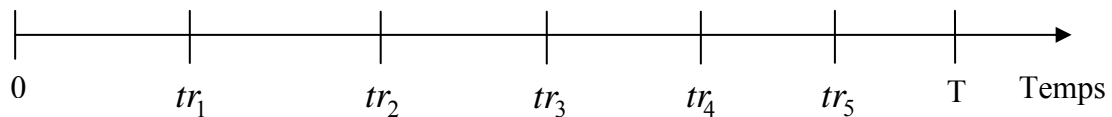
² Degree of Dynamism

- tr_i : instant de réception de la demande du client i .

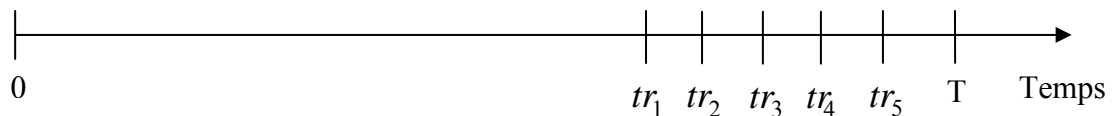
Problème 1 : (5 clients statiques, 5 clients dynamiques)



Problème 2 : (5 clients statiques, 5 clients dynamiques)



Problème 3 : (5 clients statiques, 5 clients dynamiques)



Problème 4 : (4 clients statiques, 4 clients dynamiques)

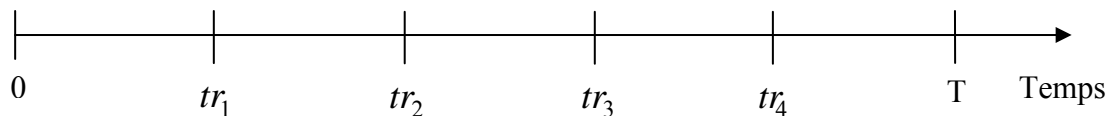


Figure 4.2 : Différents problèmes ayant les mêmes valeurs dod .

Nous remarquons à travers la Figure 4.2 que des problèmes ayant la même valeur dod peuvent avoir différents comportements. Deux problèmes ayant un nombre total de clients différents peuvent, dans ce cas, avoir le même dod et des comportements réels totalement distincts (ex : problème 1 et problème 4). Il est aussi évident que les trois premiers problèmes, ayant le même valeur de dod , ne peuvent en aucun cas être traités de la même façon par le planificateur pour donner les mêmes résultats. Il est clair que le fait d'avoir l'information concernant le cinquième client au début plutôt qu'à la fin de l'horizon de planification augmente considérablement les chances d'obtenir une meilleure solution.

4.3.2 Degré de dynamisme effectif ($edod^3$)

Le $edod$ est défini par [Larsen 2001]. En considérant :

- Un horizon de planification entre 0 et T .

³ Effective Degree of Dynamism

- Les demandes statiques sont reçues avant le commencement de l'horizon de planification et leur nombre est noté n_{sta}
- Le temps de réception des demandes dynamiques est tr_i tel que $0 < tr_i \leq T$ où i est le numéro de la demande.
- Le nombre des demandes dynamiques reçues est noté n_{dyn} .
- Le nombre total des demandes est noté n_{tot} avec $n_{tot} = n_{sta} + n_{dyn}$.

Pour définir :

$$edod = \frac{\sum_{i=1}^{n_{dyn}} \frac{tr_i}{T}}{n_{tot}} \quad (4.2)$$

Le $edod$ est compris entre 0 et 1. Lorsqu'il est égal à 0, le problème est totalement statique et lorsqu'il est égale à 1, le problème est totalement dynamique.

4.3.3 Degré de Dynamisme effectif avec fenêtres de temps ($edod-tw^4$)

Le degré de dynamisme effectif avec fenêtres de temps permet de donner une notion et une valeur d'urgence à une demande en fonction de $(l_i - tr_i)$ qui représente l'horizon de décision pour traiter la demande i . Plus cet horizon de décision est petit plus la demande est urgente.

$$edod - tw = \frac{\sum_{i=1}^{n_{tot}} \left(\frac{T - (l_i - tr_i)}{T} \right)}{n_{tot}} \quad (4.3)$$

4.3.4 Classification du DVRP

Nous présentons ici la classification du DVRP introduit dans [Larsen 2001]. Cette classification subdivise les problèmes dynamiques en trois classes : les problèmes faiblement dynamiques, les problèmes moyennement dynamiques et les problèmes fortement dynamiques comme le montre la Figure 4.3.

⁴ Effective Degree of Dynamism for problems with Time Windows

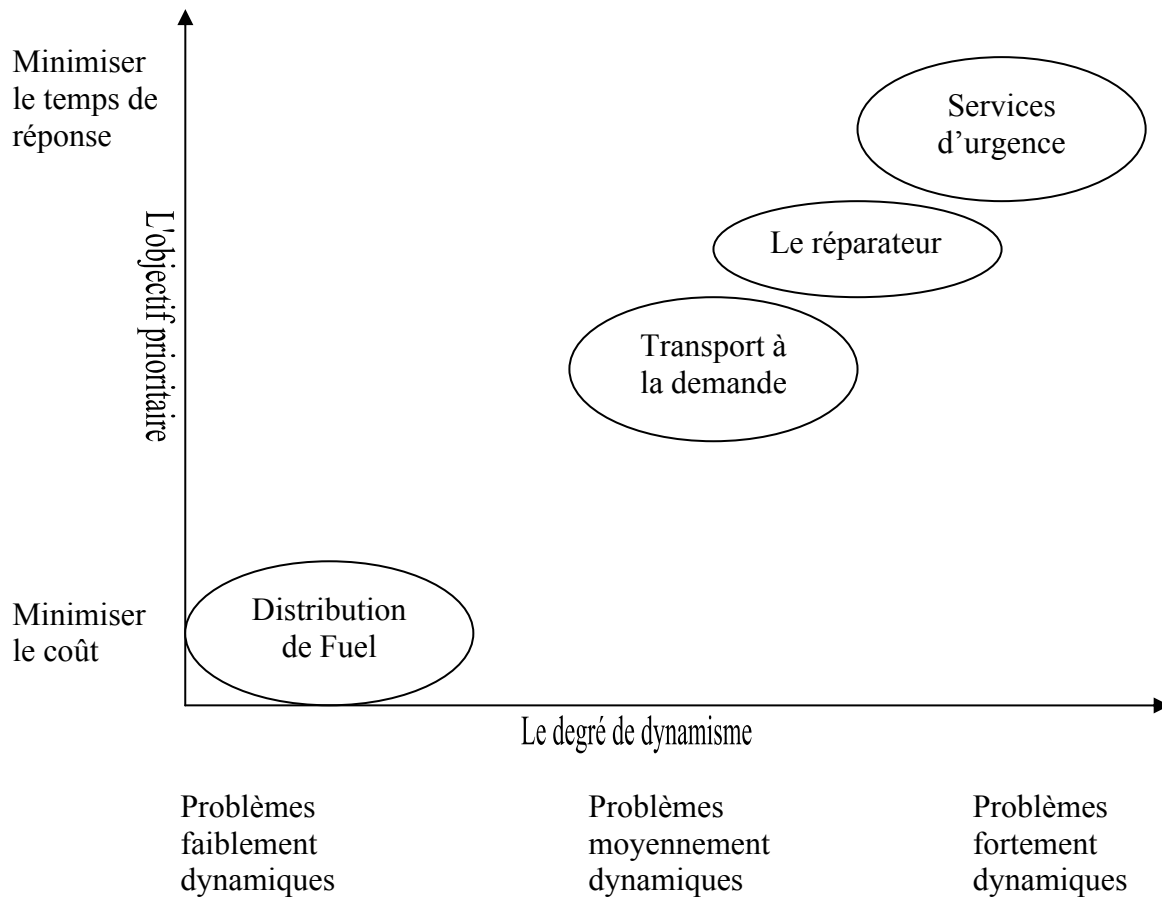


Figure 4.3 : Classification du DVRP.

- **Les problèmes faiblement dynamiques**

Dans ce type de problèmes plus de 80% des clients sont statiques et connus avant la planification. Parmi ces problèmes, nous pouvons citer la distribution de Fuel, les installateurs de services à domicile,... L'objectif principal de ce type de problèmes est de minimiser le coût total des tournées comme par exemple la distance totale parcourue par les véhicules.

- **Les problèmes moyennement dynamiques**

Dans ce type de problèmes les nombres des clients statiques et dynamiques sont proches. Ces problèmes incluent le transport à la demande, la distribution de courrier, la distribution de nourriture aux revendeurs, le service de dépannage de particuliers... L'objectif principal de ce type de problème est de minimiser le compromis entre le coût total des tournées et le temps de réponse aux demandes.

- **Les problèmes fortement dynamiques**

Ce type de problèmes concerne essentiellement les services d'urgences comme les ambulances, les pompiers et la police. L'objectif principal de ce type de problèmes est de minimiser le temps de réponse aux demandes.

4.4 Formulation mathématique générale du DVRP

Dans le premier chapitre nous avons présenté la formulation mathématique du problème VRP. Pour adapter cette dernière au problème dynamique DVRP nous ajoutons les éléments suivants concernant la réception d'une nouvelle demande:

- Variable de données :

- tr_i : instant de réception de la demande urgente

- Contraintes :

- 1^{ère} contrainte : $0 < tr_i \leq T \quad \forall i \in N'$ (4.1)

où T est l'instant de fermeture du dépôt.

- 2^{ème} contrainte : $tr_i = \alpha.e_i, \alpha \in [0, \dots, 1], \forall i \in N'$ (4.2)

La première contrainte permet de respecter le fait que les demandes dynamiques doivent parvenir au planificateur après la sortie des véhicules et avant la fin de l'horizon de planification.

La deuxième contrainte permet de vérifier que l'instant de réception de la demande dynamique doit être inférieur à la limite inférieure e_i de sa fenêtre de temps pour pouvoir être traitée.

4.5 Les méthodes de résolution du DVRP et de ses variantes

Les méthodes de résolution du DVRP et de ses variantes peuvent être divisées en deux grandes familles de méthodes : la famille des méthodes séquentielles et la famille des méthodes parallèles comme le montre la Figure 4.4.

4.5.1 Les méthodes séquentielles

Les méthodes séquentielles peuvent être composées de trois grands ensembles :

- Les heuristiques simples

- Les heuristiques d'insertion
- Les méthaheuristiques

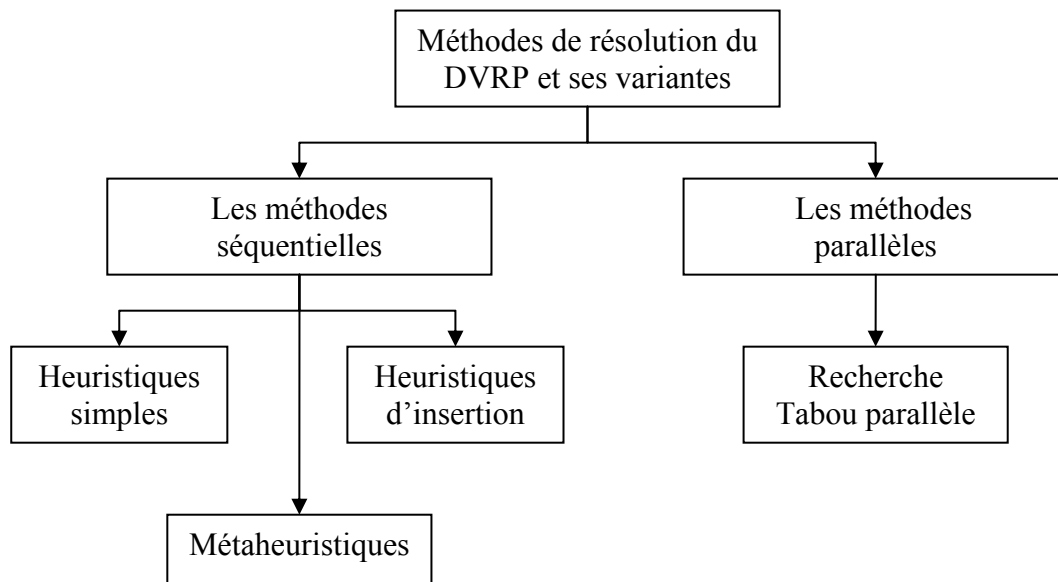


Figure 4.4 : Les méthodes de résolution du DVRP et de ses variantes.

4.5.1.1 Les heuristiques simples

Parmi les heuristiques simples les plus connues, nous pouvons citer celles présentées dans les travaux de [Larsen 2001] et [Bianchi 2000] et qui traitent le problème DTRP⁵ introduit dans le premier chapitre :

- **FCFS** (First Come First Served) : Les clients sont servis dans l'ordre où leurs demandes sont parvenues au planificateur.
- **SQM** (Stochastic Queue Median) : Cette heuristique consiste à confectionner une route médiane de la zone de service. Le réparateur quitte directement cette médiane pour aller servir un client et y retourne, si cela est possible, après que le service ait été achevé pour y attendre une nouvelle demande. Le retour à la médiane est possible s'il n'y a pas d'autres clients demandant des services et qui se trouvent aux alentours du client qui est en train d'être servi.
- **NN** ((Nearest Neighbor) : Après avoir servi un client, le réparateur passe au client le plus proche non encore servi.

⁵ Dynamic Travelling Repairman Problem

- **PART** (PARTitionning policy) : L'espace de service est subdivisé en plusieurs zones dans lesquelles est appliquée l'heuristique FCFS pour le traitement de demandes locales.
- **TSP** (Travelling Salesman Problem strategy) : En recevant les demandes, des ensembles composés de n clients sont formés et mis dans une file d'attente. Après, chaque ensemble est traité séparément comme étant un TSP.
- **mod TSP** (modified TSP) : L'espace de service est subdivisé en plusieurs zones dans lesquelles est appliquée l'heuristique TSP. Dans ce cas, une file d'attente est utilisée pour déterminer l'ordre dans lequel les zones seront traitées comme le montre la Figure 4.5 suivante.

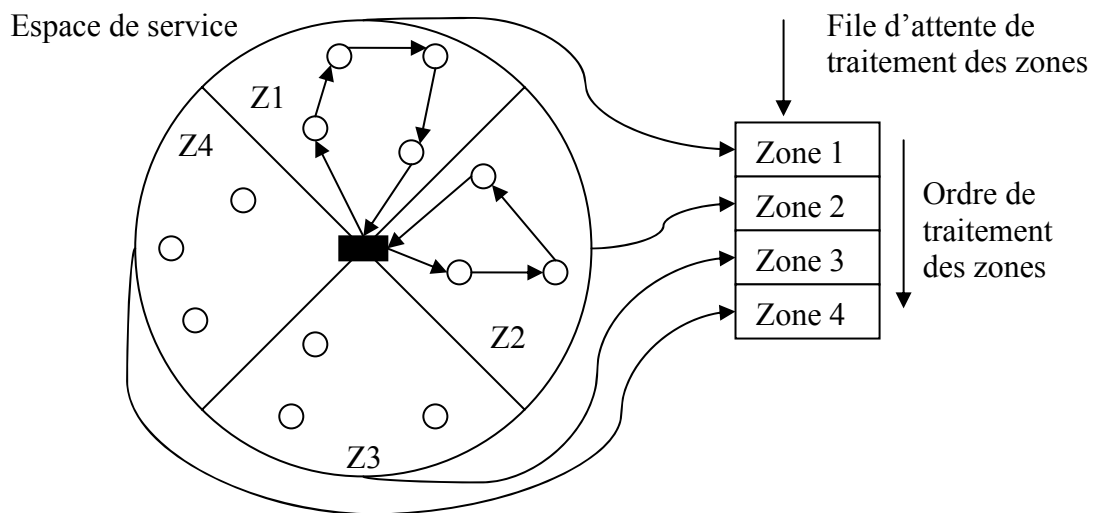


Figure 4.5 : L'heuristique mod TSP

- **GEN** (GENeration strategy) : cette heuristique a été utilisée pour la résolution du DTRP avec un seul véhicule. Son principe est le suivant :
 - Le véhicule est immobilisé en un point d'attente dans son espace de service.
 - Dès qu'une demande survient, le véhicule se déplace pour la satisfaire en créant ainsi la première génération.
 - Une fois tous les clients d'une génération servis, on vérifie s'il n'y a pas d'autres demandes reçues qui formeraient une nouvelle génération à servir en appliquant l'heuristique TSP. Le cas échéant, le véhicule revient s'immobiliser

en son point d'attente en vue de servir de nouvelles générations possibles en utilisant la même procédure.

Le fonctionnement de cette heuristique est illustré par la Figure 4.6 suivante.

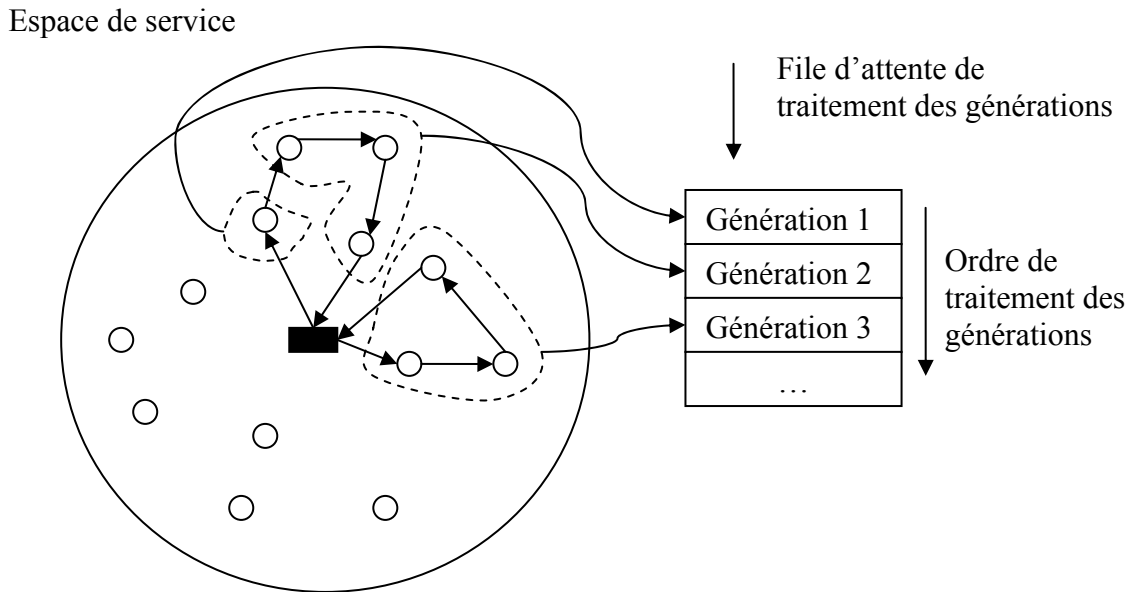


Figure 4.6 : L'heuristique GEN

Ces heuristiques ont été appliquées sur de petites instances de DTRP et elles deviennent instables dès que le nombre de demandes dynamiques augmente.

Pour la résolution du DTSPTW [Larsen et al 2002] proposent une nouvelle heuristique dynamique rapide. La journée de service est divisée en plusieurs tranches dans lesquelles apparaissent de nouvelles demandes dynamiques. Le principe de cette heuristique est le suivant :

- Résoudre le problème TSPTW pour les clients statiques.
- A $t = t_0$, le véhicule quitte le dépôt pour aller servir les clients dans l'ordre défini par l'étape précédente.
- Les clients dynamiques sont ajoutés aux clients non servis au fur et à mesure que le temps évolue.
- Chaque fois que de nouveaux clients d'une tranche horaire sont ajoutés, on relance une heuristique de résolution du TSPTW appliquée aux clients restants.

L'heuristique utilisée par Larsen pour la résolution du TSPTW utilise l'opérateur $3-opt$ conçu par [Lin 1965] pour effectuer une recherche locale sur la solution. Cette heuristique prend comme solution initiale l'ordonnancement des clients triés dans l'ordre croissant des dates e_i qui sont les débuts des fenêtres de temps. Toutefois, Larsen propose aussi, dans le cas des TSPTW de grandes tailles, l'utilisation d'opérateurs plus appropriés et plus pertinents comme l'opérateur $Or-opt$ [Or 1976].

Dans ses travaux, Larsen distingue deux cas envisageables lorsqu'un véhicule finit le service chez un client :

- Soit le véhicule arrive au prochain client avant l'ouverture de sa fenêtre de temps et il attend, ou bien il n'y a plus de clients à servir.
- Soit la fenêtre de temps du prochain client est ouverte ou le sera avant l'arrivée du véhicule.

Pour gérer ces deux cas, Larsen propose plusieurs règles dans lesquelles il utilise la notion de point d'attente de zone de service pour permettre au véhicule d'attendre l'arrivée de nouvelles demandes [Larsen et al 2002] :

- **CURRENT** : Le véhicule ne quitte pas le client servi jusqu'au moment où il peut se diriger vers un autre client non encore servi ayant une fenêtre de temps ouverte lors de l'arrivée à ce dernier.
- **NEAREST** : Une fois le dernier client servi, le véhicule se dirige vers le point d'attente le plus proche pour attendre de nouvelles demandes.
- **BUSIEST** : Une fois le dernier client servi, le véhicule se dirige vers le point d'attente ayant le plus grand taux d'arrivée de demandes dynamiques.
- **HI-REQ** (Highest expected number of immediate REQuests) : Une fois le dernier client servi, le véhicule se dirige vers le point d'attente ayant le score d'attraction le plus élevé. Le score d'attraction d'une zone de service est le nombre de demandes dynamiques prévues dans cette zone pendant une tranche de temps.

Les deux dernières règles ont été utilisées pour traiter le cas du ADTSPTW (A-priori DTSPWT) car elles sont basées sur une manipulation des données de l'historique concernant les taux d'apparition des demandes dynamiques.

- **STATIC** : Cette règle a le même principe que la règle CURRENT sauf que tous les clients sont considérés comme des clients statiques.
- **REOPT** : Cette règle consiste à satisfaire tous les clients dès qu'ils apparaissent et sans tenir compte de l'ouverture de leurs fenêtres de temps. C'est-à-dire qu'ils peuvent être servis en avance moyennant une certaine pénalité de service. En utilisant cette règle le véhicule ne s'arrête pas aux points d'attente.

Une des autres variantes du DVRPTW a été traitée par [Yang et al 2002]. Cette variante concerne un problème de transport de marchandises avec des camions. Chaque camion ne peut servir qu'un client à la fois : il transporte la marchandise du lieu de ramassage directement vers le lieu de livraison. Dans ce problème le planificateur peut refuser une demande si cela pourrait lui créer un problème quelconque. Dans la suite nous allons présenter le principe de chaque règle élaborée et utilisée par Yang pour la résolution de ce problème :

- **BENCH** : Cette règle consiste à calculer le coût engendrer par l'ajout d'un client dynamique à la fin de la tournée de chaque camion, puis à mettre le client sur la tournée qui aura le coût le plus faible.
- **NS** : Cette règle est identique à la précédente sauf que le client peut dans ce cas être inséré au milieu d'une tournée.
- **SE** : Dans les deux dernières règles le fait d'ajouter une nouvelle demande dynamique ne changeait pas l'ordre des anciennes demandes. Dans cette règle le planificateur réordonne toutes les demandes non servies.
- **MYOPT** : Cette règle rassemble toutes les demandes et traite le problème avec une heuristique en le considérant comme étant totalement statique.
- **OPTUN** : Cette règle est la même que la précédente sauf que le planificateur prend en compte les demandes futures en utilisant une loi de probabilité sur les positions des clients dynamiques.

4.5.1.2 Les heuristiques d'insertion

Contrairement aux heuristiques simples qui ne sont efficaces qu'avec des problèmes spécifiques comme le DTRP, les heuristiques d'insertion sont performantes sur un grand nombre de variantes du DVRP. Présenté par [Roy et al 1984] une heuristique d'insertion optimise de nouveau les tournées des véhicules lors de l'arrivée de la nouvelle demande et

cherche à insérer le nouveau client dans la meilleure position des tournées actuelles. Cette heuristique simple peut être appliquée sur plusieurs problèmes dynamiques. La Figure 4.7 illustre le principe d'une heuristique d'insertion classique.

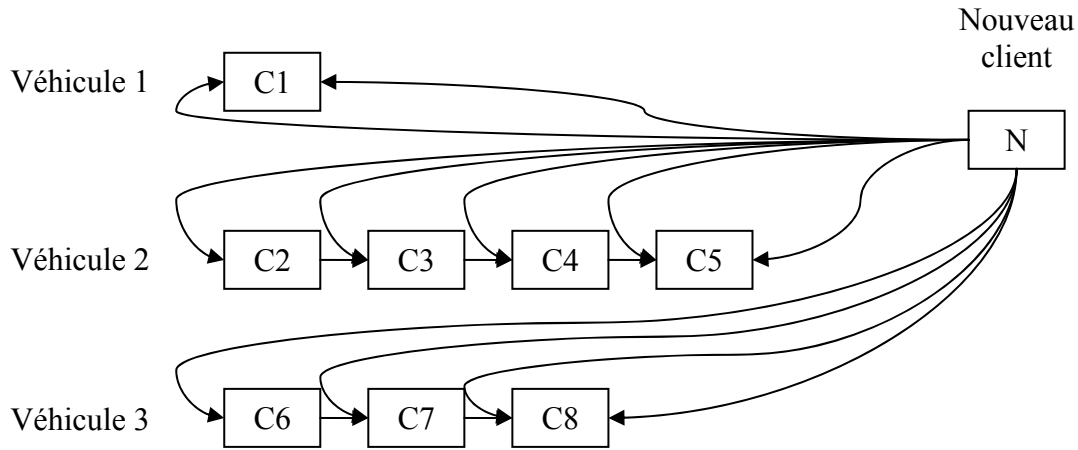


Figure 4.7 : L'heuristique d'insertion classique

Dans cette figure nous remarquons que l'heuristique teste toutes les possibilités d'insertion du nouveau client N. Pour cela elle parcourt toutes les positions d'insertion possibles dans la matrice des tournées. Cette matrice ne contient que les clients non servis, les clients déjà visités n'y sont pas représentés.

[Larsen 2001] a utilisé cette méthode pour résoudre le DVRPTW à travers un simulateur qui est composé d'un module de contrôle et d'un module de routage. Le principe du simulateur de Larsen est décrit par la Figure 4.8 suivante.

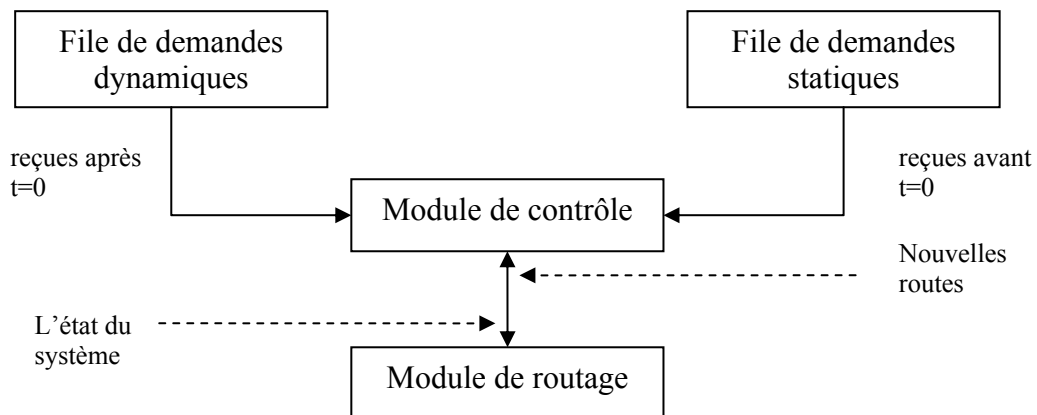


Figure 4.8 : Simulateur du système dynamique [Larsen 2001].

Le module de contrôle a pour rôle la gestion des véhicules. Ce module s'occupe de l'association des clients aux véhicules et traite les événements qui peuvent apparaître au cours du temps pour changer l'état du système. Par exemple, ces événements peuvent être l'arrivée d'une nouvelle demande ou la fin de service chez un client. Ce module vérifie l'état du système toutes les cinq minutes et l'envoie au module de routage.

Le module de routage a pour rôle de fournir la solution initiale au module de contrôle pour que ce dernier puisse affecter les clients au véhicule. Ce module de routage permet d'ordonnancer de nouveau les tournées présentes si un changement d'état du système lui est rapporté par le module de contrôle. Il peut alors dans ce cas lancer une heuristique d'insertion pour satisfaire une nouvelle demande, ou bien affecter un nouveau client à un véhicule venant de finir le service chez un client.

Dans leurs travaux [Kilby et al 1998] utilisent cette méthode d'insertion sur une famille de DVRP. Dans ces études, le temps de simulation qui n'est autre que le temps de service d'une journée était subdivisé en un total de cinquante intervalles horaires. Au cours de chaque intervalle, le planificateur rassemble toutes les nouvelles demandes, les traite et les affecte dans le prochain intervalle. Au même temps le planificateur tente d'améliorer la solution obtenue en utilisant une heuristique appliquant l'opérateur de recherche locale $2 - Opt$ de [Lin 1965].

Pour la résolution du DPDPTW, [Mitrovic-Minic et al 2004] élaborent un concept de double espace. Lors de l'apparition d'une nouvelle demande, un effort considérable de calcul est consacré à l'optimisation de la partie à court terme qui concerne le ramassage des colis chez le fournisseur, alors qu'un effort de calcul moins important est dédié à l'optimisation à long terme qui concerne la livraison du colis chez le client. La principale hypothèse de cette approche est que les parties éloignées des tournées sont susceptibles d'être sujettes à plusieurs changements et ne nécessitent donc pas un grand effort d'optimisation. Une fois la solution construite, [Mitrovic-Minic et al 2003] utilisent une des quatre procédures d'attente suivantes:

- **WF** (Wait First): Dans cette procédure d'attente le véhicule reste chez le dernier client servi et ne le quitte que pour arriver au prochain client juste au moment de l'ouverture de sa fenêtre de temps.
- **DF** (Drive First): Dès que le véhicule finit de servir le client courant, il le quitte vers le prochain à visiter. S'il arrive lors ou après l'ouverture de la fenêtre de temps de ce dernier, il le sert, sinon il attend.

Dans les deux procédures suivantes, la tournée d'un véhicule est partitionnée en zones, où chacune d'elles renferme un ensemble de clients successifs et regroupés lors de la visite du véhicule.

- **DW** (Dynamic Waiting): Cette procédure est une combinaison des deux premières. Une tournée est construite de la manière suivante :
 - Le planificateur applique la procédure DF pour gérer le véhicule dans une zone de service.
 - Une fois le service d'une zone terminé, le planificateur applique la procédure WF pour déplacer le véhicule vers une autre zone non servie.
- **ADW** (Advanced Dynamic Waiting): Le but de cette procédure est de répartir le temps d'attente sur tous les clients d'une tournée en se basant sur la procédure DW. Toutefois, le temps d'attente chez le dernier client visité dans une tournée est calculé d'une autre manière. Ce dernier peut être proportionnel au rapport de la durée d'attente totale de sa zone sur la somme de toutes les durées d'attente de toutes les tournées.

4.5.1.3 Les métaheuristiques

Les métaheuristiques ont été moins appliquées que les méthodes précédentes pour la résolution des problèmes de transport dynamiques. Parmi ces travaux nous pouvons citer ceux de [Montamenni et al 2002] qui utilisent les colonies de fourmis pour la résolution du DVRP comme suit :

- La journée est partitionnée en plusieurs intervalles de temps et dans chacun d'eux, un VRP statique issu des demandes statiques est résolu.
- Les demandes dynamiques d'un intervalle de temps sont traitées dans l'intervalle suivant. Ainsi, dans un intervalle de temps, les demandes statiques anciennes et les nouvelles demandes du précédent intervalle sont regroupées et réordonnées selon un VRP statique.

[Housroum et al 2003], [Housroum et al 2004] et [Housroum 2005] présentent une métaheuristique basée sur les algorithmes génétiques pour résoudre le DVRPTW. Cette méthode consiste à résoudre le problème « en ligne » (On-line), c'est-à-dire qu'elle traite toute nouvelle demande au moment où elle apparaît.

Une recherche Tabou couplée à une heuristique d'insertion a été utilisée par [Mitrovic-Minic et al 2001] pour la résolution du DPDPTW. Le principe de cette heuristique d'insertion a été présenté dans le paragraphe précédent.

4.5.2 Les méthodes parallèles

L'utilisation des méthodes parallèles a pour objectif principal de diminuer le temps de calcul. Néanmoins, pour pouvoir implémenter une telle méthode, la possession de matériel à architecture spéciale peut être requise comme par exemple un calculateur multiprocesseurs

Parmi les méthodes parallèles pour la résolution des problèmes de transport dynamiques que nous trouvons dans la littérature, nous pouvons citer les travaux de [Gendreau et al 1998b] et [Gendreau et al 1999]. Ces travaux permettent de résoudre respectivement le DPDPTW et le DVRPTW en utilisant un algorithme de recherche Tabou adaptative parallèle dont le pseudo-code est décrit par la figure 4.9 suivante.

```
Début  
Tant que « aucun événement »  
    Améliorer la solution courante en utilisant la recherche Tabou  
    Si un événement se produit,  
        Alors arrêter tous les processus de recherche Tabou et ajouter les  
        meilleures solutions à la mémoire adaptative qui représente un ensemble  
        de bonnes solutions.  
        Si le nouvel événement est une nouvelle demande,  
            Alors Mettre à jour la mémoire adaptative par l'insertion de la  
            nouvelle demande dans chaque solution.  
            Si l'insertion de la nouvelle demande est impossible,  
                Alors rejeter la demande  
        Sinon (l'événement est une fin de service d'un client)  
            Alors identifier la prochaine destination du véhicule en utilisant la  
            meilleure solution stockée dans la mémoire adaptative,  
            Mettre à jour les autres solutions en conséquence,  
            Relancer les processus de recherche tabou avec les nouvelles  
            solutions obtenues à partir de la mémoire adaptative.  
    Finsi  
Finsi  
FinTantque  
Fin
```

Figure 4.9 : Le Pseudo-code de la recherche Tabou adaptative parallèle [Gendreau et al 1998b]
[Gendreau et al 1999].

Cet algorithme a été conçu pour une utilisation en parallèle et testé sur une machine multiprocesseur dans laquelle chaque processeur s'occupe de lancer un seul algorithme de recherche Tabou.

En comparant les solutions données par cette méthode avec celles données par d'autres outils de résolution de problèmes de transport dynamiques de la littérature, les auteurs ont montré que l'utilisation d'une méthode parallèle avec une recherche Tabou permet d'avoir de bien meilleurs résultats. Cet algorithme a aussi été testé par [Gendreau et al 2001] pour traiter le problème dynamique des ambulances.

4.6 Approche dynamique proposée pour le 1-DPDPTW [Kammarti et al. 2006]

Le problème 1-DPDPTW est une extension du 1-PDPTW. En plus des règles de ce dernier, le 1-DPDPTW se distingue par la possibilité d'apparition de demandes dynamiques et urgentes après que le véhicule ait quitté le dépôt. Le planificateur devra alors satisfaire les nouvelles demandes en plus des anciennes tout en optimisant les critères choisis et en respectant les contraintes imposées.

Dans cette partie nous présentons les approches évolutionnistes et évolutionniste hybride pour la résolution du 1-DPDPTW. Ces approches adoptent la même représentation des solutions adoptées dans le cas statique (voir paragraphe 3.2.3.3 « codage par liste de permutation »). Elles sont aussi basées sur l'adaptation de la méthode de génération des solutions initiales, de l'opérateur de croisement, des opérateurs de mutation et des procédures de correction initialement conçus pour le cas statique pour pouvoir servir dans le cas dynamique. Dans ces approches, l'ajout d'une ou de plusieurs demandes urgentes peut être effectué à n'importe quel moment après le départ du véhicule.

Les principes des autres méthodes de résolution des problèmes de transport dynamiques sont basés essentiellement sur la date d'arrivée de la demande dynamique. Dans notre cas, nous utilisons une image directe du temps d'arrivée d'une demande dynamique qui n'est autre que la position du véhicule à cet instant précis. Nous supposons alors qu'un véhicule ne peut changer de route qu'après avoir servi le client vers lequel il se dirigeait. Les figures 4.10 et 4.11 illustrent le principe de notre 1-DPDPTW.

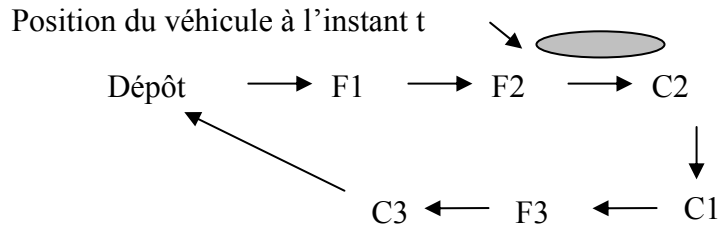


Figure 4.10 : La tournée du véhicule avant l'apparition de la demande dynamique urgente.

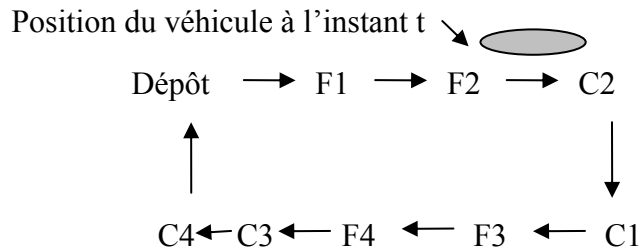


Figure 4.11 : La tournée du véhicule après l'apparition de la demande dynamique urgente.

A travers ces deux figures nous remarquons bien que l'insertion des éléments (F4 : fournisseur 4, C4 : client 4) de la demande dynamique se fait en aval du client C2.

Dans la Figure 4.12 nous détaillons la stratégie de résolution du 1-DPDPTW. Dès que le véhicule quitte le dépôt, nous vérifions en permanence l'apparition de nouvelles demandes urgentes. Si cela n'est pas le cas, le véhicule continue sa tournée normalement, sinon le circuit courant que suit le véhicule sera considéré comme solution initiale pour l'application de l'une des approches évolutionniste pour la résolution du 1-DPDPTW. Le résultat élaboré est un nouveau circuit qui sera immédiatement communiqué au véhicule.

Le fait de considérer la solution précédente du problème (problème sans demandes dynamiques), comme solution initiale de l'approche évolutionniste à appliquer permet de réduire considérablement le temps de calcul puisque cela permet de garder quelques bonnes parties de celle-ci.

4.6.1 Formulation mathématique

La formulation mathématique adoptée pour résoudre le problème 1-DPDPTW est la formulation utilisée dans la 1^{ère} approche évolutionniste pour la résolution du 1-PDPTW à laquelle nous ajoutons les contraintes (4.1) et (4.2). Nous aurons alors à minimiser la fonction (3.1) :

$$\text{MINIMISER } F_1 = \left(\begin{array}{l} \alpha_1 \sum_{i \in N} \sum_{j \in N} d_{ij} X_{ij} + \\ \alpha_2 \sum_{i \in N} \sum_{j \in N} \max(0, e_i - A_i) + \\ \alpha_3 \sum_{i \in N} \sum_{j \in N} \max(0, l_i - D_i) \end{array} \right)$$

Cette fonction fixe l'objectif des approches évolutionnistes : minimiser la somme globale de la distance totale parcourue, le total des temps d'attente évité et le total des retards.

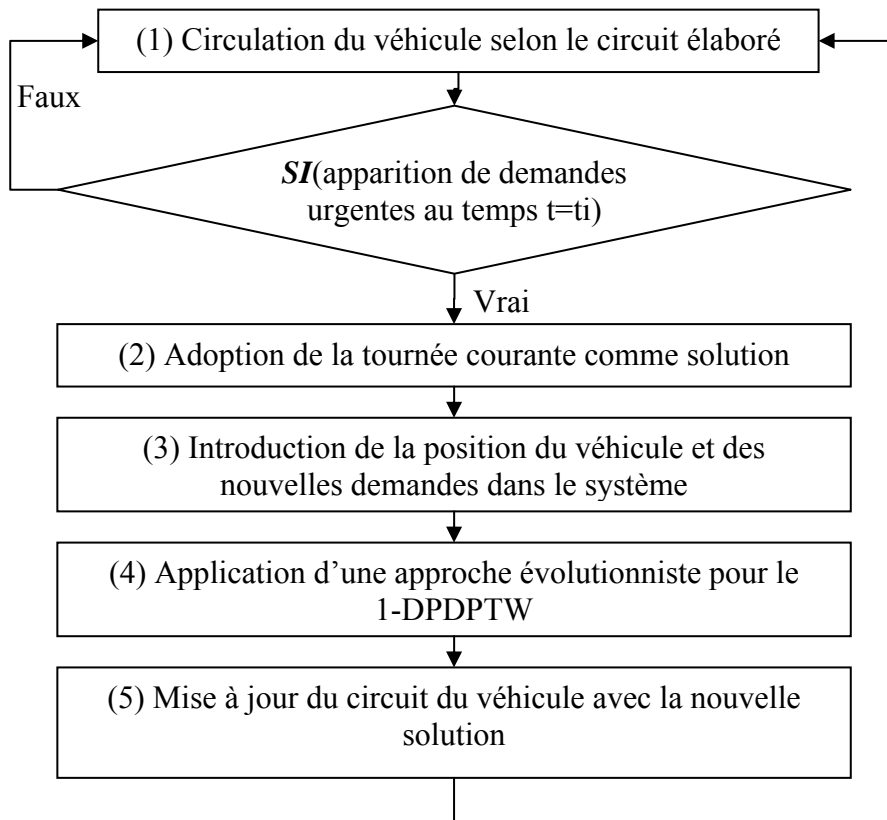


Figure 4.12 : Stratégie de résolution du 1-DPDPTW.

4.6.2 Structure générale des algorithmes évolutionnistes élaborés

Les approches évolutionnistes élaborées sont au nombre de deux. La première est une approche évolutionniste utilisant des particularités conçues pour le problème dynamique et la deuxième est une approche évolutionniste hybride utilisant la première en lui ajoutant à la fin un étage d'amélioration basé sur une recherche Tabou dédiée au problème 1-DPDPTW.

Ces deux approches ont été séparées afin de permettre à l'utilisateur de l'outil de résolution d'avoir la possibilité de choisir une d'elles. Ceci nous permettra aussi de juger les

performances de ces deux approches séparément et surtout l'apport qu'engendre l'étape d'amélioration sur le raffinement d'une solution.

Le principe de la procédure d'évolution des solutions est présenté dans la figure 3.14. Nous présentons dans la figure 4.13 la structure de l'algorithme évolutionniste hybride élaboré pour la résolution du 1-DPDPTW. Cet algorithme est appliqué après la fabrication des solutions avec la première approche évolutionniste. Il est utilisé lors de l'étape (4) de la stratégie de résolution présentée dans la figure 4.12.

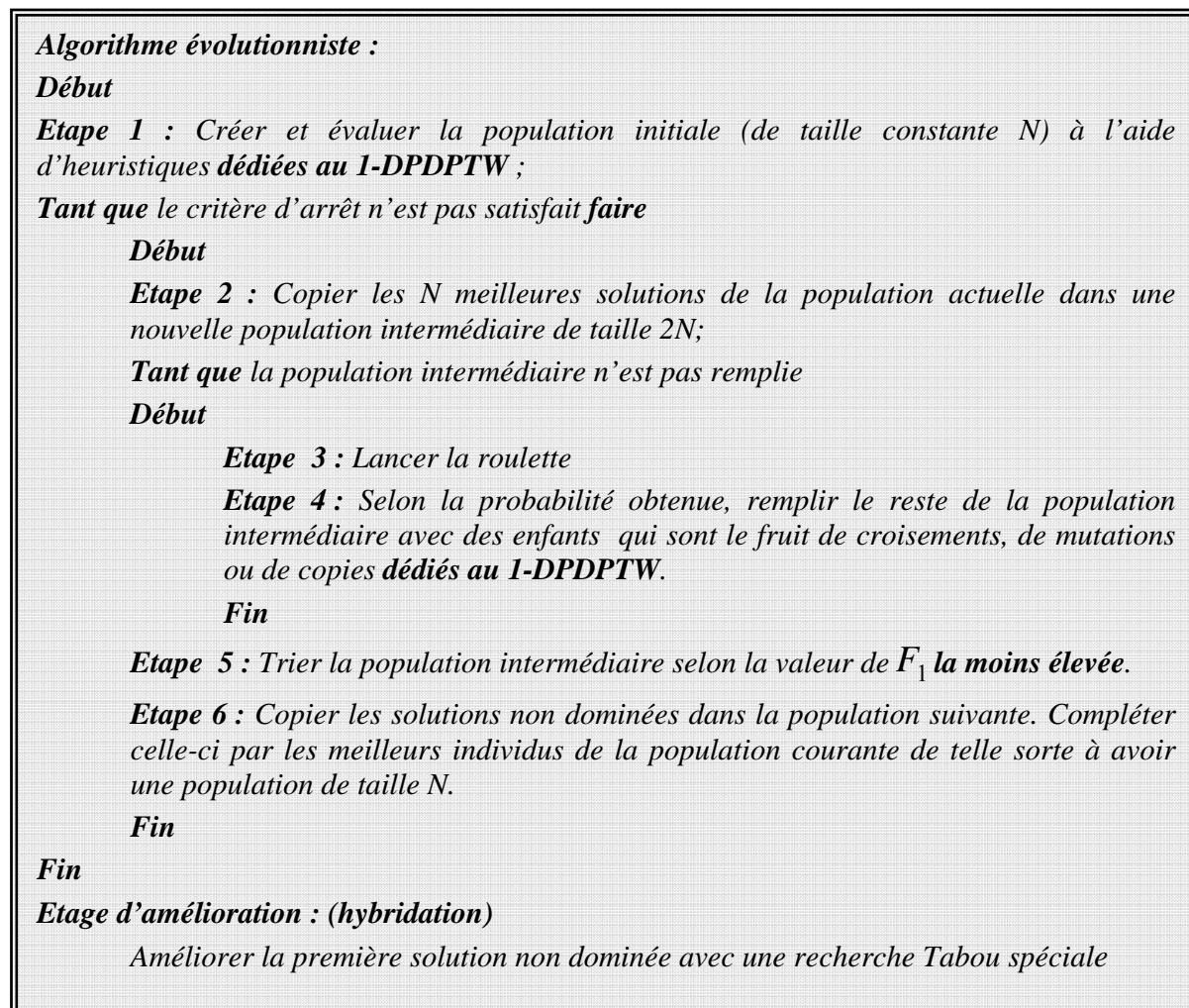


Figure 4.13 : Structure de l'algorithme de la 1^{ère} approche évolutionniste pour le 1-DPDPTW

4.6.3 Génération de la population initiale

Dans le problème 1-DPDPTW, le véhicule ne doit en aucun cas visiter un client déjà servi. Pour assurer le fait que le véhicule a visité plusieurs fournisseurs et clients, nous avons conçu de nouvelles heuristiques pour la génération de la population initiale. La figure 4.14 introduit le principe de ces heuristiques.

A travers la figure 4.14 nous observons que les sommets à gauche de la position actuelle du véhicule ($n_vehicule$) restent figés : ce sont des sommets déjà visités et ceci représente la particularité commune des heuristiques de générations de la population initiale. Dans la suite, nous détaillerons les principes de ces heuristiques qui ont pour but de minimiser respectivement la distance totale parcourue, le total des temps d'attente et le total des retards.

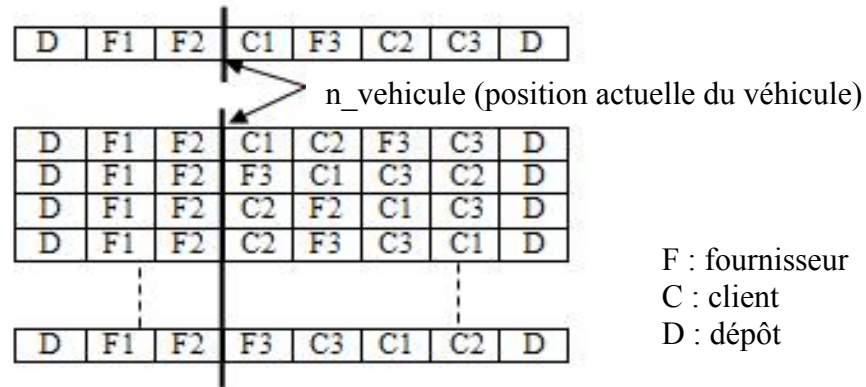


Figure 4.14 : Principe des heuristiques de génération de la population initiale pour le 1-DPDPTW

4.6.3.1 Minimisation de la distance totale parcourue (*heuristique 1*)

Pour minimiser la distance totale parcourue dans les individus de la population initiale dans le problème 1-DPDPTW, une heuristique a été élaborée. Le principe de cette heuristique est de fixer le dépôt comme sommet d'arrivée puis de s'assurer que le véhicule voyage du point $n_vehicule$ au dépôt d'arrivée en passant d'un sommet à celui qui lui est le plus proche. Cette procédure est répétée jusqu'à obtenir le nombre désiré d'individus à mettre dans la population initiale. La Figure 4.15 illustre le principe de cette heuristique.

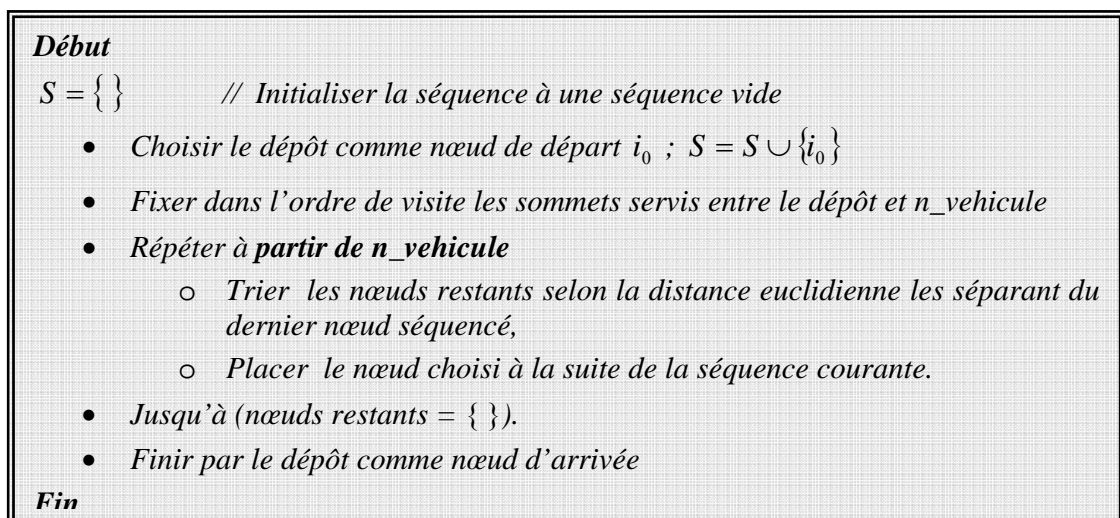


Figure 4.15 : Algorithme de l'heuristique 1.

4.6.3.2 Minimisation de la somme des temps d'attente (*heuristique2*)

Basée sur les caractéristiques temporelles du problème, cette heuristique a pour objectif de minimiser la somme des temps d'attente. Son principe est de fixer le dépôt comme sommet d'arrivée et, partant de la position $n_vehicule$, de visiter les sommets restants (à droite de $n_vehicule$) triés dans l'ordre décroissant de la somme des date de fin de fenêtre temps et du temps de service de chacun simultanément $(l_i + s_i)$.

4.6.3.3 Minimisation de la somme des retards (*heuristique3*)

Cette heuristique utilise, elle aussi, les caractéristiques temporelles du problème pour minimiser la somme des retards. Le principe de la deuxième heuristique est de fixer le dépôt comme sommet d'arrivée et de visiter les sommets restants (à droite de $n_vehicule$) triés dans l'ordre croissant de la date de fin de fenêtre de temps.

4.6.4 Conception des solutions réalisables et correction des individus non viables : les procédures de vérification et de correction

Outre l'obligation de satisfaire les contraintes de capacité du véhicule et de précedence entre les fournisseurs et les clients, les solutions proposées doivent respecter aussi le fait que les sommets servis ne doivent en aucun cas être revisités. Pour cela, nous avons élaboré un moyen de vérification et de correction des solutions non viables. Ce moyen consiste à repérer les gènes, à droite de la position $n_vehicule$, mal placés et de rectifier leur position.

Ceci nous permet d'avoir une population initiale totalement composée d'individus réalisables et de corriger les individus non réalisables après l'étape de mutation. Ce moyen est composé de deux procédures de correction : la première concernant les contraintes de précedence et la deuxième les contraintes de capacité.

4.6.4.1 Vérification et correction des individus vis-à-vis des contraintes de précedence

Pour ne plus visiter les sommets déjà servis et assurer le passage du véhicule par un fournisseur avant son client respectif, nous avons élaboré une procédure de correction. L'algorithme de la procédure de correction des individus vis-à-vis des contraintes de précedence pour le 1-DPDPTW est donné par la Figure 4.16.c de la page suivante.

Le principe de cette procédure est de parcourir le chromosome gène par gène, de la position $n_vehicule$ vers le dépôt d'arrivée, et de vérifier quand le noeud est un client si son fournisseur a bien été visité. Si ce n'est pas le cas, nous insérons ce fournisseur juste devant le client.

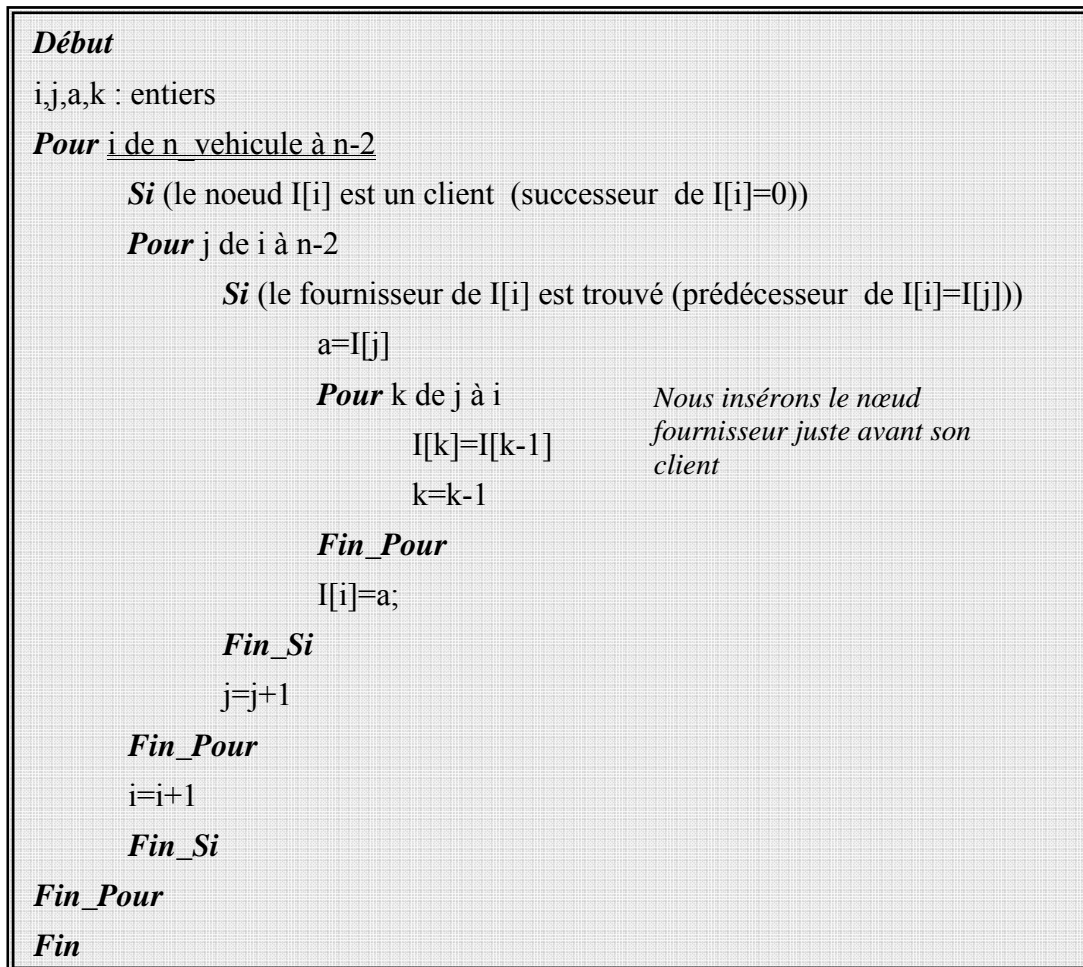


Figure 4.16.a : Correction des individus vis-à-vis des contraintes de précédence dans un 1-DPDPTW.

Exemple 4.1 :

Nous reprenons dans cet exemple les données de l'exemple 3.5 du troisième chapitre pour le traiter dans le cas dynamique 1-DPDPTW.

Soient les couples de (fournisseur, client) suivants : (3,1) ; (10,2) ; (5,7) ; (8,4) et (9,6).

Le dépôt est le nœud 0. Considérons l'individu

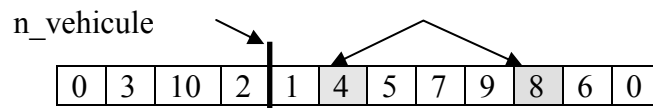


Figure 4.16.b : Individu non viable vis-à-vis des contraintes de précédences dans un 1-DPDPTW.

L'algorithme permet de localiser le gène (8) mal placé et de l'insérer avant son client (4) pour donner l'individu viable de la Figure 4.16.c.

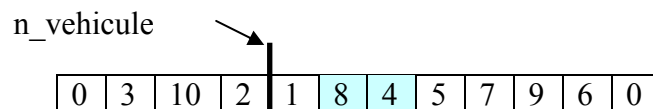


Figure 4.16.c : Individu viable vis-à-vis des contraintes de précédence dans un 1-DPDPTW.

4.6.4.2 Vérification et correction des individus vis-à-vis des contraintes de capacité

Pour ne plus visiter les sommets déjà servis et assurer le respect de la capacité maximale du véhicule, nous avons élaboré une procédure de correction. L'algorithme de correction des individus vis-à-vis des contraintes de capacité dans le cas du 1-DPDPTW est illustré dans la figure 4.17.c.

Le but de cette procédure est d'assurer, après avoir appliqué la procédure de correction de précedence pour le 1-DPDPTW, la vérification et la correction si cela est nécessaire vis à vis des contraintes de capacité. Son principe est de parcourir le chromosome gène par gène, de la position n-vehicule vers le dépôt d'arrivée, et de calculer la quantité transportée. Si celle-ci excède la capacité du véhicule, la procédure revient au fournisseur précédent qui n'est pas suivi par son client puis elle le fait immédiatement suivre par ce dernier.

Exemple 4.2 :

Nous reprenons dans cet exemple les données de l'exemple 3.6 du troisième chapitre pour le traiter dans le cas dynamique 1-DPDPTW.

Soient les couples de (fournisseur, client) suivants : (3,1) ; (10,2) ; (5,7) ; (8,4) et (9,6).

Le dépôt est le nœud 0 ; $Q = 60$; $q[1]=-20$; $q[2]=-20$; $q[3]=20$; $q[4]=-20$; $q[5]=20$; $q[6]=-20$; $q[7]=-20$; $q[8]=20$; $q[9]=20$; $q[10]=20$;

Considérons l'individu :

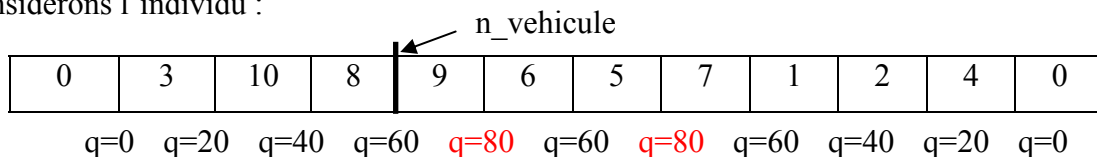


Figure 4.17.a : Individu non viable vis-à-vis des contraintes de capacité dans un 1-DPDPTW.

L'algorithme permet de localiser tout d'abord le gène (9) mal placé et de rapprocher du fournisseur précédent (8) son client (4) pour donner l'individu viable suivant :

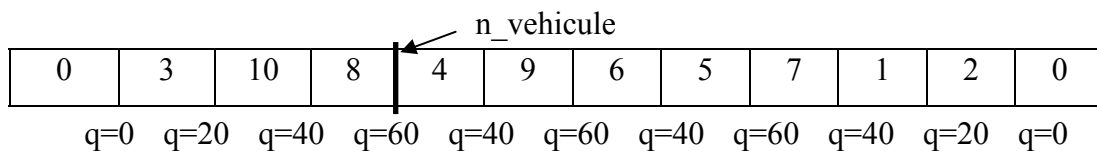


Figure 4.17.b : Individu viable vis-à-vis des contraintes de capacité dans un 1-DPDPTW.

```

Début
quant, Q : réel
i,a,j,k,f,test : entiers
quant=0
Q = quantité maximale du véhicule
Pour i de n_vehicule à n-2
    quant=quant+quantité du noeud I[i]
    Si (quant>Q)
        Pour f de i-1 à 1
            Si (le noeud I[f] est un fournisseur (prédécesseur de I[f]=0))
                test=0
                Pour k de f+1 à i
                    Si (le client de I[f] est trouvé (successeur de I[f]=I[k]))
                        test=1
                    Fin_Si
                    k=k+1
                Fin_Pour
            Si (test=0)
                Pour j de i+1 à n-1
                    Si (le client de I[f] est trouvé (successeur de
                                                                I[f]=I[j]))
                        a=I[j]
                        Pour k de j à f+1)
                            I[k]=I[k-1];
                        Fin_Pour
                        k=k-1
                        I[f+1]=a;
                    Fin_Si
                Fin_Pour
                j=j+1
            Fin_Si
        Fin_Si
        f=f-1
    Fin_Pour
    Fin_Si
    i=i+1
Fin_Pour
Fin

```

*Nous insérons
le nœud client
juste après son
fournisseur*

Figure 4.17.c : Correction des individus vis-à-vis des contraintes de capacité dans un 1-DPDPTW.

4.6.5 Croisement

Pour ne plus visiter les sommets déjà servis, assurer la viabilité des solutions et décroître le temps total de calcul, nous avons conçu un opérateur de croisement qui ne nécessite pas de procédures de correction.

Le principe de cet opérateur est basé sur celui de l'opérateur 1.X. Nous choisissons au hasard une position de croisement p impérativement à droite de la position $n_vehicule$ pour assurer le non retour aux sommets déjà visités. Après, nous fixons le dépôt comme sommet d'arrivée et nous parcourons la solution (chromosome) gène par gène du dépôt de départ vers la position de croisement p . Si le sommet est un fournisseur respectant les contraintes de capacité nous le copions dans l'enfant correspondant. Si le sommet est un client et si son fournisseur a été servi nous le copions aussi dans l'enfant correspondant. Le fait de partir du dépôt de départ et non de la position $n_vehicule$ n'influe en aucun cas sur l'ordonnement des sommets déjà visités puisqu'ils sont conformes à toutes les contraintes.

Pour compléter le premier enfant du croisement, nous parcourons le deuxième parent gène par gène. Si le sommet est un fournisseur pas encore servi dans le premier enfant et si son insertion dans ce dernier permet de vérifier la contrainte de capacité, alors nous le copions dans cet enfant. Si le sommet est un client dont le fournisseur est visité dans le premier enfant alors nous le copions dans ce dernier. Sinon nous répétons cette procédure jusqu'à compléter le premier enfant. Puis nous appliquons cette procédure pour générer le deuxième enfant en utilisant ses deux parents.

Exemple 4.3 :

Nous reprenons dans cet exemple les données de l'exemple 3.10 du troisième chapitre pour le traiter dans le cas dynamique 1-DPDPTW.

Soient les couples de (fournisseur, client) suivants : (3,1) ; (10,2) ; (5,7) ; (8,4) et (9,6). Le dépôt est le nœud 0. $Q = 60$; $q[1]=-20$; $q[2]=-20$; $q[3]=20$; $q[4]=-20$; $q[5]=20$; $q[6]=-20$; $q[7]=-20$; $q[8]=20$; $q[9]=20$; $q[10]=20$;

La Figure 4.18.a montre un exemple d'une opération de croisement.

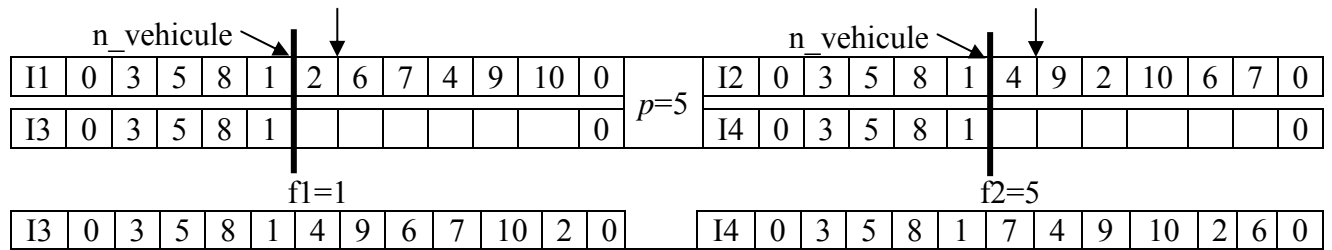


Figure 4.18.a : Croisement pour le 1-DPDPTW.

L'algorithme de l'opérateur de croisement pour le 1-DPDPTW est alors similaire à celui du 1-PDPTW (Figure 3.10.b). Sauf qu'au lieu de choisir une position de croisement au hasard sur tout le chromosome, il est impératif et primordial de la choisir aléatoirement entre la position $n_vehicule$ et le dernier sommet avant le dépôt d'arrivée. La Figure 4.18.b illustre le début de cet algorithme qui sera complété avec les mêmes instructions de la Figure 3.10.b.

```
// I1, I2 les individus parents et I3, I4 les individus enfants
// n: nombre de gènes: nombre des fournisseurs, clients et le (dépôt *2) (départ et arrivée)
Début
• Choisir au hasard une position de croisement "p" dans [n_vehicule, dernier sommet]
• Fixer le dépôt comme noeud de départ et d'arrivée
... (instructions restantes de la Figure 3.10.b)
```

Figure 4.18.b : l'algorithme de croisement pour le 1-DPDPTW.

4.6.6 Mutation

Lors de chaque étape de mutation, nous utilisons un de ces opérateurs spéciaux de mutation pour le 1-DPDPTW :

- **Mutation 1** : C'est un opérateur contrôlé basé sur une condition d'optimalité locale similaire à celui présenté dans le cas du 1-PDPTW. Son but est de minimiser la distance totale parcourue. Le principe de cet opérateur est de calculer la somme des distances parcourues pour trois nœuds successifs en parcourant l'individu de la position $n_vehicule$ vers le dépôt, puis de permuter les gènes centraux des deux groupes ayant les plus grandes sommes de distances parcourues.
- **Mutation 2** : Cet opérateur est basé sur le déplacement d'un gène quelconque et son insertion dans une position choisie au hasard. Le gène choisi et sa nouvelle position doivent être entre la position $n_vehicule$ et le dernier sommet à visiter.

4.6.7 Hybridation de l'approche évolutionniste : étage d'amélioration par la recherche Tabou spécialisée

Dans le but d'améliorer les solutions fournies par l'approche évolutionniste, nous appliquons une recherche Tabou spécialement conçue pour notre problème 1-DPDPTW. Comme dans la recherche Tabou spécialisée pour le 1-PDPTW, dans ce cas du 1-DPDPTW, la solution initiale du calcul va être la première solution non dominée résultant de l'approche évolutionniste détaillée précédemment. L'algorithme de cet étage d'hybridation est détaillé dans la Figure 4.19. Cette amélioration est basée sur l'utilisation de l'opérateur de mutation 2 de l'approche évolutionniste conçue pour le 1-DPDPTW.

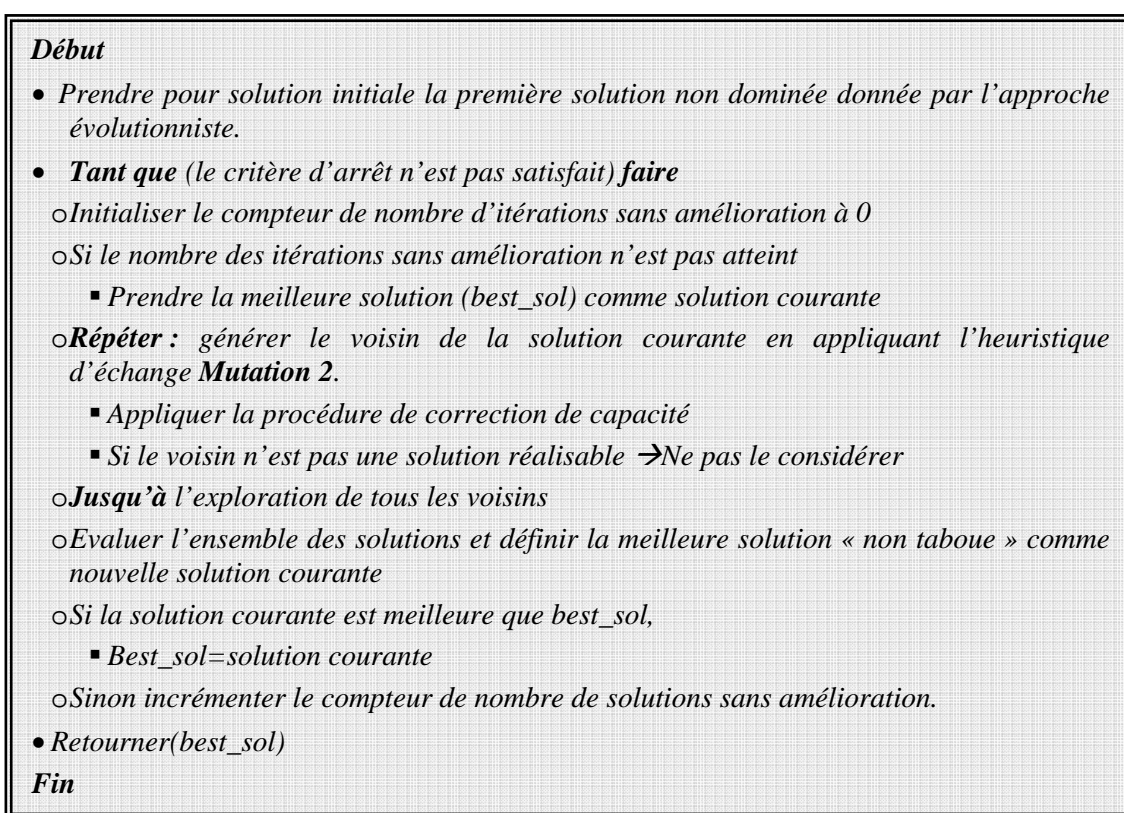


Figure 4.19 : L'algorithme de la recherche Tabou spécialisée pour le 1-DPDPTW

4.7 Conclusion

Dans ce chapitre nous avons présenté les problèmes de tournées de véhicules dynamiques et leurs caractéristiques. Nous avons aussi exposé la notion de degré de dynamisme des problèmes de tournée de véhicules dynamiques ainsi que leurs classifications et la formulation mathématique générale d'un problème de tournée dynamique. Ensuite, nous avons détaillé les méthodes de résolution du DVRP et ses variantes qui se composent essentiellement de méthodes séquentielles et de méthodes parallèles. Pour cela nous avons

exposé plusieurs méthodes élaborées par différents chercheurs. Enfin, nous avons détaillé les approches de résolution du DPDPTW élaborées. Ces approches sont basées essentiellement sur l'utilisation d'outils spéciaux dédiés au 1-DPDPTW comme les heuristiques de génération de la population initiale, les procédures de correction, les opérateurs de croisement et de mutation et l'heuristique d'échange pour l'étape d'amélioration. Tous ces outils ont été élaborés et utilisés dans le but de respecter des contraintes, améliorer les résultats et diminuer les temps de calcul comme cela sera présenté dans le chapitre suivant.

Chapitre 4 : Adaptation des outils de résolution du 1-PDPTW pour le 1-DPDPTW intégrant les contraintes dynamiques.

Chapitre 5

Système informatisé de résolution du 1-PDPTW et du 1-DPDPTW, Simulation et Résultats

5.1 Introduction

Dans ce chapitre, nous présentons les différentes étapes de notre travail de conception et de simulation à travers des exemples et des résultats. Nous détaillons en premier lieu la conception générale du système informatisé de résolution du 1-PDPTW. Ensuite, nous présentons les benchmarks utilisés et nous justifions les paramètres de simulation adoptés. Enfin, nous exposons les simulations réalisées et les résultats obtenus lors de la résolution du 1-PDPTW.

5.2 Architecture du système informatisé de résolution du 1-PDPTW

Les simulations et les expérimentations ont été réalisées sur un **PC PENTIUM 4** de **3GHZ** de fréquence. L'outil de programmation utilisé est le **MSVisual C++ 6.0** et l'outil de résolution des systèmes linéaires sous contraintes est le **CPLEX**.

Le système informatisé de résolution se compose de quinze modules interdépendants :

- Le générateur de benchmarks : sert à générer d'une façon aléatoire guidée les données sur lesquelles vont être testées les approches proposées.
- Les fichiers benchmarks : vont contenir les données des problèmes et vont être utilisés par le reste des modules.
- Le gestionnaire de benchmarks : permet de choisir le benchmark sur lequel va être testée une approche.
- Le programme calculateur de la borne inférieure de la somme des retards : calcule la borne inférieure de la somme des retards à partir des données des benchmarks.
- Le générateur de fichiers d'extension **.lp** : ces fichiers générés à partir des benchmarks sont exécutables sur le logiciel CPLEX pour l'élaboration des bornes inférieures.
- Le logiciel CPLEX : permet le calcul de la borne inférieure de la distance totale parcourue par le véhicule pour un problème donné. Ceci est fait en exécutant le fichier d'extension **.lp** du problème considéré.
- Le gestionnaire des paramètres de simulation : il permet de saisir et de modifier les paramètres des algorithmes évolutionnistes qui sont le taux de croisement, le taux de mutation, le nombre de générations et la vitesse du véhicule.

- Le programme évolutionniste AEs_1 pour le 1-PDPTW statique : il permet de résoudre le 1-PDPTW avec la 1^{ère} approche évolutionniste présentée dans le paragraphe 3.2.3 du troisième chapitre.
- Le programme évolutionniste AEs_2 pour le 1-PDPTW statique : il permet de résoudre le 1-PDPTW avec la 2^{ème} approche évolutionniste présentée dans le paragraphe 3.2.4 du troisième chapitre.
- Le programme évolutionniste hybride AEHs_1 pour le 1-PDPTW statique : il permet de résoudre le 1-PDPTW avec la 1^{ère} approche évolutionniste présentée dans le paragraphe 3.2.3 du troisième chapitre hybridée avec la recherche Tabou présentée dans le paragraphe 3.3 du même chapitre.
- Le programme évolutionniste hybride AEHs_2 pour le 1-PDPTW statique : il permet de résoudre le 1-PDPTW avec la 2^{ème} approche évolutionniste présentée dans le paragraphe 3.2.4 du troisième chapitre hybridée avec la recherche Tabou présentée dans le paragraphe 3.3 du même chapitre.
- Le programme évolutionniste AEd_1 pour le 1-PDPTW dynamique : il permet de résoudre le 1-PDPTW dynamique avec l'approche évolutionniste présentée dans le paragraphe 4.6 du troisième chapitre.
- Le programme évolutionniste hybride AEHd_1 pour le 1-PDPTW dynamique : il permet de résoudre le 1-PDPTW avec la 2^{ème} approche évolutionniste présentée dans le paragraphe 4.6 du troisième chapitre hybridée avec la recherche Tabou présentée dans le paragraphe 4.6.7 du même chapitre.
- Le gestionnaire des demandes dynamiques : il permet de saisir les demandes urgentes et toutes leurs caractéristiques (positions, fenêtre de temps, quantité, successeur, prédécesseur, position du véhicule lors de l'apparition de la demande...). La figure 5.1 suivante illustre le gestionnaire des demandes dynamiques.

The screenshot shows a window titled "Tache urgente" with a blue title bar. It is divided into two main columns: "Fournisseur" (Supplier) and "Client" (Client). Each column has input fields for "Coordonnées" (Coordinates) with X and Y values, "Quantité" (Quantity) with q, "Fenêtre de temps" (Time window) with e and f, and "Temps de service" (Service time) with s. The "Fournisseur" s field contains the value 10. Below these fields is a "Conditions:" box containing the following constraints: X, Y : 0 -> 100; q : 0 -> 90; f : 500 -> 1000; e : 0 -> 426; s = (0 , 90). To the right of the "Conditions" box is a "Prochain arrêt du véhicule" (Next vehicle stop) field with the value 0. At the top right are "OK" and "Quitter" buttons. At the bottom right is a "Nouvelle Commande" (New Order) button.

Figure 5.1 : Le gestionnaire des demandes dynamiques

- Les fichiers solutions : vont contenir les solutions proposées.

5.3 Les benchmarks

Vu le nombre restreint de travaux concernant le PDPTW dans la littérature, le nombre d’instances de problèmes et de jeux de tests pour évaluer les performances des approches de résolution est réduit. Parmi, ces instances nous citons les benchmarks de Li et Lim [Li et al 2001]. De plus, vu que dans ces travaux les chercheurs ont traité des objectifs différents, a fait que ces benchmarks peuvent être dans certains cas inadéquats voir sans intérêt à appliquer.

Dans notre cas, nous traitons le problème du 1-PDPTW statique et dynamique, et après plusieurs tests basés sur les benchmarks de Li et Lim, nous avons vu qu’il était nécessaire de générer de nouveaux benchmarks dédiés au 1-PDPTW.

Dans ce qui suit, nous allons présenter les benchmarks de Li et Lim et nos benchmarks élaborés en justifiant les raisons de leurs utilisations.

5.3.1 Les benchmarks utilisés (Li et Lim)

Pour tester notre approche nous avons utilisé, au départ, un ensemble d’instances de problèmes générés par Li et Lim [Li et al 2001] à partir de ceux élaborés par Solomon

[Solomon 1987]. Gardant la même classification des problèmes de Solomon : C1, C2, R1, R2, RC1 et RC2, Li et Lim ont aussi généré six classes de problèmes : LC1, LC2, LR1, LR2, LRC1 et LRC2. Tous ces problèmes ont environ 100 fournisseurs et clients avec plusieurs nœuds factices pour les besoins de couplage si cela est nécessaire. Ils renferment aussi les fenêtres de temps, la capacité de transport, les quantités de biens demandés pour chaque sommet et les contraintes de couplage (fournisseur, client). Dans les problèmes de type LC, les fournisseurs et les clients sont regroupés. Alors que dans les problèmes de type LR, ils sont générés aléatoirement. Naturellement, dans les problèmes de types LRC, fournisseurs et clients sont partiellement regroupés et partiellement affectés au hasard, et tandis que LC1, LR1 et LRC1 ont un court horizon d'ordonnancement, LC2, LR2 et LRC2 ont un horizon plus important.

Dans ces instances, chaque fichier de problème contient trois groupes de données :

- Le groupe concernant les clients et les fournisseurs :
 - Id : identificateur du client ou du fournisseur,
 - (x,y) : position du client ou du fournisseur,
 - d : quantité de service demandée par le client ou par le fournisseur,
 - (e,l) : fenêtre de temps du client ou du fournisseur dans laquelle il souhaite être servi,
 - s : durée de service chez un client ou un fournisseur,
 - $pred(i)$: prédécesseur du client ou du fournisseur i ,
 - $succ(i)$: successeur du client ou du fournisseur i .
- Le groupe concernant le dépôt :
 - (x,y) : la position du dépôt,
 - (e,l) : horaires d'ouverture et de fermeture du dépôt.
- Le groupe concernant les véhicules :
 - C : la capacité du véhicule,
 - V : la vitesse du véhicule.

Les figures 5.2 et 5.3 illustrent quelques uns de ces problèmes.

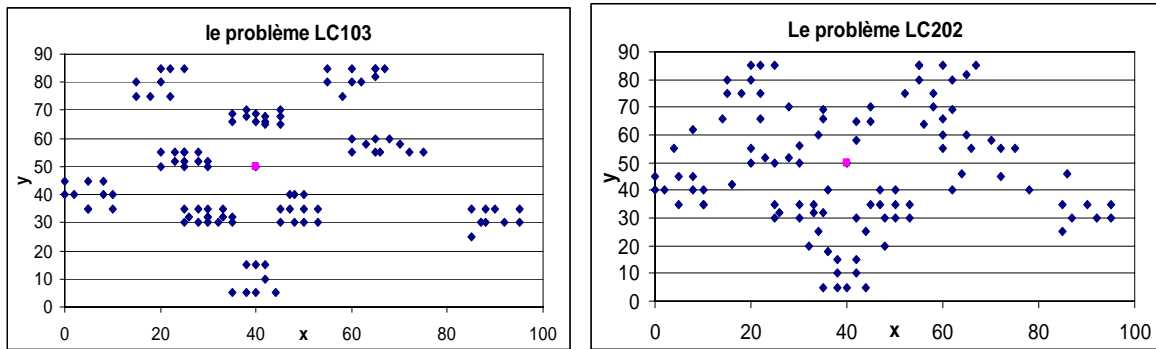


Figure 5.2 : Jeux de tests de Li et Lim pour le PDPTW

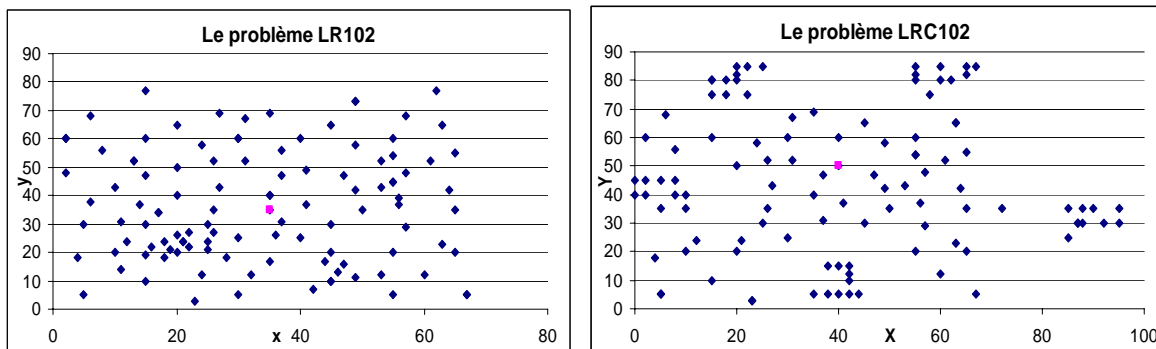


Figure 5.3 : Jeux de tests de Li et Lim pour le PDPTW

Ces instances ont été initialement conçues pour le m-PDPTW que Li et Lim traitaient. Ils avaient pour objectif de minimiser la distance totale parcourue en utilisant un nombre minimum de véhicule. Ce qui était prévisible, l'application de ces données sur un 1-PDPTW a donné des résultats inacceptables vis-à-vis des contraintes de temps, mais elle nous a permis de juger l'efficacité de nos approches quant à la minimisation de la distance totale parcourue, et le respect des contraintes de précédence et de capacité (voir tableau 5.1 du paragraphe 5.5). Pour cette raison, nous avons jugé nécessaire de générer nos propres instances pour mieux évaluer nos approches vis-à-vis des données temporelles.

5.3.2 Les benchmarks élaborées [Kammarti et al 2006]

Pour la génération de benchmarks dédiés au 1-PDPTW, nous nous sommes inspiré de ceux de Solomon et de Li et Lim. Nous avons généré quarante instances de problèmes classés selon cinq types : P, PD, PT et PDT. Ces problèmes peuvent avoir 10, 16, 20, 26 ou 30 fournisseurs et clients et donnent les fenêtres de temps, la capacité de transport, les quantités de biens demandés pour chaque sommet et les contraintes de couplage (fournisseur, client). Le fait d'avoir moins de sommets à visiter pourrait permettre de respecter toutes les contraintes temporelles.

Les caractéristiques de ces problèmes sont les suivantes :

- Type P : les sommets sont générés aléatoirement et les largeurs des fenêtres de temps sont importantes et dispersées.
- Type PD : les sommets sont regroupés et les largeurs des fenêtres de temps sont importantes.
- Type PT : les sommets sont générés aléatoirement et les largeurs des fenêtres de temps sont moins importantes.
- Type PDT : les sommets sont regroupés et les largeurs des fenêtres de temps sont moins importantes.

Ces problèmes sont notés de la façon suivante : type|nombre de sommets_version.
(Exemple : PDT30_1 : 1^{ère} version du problème de type PDT avec 30 sommets).

Dans ces problèmes, les coordonnées géographiques sont exprimées en kilomètres et les données temporelles en minutes.

Les figures 5.4 et 5.5 illustrent la dispersion géographique des sommets pour les problèmes P30_1, PD30_1, PT30_1 et PDT30_1.

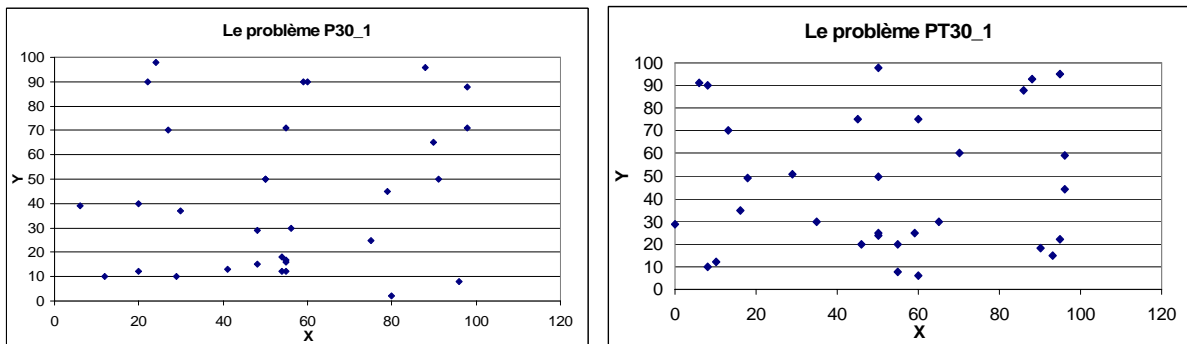


Figure 5.4 : Jeux de tests élaborés pour le 1-PDPTW

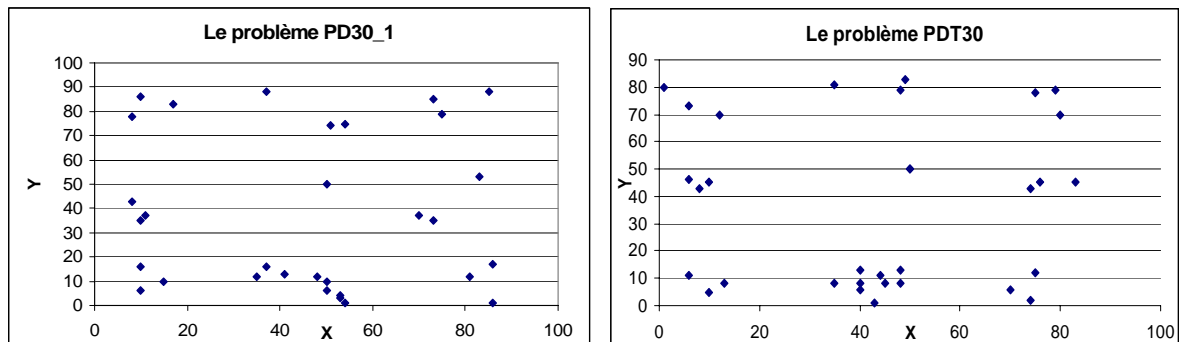


Figure 5.5 : Jeux de tests élaborés pour le 1-PDPTW

Nous présentons dans la figure 5.6 l'instance du problème PDT10_1 où nous pouvons montrer que nos instances comportent toutes les données nécessaires à la confection d'un problème 1-PDPTW.

Véhicule	1	150	0						
Dépôt	0	50	50	0	0	600	0	0	0
Sommets	1	75	12	10	0	300	10	0	2
	2	73	46	-10	300	600	10	1	0
	3	83	78	15	0	300	10	0	4
	4	43	71	-15	300	600	10	3	0
	5	5	80	15	60	360	10	0	6
	6	5	39	-15	120	420	10	5	0
	7	6	12	10	0	600	10	0	8
	8	44	0	-10	180	480	10	7	0
	9	41	4	15	180	480	10	0	10
	10	37	6	-15	180	480	10	9	0
Id	X	Y	D	e	l	s	pred	succ	

Figure 5.6 : L'instance PDT10_1 élaborée.

Dans l'instance ci-dessus :

- Nous disposons d'un seul véhicule de capacité 150,
- Le dépôt est au point de coordonnées (50,50) et est disponible dans la fenêtre de temps [0,600],
- Les positions, les demandes, les temps de services, les fenêtres de temps des sommets à visiter et les contraintes de précédence forment le reste du tableau.

Il est à noter que :

- si la valeur du prédécesseur ou du successeur est égale à 0, cela signifie que le sommet n'a respectivement pas de prédécesseur ou de successeur.
- Les valeurs de quantité de service négatives indiquent que le sommet est un client et que le véhicule va décharger des biens.

Vu les tailles importantes de ces instances elles seront exposées dans l'annexe 1.

5.4 Les paramètres de simulation adoptés

Afin d'augmenter les chances d'obtention de solutions de bonne qualité en un temps minimum, nous avons fixé la taille des populations entre 50 et 250 individus selon la taille et la nature du problème traité. De plus, pour les problèmes proposés par Li et Lim [Li et al 2001], des tests préliminaires nous ont aidé à remarquer que la convergence est obtenue la

plupart du temps aux alentours de la 1500^{ème} itération pour ceux résolus par une des approches évolutionnistes. Quant aux problèmes élaborés par notre générateur de benchmark, la convergence des approches de résolution est aux alentours de la 60^{ème} itération. Ceci est montré dans les figures 5.7 et 5.8.

Sachant que notre étage d'hybridation est une recherche Tabou spéciale qui a pour solution initiale la première solution non dominée offerte par une approche évolutionniste, elle risque alors de se trouver prisonnière d'un minimum local dès le début de la recherche. Pour y remédier, nous prenons comme solution initiale la première solution donnée par une des approches évolutionnistes à une itération qui précède son nombre maximum de générations. Ceci offrirait probablement plus de chance d'être améliorée et de conduire vers un meilleur résultat.

Après de nombreux tests, il s'est avéré qu'augmenter le nombre des itérations, pour la résolution les problèmes élaborés par notre générateur de benchmark, peut augmenter la chance d'obtenir un meilleur résultat fruit d'une mutation contrôlée. Pour cela nous avons fixé, le nombre maximal de génération à effectuer égale à :

- 2000 pour les problèmes proposés par Li et Lim traités par l'approche AEs_1;
- 3000 pour les problèmes proposés par Li et Lim traités par l'approche AEs_2 ;
- 1500 pour les problèmes proposés par Li et Lim traités par une des approches évolutionnistes hybrides;
- 1500 pour les problèmes élaborés par notre générateur d'instances et résolus par une des approches évolutionnistes ;
- 1300 pour les problèmes élaborés par notre générateur d'instances et résolus par une des approches évolutionnistes hybrides.

D'autres tests ont aussi permis de fixer le nombre d'itérations de la recherche Tabou à :

- 1000 pour les problèmes proposés par Li et Lim ;
- 300 pour les problèmes élaborés par notre générateur d'instances
- 10 pour le nombre des itérations sans amélioration (n_{max}).
- $\alpha_1 = \alpha_2 = \alpha_3$

Le nombre d'itérations à effectuer peut varier d'un problème à un autre sans pour autant dépasser ces limites ci-dessus fixées. L'annexe 2 présente en détails les nombres d'itérations de chaque problème.

Dans la littérature, il est souvent conseillé de prendre une grande probabilité de croisement P_c de l'ordre de 0.75, et une petite probabilité de mutation P_m de l'ordre de 0.05. Dans notre cas, et après avoir fait plusieurs simulations préliminaires, nous avons fixé $P_c = 0.75$ et $P_m = 0.2$. La grande valeur de la probabilité de mutation P_m a pour cause l'utilisation d'opérateurs de mutation spéciaux qui augmentent les chances d'obtention de meilleurs individus. Lors d'une opération de mutation les probabilités de choix entre les opérateurs concernés sont identiques.

La figure 5.7 illustre l'évolution du critère global du problème LRC105 de Li et Lim résolu par le programme AEs_1.

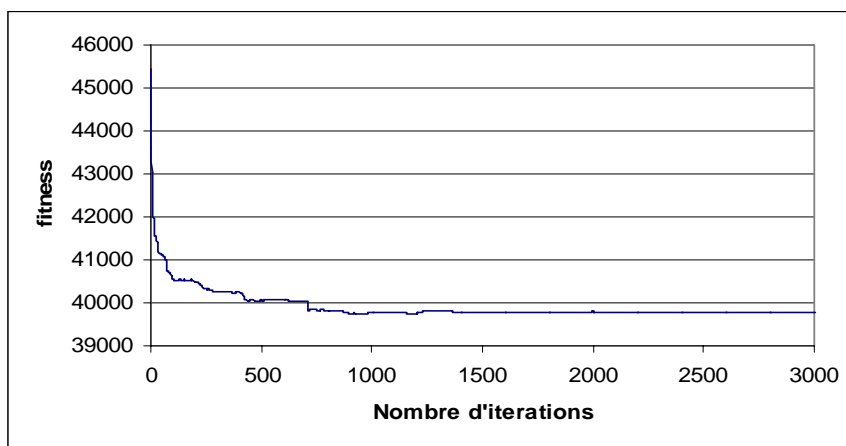


Figure 5.7 : Convergence du critère global pour le problème LRC105 résolu par AEs_1.

La figure 5.8 illustre l'évolution du critère global du problème PDT30 résolu par le programme AEs_1.

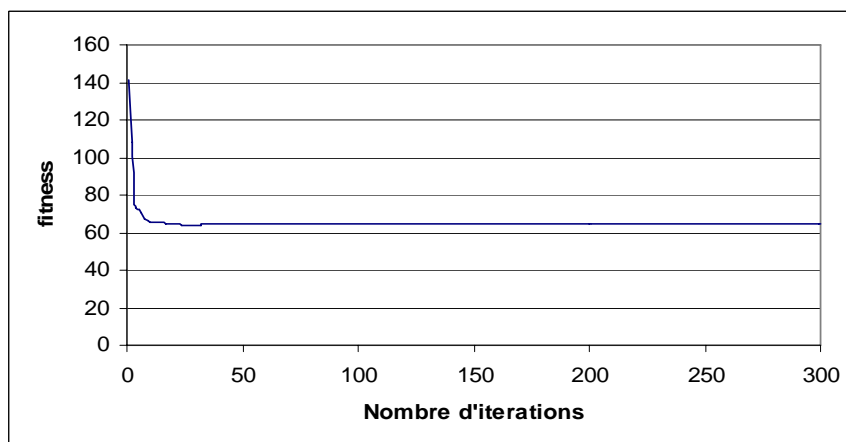


Figure 5.8 : Convergence du critère global pour le problème PDT30 résolu par AEs_1.

La Figure 5.9 montre l'évolution de la solution durant l'application de la recherche Tabou spéciale pour le problème LRC102 après 50 itérations de l'approche évolutionniste AEs_1.

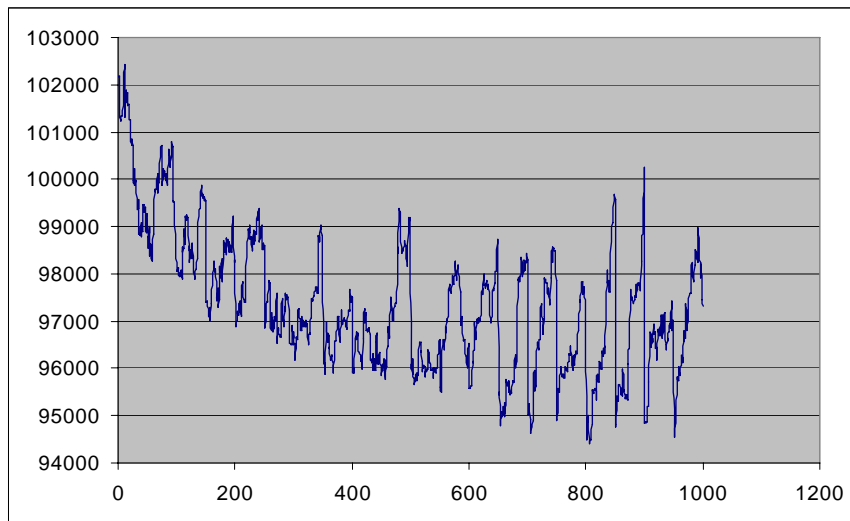


Figure 5.9 : Evolution de la valeur de la solution avec la recherche Tabou spéciale pour le problème LRC102.

5.5 Le 1-PDPTW statique : Simulations et Résultats

Dans cette partie nous exposons les différents résultats obtenus pour la résolution des problèmes de Li et Lim et ceux élaborés par notre générateur de benchmark.

Nous adoptons dans ce qui suit la notation suivante :

- Nd : Nombre des solutions Pareto optimales,
- T_D : Distance totale parcourue,
- T_W : Temps d'attente total,
- T_T : Somme des retards,
- T_D_Li : Distance totale parcourue dans [Li et al 2001],
- $\tilde{f}_1(x)$: Valeur homogénéisée de la distance totale,
- $\tilde{f}_2(x)$: Valeur homogénéisée de la somme des retards.
- $\tilde{f}_G(x)$: Valeur homogénéisée totale.

5.5.1 Les benchmarks de Li et Lim : simulations et résultats

Nous présentons ici quelques résultats des simulations réalisées pour les problèmes de Li et Lim. Nous commencerons par présenter des résultats obtenus par l'approche AEHs_1, ensuite nous exposerons les résultats donnés par l'approche AEs_2.

Le tableau 5.1 illustre quelques uns des résultats donnés pour le 1-PDPTW par l'approche AEHs_1.

Prob	Nd	T_D	T_W	T_T	T_D_Li
LC101	4	708,91	64,78	456043	828,94
LC102	8	745,015	0	443049,1	828.94
LC201	10	734,28	1493,61	337113,563	591.56
LC202	15	800,11	59,68	296089,594	591.56
LR101	70	937,39	347,07	88385,719	1650.78
LR102	19	859,00	179,99	81963,836	1487.57
LR201	7	859,05	2226,84	44442,500	1263.84
LR202	25	971,81	2298,62	42733,805	1197.67
LRC101	26	926,30	140,56	84927,188	1708.80
LRC102	31	991,62	280,54	84132,336	1563.55
LRC201	33	1087,79	2124,02	52520,789	1468.96
LRC202	22	1092,29	2737,73	51340,402	1374.27

Tableau 5.1 : Solutions pour le 1-PDPTW basées sur des problèmes de Li et Lim (AEHs_1)

La figure 5.10 et le tableau 5.2 montrent respectivement l'évolution de la valeur de $\tilde{f}_G(x)$ et les résultats obtenus en utilisant l'approche AEs_2.

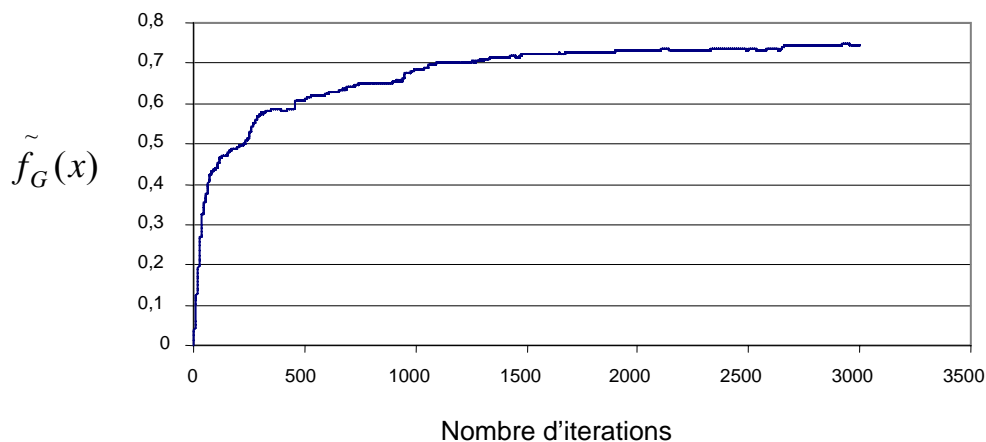


Figure 5.10 : Convergence du critère global pour le problème LRC205 résolu par AEs_2

prob	Borne inférieure distance	Borne inférieure retard	Nd	$\tilde{f}_G(x)$	$\tilde{f}_1(x)$	$\tilde{f}_2(x)$	T_d	T_t	T_w
LRC101	586,895	44565	8	0,4301755	0,543964	0,316387	968,331665	95575,125	0
LRC102	591,664	42198	3	0,4783075	0,57456	0,382055	961,638794	87865,13281	0
LRC103	591,809	40395	2	0,359202	0,445812	0,272592	957,395935	88244,11719	0
LRC104	590,38	39107	1	0,484756	0,572442	0,39707	905,363953	80352,75	0
LRC105	585,424	43032	15	0,403163	0,505825	0,300501	942,282593	93515,4375	0
LRC106	590,347	43043	11	0,4125895	0,502792	0,322387	982,433411	91988,52344	0
LRC107	588,664	41448	5	0,435573	0,559141	0,312005	888,981262	84920,94531	0
LRC108	590,785	40359	1	0,478105	0,558081	0,398129	958,745728	86379,27344	0
LRC201	590,411	7742	34	0,5694785	0,702211	0,436746	1146,70874	75028,34375	0
LRC202	592,593	1563	50	0,6677335	0,784979	0,550488	1100,588989	65068,96875	0
LRC203	591,415	1487	47	0,5917085	0,682603	0,500814	1071,309448	50764,78906	0
LRC204	591,667	1456	17	0,528145	0,617202	0,439088	1031,587646	44051,49219	0
LRC205	591,096	4675	26	0,711225	0,808443	0,614007	1050,627075	63513,5625	0
LRC206	594,312	4466	23	0,645961	0,731337	0,560585	1158,24707	66111,82813	0
LRC207	592,195	2640	14	0,6158625	0,663563	0,568162	1062,383179	48962,39063	0
LRC208	593,465	1716	17	0,673609	0,758359	0,588859	1071,527832	47267,10547	0

Tableau 5.2 : Solutions pour le 1-PDPTW basées sur des problèmes de Li et Lim (AEs_2)

Nous remarquons que la valeur du retard total est considérable. Ceci est du aux données fournies par les instances de Li et Lim. Il y a plusieurs cas dans lesquels nous sommes dans l'obligation d'accuser un retard. Nous citons par exemple le sommet 1 du problème LC101 qui a [912,967] pour fenêtre de temps et 90 pour temps de service ce qui engendrera nécessairement un retard. Nous pouvons aussi remarquer que le fait de servir tous les sommets par un seul véhicule engendrera sans doute un retard puisqu'il faut plusieurs véhicules pour essayer de satisfaire tous les sommets dans leurs fenêtres de temps.

Nous ne pouvons pas correctement comparer nos résultats à ceux trouvés dans [Li et al 2001] à cause des incompatibilités entre les objectifs à optimiser. Néanmoins, nous mentionnons que nous avons obtenu dans plusieurs cas la meilleure distance totale parcourue. Pour cette raison, nous avons jugé nécessaire de générer nos propres instances pour mieux évaluer nos approches vis-à-vis des données temporelles.

Les données du tableau 5.2 montrent que les résultats obtenus sont acceptables vis à vis du critère de minimisation de la distance totale parcourue. Prenons l'exemple du problème

LRC205 dans lequel $\tilde{f}_1(x) = 0,808443$ qui peut être considéré comme un bon résultat. En fait, en comparant les résultats des problèmes communs des tableaux 5.2 et 5.1, nous remarquons que les valeurs de la distance totale parcourue sont voisines. Ces données permettent aussi de montrer que la borne inférieure de retard peut être améliorée pour se rapprocher de la valeur optimale que l'on cherche à obtenir pour les problèmes de grandes tailles.

Nous rappelons que les bornes inférieures ont été élaborées séparément et indépendamment les une des autres et quelles ne tiennent pas compte des contraintes de précédence, de capacité et de fenêtres de temps. Il est alors très fort probable, vu le fort antagonisme des critères à optimiser, que les valeurs d'une borne inférieure de distance et une autre de retard ne soient pas atteintes dans une même solution.

5.5.2 Les benchmarks P, PD, PT, PDT : simulations et résultats

Dans cette partie, nous exposons les résultats obtenus pour les benchmarks élaborés par le générateur d'instances. Ces résultats ont été obtenus par l'application des différentes approches AEs_1, AEs_2, AEHs_1 et AEHs_2. Nous signalons, pour une meilleure lecture des résultats, que la distance T_d est exprimée en kilomètres et les temps d'attentes T_w et de retard T_t en heures.

5.5.2.1 Résultats

Le tableau 5.3 illustre les résultats donnés pour les problèmes P résolus par l'approche AEs_1.

Approche	AEs_1						
	Problème	T_d	T_w	T_t	Fitness_glob	Nd	Solution
	P10_1	34,80691	13,93985	0	48,74676	16	0-1-2-3-4-5-6-7-8-9-10-0
	P10_2	38,0329	17,2692	0	55,3021	21	0-7-8-9-10-1-2-3-5-6-4-0
	P16_1	43,263	17,4345	0	60,6975	25	0-3-5-6-4-1-2-15-16-13-14-11-12-9-10-7-8-0
	P16_2	46,6906	18,6468	0	65,3374	24	0-13-14-11-12-9-10-7-8-5-6-15-16-1-2-3-4-0
	P20_1	59,9669	9,66024	0	69,62714	45	0-13-14-1-2-5-6-7-9-11-10-12-19-15-17-20-18-16-3-4-8-0
	P20_2	46,82917	19,44445	0	66,27362	51	0-13-14-11-12-9-10-7-8-5-6-3-4-1-2-15-19-20-17-16-18-0
	P26_1	57,7953	18,883	1,0128	77,69109	77	0-11-7-8-9-12-10-15-13-14-16-17-18-23-25-21-19-20-22-26-24-1-2-3-5-6-4-0
	P26_2	67,3976	4,79674	0	72,19434	53	0-3-1-2-9-7-8-13-14-15-17-16-18-19-21-20-25-26-22-23-24-4-5-6-10-11-12-0
	P30_1	71,3236	1,75002	0	73,07362	22	0-7-8-9-3-1-2-13-14-15-17-18-16-29-27-25-30-23-24-26-28-19-20-21-22-5-6-4-10-11-12-0
	P30_2	68,4094	2,048	0	70,4574	250	0-1-3-2-7-8-9-13-15-14-17-16-18-19-21-20-27-28-29-30-25-26-23-22-24-5-4-6-11-10-12-0

Tableau 5.3 : Solutions pour la famille des benchmarks P résolus par AEs_1.

Nous remarquons à partir des résultats présentés dans le tableau 5.3 que les benchmarks élaborés sont mieux adaptés au problème du 1-PDPTW : les distances parcourues, les sommes des temps d’attentes évités et les sommes des retards sont acceptables. Nous remarquons aussi que nous arrivons à atteindre plusieurs fois la borne inférieure de retard.

Le tableau 5.4 illustre quelques uns des résultats donnés pour les problèmes P résolus par les approches AEs_1, AEs_2, AEHs_1 et AEHs_2.

Approche Problème	AEs_1			AEHs_1			AEs_2			AEHs_2			b_inf _dist	b_inf _ret
	T_d	T_w	T_t	T_d	T_w	T_t	T_d	T_w	T_t	T_d	T_w	T_t		
P10_1	34,807	13,94	0	34,807	13,94	0	34,807	13,94	0	34,807	13,94	0	32,55	0
P10_2	38,033	17,269	0	38,033	17,269	0	38,033	17,269	0	39,121	17,045	0	32,29	0
P16_1	43,263	17,435	0	42,274	17,483	0	41,269	20,894	0	41,269	20,894	0	32,8	0
P16_2	46,691	18,647	0	45,179	18,697	0	43,529	20,618	0	46,251	19,134	0	33,9	0
P20_1	59,967	9,6602	0	58,408	9,3882	0	48,599	20,998	0	44,665	24,052	0,01	38,96	0
P20_2	46,829	19,444	0	46,829	19,444	0	49,17	18,791	0	52,547	17,724	0	39,26	0
P26_1	57,795	18,883	1,013	54,835	17,39	0,844	59,219	16,828	0	65,783	7,32	0	42,03	0
P26_2	67,398	4,7967	0	67,397	4,7967	0	64,571	22,986	0	54,337	28,85	0	42,69	0
P30_1	71,324	1,75	0	64,912	1,8726	0	66,431	3,5171	0	64,002	9,11	0	44,6	0
P30_2	68,409	2,048	0	73,902	0,3835	0	65,153	19,282	0	65,12	19,391	0	43,05	0

Tableau 5.4 : Solutions pour la famille des benchmarks P.

Le tableau 5.4 montre les valeurs de la distance totale parcourue, le temps d’attente évité et le retard total de chaque première solution dominante au sens de Pareto de la dernière population pour une approche évolutionniste et de la solution donnée par l’étage d’hybridation pour les approches évolutionnistes hybrides.

A travers ces résultats nous pouvons remarquer que les méthodes AEs_2 et AEHs_2, basées sur l’utilisation de bornes inférieures de distance et de retard, ont tendance à orienter la recherche vers la minimisation de la somme de la distance totale parcourue et du retard total au dépend du temps d’attente qui a pour la majorité des cas des valeurs plus importantes que celles élaborées par les approches AEs_1 et AEHs_1. L’avantage des approches AEs_2 et AEHs_2 réside donc dans une meilleure optimisation de la somme la distance totale parcourue et du retard total et pourront être utilisés pour les problèmes dont les limites inférieures e_i des fenêtres de temps des clients sont flexibles et les pénalités de service en avance peu conséquentes.

Le reste des résultats concernant les autres problèmes est présenté dans l’annexe 3.

5.5.2.2 L'apport de l'étage d'hybridation

Le tableau 5.5 illustre les résultats obtenus pour les familles des problèmes P, PD, PT et PDT résolus par les approches AEs_1, AEs_2, AEHs_1 et AEHs_2. Dans ce tableau, les cases grisées indiquent les solutions ayant une meilleure fonction fitness F_1 (la moins élevée).

Approche Problème	AEs_1		AEHs_1		AEs_2		AEHs_2	
	Fitness _glob	Nd	Fitness _glob	Nd	Fitness _glob	Nd	Fitness _glob	Nd
P10_1	48,747	16	48,747	15	48,747	10	48,747	8
P10_2	55,302	21	55,302	25	55,302	15	56,166	21
P16_1	60,698	25	59,757	28	62,162	24	62,162	28
P16_2	65,337	24	63,877	28	64,147	33	62,889	32
P20_1	69,627	45	67,796	35	69,597	48	68,729	34
P20_2	66,274	51	66,274	42	67,961	53	70,271	48
P26_1	77,691	77	73,068	43	76,047	65	73,103	65
P26_2	72,194	53	72,194	46	87,556	41	83,187	103
P30_1	73,074	22	66,784	6	69,948	36	73,113	18
P30_2	70,457	250	74,286	21	84,434		84,512	120
PD10_1	44,362	21	44,362	21	44,362	19	44,362	19
PD10_2	47,258	20	47,258	21	47,474	16	47,258	21
PD16_1	52,074	25	53,023	31	55,107	19	52,074	25
PD16_2	51,598	35	51,598	33	54,517	27	51,598	24
PD20_1	52,412	119	48,992	104	57,894	250	50,497	117
PD20_2	54,746	50	52,094	51	52,192	49	52,072	53
PD26_1	60,086	40	60,360	66	65,601	38	58,855	36
PD26_2	63,823	64	57,723	96	60,325	250	65,339	242
PD30_1	59,127	11	59,127	8	71,518	30	71,518	30
PD30_2	53,573	53	61,302	36	67,936	107	67,554	32
PT10_1	47,211	25	47,211	31	47,211	28	47,211	28
PT10_2	48,003	22	48,003	24	48,003	18	48,003	15
PT16_1	61,540	40	59,397	43	64,885	27	58,297	26
PT16_2	57,344	75	53,595	91	53,595	21	54,986	21
PT20_1	65,548	41	65,267	36	71,212	35	67,318	30
PT20_2	66,381	91	66,381	93	69,194	31	69,146	31
PT26_1	67,272	80	62,798	250	62,798	250	63,883	250
PT26_2	64,135	250	64,301	78	69,913	250	69,913	250
PT30_1	72,453	20	71,640	19	83,174	172	78,054	250
PT30_2	63,215	4	63,141	3	74,987	30	73,471	7
PDT10_1	45,684	20	45,684	20	45,684	22	45,684	17
PDT10_2	46,339	24	46,339	28	46,339	23	48,068	22
PDT16_1	55,657	44	53,738	33	60,037	25	54,469	20
PDT16_2	55,490	31	55,490	38	58,899	29	54,654	21
PDT20_1	54,283	113	54,283	82	56,653	111	56,653	109
PDT20_2	57,312	83	57,312	119	65,603	131	57,312	95
PDT26_1	56,917	250	56,510	91	59,201	250	58,769	250
PDT26_2	55,660	250	53,902	250	65,214	250	65,219	51
PDT30_1	56,778	7	53,103	5	65,065	96	63,502	56
PDT30_2	56,119	3	56,094	3	63,326	38	63,056	40

Tableau 5.5 : Solutions des problèmes P, PD, PT et PDT.

A partir de ce tableau, nous remarquons dans 80% des cas que l'approche AEHs_1 fournit la solution ayant la meilleure somme de la distance totale parcourue, le temps d'attente total et le retard total. Ceci est essentiellement dû à l'utilisation de la recherche Tabou pour l'amélioration de la solution donnée par l'approche évolutionniste qui a permis, à l'aide de la nouvelle heuristique d'échange, de brasser intelligemment le voisinage à la recherche de meilleures solutions. La figure 5.11 suivante montre l'efficacité des approches les une par rapport aux autres pour la minimisation de la somme de la distance totale parcourue, le temps d'attente total et le retard total F_1 .

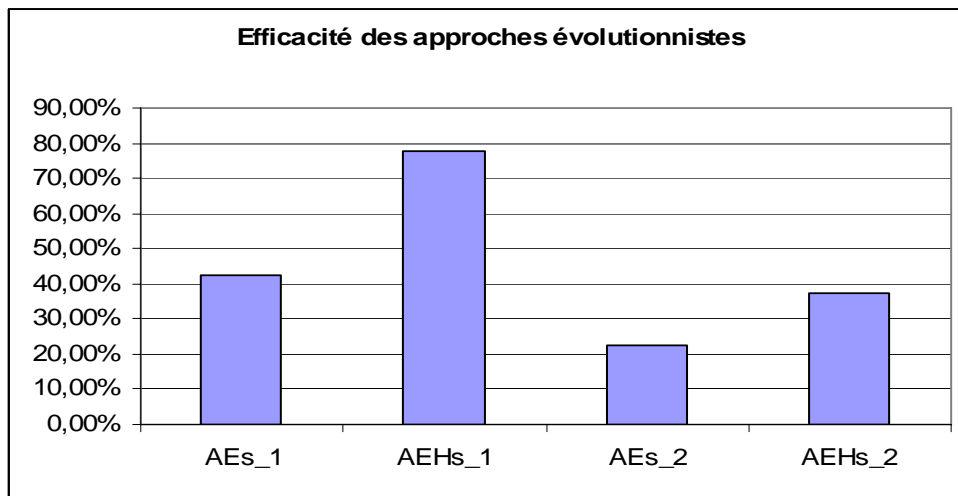


Figure 5.11 : Efficacité des approches pour la minimisation de F_1 .

La figure 5.12 suivante montre l'efficacité des approches AEs_2 et AEHs_2 l'une par rapport à l'autre pour la minimisation de la somme de la distance totale parcourue et du retard total F_2 .

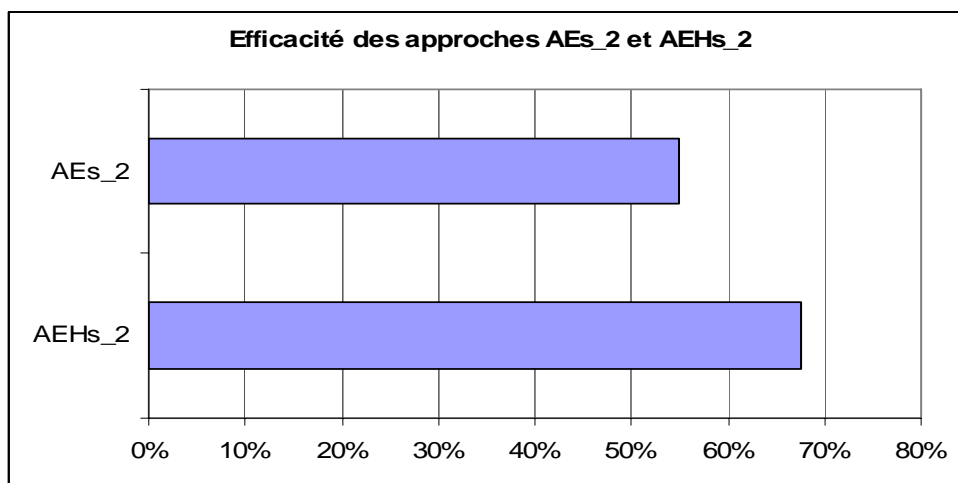


Figure 5.12 : Efficacité des approches AEs_2 et AEHs_2 pour la minimisation de F_2 .

5.6 Le 1-PDPTW dynamique : Simulations et Résultats

Dans cette partie, nous présentons les benchmarks utilisés pour traiter le 1-DPDPTW. Nous y exposons aussi les résultats obtenus par l'application des approches AEd_1 et AEHd_1.

5.6.1 Les benchmarks utilisés

Pour tester le 1-DPDPTW, nous nous sommes basés sur les benchmarks précédemment élaborés. Pour cela, pour un benchmark donné, nous considérons une partie des sommets comme sommets statiques et le reste des sommets dynamiques. La partie constituant les sommets statiques doit comprendre le dépôt et est constituée des $n-m$ premiers sommets, où m est le nombre de sommets dynamiques. Ceci doit impérativement respecter la condition que chaque client a un seul fournisseur et qu'un fournisseur n'a qu'un seul client.

Ces problèmes sont notés de la façon suivante : type|nombre de sommets|d_version. (Exemple : PDT30d_1 : 1^{ère} version du problème dynamique de type PDT avec 30 sommets).

La figure 5.13 montre un exemple benchmark dynamique PDT10d_1.

Véhicule	1	150	0						
Dépôt	0	50	50	0	0	600	0	0	0
Sommets statiques	1	75	12	10	0	300	10	0	2
	2	73	46	-10	300	600	10	1	0
	3	83	78	15	0	300	10	0	4
	4	43	71	-15	300	600	10	3	0
	5	5	80	15	60	360	10	0	6
	6	5	39	-15	120	420	10	5	0
Sommets dynamiques	7	6	12	10	0	600	10	0	8
	8	44	0	-10	180	480	10	7	0
	9	41	4	15	180	480	10	0	10
	10	37	6	-15	180	480	10	9	0
Id	X	Y	D	e	l	s	pred	succ	

Figure 5.13 : L'instance PDT10d_1 pour le 1-DPDPTW.

Le nombre des sommets dépend uniquement de la taille du problème initial (cas statique). Le tableau 5.6 suivant illustre le nombre des tâches urgentes pour les problèmes considérés ainsi que le degrés de dynamismes de ces derniers. Les problèmes utilisés sont les familles Pd_1, PDd_1, PTd_1 et PDTd_1.

taille du problème	nombre de sommets urgents	<i>dod</i>
10	4	0,4
16	4	0,25
20	4	0,2
26	4	0,15384615
30	6	0,2

Tableau 5.6 : Nombre des sommets dynamiques par problème et degrés de dynamisme.

Remarquons que le nombre de sommets dynamiques doit être pair car une demande dynamique comporte à la fois un fournisseur et un client.

5.6.2 L'acquisition des demandes dynamiques

Contrairement à plusieurs études où les demandes urgentes sont introduites dès le départ dans le benchmark mais ne sont validées que si leur temps d'apparition est atteint, notre outil de résolution nous permet de simuler l'arrivée des demandes urgentes et de les saisir après que le système informatisé ait élaboré une solution que le véhicule est en train de suivre. Ceci nous permet de ne mettre, dans les benchmarks du 1-DPDPTW, que les demandes statiques et nous laisse la liberté d'introduire les demandes dynamiques au choix. (Voir Tableau 5.7)

Dans notre cas, pour tester l'efficacité de nos approches de résolution, nos demandes dynamiques seront le reste des données des benchmarks du cas statique.

5.6.3 Résultats

Nous signalons, pour une meilleure lecture des résultats, que la distance T_d est exprimée en kilomètres et les temps d'attente T_w , de retard T_t et d'apparition de demande urgente tr en heures.

Le tableau 5.7 illustre les temps d'apparition des demandes dynamiques pour les différents problèmes traités.

On note tr_i : le moment d'apparition de la demande dynamique i exprimé en heures.

Approche	tr_1	tr_2	tr_3
Problème			
P10d_1	0,1	0,5	
P16d_1	0,2	1,5	
P20d_1	1,5	3	
P26d_1	2	4	
P30d_1	1	4,5	5,5
PD10d_1	1,2	1,5	
PD16d_1	2,2	2,8	
PD20d_1	3	3,5	
PD26d_1	4,1	4,5	
PD30d_1	3,6	5	5,5
PT10d_1	0,7	1,3	
PT16d_1	1,8	3	
PT20d_1	2,5	3,5	
PT26d_1	4	4,7	
PT30d_1	3,8	4,5	5,8
PDT10d_1	0,5	1,6	
PDT16d_1	1,6	2,5	
PDT20d_1	2,3	3,3	
PDT26d_1	3,5	4,7	
PDT30d_1	4,1	4,9	5,4

Tableau 5.7 : Caractéristiques temporelles des demandes dynamiques.

Le tableau 5.8 illustre les résultats obtenus pour les familles des problèmes Pd, PDD, PTd et PDTd résolus par les approches AEd_1 et AEHd_1. Ce tableau montre les valeurs de la somme la distance totale parcourue, du temps d'attente et du retard total de chaque première solution dominante au sens de Pareto de la dernière population pour une approche évolutionniste et de la solution donnée par l'étage d'hybridation pour les approches évolutionnistes hybrides.

Approche	AEd_1					AEHd_1				
	T_d	T_w	T_t	Fitness_glob	Nd	T_d	T_w	T_t	Fitness_glob	n_sol
P10d_1	34,807	13,940	0	48,747	25	34,807	13,940	0	48,747	17
P16d_1	42,192	17,945	0	60,137	10	42,192	17,945	0	60,137	12
P20d_1	54,292	19,147	0	73,439	3	57,782	23,609	0	81,391	2
P26d_1	60,998	16,972	1,272	79,243	37	60,613	16,946	0,3685	77,927	24
P30d_1	75,141	2,309	0	77,450	2	77,040	2,047	0	79,087	2
PD10d_1	35,688	9,206	0	44,894	2	35,688	9,206	0	44,894	2
PD16d_1	44,488	15,026	0	59,514	1	49,047	13,107	0	62,154	2
PD20d_1	55,208	12,924	0	68,132	6	55,126	12,924	0	68,050	7
PD26d_1	51,344	13,250	0	64,595	1	51,344	13,250	0	64,595	2
PD30d_1	48,936	23,151	0	72,087	1	50,027	21,916	0	71,944	1
PT10d_1	33,594	15,364	0	48,957	4	33,594	15,364	0	48,957	4
PT16d_1	54,405	14,566	0	68,971	1	54,405	14,566	0	68,971	1
PT20d_1	49,052	19,547	0	68,599	1	49,052	19,547	0	68,599	1
PT26d_1	68,020	5,614	0	73,634	1	70,835	3,004	0	73,839	1
PT30d_1	77,936	3,118	0	81,194	2	75,883	7,001	0,0816	82,966	2
PDT10d_1	36,070	11,571	0	47,641	1	35,882	11,578	0	47,461	3
PDT16d_1	40,691	15,187	0	55,877	1	40,691	15,187	0	55,877	1
PDT20d_1	50,057	16,056	0	66,113	1	45,185	13,973	0	59,158	1
PDT26d_1	65,604	1,699	0	67,303	1	65,604	1,699	0	67,303	1
PDT30d_1	44,872	20,497	0	65,369	1	69,540	2,103	0	71,643	1

Tableau 5.8 : Solutions des problèmes Pd, PDd, PTd et PDTd.

A partir du tableau précédent, nous remarquons que dans la majorité des cas, les deux approches AEd_1 et AEHd_1 utilisées convergent vers la même solution. Nous pouvons distinguer que les sommets ont été visités sans accuser un retard mais engendrant un temps d'attente pénalisant. Nous pouvons aussi remarquer que le nombre des solutions dominantes au sens de Pareto a considérablement diminué par rapport au cas statique. Ceci est dû au fait de l'apparition des demandes en retard qui réduit d'une manière importante l'espace de recherche et le nombre de solutions réalisables possibles. La figure 5.14 montre l'efficacité des ces approches.

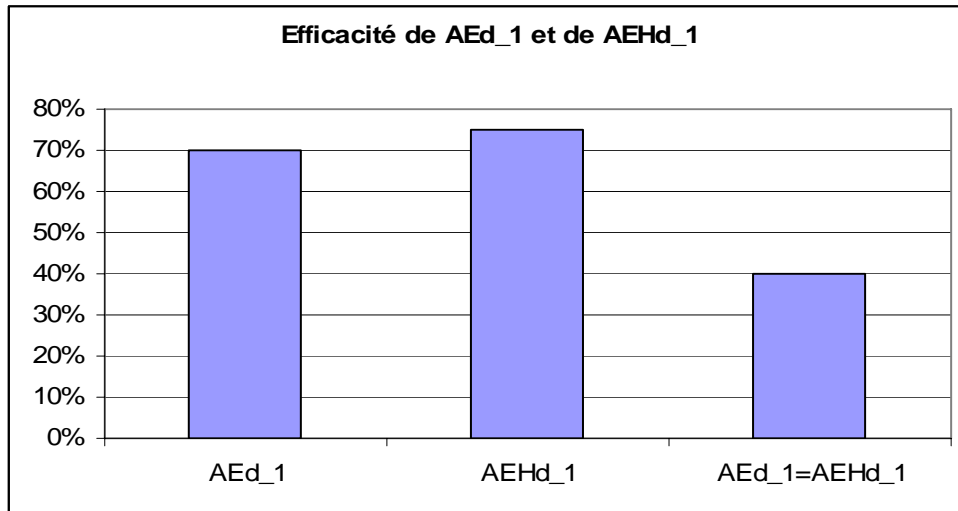


Figure 5.14 : Efficacité des approches AEd_1 et AEHd_1 pour la minimisation de F_1 .

Le tableau 5.9 montre une comparaison entre les résultats obtenus pour le 1-PDPTW et le 1-DPDPTW.

Approche Problème	AEd_1	AEHd_1	AEs_1	AEHs_1	AEs_2	AEHs_2
	Fitness _glob	Fitness _glob	Fitness _glob	Fitness _glob	Fitness _glob	Fitness _glob
P10_1	48,747	48,747	48,747	48,747	48,747	48,747
P16_1	60,137	60,137	60,698	59,757	62,162	62,162
P20_1	73,439	81,391	69,627	67,796	69,597	68,729
P26_1	79,243	77,927	77,691	73,068	76,047	73,103
P30_1	77,450	79,087	73,074	66,784	69,948	73,113
PD10_1	44,894	44,894	44,362	44,362	44,362	44,362
PD16_1	59,514	62,154	52,074	53,023	55,107	52,074
PD20_1	68,132	68,050	52,412	48,992	57,894	50,497
PD26_1	64,595	64,595	60,086	60,36	65,601	58,855
PD30_1	72,087	71,944	59,127	59,127	71,518	71,518
PT10_1	48,957	48,957	47,211	47,211	47,211	47,211
PT16_1	68,971	68,971	61,54	59,397	64,885	58,297
PT20_1	68,599	68,599	65,548	65,267	71,212	67,318
PT26_1	73,634	73,839	67,272	62,798	62,798	63,883
PT30_1	81,194	82,966	72,453	71,64	83,174	78,054
PDT10_1	47,641	47,461	45,684	45,684	45,684	45,684
PDT16_1	55,877	55,877	55,657	53,738	60,037	54,469
PDT20_1	66,113	59,158	54,283	54,283	56,653	56,653
PDT26_1	67,303	67,303	56,917	56,51	59,201	58,769
PDT30_1	65,369	71,643	56,778	53,103	65,065	63,502

Tableau 5.9 : Tableau récapitulatif des résultats du 1-PDPTW et du 1-DPDPTW.

A partir de ce tableau nous pouvons remarquer que les résultats donnés par les approches dynamiques sont satisfaisants. Dans le premier cas, les approches dynamiques ont pu atteindre la solution trouvée pour le cas statique. La figure 5.15 suivante montre l'écart

entre les meilleures solutions des approches dynamiques et les meilleures solutions des approches statiques.

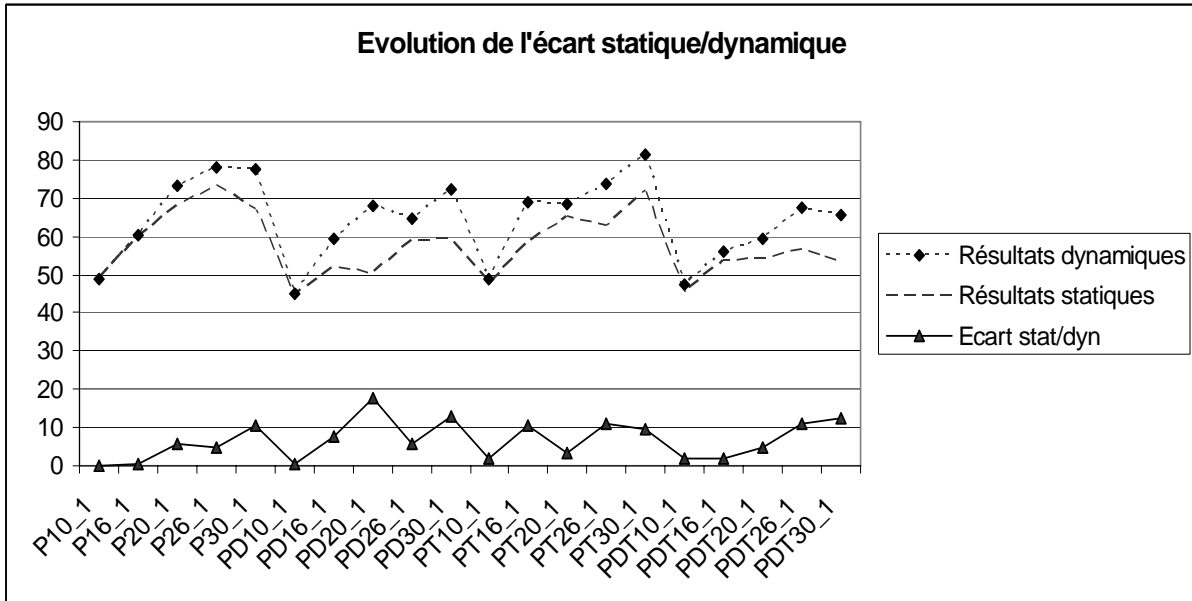


Figure 5.15 : Evolution de l'écart statique/dynamique.

La figure 5.16 suivante montre l'évolution de l'écart entre les meilleurs résultats des approches statiques et les meilleurs résultats des approches dynamiques.

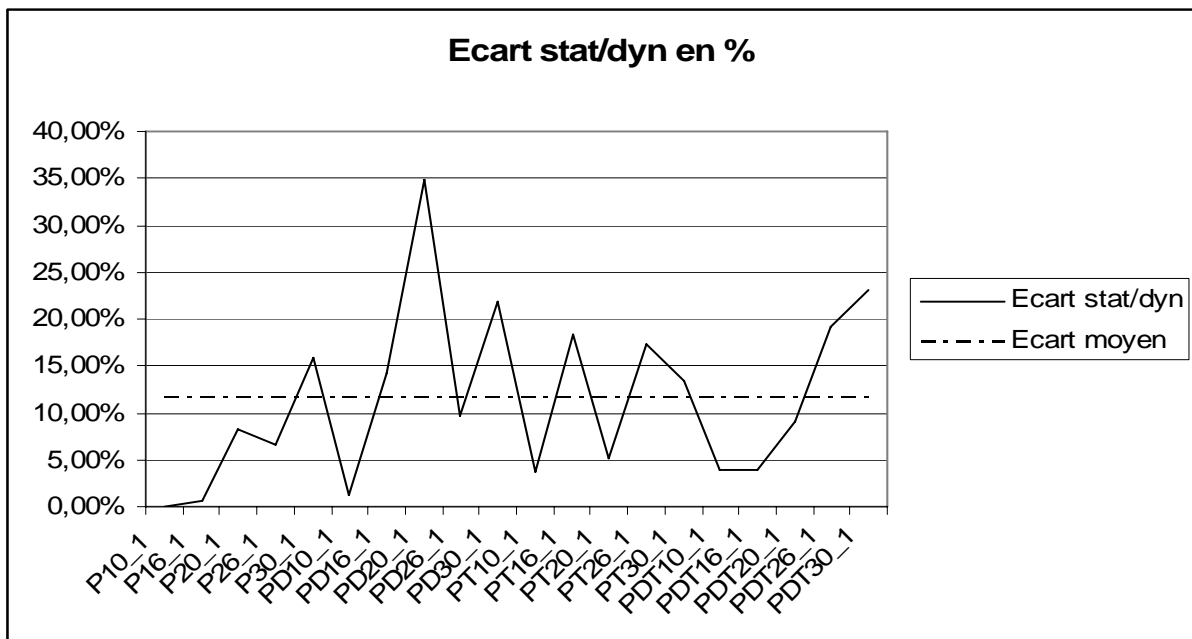


Figure 5.16 : Evolution de l'écart statique/dynamique en pourcentage.

Nous pouvons remarquer, à partir des deux dernières figures, que les approches dynamiques présentent une bonne réactivité face aux demandes urgentes. Ceci est confirmé

par un écart moyen de 12,3% et par l'allure et les valeurs des approches dynamiques qui suivent de près celles des approches statiques.

5.7 Difficulté des benchmarks & Temps de calcul requis

Le temps de calcul requis dépend essentiellement de la taille du problème, sa difficulté et du nombre d'itérations à effectuer. Nous présentons dans ce qui suit la manière avec laquelle nous jugeons la difficulté d'un problème et nous exposons les résultats concernant les temps de calcul obtenus.

5.7.1 Difficulté des benchmarks

Pour mieux évaluer les performances de nos approches, nous avons essayé d'estimer un degré de difficulté à chacun des problèmes testés. Ce degré de difficulté est essentiellement basé sur les positions géographiques des sommets, sur les fenêtres de temps et sur le nombre de clients à visiter.

Le principe de calcul du degré de difficulté d'un problème est le suivant :

- Soit N le nombre de clients, de fournisseurs et du dépôt.
- Calculer la distance moyenne entre les sommets du problème :

$$dist_moy = \frac{\sum_{i \in N} \sum_{j \in N} d_{ij}}{N \times (N - 1)} \quad (5.1)$$

- Calculer la moyenne des largeurs des fenêtres de temps :

$$TW_moy = \frac{\sum_{i \in N} (l_i - e_i)}{N} \quad (5.2)$$

- Calculer la distance du trajet moyen :

$$trajet_moyen = dist_moy(N + 1) \quad (5.3)$$

On divise par $N+1$ pour que le véhicule puisse retourner au dépôt.

- Diviser la valeur du trajet moyen par la valeur de la moyenne des largeurs des fenêtres des temps pour trouver la vitesse moyenne requise :

$$V_moy = \frac{trajet_moyen}{TW_moy} \quad (5.4)$$

- Diviser la vitesse obtenue par la vitesse de simulation V_{sim} pour obtenir le degré de difficulté.

$$Difficulté = \frac{V_{\text{moy}}}{V_{sim}}$$

- Le problème ayant un rapport plus grand est celui le plus difficile. Ceci peut être interprété par le fait qu'un problème nécessitant un véhicule plus rapide est un problème plus difficile qu'un problème se contentant d'un véhicule plus lent.

Le tableau 5.9 suivant montre les étapes de calcul des degrés de difficulté obtenus pour les problèmes testés.

problème	N (cl. + fo.+dè.)	$dist_{moy}$	TW_{moy}	$Trajet_{moyen}$	V_{sim}	V_{moy}	$Difficulté$
PT10_1	11	4,32	321,81	51,84	35	9,66533047	0,2761523
PDT10_1	11	4,9	354,54	58,8	35	9,95092232	0,28431207
PD10_1	11	4,81	344,72	57,72	35	10,0464145	0,28704041
P10_1	11	4,92	314,63	59,04	35	11,2589391	0,32168397
PT16_1	17	5,13	352,94	92,34	35	15,6978523	0,44851007
PDT16_1	17	5,3	352,94	95,4	35	16,2180541	0,46337297
PD16_1	17	4,94	320,23	88,92	35	16,6605252	0,47601501
P16_1	17	5,04	304,29	90,72	35	17,8881988	0,51109139
PDT20_1	21	4,83	342,85	106,26	35	18,5958874	0,53131107
PD20_1	21	4,87	331,42	107,14	35	19,3965361	0,55418675
P20_1	21	4,84	279,42	106,48	35	22,864505	0,65327157
PT20_1	21	5,03	282,85	110,66	35	23,4739261	0,6706836
PD26_1	27	4,87	324,74	136,36	35	25,1943093	0,71983741
PDT26_1	27	5,2	344,44	145,6	35	25,3629079	0,72465451
PT26_1	27	5,33	344,44	149,24	35	25,9969806	0,74277087
PDT30_1	31	4,67	338,7	149,44	35	26,4729849	0,756371
PD30_1	31	4,83	302,8	154,56	35	30,6261559	0,87503303
P26_1	27	5,31	281,92	148,68	35	31,6430193	0,90408627
P30_1	31	4,76	284,54	152,32	35	32,11921	0,91769171
PT30_1	31	5,06	274,83	161,92	35	35,3498526	1,00999579

Tableau 5.9 : Calcul des degrés de difficulté.

5.7.2 Temps de calcul

Une approche de résolution efficace n'est pas seulement une approche qui donne que de bonnes solutions, elle doit les fournir à temps pour le décideur. Une bonne information qui arrive en retard peut ne plus intéresser le décideur surtout dans le cas des problèmes dynamiques. Dans ce qui suit nous présentons, à travers le tableau 5.10, les temps de calcul obtenus en appliquant nos approches AEs_1, AEH_1, AEs_2 et AEHs_2.

Nous rappelons que dans le cas 1-DPDPTW que la saisie des demandes dynamiques est faite directement par l'utilisateur ce qui ne nous permet pas de juger directement le temps de calcul, mais, savoir que les approches dynamiques sont des adaptations de celles conçues pour le cas statique, nous permet de confirmer que ces temps sont proches les uns des autres.

problème	temps moyen de calcul (seconde)			
	AEs_1	AEHs_1	AEs_2	AEHs_2
PT10_1	4,26	2,96	8,46	4,64
PDT10_1	4,26	2,7	8,01	4,031
PD10_1	4,14	1,53	7,15	3,75
P10_1	2,19	1,25	4,84	4,23
PT16_1	11,04	14,95	16,35	15,71
PDT16_1	12,41	15,64	11,01	8,83
PD16_1	10,2	12,29	15,31	14,53
P16_1	10,89	12,87	16,87	16,42
PDT20_1	24,35	24,53	24,09	13,73
PD20_1	24,87	35,68	47,65	40,54
P20_1	13,7	17,43	29,9	24,125
PT20_1	13,9	18,15	31,21	26,4
PD26_1	15,5	20,71	25,1	21,76
PDT26_1	30,09	33,62	35,76	22,46
PT26_1	32,68	29,125	53,609	46,59
PDT30_1	15,28	25,03	51,98	56,81
PD30_1	25,76	25,95	26,64	28,34
P26_1	22,06	26,54	56,26	47,98
P30_1	23,23	25,01	24,82	37,34
PT30_1	33,14	23,92	73,46	52,64
Moyenne	16,6975	18,49425	28,42395	24,5428

Tableau 5.10 : Les temps moyens de calcul.

A partir de ce tableau, nous constatons que les temps de calcul mis par les différentes approches sont petits et conviennent aux problèmes à aspects dynamiques. Ces temps de calcul intéressants ont été obtenus grâce aux opérateurs génétiques et l'heuristique d'échange originaux qui permettent de diminuer l'espace de recherche et de générer des solutions directement viables.

La figure 5.17 montre l'évolution du temps de calcul mis par chacune des approches en fonction de la difficulté du problème. Nous pouvons remarquer que l'approche AEHs_1 qui possède les meilleures performances en terme de qualité des solutions est la deuxième approche la plus rapide avec une moyenne de 18,49 secondes pour la résolution d'un problème.

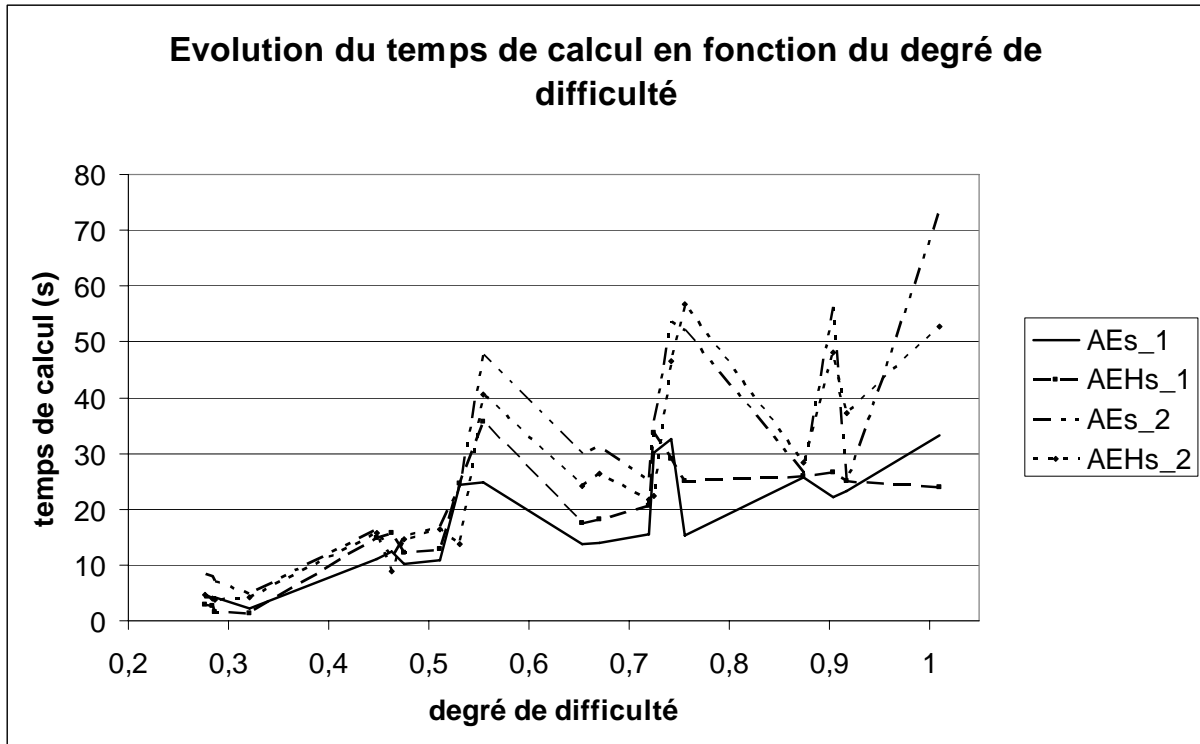


Figure 5.17 : Evolution du temps de calcul en fonction du degré de difficulté.

5.8 Conclusion

Dans ce chapitre, nous avons présenté les résultats trouvés pour les différents problèmes du 1-PDPTW et du 1-DPDPTW traités. Pour élaborer ces résultats nous avons utilisé les approches AEs_1, AEHs_1, AEs_2 et AEHs_2 pour la résolution des problèmes statiques et les approches AEd_1 et AEHd_1 pour la résolution des problèmes dynamiques.

Pour tester ces différentes approches nous avons opté pour la construction d'un ensemble de benchmarks qui représentent des cas permettant de juger l'efficacité et la réactivité de approches de résolution.

Les bons résultats trouvés ont montré l'efficacité de ces approches en qualité des solutions fournies et en temps de calcul mis pour les élaborer. D'après ces résultats l'approche AEHs_1 semble être la plus efficace pour la résolution du 1-PDPTW. Alors que dans le cas dynamique les deux approches AEd_1 et AEHd_1 se valent vis-à-vis du temps de calcul et de la qualité des solutions fournies.

Conclusion Générale

Les problèmes de tournées de véhicules font partie intégrante de la vie quotidienne des décideurs et des planificateurs. Malgré leur simplicité apparente, les problèmes de transport à un seul véhicule sont des problèmes NP-difficiles, et leur étude est utile à la compréhension et à la modélisation des problèmes à plusieurs véhicules.

Le travail de recherche que nous présentons, traite un problème de collecte et de distribution à un seul véhicule 1-PDPTW dans les cas statiques et dynamiques. Ce problème, fréquemment rencontré dans notre vie courante (ramassage scolaire, transport de fonds, livraison alimentaire, ramassage de déchets) consiste à confectionner une route pour le véhicule qui lui permette de servir tous les clients tout en minimisant le coût de la tournée et en respectant la capacité de ce dernier et les fenêtres de temps des clients.

Avant d'aborder la résolution de ce problème, nous avons présenté les différents problèmes de d'élaboration de tournées de véhicules. Nous avons détaillé le VRP (Vehicle Routing Problem) qui constitue le problème originel des différentes variantes aussi présentées et dont fait partie le 1-PDPTW. Nous avons ainsi présenté un état de l'art de ces deux problèmes ainsi que leurs formulations mathématiques respectives.

Ensuite, après avoir rappelé les principes et les caractéristiques des algorithmes évolutionnistes et de la recherche Tabou qui sont les outils de base utilisés dans nos recherches, nous avons proposé plusieurs approches de résolution conçues pour le 1-PDPTW et le 1-DPDPTW. Nous avons ainsi conçu six approches de résolution, quatre pour le cas statique du 1-PDPTW et deux pour le cas dynamique. Ces approches utilisant le principe de la Pareto optimalité permettent d'offrir aux décideurs et aux planificateurs un ensemble de bonnes solutions.

Le troisième chapitre, consacré au 1-PDPTW statique, a présenté les détails des quatre approches AEs_1, AEHs_1, AEs_2 et AEHs_2, où la première est un algorithme évolutionniste principalement basé sur l'utilisation d'opérateurs génétiques spéciaux, la troisième est aussi une méthode évolutionniste essentiellement basée sur l'utilisation des

bornes inférieures. Quant aux deuxième et quatrième approches, elles consistent en l'hybridation de AEs_1 et AEs_2 par une recherche Tabou spécialement conçue pour notre problème.

Les approches AEd_1 et AEHd_1 traitant le 1-DPDPTW ont fait l'objet du quatrième chapitre. Ces deux approches sont une adaptation des méthodes conçues pour le cas statique. En plus de l'état de l'art sur les problèmes de transport à composantes dynamiques, ce chapitre présente en détails les modifications apportées à l'approche AEs_1 et à la recherche Tabou pour adapter AEd_1 et AEHd_1 au 1-DPDPTW.

Dans le cinquième et dernier chapitre, nous avons présenté l'outil informatique développé pour la résolution du 1-PDPTW statique et dynamique. Nous avons aussi exposé les résultats des simulations effectuées sur des benchmarks issus de la littérature. Les résultats élaborés sur ces benchmarks, spécialement conçus pour le m-PDPTW, nous ont incité à créer nos propres instances. Celles-ci ont été conçues de manière à englober les principales configurations possibles de problèmes. Les tests effectués sur ces problèmes ont montré que l'approche AEHs_1 est la plus efficace pour la résolution du cas statique du 1-PDPTW et que les approches AEd_1 et AEHd_1 garantissent l'obtention d'un résultat très satisfaisant dans le cas de la résolution du 1-DPDPTW.

Enfin, dans une perspective d'amélioration des performances de nos approches, nous nous proposons :

- D'utiliser des machines multiprocesseurs pour le calcul et la simulation. Cette implémentation logicielle et matérielle augmenterait, à notre avis, fortement les chances d'obtention de meilleures solutions. En fait, vu les caractères fortement antagonistes des critères que l'on cherche à optimiser, la recherche a toujours tendance à favoriser un critère par rapport à un autre. En utilisant une machine ayant un nombre de processeurs égal au nombre de critères à optimiser et en mixant les solutions entre elles lors de l'évolution du calcul nous garantirons une bonne diversité et de bons résultats.
- De développer nos approches pour traiter le cas m-PDPTW afin de pouvoir tester les outils originaux développés, leur comportement et leurs performances vis-à-vis d'un problème faisant appel à plusieurs véhicules.

Conclusion Générale

- D'améliorer les bornes inférieures conçues afin qu'elles puissent respecter les contraintes de capacité, de précedence et de fenêtrés de temps et ainsi augmenter les performances des approches AEs_2 et AEHs_2.
- De tester nos approches sur un cas réel de distribution de marchandise.

Annexes

ANNEXE 1 : Les benchmarks élaborés**P10 1**

1	150	0						
0	50	50	0	0	600	0	0	0
1	81	22	15	59	339	10	0	2
2	96	48	-15	411	546	10	1	0
3	93	80	10	68	327	10	0	4
4	55	70	-10	305	651	10	3	0
5	1	78	5	95	335	10	0	6
6	17	48	-5	200	465	10	5	0
7	22	17	5	59	590	10	0	8
8	51	0	-5	210	450	10	7	0
9	48	10	15	231	516	10	0	10
10	45	27	-15	194	474	10	9	0

P16 1

1	150	0						
0	50	50	0	0	600	0	0	0
1	96	27	5	110	271	10	0	2
2	76	12	-5	95	325	10	1	0
3	75	64	5	322	635	10	0	4
4	98	59	-5	355	612	10	3	0
5	73	73	15	60	355	10	0	6
6	81	88	-15	32	333	10	5	0
7	55	86	10	304	565	10	0	8
8	55	88	-10	312	592	10	7	0
9	1	96	10	65	310	10	0	10
10	8	85	-10	64	398	10	9	0
11	17	65	5	210	414	10	0	12
12	18	60	-5	188	388	10	11	0
13	18	27	15	118	598	10	0	14
14	27	22	-15	30	601	10	13	0
15	46	10	10	180	423	10	0	16
16	58	22	-10	250	448	10	15	0

P20 1

1	150	0						
0	50	50	0	0	600	0	0	0
1	70	2	15	116	282	10	0	2
2	70	18	-15	99	328	10	1	0
3	88	53	15	409	600	10	0	4
4	90	48	-15	319	584	10	3	0
5	75	73	15	9	336	10	0	6
6	74	98	-15	110	246	10	5	0
7	37	90	5	365	642	10	0	8
8	64	70	-5	339	636	10	7	0
9	4	75	10	129	346	10	0	10
10	27	83	-10	113	335	10	9	0
11	3	55	15	173	416	10	0	12
12	27	60	-15	222	439	10	11	0
13	22	2	15	19	563	10	0	14
14	12	16	-15	101	604	10	13	0
15	43	12	15	203	473	10	0	16
16	53	16	-15	225	494	10	15	0
17	40	8	10	258	495	10	0	18
18	60	6	-10	231	507	10	17	0
19	48	27	5	246	459	10	0	20
20	37	2	-5	276	445	10	19	0

Annexes

P26 1

1	150	0						
0	50	50	0	0	600	0	0	0
1	75	12	15	95	270	10	0	2
2	76	17	-15	61	325	10	1	0
3	96	13	-10	81	332	10	0	4
4	73	45	-10	408	643	10	3	0
5	80	58	15	366	587	10	0	6
6	86	48	-15	398	645	10	5	0
7	93	78	5	90	316	10	0	8
8	98	88	-5	25	350	10	7	0
9	90	86	5	32	253	10	0	10
10	35	98	-5	335	641	10	9	0
11	60	70	15	303	639	10	0	12
12	50	86	-15	399	616	10	11	0
13	2	83	10	114	336	10	0	14
14	15	91	-10	86	303	10	13	0
15	24	91	5	67	413	10	0	16
16	25	45	-5	148	469	10	15	0
17	22	46	15	187	376	10	0	18
18	4	43	-15	211	360	10	17	0
19	15	8	5	115	619	10	0	20
20	1	0	-5	85	555	10	19	0
21	22	17	5	113	569	10	0	22
22	49	3	-5	219	490	10	21	0
23	55	25	15	277	510	10	0	24
24	63	5	-15	272	421	10	23	0
25	46	22	15	273	512	10	0	26
26	60	13	-15	240	462	10	25	0

P30 1

1	150	0						
0	50	50	0	0	600	0	0	0
1	80	2	5	54	244	10	0	2
2	75	25	-5	47	250	10	1	0
3	96	8	5	103	250	10	0	4
4	90	65	-5	395	541	10	3	0
5	79	45	15	338	633	10	0	6
6	91	50	-15	312	611	10	5	0
7	98	71	5	7	354	10	0	8
8	98	88	-5	105	244	10	7	0
9	88	96	5	60	285	10	0	10
10	60	90	-5	401	600	10	9	0
11	59	90	15	371	636	10	0	12
12	55	71	-15	345	567	10	11	0
13	22	90	15	67	361	10	0	14
14	24	98	-15	96	414	10	13	0
15	27	70	15	96	300	10	0	16
16	20	40	-15	204	476	10	15	0
17	30	37	15	207	433	10	0	18
18	6	39	-15	163	441	10	17	0
19	29	10	10	38	623	10	0	20
20	12	10	-10	43	544	10	19	0
21	20	12	15	43	650	10	0	22
22	48	29	-15	238	539	10	21	0
23	41	13	10	244	496	10	0	24
24	48	15	-10	242	449	10	23	0
25	55	16	5	220	486	10	0	26
26	55	12	-5	233	521	10	25	0
27	55	17	5	231	453	10	0	28
28	54	12	-5	203	508	10	27	0
29	56	30	15	238	453	10	0	30
30	54	18	-15	281	484	10	29	0

Annexes

P10 2

1	150	0						
0	50	50	0	0	600	0	0	0
1	81	22	10	75	353	10	0	2
2	100	56	-10	318	567	10	1	0
3	95	91	15	71	330	10	0	4
4	37	71	-15	360	599	10	3	0
5	11	98	5	137	391	10	0	6
6	20	60	-5	122	430	10	5	0
7	4	27	15	21	601	10	0	8
8	40	6	-15	246	530	10	7	0
9	45	6	15	256	436	10	0	10
10	60	15	-15	288	472	10	9	0

P16 2

1	150	0						
0	50	50	0	0	600	0	0	0
1	75	18	10	61	297	10	0	2
2	93	25	-10	29	293	10	1	0
3	93	40	15	408	631	10	0	4
4	95	63	-15	388	582	10	3	0
5	74	98	5	11	348	10	0	6
6	98	76	-5	114	274	10	5	0
7	37	86	10	371	576	10	0	8
8	35	90	-10	410	583	10	7	0
9	27	78	10	69	329	10	0	10
10	29	96	-10	80	382	10	9	0
11	6	58	10	187	466	10	0	12
12	16	54	-10	204	379	10	11	0
13	27	20	15	52	597	10	0	14
14	22	3	-15	40	652	10	13	0
15	51	12	5	199	471	10	0	16
16	64	12	-5	212	427	10	15	0

P20 2

1	150	0						
0	50	50	0	0	600	0	0	0
1	96	10	15	105	318	10	0	2
2	74	30	-15	107	279	10	1	0
3	76	54	5	309	569	10	0	4
4	96	59	-5	375	609	10	3	0
5	78	93	10	106	338	10	0	6
6	74	70	-10	79	333	10	5	0
7	53	86	5	347	619	10	0	8
8	56	90	-5	337	548	10	7	0
9	6	88	10	116	324	10	0	10
10	27	80	-10	100	395	10	9	0
11	10	61	15	229	412	10	0	12
12	17	55	-15	203	446	10	11	0
13	10	25	10	84	630	10	0	14
14	3	22	-10	89	589	10	13	0
15	64	0	10	241	532	10	0	16
16	46	22	-10	246	526	10	15	0
17	40	27	15	186	461	10	0	18
18	45	25	-15	247	489	10	17	0
19	40	18	5	223	424	10	0	20
20	37	30	-5	235	420	10	19	0

Annexes

P26 2

1	150	0						
0	50	50	0	0	600	0	0	0
1	73	12	10	93	256	10	0	2
2	83	2	-10	115	299	10	1	0
3	75	22	5	49	279	10	0	4
4	91	50	-5	401	546	10	3	0
5	96	60	5	388	643	10	0	6
6	75	55	-5	304	541	10	5	0
7	81	98	10	7	302	10	0	8
8	88	96	-10	59	300	10	7	0
9	81	88	5	51	353	10	0	10
10	60	71	-5	413	594	10	9	0
11	49	88	10	395	601	10	0	12
12	39	76	-10	364	593	10	11	0
13	4	70	5	66	338	10	0	14
14	15	100	-5	114	369	10	13	0
15	20	81	10	89	398	10	0	16
16	25	50	-10	122	420	10	15	0
17	29	54	10	238	393	10	0	18
18	15	59	-10	201	452	10	17	0
19	8	29	10	48	639	10	0	20
20	13	1	-10	87	599	10	19	0
21	13	17	15	68	540	10	0	22
22	43	11	-15	247	533	10	21	0
23	60	24	15	289	472	10	0	24
24	40	15	-15	229	536	10	23	0
25	37	6	15	283	472	10	0	26
26	45	6	-15	266	460	10	25	0

P30 2

1	150	0						
0	50	50	0	0	600	0	0	0
1	80	3	15	103	334	10	0	2
2	96	0	-15	52	304	10	1	0
3	88	4	10	17	282	10	0	4
4	91	37	-10	374	632	10	3	0
5	75	45	15	349	612	10	0	6
6	86	44	-15	375	643	10	5	0
7	98	75	10	81	243	10	0	8
8	100	83	-10	76	286	10	7	0
9	90	90	10	55	315	10	0	10
10	45	88	-10	340	628	10	9	0
11	45	75	15	394	599	10	0	12
12	40	83	-15	336	594	10	11	0
13	8	81	10	106	322	10	0	14
14	25	98	-10	159	368	10	13	0
15	0	83	10	87	349	10	0	16
16	3	45	-10	238	471	10	15	0
17	27	65	5	197	388	10	0	18
18	20	39	-5	156	372	10	17	0
19	15	25	5	55	643	10	0	20
20	17	18	-5	42	641	10	19	0
21	8	12	10	25	605	10	0	22
22	39	12	-10	219	524	10	21	0
23	40	3	10	255	473	10	0	24
24	40	29	-10	217	453	10	23	0
25	48	5	15	204	464	10	0	26
26	44	5	-15	205	518	10	25	0
27	39	11	5	232	484	10	0	28
28	50	8	-5	212	537	10	27	0
29	61	4	5	186	511	10	0	30
30	65	6	-5	276	489	10	29	0

Annexes

PD10 1

1	150	0						
0	50	50	0	0	600	0	0	0
1	81	22	15	59	339	10	0	2
2	96	48	-15	411	546	10	1	0
3	93	80	10	68	327	10	0	4
4	55	70	-10	305	651	10	3	0
5	1	78	5	95	335	10	0	6
6	17	48	-5	200	465	10	5	0
7	22	17	5	59	590	10	0	8
8	51	0	-5	210	450	10	7	0
9	48	10	15	231	516	10	0	10
10	45	27	-15	194	474	10	9	0

PD16 1

1	150	0						
0	50	50	0	0	600	0	0	0
1	96	27	5	110	271	10	0	2
2	76	12	-5	95	325	10	1	0
3	75	64	5	322	635	10	0	4
4	98	59	-5	355	612	10	3	0
5	73	73	15	60	355	10	0	6
6	81	88	-15	32	333	10	5	0
7	55	86	10	304	565	10	0	8
8	55	88	-10	312	592	10	7	0
9	1	96	10	65	310	10	0	10
10	8	85	-10	64	398	10	9	0
11	17	65	5	210	414	10	0	12
12	18	60	-5	188	388	10	11	0
13	18	27	15	118	598	10	0	14
14	27	22	-15	30	601	10	13	0
15	46	10	10	180	423	10	0	16
16	58	22	-10	250	448	10	15	0

PD20 1

1	150	0						
0	50	50	0	0	600	0	0	0
1	70	2	15	116	282	10	0	2
2	70	18	-15	99	328	10	1	0
3	88	53	15	409	600	10	0	4
4	90	48	-15	319	584	10	3	0
5	75	73	15	9	336	10	0	6
6	74	98	-15	110	246	10	5	0
7	37	90	5	365	642	10	0	8
8	64	70	-5	339	636	10	7	0
9	4	75	10	129	346	10	0	10
10	27	83	-10	113	335	10	9	0
11	3	55	15	173	416	10	0	12
12	27	60	-15	222	439	10	11	0
13	22	2	15	19	563	10	0	14
14	12	16	-15	101	604	10	13	0
15	43	12	15	203	473	10	0	16
16	53	16	-15	225	494	10	15	0
17	40	8	10	258	495	10	0	18
18	60	6	-10	231	507	10	17	0
19	48	27	5	246	459	10	0	20
20	37	2	-5	276	445	10	19	0

Annexes

PD26 1

1	150	0						
0	50	50	0	0	600	0	0	0
1	75	12	15	95	270	10	0	2
2	76	17	-15	61	325	10	1	0
3	96	13	10	81	332	10	0	4
4	73	45	-10	408	643	10	3	0
5	80	58	15	366	587	10	0	6
6	86	48	-15	398	645	10	5	0
7	93	78	5	90	316	10	0	8
8	98	88	-5	25	350	10	7	0
9	90	86	5	32	253	10	0	10
10	35	98	-5	335	641	10	9	0
11	60	70	15	303	639	10	0	12
12	50	86	-15	399	616	10	11	0
13	2	83	10	114	336	10	0	14
14	15	91	-10	86	303	10	13	0
15	24	91	5	67	413	10	0	16
16	25	45	-5	148	469	10	15	0
17	22	46	15	187	376	10	0	18
18	4	43	-15	211	360	10	17	0
19	15	8	5	115	619	10	0	20
20	1	0	-5	85	555	10	19	0
21	22	17	5	113	569	10	0	22
22	49	3	-5	219	490	10	21	0
23	55	25	15	277	510	10	0	24
24	63	5	-15	272	421	10	23	0
25	46	22	15	273	512	10	0	26
26	60	13	-15	240	462	10	25	0

PD30 1

1	150	0						
0	50	50	0	0	600	0	0	0
1	80	2	5	54	244	10	0	2
2	75	25	-5	47	250	10	1	0
3	96	8	5	103	250	10	0	4
4	90	65	-5	395	541	10	3	0
5	79	45	15	338	633	10	0	6
6	91	50	-15	312	611	10	5	0
7	98	71	5	7	354	10	0	8
8	98	88	-5	105	244	10	7	0
9	88	96	5	60	285	10	0	10
10	60	90	-5	401	600	10	9	0
11	59	90	15	371	636	10	0	12
12	55	71	-15	345	567	10	11	0
13	22	90	15	67	361	10	0	14
14	24	98	-15	96	414	10	13	0
15	27	70	15	96	300	10	0	16
16	20	40	-15	204	476	10	15	0
17	30	37	15	207	433	10	0	18
18	6	39	-15	163	441	10	17	0
19	29	10	10	38	623	10	0	20
20	12	10	-10	43	544	10	19	0
21	20	12	15	43	650	10	0	22
22	48	29	-15	238	539	10	21	0
23	41	13	10	244	496	10	0	24
24	48	15	-10	242	449	10	23	0
25	55	16	5	220	486	10	0	26
26	55	12	-5	233	521	10	25	0
27	55	17	5	231	453	10	0	28
28	54	12	-5	203	508	10	27	0
29	56	30	15	238	453	10	0	30
30	54	18	-15	281	484	10	29	0

PD10 2

1	150	0						
0	50	50	0	0	600	0	0	0
1	81	22	10	75	353	10	0	2
2	100	56	-10	318	567	10	1	0
3	95	91	15	71	330	10	0	4
4	37	71	-15	360	599	10	3	0
5	11	98	5	137	391	10	0	6
6	20	60	-5	122	430	10	5	0
7	4	27	15	21	601	10	0	8
8	40	6	-15	246	530	10	7	0
9	45	6	15	256	436	10	0	10
10	60	15	-15	288	472	10	9	0

PD16 2

1	150	0						
0	50	50	0	0	600	0	0	0
1	75	18	10	61	297	10	0	2
2	93	25	-10	29	293	10	1	0
3	93	40	15	408	631	10	0	4
4	95	63	-15	388	582	10	3	0
5	74	98	5	11	348	10	0	6
6	98	76	-5	114	274	10	5	0
7	37	86	10	371	576	10	0	8
8	35	90	-10	410	583	10	7	0
9	27	78	10	69	329	10	0	10
10	29	96	-10	80	382	10	9	0
11	6	58	10	187	466	10	0	12
12	16	54	-10	204	379	10	11	0
13	27	20	15	52	597	10	0	14
14	22	3	-15	40	652	10	13	0
15	51	12	5	199	471	10	0	16
16	64	12	-5	212	427	10	15	0

PD20 2

1	150	0						
0	50	50	0	0	600	0	0	0
1	96	10	15	105	318	10	0	2
2	74	30	-15	107	279	10	1	0
3	76	54	5	309	569	10	0	4
4	96	59	-5	375	609	10	3	0
5	78	93	10	106	338	10	0	6
6	74	70	-10	79	333	10	5	0
7	53	86	5	347	619	10	0	8
8	56	90	-5	337	548	10	7	0
9	6	88	10	116	324	10	0	10
10	27	80	-10	100	395	10	9	0
11	10	61	15	229	412	10	0	12
12	17	55	-15	203	446	10	11	0
13	10	25	10	84	630	10	0	14
14	3	22	-10	89	589	10	13	0
15	64	0	10	241	532	10	0	16
16	46	22	-10	246	526	10	15	0
17	40	27	15	186	461	10	0	18
18	45	25	-15	247	489	10	17	0
19	40	18	5	223	424	10	0	20
20	37	30	-5	235	420	10	19	0

PD26 2

1	150	0						
0	50	50	0	0	600	0	0	0
1	73	12	10	93	256	10	0	2
2	83	2	-10	115	299	10	1	0
3	75	22	5	49	279	10	0	4
4	91	50	-5	401	546	10	3	0
5	96	60	5	388	643	10	0	6
6	75	55	-5	304	541	10	5	0
7	81	98	10	7	302	10	0	8
8	88	96	-10	59	300	10	7	0
9	81	88	5	51	353	10	0	10
10	60	71	-5	413	594	10	9	0
11	49	88	10	395	601	10	0	12
12	39	76	-10	364	593	10	11	0
13	4	70	5	66	338	10	0	14
14	15	100	-5	114	369	10	13	0
15	20	81	10	89	398	10	0	16
16	25	50	-10	122	420	10	15	0
17	29	54	10	238	393	10	0	18
18	15	59	-10	201	452	10	17	0
19	8	29	10	48	639	10	0	20
20	13	1	-10	87	599	10	19	0
21	13	17	15	68	540	10	0	22
22	43	11	-15	247	533	10	21	0
23	60	24	15	289	472	10	0	24
24	40	15	-15	229	536	10	23	0
25	37	6	15	283	472	10	0	26
26	45	6	-15	266	460	10	25	0

PD30 2

1	150	0						
0	50	50	0	0	600	0	0	0
1	80	3	15	103	334	10	0	2
2	96	0	-15	52	304	10	1	0
3	88	4	10	17	282	10	0	4
4	91	37	-10	374	632	10	3	0
5	75	45	15	349	612	10	0	6
6	86	44	-15	375	643	10	5	0
7	98	75	10	81	243	10	0	8
8	100	83	-10	76	286	10	7	0
9	90	90	10	55	315	10	0	10
10	45	88	-10	340	628	10	9	0
11	45	75	15	394	599	10	0	12
12	40	83	-15	336	594	10	11	0
13	8	81	10	106	322	10	0	14
14	25	98	-10	159	368	10	13	0
15	0	83	10	87	349	10	0	16
16	3	45	-10	238	471	10	15	0
17	27	65	5	197	388	10	0	18
18	20	39	-5	156	372	10	17	0
19	15	25	5	55	643	10	0	20
20	17	18	-5	42	641	10	19	0
21	8	12	10	25	605	10	0	22
22	39	12	-10	219	524	10	21	0
23	40	3	10	255	473	10	0	24
24	40	29	-10	217	453	10	23	0
25	48	5	15	204	464	10	0	26
26	44	5	-15	205	518	10	25	0
27	39	11	5	232	484	10	0	28
28	50	8	-5	212	537	10	27	0
29	61	4	5	186	511	10	0	30
30	65	6	-5	276	489	10	29	0

Annexes

PT10 1

1	150	0							
0	50	50	0	0	600	0	0	0	
1	73	24	5	0	300	10	0	2	
2	73	63	-5	300	600	10	1	0	
3	81	85	5	60	300	10	0	4	
4	50	75	-5	360	600	10	3	0	
5	27	71	5	60	300	10	0	6	
6	3	46	-5	180	420	10	5	0	
7	8	25	5	60	600	10	0	8	
8	55	12	-5	180	480	10	7	0	
9	61	20	10	180	480	10	0	10	
10	41	17	-10	240	480	10	9	0	

PT16 1

1	150	0							
0	50	50	0	0	600	0	0	0	
1	79	11	15	0	300	10	0	2	
2	81	6	-15	0	300	10	1	0	
3	88	53	15	300	600	10	0	4	
4	75	53	-15	300	600	10	3	0	
5	98	80	5	0	300	10	0	6	
6	74	70	-5	0	300	10	5	0	
7	65	85	10	300	600	10	0	8	
8	37	75	-10	300	600	10	7	0	
9	12	86	10	60	360	10	0	10	
10	30	91	-10	60	360	10	9	0	
11	12	64	15	120	420	10	0	12	
12	1	48	-15	120	420	10	11	0	
13	30	15	10	0	600	10	0	14	
14	1	20	-10	0	600	10	13	0	
15	59	17	5	180	480	10	0	16	
16	39	13	-5	180	480	10	15	0	

PT20 1

1	150	0							
0	50	50	0	0	600	0	0	0	
1	93	20	15	0	300	10	0	2	
2	88	12	-15	0	300	10	1	0	
3	88	49	10	360	540	10	0	4	
4	83	59	-10	360	540	10	3	0	
5	98	74	15	0	240	10	0	6	
6	86	81	-15	0	240	10	5	0	
7	49	71	10	300	540	10	0	8	
8	48	73	-10	360	600	10	7	0	
9	18	76	5	120	360	10	0	10	
10	2	80	-5	120	300	10	9	0	
11	22	59	5	180	360	10	0	12	
12	17	51	-5	120	360	10	11	0	
13	2	5	15	0	540	10	0	14	
14	3	24	-15	0	540	10	13	0	
15	45	0	15	240	480	10	0	16	
16	51	8	-15	180	420	10	15	0	
17	39	27	15	180	420	10	0	18	
18	56	27	-15	180	480	10	17	0	
19	53	11	5	180	480	10	0	20	
20	64	2	-5	240	420	10	19	0	

Annexes

PT26 1

1	150	0							
0	50	50	0	0	600	0	0	0	
1	98	29	5	0	300	10	0	2	
2	91	20	-5	0	300	10	1	0	
3	79	6	15	0	300	10	0	4	
4	91	46	-15	300	600	10	3	0	
5	76	45	5	300	600	10	0	6	
6	96	64	-5	300	600	10	5	0	
7	75	91	5	0	300	10	0	8	
8	75	93	-5	0	300	10	7	0	
9	79	88	5	0	300	10	0	10	
10	41	93	-5	300	600	10	9	0	
11	64	93	10	300	600	10	0	12	
12	60	75	-10	300	600	10	11	0	
13	3	83	10	60	360	10	0	14	
14	4	88	-10	60	360	10	13	0	
15	25	81	15	60	360	10	0	16	
16	4	35	-15	120	420	10	15	0	
17	13	55	5	120	420	10	0	18	
18	20	45	-5	120	420	10	17	0	
19	13	17	5	0	600	10	0	20	
20	6	22	-5	0	600	10	19	0	
21	20	4	5	0	600	10	0	22	
22	63	12	-5	180	480	10	21	0	
23	43	24	10	180	480	10	0	24	
24	40	22	-10	180	480	10	23	0	
25	60	4	10	180	480	10	0	26	
26	63	15	-10	180	480	10	25	0	

PT30 1

1	150	0							
0	50	50	0	0	600	0	0	0	
1	93	15	15	60	300	10	0	2	
2	95	22	-15	0	240	10	1	0	
3	90	18	15	60	240	10	0	4	
4	96	59	-15	300	600	10	3	0	
5	70	60	15	360	600	10	0	6	
6	96	44	-15	300	540	10	5	0	
7	95	95	10	0	300	10	0	8	
8	88	93	-10	60	300	10	7	0	
9	86	88	15	60	240	10	0	10	
10	45	75	-15	300	540	10	9	0	
11	50	98	5	360	600	10	0	12	
12	60	75	-5	300	540	10	11	0	
13	13	70	10	60	360	10	0	14	
14	6	91	-10	120	360	10	13	0	
15	8	90	15	120	300	10	0	16	
16	18	49	-15	180	420	10	15	0	
17	16	35	10	180	360	10	0	18	
18	29	51	-10	120	360	10	17	0	
19	0	29	10	60	540	10	0	20	
20	10	12	-10	60	540	10	19	0	
21	8	10	5	60	540	10	0	22	
22	59	25	-5	240	420	10	21	0	
23	50	25	15	180	480	10	0	24	
24	55	20	-15	240	420	10	23	0	
25	65	30	5	180	480	10	0	26	
26	60	6	-5	180	420	10	25	0	
27	55	8	5	240	480	10	0	28	
28	50	24	-5	180	480	10	27	0	
29	35	30	10	180	420	10	0	30	
30	46	20	-10	180	420	10	29	0	

Annexes

PT10 2

1	150	0						
0	50	50	0	0	600	0	0	0
1	100	24	10	0	300	10	0	2
2	83	35	-10	300	600	10	1	0
3	88	81	5	0	300	10	0	4
4	46	75	-5	300	600	10	3	0
5	6	91	15	60	360	10	0	6
6	20	44	-15	120	420	10	5	0
7	27	12	15	0	600	10	0	8
8	50	4	-15	180	480	10	7	0
9	58	11	15	180	480	10	0	10
10	50	20	-15	180	480	10	9	0

PT16 2

1	150	0						
0	50	50	0	0	600	0	0	0
1	91	10	15	0	300	10	0	2
2	95	15	-15	0	300	10	1	0
3	91	39	10	300	600	10	0	4
4	91	50	-10	300	600	10	3	0
5	98	78	10	0	300	10	0	6
6	70	80	-10	0	300	10	5	0
7	59	73	10	300	600	10	0	8
8	56	70	-10	300	600	10	7	0
9	0	75	10	60	360	10	0	10
10	12	81	-10	60	360	10	9	0
11	12	56	10	120	420	10	0	12
12	2	50	-10	120	420	10	11	0
13	3	18	5	0	600	10	0	14
14	8	10	-5	0	600	10	13	0
15	65	13	15	180	480	10	0	16
16	55	6	-15	180	480	10	15	0

PT20 2

1	150	0						
0	50	50	0	0	600	0	0	0
1	100	24	5	0	300	10	0	2
2	83	15	-5	0	300	10	1	0
3	81	39	5	300	600	10	0	4
4	90	58	-5	300	600	10	3	0
5	100	85	15	0	300	10	0	6
6	78	98	-15	0	300	10	5	0
7	50	96	15	300	600	10	0	8
8	39	81	-15	300	600	10	7	0
9	11	73	5	60	360	10	0	10
10	2	90	-5	60	360	10	9	0
11	4	41	15	120	480	10	0	12
12	5	49	-15	120	420	10	11	0
13	16	8	10	0	600	10	0	14
14	5	3	-10	0	600	10	13	0
15	56	2	15	180	480	10	0	16
16	37	22	-15	180	480	10	15	0
17	44	20	10	180	480	10	0	18
18	64	12	-10	180	480	10	17	0
19	39	22	15	180	480	10	0	20
20	48	22	-15	180	480	10	19	0

Annexes

PT26 2

1	150	0						
0	50	50	0	0	600	0	0	0
1	78	29	5	0	300	10	0	2
2	70	6	-5	0	300	10	1	0
3	81	6	5	0	300	10	0	4
4	81	39	-5	300	600	10	3	0
5	96	54	5	300	600	10	0	6
6	75	59	-5	300	600	10	5	0
7	70	96	10	0	300	10	0	8
8	83	98	-10	0	300	10	7	0
9	96	81	15	0	300	10	0	10
10	50	80	-15	300	600	10	9	0
11	50	70	10	300	600	10	0	12
12	50	93	-10	300	600	10	11	0
13	27	98	5	60	360	10	0	14
14	20	70	-5	60	360	10	13	0
15	6	90	10	60	360	10	0	16
16	3	50	-10	120	420	10	15	0
17	10	43	15	120	420	10	0	18
18	1	50	-15	120	420	10	17	0
19	25	22	15	0	600	10	0	20
20	13	24	-15	0	600	10	19	0
21	20	25	15	0	600	10	0	22
22	50	2	-15	180	480	10	21	0
23	64	25	10	180	480	10	0	24
24	54	30	-10	180	480	10	23	0
25	46	12	10	180	480	10	0	26
26	48	22	-10	180	480	10	25	0

PT30 2

1	150	0						
0	50	50	0	0	600	0	0	0
1	71	2	10	0	300	10	0	2
2	93	8	-10	0	300	10	1	0
3	88	18	15	0	300	10	0	4
4	73	60	-15	300	600	10	3	0
5	83	61	5	300	600	10	0	6
6	86	58	-5	300	600	10	5	0
7	75	74	10	0	300	10	0	8
8	83	70	-10	0	300	10	7	0
9	98	100	10	0	300	10	0	10
10	63	74	-10	300	600	10	9	0
11	50	83	5	300	600	10	0	12
12	46	74	-5	300	600	10	11	0
13	16	85	15	60	360	10	0	14
14	17	81	-15	60	360	10	13	0
15	6	75	10	60	360	10	0	16
16	5	54	-10	120	420	10	15	0
17	29	50	10	120	420	10	0	18
18	15	45	-10	120	480	10	17	0
19	6	17	10	0	600	10	0	20
20	18	30	-10	0	600	10	19	0
21	20	17	15	0	600	10	0	22
22	45	0	-15	180	480	10	21	0
23	54	17	10	180	480	10	0	24
24	50	20	-10	180	480	10	23	0
25	48	22	15	180	480	10	0	26
26	46	20	-15	180	480	10	25	0
27	49	8	5	180	480	10	0	28
28	54	25	-5	180	480	10	27	0
29	35	6	15	180	480	10	0	30
30	64	10	-15	180	480	10	29	0

Annexes

PDT10 1

1	150	0						
0	50	50	0	0	600	0	0	0
1	75	12	10	0	300	10	0	2
2	73	46	-10	300	600	10	1	0
3	83	78	15	0	300	10	0	4
4	43	71	-15	300	600	10	3	0
5	5	80	15	60	360	10	0	6
6	5	39	-15	120	420	10	5	0
7	6	12	10	0	600	10	0	8
8	44	0	-10	180	480	10	7	0
9	41	4	15	180	480	10	0	10
10	37	6	-15	180	480	10	9	0

PDT16 1

1	150	0						
0	50	50	0	0	600	0	0	0
1	86	17	5	0	300	10	0	2
2	75	0	-5	0	300	10	1	0
3	81	45	10	300	600	10	0	4
4	79	53	-10	300	600	10	3	0
5	88	80	10	0	300	10	0	6
6	81	74	-10	0	300	10	5	0
7	50	85	10	300	600	10	0	8
8	44	76	-10	300	600	10	7	0
9	6	76	10	60	360	10	0	10
10	4	85	-10	60	360	10	9	0
11	8	46	5	120	420	10	0	12
12	13	37	-5	120	420	10	11	0
13	13	12	10	0	600	10	0	14
14	12	16	-10	0	600	10	13	0
15	37	3	15	180	480	10	0	16
16	37	3	-15	180	480	10	15	0

PDT20 1

1	150	0						
0	50	50	0	0	600	0	0	0
1	81	12	5	0	300	10	0	2
2	76	10	-5	0	300	10	1	0
3	83	49	10	300	600	10	0	4
4	75	44	-10	300	600	10	3	0
5	74	71	10	0	300	10	0	6
6	75	75	-10	0	300	10	5	0
7	48	83	15	300	600	10	0	8
8	40	73	-15	300	600	10	7	0
9	12	75	10	60	360	10	0	10
10	2	83	-10	60	360	10	9	0
11	6	43	15	120	420	10	0	12
12	3	35	-15	120	420	10	11	0
13	12	8	5	0	600	10	0	14
14	0	10	-5	0	600	10	13	0
15	40	3	5	180	480	10	0	16
16	35	12	-5	180	480	10	15	0
17	35	8	15	180	480	10	0	18
18	45	5	-15	180	480	10	17	0
19	43	8	5	180	480	10	0	20
20	35	4	-5	180	480	10	19	0

Annexes

PDT26 1

1	150	0						
0	50	50	0	0	600	0	0	0
1	79	8	15	0	300	10	0	2
2	76	3	-15	0	300	10	1	0
3	75	12	5	0	300	10	0	4
4	70	50	-5	300	600	10	3	0
5	75	41	15	300	600	10	0	6
6	86	37	-15	300	600	10	5	0
7	76	70	10	0	300	10	0	8
8	85	81	-10	0	300	10	7	0
9	76	86	5	0	300	10	0	10
10	50	88	-5	300	600	10	9	0
11	37	86	5	300	600	10	0	12
12	41	78	-5	300	600	10	11	0
13	16	86	5	60	360	10	0	14
14	2	83	-5	60	360	10	13	0
15	8	78	15	60	360	10	0	16
16	11	37	-15	120	420	10	15	0
17	8	45	10	120	420	10	0	18
18	8	50	-10	120	420	10	17	0
19	8	12	10	0	600	10	0	20
20	6	1	-10	0	600	10	19	0
21	1	8	10	0	600	10	0	22
22	51	12	-10	180	480	10	21	0
23	45	1	15	180	480	10	0	24
24	50	17	-15	180	480	10	23	0
25	53	8	5	180	480	10	0	26
26	44	6	-5	180	480	10	25	0

PDT30 1

1	150	0						
0	50	50	0	0	600	0	0	0
1	70	6	15	0	300	10	0	2
2	74	2	-15	0	300	10	1	0
3	75	12	15	0	300	10	0	4
4	76	45	-15	300	600	10	3	0
5	74	43	15	300	600	10	0	6
6	83	45	-15	300	600	10	5	0
7	75	78	5	0	300	10	0	8
8	80	70	-5	0	300	10	7	0
9	79	79	15	0	300	10	0	10
10	49	83	-15	300	600	10	9	0
11	48	79	5	300	600	10	0	12
12	35	81	-5	300	600	10	11	0
13	6	73	15	60	360	10	0	14
14	1	80	-15	60	360	10	13	0
15	12	70	5	60	360	10	0	16
16	6	46	-5	120	420	10	15	0
17	10	45	15	120	420	10	0	18
18	8	43	-15	120	420	10	17	0
19	6	11	10	0	600	10	0	20
20	10	5	-10	0	600	10	19	0
21	13	8	10	0	600	10	0	22
22	40	8	-10	180	480	10	21	0
23	35	8	10	180	480	10	0	24
24	48	13	-10	180	480	10	23	0
25	45	8	5	180	480	10	0	26
26	43	1	-5	180	480	10	25	0
27	40	13	5	180	480	10	0	28
28	48	8	-5	180	480	10	27	0
29	44	11	10	180	480	10	0	30
30	40	6	-10	180	480	10	29	0

PDT10 2

1	150	0						
0	50	50	0	0	600	0	0	0
1	88	12	15	0	300	10	0	2
2	74	43	-15	300	600	10	1	0
3	75	88	5	0	300	10	0	4
4	48	88	-5	300	600	10	3	0
5	16	81	5	60	360	10	0	6
6	5	35	-5	120	420	10	5	0
7	0	10	10	0	600	10	0	8
8	46	17	-10	180	480	10	7	0
9	35	16	10	180	480	10	0	10
10	50	4	-10	180	480	10	9	0

PDT16 2

1	150	0						
0	50	50	0	0	600	0	0	0
1	86	17	10	0	300	10	0	2
2	70	13	-10	0	300	10	1	0
3	86	50	15	300	600	10	0	4
4	70	45	-15	300	600	10	3	0
5	86	75	15	0	300	10	0	6
6	83	73	-15	0	300	10	5	0
7	54	88	15	300	600	10	0	8
8	45	81	-15	300	600	10	7	0
9	17	81	5	60	360	10	0	10
10	18	73	-5	60	360	10	9	0
11	17	40	5	120	420	10	0	12
12	3	45	-5	120	420	10	11	0
13	0	4	15	0	600	10	0	14
14	8	8	-15	0	600	10	13	0
15	43	17	10	180	480	10	0	16
16	51	6	-10	180	480	10	15	0

PDT20 2

1	150	0						
0	50	50	0	0	600	0	0	0
1	86	2	5	0	300	10	0	2
2	88	12	-5	0	300	10	1	0
3	83	41	15	300	600	10	0	4
4	70	35	-15	300	600	10	3	0
5	81	86	15	0	300	10	0	6
6	78	71	-15	0	300	10	5	0
7	48	75	15	300	600	10	0	8
8	53	73	-15	300	600	10	7	0
9	10	86	15	60	360	10	0	10
10	16	70	-15	60	360	10	9	0
11	18	53	10	120	420	10	0	12
12	8	50	-10	120	420	10	11	0
13	6	0	5	0	600	10	0	14
14	1	12	-5	0	600	10	13	0
15	46	0	15	180	480	10	0	16
16	45	2	-15	180	480	10	15	0
17	43	10	10	180	480	10	0	18
18	41	4	-10	180	480	10	17	0
19	35	15	5	180	480	10	0	20
20	49	5	-5	180	480	10	19	0

Annexes

PDT26 2

1	150	0							
0	50	50	0	0	600	0	0	0	
1	83	4	15	0	300	10	0	2	
2	88	6	-15	0	300	10	1	0	
3	88	4	5	0	300	10	0	4	
4	70	35	-5	300	600	10	3	0	
5	86	51	10	300	600	10	0	6	
6	83	46	-10	300	600	10	5	0	
7	81	75	15	0	300	10	0	8	
8	78	80	-15	0	300	10	7	0	
9	83	71	5	0	300	10	0	10	
10	37	81	-5	300	600	10	9	0	
11	35	81	15	300	600	10	0	12	
12	45	83	-15	300	600	10	11	0	
13	15	85	5	60	360	10	0	14	
14	2	73	-5	60	360	10	13	0	
15	13	74	15	60	360	10	0	16	
16	6	40	-15	120	420	10	15	0	
17	12	41	10	120	420	10	0	18	
18	11	43	-10	120	420	10	17	0	
19	12	18	10	0	600	10	0	20	
20	4	5	-10	0	600	10	19	0	
21	18	10	5	0	600	10	0	22	
22	49	0	-5	180	480	10	21	0	
23	37	13	15	180	480	10	0	24	
24	46	3	-15	180	480	10	23	0	
25	37	10	5	180	480	10	0	26	
26	46	2	-5	180	480	10	25	0	

PDT30 2

1	150	0							
0	50	50	0	0	600	0	0	0	
1	71	17	5	0	300	10	0	2	
2	83	8	-5	0	300	10	1	0	
3	78	0	15	0	300	10	0	4	
4	75	40	-15	300	600	10	3	0	
5	79	54	5	300	600	10	0	6	
6	86	37	-5	300	600	10	5	0	
7	81	81	5	0	300	10	0	8	
8	83	75	-5	0	300	10	7	0	
9	81	83	10	0	300	10	0	10	
10	45	79	-10	300	600	10	9	0	
11	45	70	5	300	600	10	0	12	
12	48	88	-5	300	600	10	11	0	
13	6	85	10	60	360	10	0	14	
14	0	80	-10	60	360	10	13	0	
15	13	85	5	60	360	10	0	16	
16	16	40	-5	120	420	10	15	0	
17	12	50	10	120	420	10	0	18	
18	18	48	-10	120	420	10	17	0	
19	11	18	15	0	600	10	0	20	
20	18	17	-15	0	600	10	19	0	
21	18	13	10	0	600	10	0	22	
22	44	6	-10	180	480	10	21	0	
23	46	16	5	180	480	10	0	24	
24	37	6	-5	180	480	10	23	0	
25	39	8	10	180	480	10	0	26	
26	45	2	-10	180	480	10	25	0	
27	35	6	15	180	480	10	0	28	
28	44	3	-15	180	480	10	27	0	
29	45	8	15	180	480	10	0	30	
30	39	17	-15	180	480	10	29	0	

ANNEXE 2 : Les paramètres de simulation

Les tableaux suivants contiennent les nombres d'itérations utilisés par chaque approche.

	AEs_1	AEHs_1		AEs_2	AEHs_2	
		AE	RT		AE	RT
P10	300	150	300	700	600	200
P16	1000	1000	300	1500	1300	300
P20	1000	1000	300	1500	1300	300
P26	1000	1000	300	1500	1300	300
P30	1500	1300	300	1500	1300	300

	AEs_1	AEHs_1		AEs_2	AEHs_2	
		AE	RT		AE	RT
PD10	500	150	300	1000	800	300
PD16	1000	1000	300	1500	1200	300
PD20	1000	1000	300	1500	1200	300
PD26	1000	1000	300	1500	1200	300
PD30	1500	1300	300	1500	1200	300

	AEs_1	AEHs_1		AEs_2	AEHs_2	
		AE	RT		AE	RT
PT10	500	300	300	1000	500	300
PT16	1000	1000	300	1500	1200	300
PT20	1000	1000	300	1500	1200	300
PT26	1000	1000	300	1500	1200	300
PT30	2000	1300	300	2000	1300	300

	AEs_1	AEHs_1		AEs_2	AEHs_2	
		AE	RT		AE	RT
PDT10	500	300	300	1000	500	300
PDT16	1000	1000	300	1500	500	300
PDT20	1000	1000	300	1500	500	300
PDT26	1000	1000	300	1500	500	300
PDT30	1500	1300	600	1500	1300	600

ANNEXE 3 : Résultats des simulations

Approche Problème	AEs 1					Solution
	T d	T w	T t	Fitness_glob	n_sol	
P10_1	34,806911	13,939849	0	48,74676	16	0-1-2-3-4-5-6-7-8-9-10-0
P10_2	38,0329	17,2692	0	55,3021	21	0-7-8-9-10-1-2-3-5-6-4-0
P16_1	43,263	17,4345	0	60,6975	25	0-3-5-6-4-1-2-15-16-13-14-11-12-9-10-7-8-0
P16_2	46,6906	18,6468	0	65,3374	24	0-13-14-11-12-9-10-7-8-5-6-15-16-1-2-3-4-0
P20_1	59,9669	9,66024	0	69,62714	45	0-13-14-1-2-5-6-7-9-11-10-12-19-15-17-20-18-16-3-4-8-0
P20_2	46,82917	19,44445	0	66,27362	51	0-13-14-11-12-9-10-7-8-5-6-3-4-1-2-15-19-20-17-16-18-0
P26_1	57,7953	18,883	1,01279	77,69109	77	0-11-7-8-9-12-10-15-13-14-16-17-18-23-25-21-19-20-22-26-24-1-2-3-5-6-4-0
P26_2	67,3976	4,79674	0	72,19434	53	0-3-1-2-9-7-8-13-14-15-17-16-18-19-21-20-25-26-22-23-24-4-5-6-10-11-12-0
P30_1	71,3236	1,75002	0	73,07362	22	0-7-8-9-3-1-2-13-14-15-17-18-16-29-27-25-30-23-24-26-28-19-20-21-22-5-6-4-10-11-12-0
P30_2	68,4094	2,048	0	70,4574	250	0-1-3-2-7-8-9-13-15-14-17-16-18-19-21-20-27-28-29-30-25-26-23-22-24-5-4-6-11-10-12-0
PD10_1	34,8956	9,4666	0	44,3622	21	0-3-4-5-6-7-9-10-8-1-2-0
PD10_2	36,913181	10,344572	0	47,257753	20	0-3-4-5-6-7-8-9-10-1-2-0
PD16_1	37,4971	14,5768	0	52,0739	25	0-7-9-10-11-12-13-14-15-16-1-2-3-4-5-6-8-0
PD16_2	36,722836	14,875284	0	51,59812	35	0-5-6-3-4-1-2-15-16-13-14-11-12-9-10-7-8-0
PD20_1	47,7288	4,68298	0	52,41178	119	0-5-6-9-10-11-12-13-14-15-1-2-17-16-19-20-18-3-4-7-8-0
PD20_2	40,2293	14,5169	0	54,7462	50	0-13-14-11-12-9-10-7-8-5-6-3-4-1-2-17-18-19-15-20-16-0
PD26_1	57,8437	2,2426	0	60,0863	40	0-13-15-14-7-9-8-3-1-2-25-23-19-21-20-16-17-18-24-22-26-4-5-6-11-10-12-0
PD26_2	57,518	6,30468	0	63,82268	64	0-13-14-15-7-9-8-3-1-2-4-17-18-16-21-19-20-25-26-23-24-22-5-6-10-11-12-0
PD30_1	58,1673	0,95962	0	59,12692	11	1-2-3-7-9-8-15-13-14-17-16-18-19-20-21-23-25-29-27-26-28-22-24-30-11-10-12-4-5-6-0
PD30_2	50,8968	2,61626	0,0597925	53,5728525	53	0-13-15-14-17-16-18-19-21-20-29-30-23-22-25-27-26-28-24-1-3-2-7-8-9-10-11-12-5-6-4-0
PT10_1	30,6322	16,5785	0	47,2107	25	0-9-1-2-3-4-5-6-7-10-8-0
PT10_2	36,720341	11,28221	0	48,002551	22	0-1-2-3-4-5-6-7-8-9-10-0
PT16_1	46,5289	15,0113	0	61,5402	40	0-1-2-3-5-7-6-4-15-16-13-14-11-12-9-10-8-0
PT16_2	51,0185	6,32588	0	57,34438	75	0-1-2-5-6-9-10-11-12-13-14-15-16-3-4-7-8-0
PT20_1	57,8425	7,7051	0	65,5476	41	0-5-6-1-2-13-14-9-10-11-12-17-15-16-19-20-18-3-4-7-8-0
PT20_2	58,2888	8,09256	0	66,38136	91	0-1-2-5-6-7-8-9-10-11-12-13-14-15-16-19-17-20-18-3-4-0
PT26_1	67,0728	0,199551	0	67,272351	80	0-3-1-2-9-7-8-15-13-14-17-18-16-19-20-21-25-22-26-23-24-10-11-12-5-4-6-0
PT26_2	63,9029	0,232257	0	64,135157	250	0-1-2-3-9-7-8-13-15-14-17-16-18-21-19-20-22-25-26-23-24-4-5-6-11-10-12-0
PT30_1	69,7633	2,69	0	72,4533	20	0-3-1-2-7-8-9-13-15-14-19-21-20-17-16-18-29-30-23-24-22-25-26-27-28-10-11-12-5-4-6-0
PT30_2	63,2147	0	0	63,2147	4	0-1-2-3-7-8-9-13-14-15-16-17-18-19-20-21-29-22-27-30-23-28-25-26-24-5-6-4-10-11-12-0
PDT10_1	33,119518	12,564718	0	45,684236	20	0-3-4-5-6-7-9-10-8-1-2-0
PDT10_2	33,604	12,7353	0	46,3393	24	0-3-4-5-6-7-9-8-10-1-2-0
PDT16_1	45,5314	10,1256	0	55,657	44	0-1-2-5-6-7-8-9-10-11-12-13-14-15-16-3-4-0
PDT16_2	42,6032	12,8869	0	55,4901	31	0-1-2-3-5-6-7-8-9-10-11-12-13-14-15-16-4-0
PDT20_1	45,1691	9,11427	0	54,28337	113	0-1-2-5-6-7-8-9-10-11-12-13-14-17-19-18-15-20-16-3-4-0
PDT20_2	53,8742	3,4376	0	57,3118	83	0-1-2-5-6-9-10-11-12-13-14-19-17-18-15-16-20-3-4-7-8-0
PDT26_1	56,2748	0,641923	0	56,916723	250	0-1-2-3-7-8-9-13-14-15-17-18-16-19-21-20-25-23-26-22-24-5-6-4-10-11-12-0
PDT26_2	54,9618	0,697847	0	55,659647	250	0-1-3-2-9-7-8-13-15-14-17-18-16-19-20-21-23-25-24-26-22-11-10-12-5-6-4-0
PDT30_1	56,6696	0,108431	0	56,778031	7	0-1-2-3-7-9-8-15-13-14-16-17-18-23-21-19-20-27-29-25-26-30-22-28-24-5-4-6-11-10-12-0
PDT30_2	56,1189	0	0	56,1189	3	0-1-3-2-7-9-8-15-13-14-17-18-16-19-20-21-27-25-28-26-22-29-23-24-30-11-10-12-5-6-4-0

Annexes

Approche Problème	AEHs_1					Solution
	T_d	T_w	T_t	Fitness_glob	n_sol	
P10_1	34,806911	13,939849	0	48,74676	15	0-1-2-3-4-5-6-7-8-9-10-0
P10_2	38,0329	17,2692	0	55,3021	25	0-7-8-9-10-1-2-3-5-6-4-0
P16_1	42,273739	17,483009	0	59,756748	28	0-3-5-6-4-1-2-15-16-13-14-11-12-9-10-7-8-0
P16_2	45,179363	18,697428	0	63,876791	28	0-5-6-9-7-8-10-11-12-13-14-15-16-1-2-3-4-0
P20_1	58,4081	9,38822	0	67,79632	35	0-5-6-1-2-13-14-11-9-10-7-12-19-15-17-20-18-16-3-4-8-0
P20_2	46,82917	19,44445	0	66,27362	42	0-13-14-11-12-9-10-7-8-5-6-3-4-1-2-15-19-20-17-16-18-0
P26_1	54,834553	17,389782	0,843605	73,06794	43	0-11-7-8-9-12-10-15-13-14-16-17-18-21-19-20-22-25-23-26-24-1-2-3-5-6-4-0
P26_2	67,397461	4,796754	0	72,194215	46	0-3-1-2-9-7-8-13-14-15-17-16-18-19-21-20-25-26-22-23-24-4-5-6-10-11-12-0
P30_1	64,911552	1,872551	0	66,784103	6	0-7-8-9-13-14-15-29-1-3-2-25-27-30-28-26-23-24-19-21-20-17-16-18-22-5-6-4-10-11-12-0
P30_2	73,902428	0,383478	0	74,285906	21	0-19-21-20-1-3-2-7-8-9-13-15-14-17-16-18-22-27-28-29-30-25-26-23-24-11-12-10-5-6-4-0
PD10_1	34,8956	9,4666	0	44,3622	21	0-3-4-5-6-7-9-10-8-1-2-0
PD10_2	36,913181	10,344572	0	47,257753	21	0-3-4-5-6-7-8-9-10-1-2-0
PD16_1	39,1896	13,8337	0	53,0233	31	0-1-2-15-16-13-14-11-12-9-10-7-8-5-6-3-4-0
PD16_2	36,722836	14,875284	0	51,59812	33	0-5-6-3-4-1-2-15-16-13-14-11-12-9-10-7-8-0
PD20_1	43,074326	5,917346	0	48,991672	104	0-5-6-9-10-11-12-13-14-19-20-15-16-17-18-1-2-3-4-7-8-0
PD20_2	37,214447	14,879236	0	52,093683	51	0-5-6-3-4-1-2-17-18-19-20-15-16-13-14-11-12-9-10-7-8-0
PD26_1	57,8625	57,8625	0	115,725	66	0-3-1-2-7-9-8-13-14-15-17-18-16-19-21-20-22-23-25-24-26-4-5-6-11-10-12-0
PD26_2	50,3135	3,70481	3,70481	57,72312	96	0-3-1-2-7-8-9-13-14-15-17-18-16-21-19-20-25-26-23-24-22-5-6-4-10-11-12-0
PD30_1	58,1673	0,95962	0	59,12692	8	1-2-3-7-9-8-15-13-14-17-16-18-19-20-21-23-25-29-27-26-28-22-24-30-11-10-12-4-5-6-0
PD30_2	60,4861	0,815444	0	61,301544	36	0-9-7-8-1-3-2-23-24-29-30-17-15-13-14-16-18-19-21-20-22-25-27-26-28-11-12-10-5-6-4-0
PT10_1	30,6322	16,5785	0	47,2107	31	0-9-1-2-3-4-5-6-7-10-8-0
PT10_2	36,720341	11,28221	0	48,002551	24	0-1-2-3-4-5-6-7-8-9-10-0
PT16_1	46,7819	12,615	0	59,3969	43	0-15-1-2-16-13-14-11-12-9-10-5-3-4-6-7-8-0
PT16_2	38,9766	14,6183	0	53,5949	91	0-9-10-11-12-13-14-15-16-1-2-3-4-5-6-7-8-0
PT20_1	58,1837	7,08348	0	65,26718	36	0-5-6-1-2-13-14-9-10-11-12-17-18-19-15-16-20-3-4-7-8-0
PT20_2	58,2888	8,09256	0	66,38136	93	0-1-2-5-6-7-8-9-10-11-12-13-14-15-16-19-17-20-18-3-4-0
PT26_1	62,3633	0,434377	0	62,797677	250	0-3-1-2-9-7-8-15-13-14-17-18-16-19-20-21-23-24-25-22-26-5-4-6-11-10-12-0
PT26_2	64,0765	0,224804	0	64,301304	78	0-1-3-2-9-7-8-13-15-14-17-16-18-21-19-20-22-25-26-23-24-4-5-6-11-10-12-0
PT30_1	69,7633	1,87708	0	71,64038	19	0-3-1-2-7-8-9-13-15-14-19-21-20-17-16-18-29-30-23-24-22-25-26-27-28-10-11-12-5-4-6-0
PT30_2	63,141438	0	0	63,141438	3	0-1-2-3-7-8-9-13-14-15-16-17-18-19-21-20-29-22-27-30-23-24-25-26-28-5-6-4-10-11-12-0
PDT10_1	33,119518	12,564718	0	45,684236	20	0-3-4-5-6-7-9-10-8-1-2-0
PDT10_2	33,604	12,7353	0	46,3393	28	0-3-4-5-6-7-9-8-10-1-2-0
PDT16_1	43,4867	10,2511	0	53,7378	33	0-5-6-1-2-15-16-13-14-11-12-9-10-7-8-3-4-0
PDT16_2	42,6032	12,8869	0	55,4901	38	0-1-2-3-5-6-7-8-9-10-11-12-13-14-15-16-4-0
PDT20_1	45,1691	9,11427	0	54,28337	82	0-1-2-5-6-7-8-9-10-11-12-13-14-17-19-18-15-20-16-3-4-0
PDT20_2	53,8742	3,4376	0	57,3118	119	0-1-2-5-6-9-10-11-12-13-14-19-17-18-15-16-20-3-4-7-8-0
PDT26_1	55,834892	0,674966	0	56,509858	91	0-1-2-3-7-8-9-13-14-15-17-18-16-19-21-20-23-25-26-22-24-5-6-4-10-11-12-0
PDT26_2	52,9605	0,9416	0	53,9021	250	0-1-3-2-9-7-8-13-15-14-17-18-16-19-20-21-23-25-24-26-22-4-5-6-11-10-12-0
PDT30_1	53,1026	0	0	53,1026	5	0-1-2-3-7-9-8-15-13-14-17-16-18-19-20-21-23-27-29-28-25-26-30-22-24-5-4-6-11-10-12-0
PDT30_2	56,0935	0	0	56,0935	3	0-1-3-2-7-9-8-15-13-14-17-18-16-19-20-21-27-25-23-29-26-28-22-24-30-11-10-12-5-6-4-0

Annexes

Approche	AEs 2										
	Problème	T_d	T_d homogén	T_t	T_t homogén	T_w	b_inf_dist	b_inf_ret	Fitness_glob	Fitness_glob_homogénéisée	n_sol
P10_1	34,807	0,572656	0	1	13,9398	32,55	0	48,7468	0,786328	10	0-1-2-3-4-5-6-7-8-9-10-0
P10_2	38,0329	0,559339	0	1	17,2692	32,29	0	55,3021	0,7796695	15	0-7-8-9-10-1-2-3-5-6-4-0
P16_1	41,2688	0,569844	0	1	20,8936	32,8	0	62,1624	0,784922	24	0-3-4-5-6-7-8-9-10-11-12-13-14-15-16-1-2-0
P16_2	43,5291	0,567044	0	1	20,6176	33,9	0	64,1467	0,783522	33	0-13-14-15-16-1-2-3-11-12-9-7-8-10-5-6-4-0
P20_1	48,5994	0,570217	0	1	20,998	38,96	0	69,5974	0,7851085	48	0-5-6-3-4-1-2-19-15-17-16-18-20-13-14-11-9-12-10-7-8-0
P20_2	49,1701	0,614854	0	1	18,791	39,26	0	67,9611	0,807427	53	0-13-14-9-11-12-10-7-8-5-6-3-4-1-2-15-19-20-17-16-18-0
P26_1	59,2185	0,587091	0	1	16,8284	42,03	0	76,0469	0,7935455	65	0-11-7-8-9-12-10-15-13-14-16-17-18-21-19-20-22-1-3-2-23-25-26-24-4-5-6-0
P26_2	64,5705	0,605828	0	1	22,9858	42,69	0	87,5563	0,802914	41	0-3-1-2-23-11-12-15-17-16-18-13-14-7-8-9-10-5-4-6-19-21-20-25-26-22-24-0
P30_1	66,4309	0,596078	0	1	3,51714	44,6	0	69,94804	0,798039	36	0-7-8-9-3-1-2-29-17-15-13-14-18-16-19-20-21-23-24-22-30-27-25-28-26-5-6-4-10-11-12-0
P30_2	65,1527	0,498106	0	1	19,2817	43,05	0	84,4344	0,749053		0-19-20-21-23-24-22-27-25-26-28-29-30-1-3-2-4-7-8-9-10-13-15-14-11-12-17-18-16-5-6-0
PD10_1	34,8956	0,542653	0	1	9,4666	34,3	0	44,3622	0,7713265	19	0-3-4-5-6-7-9-10-8-1-2-0
PD10_2	35,6353	0,569211	0	1	11,8387	32,59	0	47,474	0,7846055	16	0-5-6-7-8-9-10-1-2-3-4-0
PD16_1	37,2011	0,523545	0	1	17,9056	33,46	0	55,1067	0,7617725	19	0-3-4-1-2-15-16-13-14-11-12-9-10-7-8-5-6-0
PD16_2	36,4912	0,524884	0	1	18,0256	34,6	0	54,5168	0,762442	27	0-3-4-1-2-15-16-13-14-11-12-9-10-7-8-5-6-0
PD20_1	42,6031	0,562168	0	1	15,2905	34,88	0	57,8936	0,781084	250	0-1-2-3-4-5-6-7-8-9-10-11-12-13-14-19-20-15-16-17-18-0
PD20_2	36,9418	0,515885	0	1	15,2504	34,16	0	52,1922	0,7579425	49	0-3-5-6-4-1-2-17-18-19-15-20-16-13-14-11-12-9-10-7-8-0
PD26_1	49,9146	0,568784	0	1	15,6865	34,45	0	65,6011	0,784392	38	0-5-7-9-8-11-12-10-13-14-15-17-18-16-19-21-20-22-23-1-3-2-25-24-26-4-6-0
PD26_2	45,2476	0,520345	0	1	15,0776	26,66	0	60,3252	0,7601725	250	0-7-8-9-10-11-12-13-14-15-17-18-16-21-19-20-23-1-2-3-24-25-26-22-5-4-6-0
PD30_1	48,3678	0,547098	0	1	23,1506	32,26	0	71,5184	0,773549	30	0-5-6-1-2-3-4-7-9-8-11-12-10-15-13-14-17-16-18-19-20-21-23-25-29-30-27-26-28-22-24-0
PD30_2	48,5981	0,684759	0	1	19,3377	30,51	0	67,9358	0,8423795	107	0-9-7-8-5-6-3-1-2-4-10-11-12-13-15-14-17-16-18-19-21-20-27-25-28-23-26-24-22-29-30-0
PT10_1	30,6322	0,563936	0	1	16,5785	26,2	0	47,2107	0,781968	28	0-9-1-2-3-4-5-6-7-10-8-0
PT10_2	36,7203 41	0,526844	0	1	11,28221	32,35	0	48,002551	0,763422	18	0-1-2-3-4-5-6-7-8-9-10-0
PT16_1	47,5308	0,553175	0	1	17,3539	37,33	0	64,8847	0,7765875	27	0-15-1-2-3-4-5-6-7-8-9-10-11-12-16-13-14-0
PT16_2	38,9766	0,594917	0	1	14,6183	37,33	0	53,5949	0,7974585	21	0-9-10-11-12-13-14-15-16-1-2-3-4-5-6-7-8-0
PT20_1	48,1978	0,519506	0	1	23,0137	40,88	0	71,2115	0,759753	35	0-3-4-5-6-7-8-9-10-11-12-13-14-17-18-19-15-16-20-1-2-0
PT20_2	54,3099	0,543177	0	1	14,8839	43,93	0	69,1938	0,7715885	31	0-3-5-6-4-1-2-15-16-19-17-20-18-13-14-11-12-9-10-7-8-0
PT26_1	62,3633	0,634851	0	1	0,434377	41,6	0	62,797677	0,8174255	250	0-3-1-2-9-7-8-15-13-14-17-18-16-19-20-21-23-24-25-22-26-5-4-6-11-10-12-0
PT26_2	59,8632	0,625654	0	1	10,0497	40,98	0	69,9129	0,812827	250	0-1-3-2-23-4-5-9-7-8-13-15-14-17-16-18-21-19-20-22-25-26-24-6-11-10-12-0
PT30_1	57,9188	0,670006	0	1	25,2554	42	0	83,1742	0,835003	172	0-5-25-3-1-2-6-4-7-8-9-11-12-10-13-15-14-16-17-18-19-21-20-29-30-23-22-24-26-27-28-0
PT30_2	55,3988	0,6452	0	1	19,5886	41,65	0	74,9874	0,8226	30	0-1-2-3-5-6-4-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-29-22-27-30-23-24-25-26-28-0
PDT10_1	33,1196	0,523471	0	1	12,5647	29,7	0	45,6843	0,7617355	22	0-3-4-5-6-7-9-10-8-1-2-0
PDT10_2	33,604	0,620515	0	1	12,7353	36,2	0	46,3393	0,8102575	23	0-3-4-5-6-7-9-8-10-1-2-0
PDT16_1	40,8465	0,521513	0	1	19,19	33,7	0	60,0365	0,7607565	25	0-7-8-5-6-3-4-1-2-15-16-13-14-11-12-9-10-0
PDT16_2	41,3397	0,522848	0	1	17,5593	36,13	0	58,899	0,761424	29	0-3-4-5-6-7-8-9-10-11-12-13-14-15-16-1-2-0
PDT20_1	41,1839	0,53478	0	1	15,4688	35,36	0	56,6527	0,76739	111	0-1-2-3-4-5-6-7-8-9-10-11-12-13-14-17-19-18-15-20-16-0
PDT20_2	44,6747	0,533132	0	1	20,9284	37,91	0	65,6031	0,766566	131	0-7-8-5-6-3-4-1-2-19-17-20-15-16-18-13-14-11-12-9-10-0
PDT26_1	55,429	0,639321	0	1	3,77198	34,08	0	59,20098	0,8196605	250	0-1-2-3-7-8-9-11-13-14-15-17-18-16-19-21-20-23-22-25-26-24-5-6-4-10-12-0
PDT26_2	45,0328	0,572709	0	1	20,1814	30,12	0	65,2142	0,7863545	250	0-1-3-2-4-5-6-9-7-8-11-12-10-13-15-14-17-18-16-19-20-21-23-25-24-26-22-0
PDT30_1	46,1708	0,541475	0	1	18,8938	28,35	0	65,0646	0,7707375	96	0-7-9-8-5-1-2-3-6-4-11-10-12-15-13-14-16-17-18-19-20-21-23-22-27-29-25-30-26-28-24-0
PDT30_2	49,8453	0,676497	0	1	13,4804	31,8	0	63,3257	0,8382485	38	0-1-3-2-5-7-9-8-11-10-12-15-13-14-17-18-16-19-20-21-27-25-22-28-26-29-23-24-30-6-4-0

Annexes

Approche	AEHs 2										
	Problème	T_d	T_d homogén	T_t	T_t homogén	T_w	b_inf_dist	b_inf_ret	Fitness_glob	Fitness_glob homogénéisée	n_sol
P10_1	34,807	0,595112	0	1	13,9398	32,55	0	48,7468	0,797556	8	0-1-2-3-4-5-6-7-8-9-10-0
P10_2	39,1208	0,545565	0	1	17,0448	32,29	0	56,1656	0,7727825	21	0-1-2-3-4-5-6-7-8-9-10-0
P16_1	41,2688	0,569844	0	1	20,8936	32,8	0	62,1624	0,784922	28	0-3-4-5-6-7-8-9-10-11-12-13-14-15-16-1-2-0
P16_2	41,814606	0,537681	0	1	21,074863	33,9	0	62,889469	0,7688405		0-5-7-9-8-10-11-12-13-14-15-16-1-2-3-4-6-0
P20_1	44,6651	0,559663	0,01168	0,9785	24,052122	38,96	0	68,728904	0,7690815	34	0-19-15-17-18-1-2-16-20-13-14-11-9-12-10-7-8-5-6-3-4-0
P20_2	52,5473	0,586104	0	1	17,7239	39,26	0	70,2712	0,793052	48	0-9-10-11-12-13-14-19-15-1-2-16-17-20-18-3-4-5-7-8-6-0
P26_1	65,782959	0,55059	0	1	7,320022	42,03	0	73,102981	0,775295	65	0-1-2-3-7-8-9-10-15-13-14-16-17-18-19-20-21-25-23-26-24-22-11-12-5-6-4-0
P26_2	54,3371	0,672721	0	1	28,8503	42,69	0	83,1874	0,8363605	103	0-23-1-2-3-4-5-6-7-8-9-10-11-12-17-18-13-14-15-16-19-21-20-25-26-22-24-0
P30_1	64,00251	0,658671	0	1	9,110229	44,6	0	73,112739	0,8293355		0-15-13-14-9-7-8-5-6-3-1-2-29-17-16-18-19-20-21-23-24-27-25-26-28-30-22-4-10-11-12-0
P30_2	65,1202	0,498374	0	1	19,3913	43,05	0	84,5115	0,749187	120	0-19-20-21-23-27-24-22-25-26-28-29-30-1-3-2-4-7-8-9-10-13-15-14-11-12-17-18-16-5-6-0
PD10_1	34,8956	0,542653	0	1	9,4666	34,3	0	44,3622	0,7713265	19	0-3-4-5-6-7-9-10-8-1-2-0
PD10_2	36,913235	0,625762	0	1	10,344564	32,59	0	47,257799	0,812881	21	0-3-4-5-6-7-8-9-10-1-2-0
PD16_1	37,497108	0,523545	0	1	14,576831	33,46	0	52,073939	0,7617725	25	0-7-9-10-11-12-13-14-15-16-1-2-3-4-5-6-8-0
PD16_2	36,722919	0,524884	0	1	14,875277	34,6	0	51,598196	0,762442	24	0-5-6-3-4-1-2-15-16-13-14-11-12-9-10-7-8-0
PD20_1	39,0838	0,601561	0	1	11,4133	34,88	0	50,4971	0,8007805	117	0-5-6-7-8-9-10-11-12-13-14-19-20-17-18-15-16-1-2-3-4-0
PD20_2	37,191856	0,512275	0	1	14,880527	34,16	0	52,072383	0,7561375	53	0-5-6-3-4-1-2-17-18-19-15-20-16-13-14-11-12-9-10-7-8-0
PD26_1	47,1924	0,594044	0	1	11,6624	34,45	0	58,8548	0,797022	36	7-9-8-11-12-10-13-14-15-17-18-16-19-21-20-22-23-25-24-26-1-2-3-4-5-6-0
PD26_2	42,1017	0,555781	0	1	23,2374	26,66	0	65,3391	0,7778905		0-3-1-2-5-6-4-7-8-9-10-11-12-13-14-15-17-18-16-21-19-20-25-26-23-24-22-0
PD30_1	48,3678	0,547098	0	1	23,1506	32,26	0	71,5184	0,773549	30	0-5-6-1-2-3-4-7-9-8-11-12-10-15-13-14-17-16-18-19-20-21-23-25-29-30-27-26-28-22-24-0
PD30_2	48,2164	0,687401	0	1	19,3377	30,51	0	67,5541	0,8437005	32	0-9-7-8-5-6-3-1-2-4-10-11-12-13-15-14-16-17-18-19-21-20-27-25-22-29-30-23-26-24-28-0
PT10_1	30,6322	0,563936	0	1	16,5785	26,2	0	47,2107	0,781968	28	0-9-1-2-3-4-5-6-7-10-8-0
PT10_2	36,720341	0,526844	0	1	11,28221	32,35	0	48,002551	0,763422	15	0-1-2-3-4-5-6-7-8-9-10-0
PT16_1	45,3861	0,575054	0	1	12,9112	37,33	0	58,2973	0,787527	26	0-1-2-15-16-13-14-11-12-9-10-7-5-3-4-6-8-0
PT16_2	40,4038	0,578512	0	1	14,5824	37,33	0	54,9862	0,789256	21	0-11-12-9-10-13-14-15-16-1-2-3-4-5-6-7-8-0
PT20_1	47,9821	0,521856	0	1	19,3363	40,88	0	67,3184	0,760928	30	0-1-2-3-4-5-6-7-8-9-10-11-12-13-14-17-15-16-19-20-18-0
PT20_2	54,4903	0,541526	0	1	14,6557	43,93	0	69,146	0,770763	31	0-3-5-6-4-1-2-15-19-17-18-20-16-13-14-11-12-9-10-7-8-0
PT26_1	57,1498	0,68054	0	1	6,73343	41,6	0	63,88323	0,84027	250	0-3-1-2-9-7-8-11-10-15-13-14-17-18-16-19-20-21-23-24-25-22-26-5-4-6-12-0
PT26_2	59,8632	0,625654	0	1	10,4097	40,98	0	69,9129	0,812827	250	0-1-3-2-23-4-5-9-7-8-13-15-14-17-16-18-21-19-20-22-25-26-24-6-11-10-12-0
PT30_1	64,7497	0,628043	0	1	13,3041	42	0	78,0538	0,8140215	250	0-25-3-1-2-5-9-7-8-11-13-15-14-17-18-16-19-21-20-29-23-30-27-26-24-28-22-6-4-12-10-0
PT30_2	59,6625	0,615673	0	1	13,8089	41,65	0	73,4714	0,8078365	7	0-1-2-3-5-6-7-8-9-11-12-13-14-15-16-17-18-19-20-21-29-22-27-30-23-24-25-26-28-4-10-0
PDT10_1	33,1196	0,523471	0	1	12,5647	29,7	0	45,6843	0,7617355	17	0-3-4-5-6-7-9-10-8-1-2-0
PDT10_2	36,6878	0,551258	0	1	11,3805	36,2	0	48,0683	0,775629	22	0-1-2-3-4-5-6-7-9-10-8-0
PDT16_1	39,5108	0,538567	0	1	14,9579	33,7	0	54,4687	0,7692835	20	0-3-5-6-4-1-2-15-16-13-14-11-12-9-10-7-8-0
PDT16_2	41,7022	0,518263	0	1	12,9522	36,13	0	54,6544	0,7591315	21	0-1-2-15-16-13-14-11-12-9-10-7-8-5-6-3-4-0
PDT20_1	41,1839	0,53478	0	1	15,4688	35,36	0	56,6527	0,76739	109	0-1-2-3-4-5-6-7-8-9-10-11-12-13-14-17-19-18-15-20-16-0
PDT20_2	53,8742	0,42808	0	1	3,4376	37,91	0	57,3118	0,71404	95	0-1-2-5-6-9-10-11-12-13-14-19-17-18-15-16-20-3-4-7-8-0
PDT26_1	54,9706	0,642498	0	1	3,79817	34,08	0	58,76877	0,821249	250	0-1-2-3-7-8-9-11-13-14-15-17-18-16-19-21-20-23-25-26-22-24-5-6-4-10-12-0
PDT26_2	44,9788	0,573258	0	1	20,2398	30,12	0	65,2186	0,786629	51	0-1-3-2-4-5-6-9-7-8-10-11-12-13-15-14-17-18-16-19-20-21-23-25-24-26-22-0
PDT30_1	43,0055	0,574966	0	1	20,4968	28,35	0	63,5023	0,787483	56	0-1-2-3-5-4-6-7-9-8-10-11-12-15-13-14-16-17-18-19-20-21-23-22-27-29-25-30-26-28-24-0
PDT30_2	49,5751	0,679116	0	1	13,4804	31,8	0	63,0555	0,839558	40	0-1-3-2-5-7-9-8-11-10-12-15-13-14-17-18-16-19-20-21-27-25-22-29-23-30-24-28-26-6-4-0

Bibliographie Générale

- [Archetti et al. 2002] C. Archetti, A. Hertz and U. Derigs “A Tabu Search Algorithm for the Split Delivery Vehicle Routing Problem”. Mathematics of Information Technology and Complex Systems, 2002
- [Bianchi 2000] L. Bianchi. “Notes on Dynamic Vehicle Routing- The State of the Art”. Technical Report . IDSIA-05-01, 20 December 2000.
- [Bodin et al. 1981] L. Bodin and B. Golden. “Classification in vehicle routing and scheduling Networks”, 11:97-108, 1981.
- [Bodin et al.1979] L. Bodin and L. Berman. “Routing and scheduling of school buses by computer”. Transportation Science 13(2), pages 113-129, 1979.
- [Borne 1998] P. Borne, J. Rozinoer, J.Y. Dieulot, L. Dubois, “Introduction à la commande floue ”, Edition Technip, France, 1998.
- [Borne et al.1990] P. Borne, G.D. Tanguy, J.P. Richard, F. Rotella, I. Zambettakis, “Commande et optimisation de processus”, Edition Technip, France, 1990.
- [Bruggen et al.1993] V. D. Bruggen, J.K Lenstra., P.C Schuur., “Variable-depth search for the single-vehicle pickup and delivery problem with time windows”, Transportation Science 27 (3), 298-311, 1993.
- [Chérif 2002] S. Chérif, “Problèmes d’ordonnancement avec minimisation des encours”, Thèse soutenue le 5 Février 2002
- [Chu 1990] C. Chu, “Nouvelles approches analytiques et concept de mémoire artificielle pour divers problèmes d’ordonnancement”,

Thèse en automatique à l'université de Metz. Soutenue le 25 Septembre 1990.

- [Clarcke et al. 1964] G. Clarcke and J.W. Wright. "Scheduling of vehicles from a central depot to a number of delivery points". *Operations Research* 12, pages 568-581, 1964
- [Coello 2001] C.A. Coello , "A Short Tutorial on Evolutionary Multiobjective Optimization". *Proceeding of the 1st International Conference on Evolutionary Multi-Criterion Optimization, EMO 2001, Lecture Notes in Computer Science, vol.1993, pp. 21-40, Springer, Zurich, Switzerland, 2001*
- [Colomi et al.1992] A. Colomi, M. Dorigo, and V. Maniezzo. "Distributed optimization by ant colonies". *Proceeding of the first European Conference on Artificial Life (ECAL 91), pages p. 134-142, 1992.*
- [Cordeau et al. 1997] J.-F Cordeau, M. Gendreau and G. Laporte. "A tabu search heuristic for periodic and multi-depot vehicle routing problems". *Networks*, 30, pages pp. 105-119, 1997.
- [Cordeau et al. 2000] J.-F Cordeau, G. Desaulniers, J. Desrosiers, M. Solomon, and F. Soumis. "The VRP with Time Windows". *Technical Report, Département de Mathématiques et de Génie Industriel, Ecole polytechnique de Montréal, Canada, June 2000.*
- [Cordeau et al. 2003] J.-F Cordeau, G. Laporte, "The dial-a-ride problem (DARP) :Variants, modeling issues and algorithms". *4OR Quarterly Journal of the Belgian French and Italian Operations Research Societies* 1, 89–101, 2003.
- [Cordeau et al. 2004] J.-F Cordeau, M.Gendreau, A. Hertz, G. Laporte, and J.-S Sormany. "New Heuristics for the vehicles Routing Problem". *Technical Report, G-2004-33, Université de Montréal, Canada, April 2004.*

- [Dantzig et al. 1959] G.B. Dantzig and J.H. Ramser. "The Truck dispatching problem". Lehrstuhl für Wirtschaftsinformatik und Operations Research, Universität zu Köln, Mgmt Sci, 6(1) : 80-91, 1959.
- [Desrochers et al. 1990] M. Desrochers, J. K. Lenstra, and M. W. P. Savelsbergh. "A classification scheme for vehicle routing and scheduling problems". European Journal of Operational Research, 46:322-332, 1990.
- [Desrochers et al. 1991] M. Desrochers, G. Laporte, "Improvements and extensions to the Miller-Tucker-Zemlin subtour elimination constraints". Operations Research Letters, 10, 27-36, 1991.
- [Dréo et al. 2006] J. Dréo, A. Pétrowski, P. Siarry, É. D. Taillard, "Metaheuristics for Hard Optimization: Methods and Case Studies", 3-540-23022-X, Springer, 2006.
- [Dumas et al. 1986] Y. Dumas, J. Desrosiers, J. Soumis, "A dynamic programming solution of the large-scale single vehicle dial-ride problem with time windows". American Journal of Mathematical and Management Science 16, 301–325, 1986.
- [Dumas et al. 1991] Y. Dumas, J. Desrosiers, F. Soumis, "The pickup and delivery problem with time windows". European Journal of Operational Research, 54, 7-22, 1991.
- [Esquirol 2001] P. Esquirol, P. Lopez, Concepts et méthode de base en ordonnancement de la production. Chapitre dans l'ouvrage : Ordonnancement de la production, Edition Hermès, France, 2001
- [Fischetti et al. 1999] M. Fischetti, A. Lodi and P. Toth "A Branch-and-Cut Algorithm for the Multiple Depot Vehicle Scheduling Problem". Dipartimento di Elettronica e Informatica, Università di Padova, Italy, 1999.
- [Fu et al. 2003] Z. Fu, R. W. Eglese and L. Li "A tabu search heuristic for the open vehicle routing problem". LUMS working paper, 2003.

- [Gambardella et al. 2003] L.M. Gambardella, A.E. Rizzoli, F. Oliveriob, N. Casagrande, A.V. Donati, R. Montemanni and E. Lucibello. “Ant Colony Optimization for vehicle routing in advanced logistics systems” IDSIA, Galleria 2, 6928 Manno, Switzerland and AntOptima, via Fusion 4, 6900 Lugano, Switzerland, 2003.
- [Gendreau et al. 1994] M. Gendreau, A. Hertz, G. Laporte “A tabu search heuristic for the vehicle routing problem”, *Manag. Sci.*, vol. 40, pp. 1276-1290, 1994.
- [Gendreau et al 1998a] M. Gendreau and J.Y. Potvin. “Dynamic vehicle routing and dispatching” . Working paper, Université de Montréal, 1998
- [Gendreau et al. 1998b] M. Gendreau, F. Guertin, J. Y. Potvin and R. Séguin. “Neighbourhood search heuristics for a dynamic vehicle dispatching problem with pick-ups and deliveries”. Technical Report, CRT-98-10-Centre de recherche sur les Transports, Université de Montréal, 1998.
- [Gendreau et al. 1999] M. Gendreau, F. Guertin, J. Y. Potvin and E. Taillard. “Parallel Tabu Search for Real-Time Vehicle Routing and Dispatching”. *Transportation Science*, 1999.
- [Gendreau et al. 2001] M. Gendreau, G. Laporte, and F. Semet. “A dynamic model and parallel tabu search heuristic for real-time ambulance relocation”. *Parallel Computing* 27 1641-1653, 2001.
- [Glover 1989] F. Glover, “Tabu Search, Part I”, *ORSA Journal on Computing* 1(3), pp. 190-206.
- [Glover 1990] F. Glover, “Tabu Search, Part II”, *ORSA Journal on Computing* 2(1), pp. 4-32.
- [Goldberg 1989] D.E. Goldberg, “Genetic algorithms in search, optimization, and machine learning”, Advison-Wesley, 1989.
- [Golden et al. 1982] B. Goldberg, A. Assad, A. Levy and F. Gheysens “The fleet size and mix vehicle routing problem”, *Management Science and Statistics*. Working paper, n° 82-020, 1982.

- [Groth 2002] M. J. Groth “Stochastic considerations in vehicle routing Problems”. *Transport Optimization*, June 17 2002.
- [Holland 1975] J. Holland. “Adaptation in natural and artificial systems”. University of Michigan Press, Ann Arbor, Canberra ACT 2601. Australia, 1975.
- [Housroum 2005] H. Housroum. “Une approche génétique pour la résolution du problème VRPTW dynamique”. Thèse en génie informatique et automatique à l’université d’Artois. Soutenue le 3 Mai 2005.
- [Housroum et al. 2003] H. Housroum and H. Goncalves, R. Dupas and T. Hsu. “Une approche évolutionniste du problème de la gestion dynamique de tournées de véhicules avec fenêtres de temps”. *CPI’03*, Meknès, Maroc, Octobre 2003.
- [Housroum et al. 2004] H. Housroum and H. Goncalves, R. Dupas and T. Hsu. “An hybrid GA approach for solving the dynamic vehicle routing problem with time windows”. *Francoro IV*, Fribourg, Suisse, 2004.
- [Jacobs-Blecha et al. 1998] C. Jacobs-Blecha and M. Goetschalckx. “The vehicle Routing Problem with Backhauls : Properties and Solution Algorithms”. School of Industrial and systems engineering, 1992-1998.
- [Kammarti et al. 2004] R. Kammarti, S. Hammadi, P. Borne, M. Ksouri, “A new hybrid evolutionary approach for the pickup and delivery problem with time windows”. *IEEE International Conference on Systems, Man and Cybernetic*. 2004. Volume 2, P 1498-1503, Oct 2004
- [Kammarti et al. 2005a] R. Kammarti, S. Hammadi, P. Borne, M. Ksouri, “Improved tabu search in an hybrid evolutionary approach for the pickup and delivery problem with time windows”, *Intelligent Transportation Systems*, 2005. *Proceeding 2005 IEEE*, p 148-153, 2005.
- [Kammarti et al. 2005b] R. Kammarti, S. Hammadi, P. Borne, M. Ksouri, “Lower Bounds In An Hybrid Evolutionary Approach For The Pickup And Delivery Problem With Time Windows”. *IEEE*

- International Conference on Systems, Man and Cybernetics. 2005. Volume 2, P 1156-1161, Oct 2005.
- [Kammarti et al. 2006] R. Kammarti, S. Hammadi, P. Borne, M. Ksouri, "Solving the Real Time dynamic Pickup and Delivery Problem with an hybrid evolutionary approach". Multiconference on Computational Engineering in Systems Application. Volume 2, P 1520-1525, Oct 2005.
- [Kilby et al. 1998] P. Kilby, P. Prosser, and P. Shaw. "Guided Local Search for the Vehicle Routing Problem with Time Windows". META-HEURISTICS Advanced and Trends in Local Search Paradigms for Optimization, Kluwer Academic Publisher, Bosten, 1999.
- [Krumke et al. 2001] S.O. Krumke, W.E de Paepe, D. Poensgen and L. Stougie. "News from the Online Travelling Repairman". Lecture Notes in Computer Science, 2001.
- [Landrieu et al. 2001] A. Landrieu, Y. Mati, Z. Binder, "A tabu search heuristic for the single vehicle pickup and delivery problem with time windows", Journal of Intelligent Manufacturing, 12:497 - 508, 2001.
- [Laporte et al. 1992] G. Laporte. "The vehicle routing problem: An overview of exact and approximate algorithms". European Journal of Operational Research, 59:345-358, 1992.
- [Larsen 2001] A. Larsen. "The Dynamic Vehicle Routing Problem". PhD thesis, IMM-Danemark, 2001.
- [Larsen et al. 2002] A. Larsen, O. Madsen, and M. Solomon. "The A-priori Dynamic Travelling salesman Problem with Time Window". IMM, October 2002
- [Lau et al. 2001] H.C. Lau, Z. Liang, "Pickup and Delivery with Time Windows : Algorithms and Test Case Generation", In Proceedings of the 13th International Conference on Tools with Artificial Intelligence, pages 333 - 340. IEEE, 2001.

- [LeBouthillier 2000] A. LeBouthillier. “Modélisation UML pour une architecture coopérative appliquée au problème de tournées de véhicules avec fenêtres de temps”. Technical Report, Département d’informatique et de recherche opérationnelle. Faculté des arts et des sciences. Université de Montréal, Canada, Avril 2000.
- [Li et al. 2001] H. Li, A. Lim, “A metaheuristic for the pickup and delivery problem with time windows”, In IEEE International Conference on Tools with Artificial Intelligence, volume 13, pages 160–167, 2001.
- [Lim et al. 2002] A. Lim, H. Li, Rodrigues B., “Solving the pickup and delivery problem with time windows using squeaky wheel optimization with local search”, In Proceedings of 2002 Americas Conference on Information Systems, 2002.
- [Lin 1965] S. Lin, “Computer solutions of the travelling salesman problem”. Bell System Technical Journal 44, page pp 2245-2269, 1965.
- [Lund et al. 1996] K. Lund, Oli B.G. Madsen, and J. M. Rygaard, “Vehicle Routing Problems with Varying Degrees of Dynamism”. Technical Report, IMM. The Department of Mathematical Modeling. Technical University of Denmark, 1996.
- [Mechti 1995] R. Mechti, “Tournée de véhicules à la demande : problèmes et méthodes”. Technical report, Laboratoire PRiSM, Université de Versaille-St. Quentin, France, 1995.
- [Mesghouni 1999] K. Mesghouni, “Application des algorithmes évolutionnistes dans les problèmes d’optimisation en ordonnancement de production”. Thèse en automatique et informatique industrielle à l’école centrale de Lille et l’université de Lille1. Soutenue le 5 Janvier 1999.
- [Metropolis et al. 1953] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. “Equations of state calculations by fast computing machines”. Journal of Chemical Physics, 21:1087--1091, 1953.

- [Mitrovic-Minic 2001] S. Mitrovic-Minic. “The Dynamic Pickup and Delivery Problem with Time Windows”. PhD thesis, Simon Frase University, 2001.
- [Mitrovic-Minic et al. 2003] S. Mitrovic-Minic, and G. Laporte. “Waiting Strategies for the dynamic Pickup and delivery Problem with Time Windows”. Elsevier Science, Septembre 2003.
- [Mitrovic-Minic et al. 2004] S. Mitrovic-Minic, R. Krishnamurti, and G. Laporte. “Double-Horizon based heuristics for the dynamic pickup and delivery problem with time windows”. *Transportation Research* 38, pages pp 669-685, 2004.
- [Montamenni et al. 2002] R. Montamenni, L.M. Gambardella, A.E. Rizzoli, and A.V. Donati. “A new algorithm for a Dynamic Vehicle Routing Problem based on Ant Colony System”. IDSIA, Switzerland, 2002.
- [Nanry et al. 2000] W.P. Nanry, J.W. Barnes, “Solving the Pickup and Delivery Problem with Time Windows Using Reactive Tabu Search”, *Transportation Research (Part B)*, 34, 107-121, 2000.
- [Or 1976] I. Or. “Travelling salesman-type combinatorial problems and their relation to the logistics of blood banking”. PhD thesis, Departement of industrial engineering and management science. North Western University, Evanston, II, 1976.
- [Portmann 1996] M.C. Portmann, “Genetic algorithms and scheduling : a state of the art and some propositions, Proceedings of the workshop on production planning and control”, Mons, Belgium, pi-xxiv, September 9-11, 1996
- [Portmann 2001] M.C. Portmann, A. Vignier, “Algorithmes génétiques et ordonnancement”. Chapitre dans l’ouvrage : *Ordonnancement de la production*, Edition Hermès, France, 2001
- [Prins 2002] C. Prins, “Efficient heuristics for the heterogeneous fleet multitrip vehicle routing problem”. *Journal of Mathematical Modelling and Algorithms*, 1 (2), pages pp. 135-150, 2002.

- [Psaraftis 1983] H.N. Psaraftis, "Analysis of an $O(N^2)$ heuristic for the single vehicle many-to-many Euclidean dial-a-ride problem". *Transportation Research*, 17B, 133-145, 1983.
- [Psaraftis 1983a] H.N. Psaraftis, "An exact algorithm for the single vehicle many-to-many immediate request dial-a-ride problem with time windows". *Transportation Science*, 17, 351-357, 1983.
- [Ralphs 2002] T.K. Ralphs, "Parallel branch and cut for capacitated vehicle routing". *Parallel Computing archive*, Volume 29, Issue 5, 2 December 2002
- [Ralphs et al 2001] T.K. Ralphs, L. Kopman, W.R. Pulleyblank and L.E. Trotter, "On the capacitated vehicle routing problem", *Mathematical Programming* 94, 343-359, 2001
- [Rego et al. 1994] C. Rego et C. Roucairol. "Le problème de tournées de véhicules : Etude et Résolution Approchée". Technical Report, inria, Février 1994.
- [Rego et al.1996] C. Rego, C. Roucairol, "A Parallel Tabu Search Algorithm using Ejection Chains for the Vehicle Routing Problem". In I.H. Osman and J.P. Kelly (eds), *MetaHeuristics: Theory and Applications*. Kluwer, Boston, 1996.
- [Righini 2000] G. Righini, "Approximation algorithms for the vehicle routing problem with pick-up and delivery", "Note del Polo - Ricerca" n.33, July 2000 problem with time windows and precedence", Technical report, University of Copenhagen, 2000.
- [Roy et al. 1984] S. Roy, J.-M. Rousseau, G. Lapalme and J. A. Ferland. "Routing and scheduling of transportation services for disabled: summary report". Technical report, Centre de recherche sur les transports, Université de Montréal, June 1984.
- [Ryan et al. 1993] Ryan, D.M. Hjorring, C., Glover, F. "Extensions of the petal method for vehicle routing", *Journal of the Operational Research Society* 44, 289-296, 1993.

- [Saker et al. 2001] R. Saker, H.A. Abbas, C. Newton, "Solving multi-objective optimization problems using evolutionary algorithm". Proceedings of International CIMCA Conference, July 9-11, Las Vegas, Nevada, USA, 2001
- [Savelsbergh 1995b] M.W.P. Savelsbergh, "Local Search for Routing Problems with Time Windows". Annals of Operations Research 4, 285-305, 1995.
- [Savelsbergh et al. 1995a] M.P.W. Savelsbergh, M. SOL, "The general pickup and delivery problem". Transportation Science 29 (1), 17-29, 1995.
- [Sexton et al. 1983] T.R. Sexton, L.D. Bodin, "The multiple-vehicle subscriber dial-a-ride problem", Working paper MS /S 83-009, College of Business and Management, University of Maryland, 1983.
- [Sexton et al. 1985] T.R. Sexton, L.D. Bodin, "Optimizing single vehicle many-to-many operations with desired delivery times: I. scheduling", Transportation Science 19 (4), 378-410, 1985.
- [Sexton et al. 1985] T.R. Sexton, L.D. Bodin, "Optimizing single vehicle many-to-many operations with desired delivery times: II.routing", Transportation Science 19 (4), 411-435, 1985
- [Solomon 1987] M.M. Solomon, "Algorithms for the vehicle Routing and Scheduling Problem with Time Window Constraints". Operations Research, 41, 469-488, 1987.
- [Taillard 1990] E. D. Taillard, "Some efficient heuristic methods for the flow shop sequencing problem". European Journal of Operational research, Tome 47, n° 1, pages 65-74, 1990
- [Taillard 1991] E. D. Taillard, "Robust taboo search for the quadratic assignment problem". Parallel Computing, Tome 17, pages 443-455, 1991
- [Taillard 1993] E. D. Taillard, "Parallel iterative search methods for vehicle routing problems". Networks, vol. 23, pp. 661-673, 1993.

- [Taillard 1995] E. D. Taillard, "Comparison of iterative searches for the quadratic assignment problem". *Location Science*, Tome 3, n° 2, pages 87-105, 1995
- [Taillard 1997] E. D. Taillard, P. Badeau, M. Gendreau, F. Guertin and J.-Y. Potvin, "A Tabu Search Heuristic for the Vehicle Routing Problem with Soft Time Windows", *Transportation Science* 31, 170-186, 1997
- [Taillard 1999] E.D. Taillard, "A heuristic column generation method for the heterogeneous fleet VRP". *RAIRO-Operations Research*, 33, pages PP. 1-14, 1999.
- [Taillard et al. 1997] E. D. Taillard, P. Badeau, M. Gendreau, F. Guertin, J.-Y. Potvin. "A tabu search heuristic for the vehicle routing problem with soft time windows". *Transportation Science*, 31:170-186, 1997.
- [Taillard et al. 1998] E. D. Taillard, M. Gambardella, M., Gendreau, J.-Y. Potvin, "Programmation à mémoire adaptative". Technical report IDSIA-79-97, IDSIA, Lugano, 1997. Published in: *Calculateurs Parallèles, Réseaux et Systèmes Répartis* 10, 117-140, 1998.
- [Wimer 2001] M. Widmer, A. Hertz, D. Costa, "Les métaheuristiques". Chapitre dans l'ouvrage : *Ordonnancement de la production*, Edition Hermès, France, 2001
- [Witucki et al. 1997] M. Witucki, P. Dejax and M. Haouari. "Un modèle et un algorithme de résolution exacte pour le problème de tournées de véhicules multipériodique". Ecole Centrale Paris, Laboratoire Productique-Logistique, Grande Voie des Vignes, 92295 Châtenay-Malabry. Ecole Polytechnique de Tunisie, 2070 La Marsa, Tunisia 1997.
- [Woodruff et al. 1993] D. L. Woodruff, et E. Zemel. "Hashing vectors for tabu search". *Tabu search*, édité par G. Glover, M. Laguna, E. D. Taillard, et al., numéro 41 dans *Annals of Operations Research*, pages 123-137. Baltzer, Bâle, Suisse, 1993.

- [Xu et al. 1996] J. Xu, J. P. Kelly, “A network flow-based tabu search heuristic for the vehicle routing problem”, *Transportation Science* 30, 379-393, 1996.
- [Yang et al. 2002] J. Yang, P. Jaillet, and H. S. Mahmassani. “Real Time Multi-Vehicle Truckload Pick-Up and Delivery Problems”. Department of Industrial and Manufacturing Engineering, New Jersey Institute of Technology, Newark, New Jersey 07102 June 2002.

I

1-DPDPWTW ... 13, 97, 115, 116, 117, 118, 119, 120,
121, 122, 123, 124, 125, 126, 127, 129, 146,
147, 150, 154, 155, 157, 158
1-PDPTW .. 1, 13, 17, 18, 19, 42, 43, 65, 66, 68, 69,
81, 82, 86, 88, 90, 91, 92, 93, 94, 95, 97, 115,
116, 125, 126, 129, 130, 131, 132, 134, 135,
136, 139, 140, 141, 143, 146, 150, 155, 157, 158

A

ADW113
AE.....178
aléatoire27, 46, 49, 130
algorithme génétique46, 47, 53
Approche . 68, 69, 82, 115, 142, 143, 144, 148, 149,
150, 179, 180, 181, 182
Approche évolutionniste69, 82

B

BENCH110
benchmark 43, 44, 130, 132, 137, 139, 146, 147
BUSIEST109

C

chromosome .. 47, 49, 51, 73, 79, 89, 120, 122, 124,
125
classification..... 23, 98, 101, 103, 133
complexité66
contraintes 17, 23, 25, 26, 27, 28, 29, 39, 41, 42, 43,
47, 57, 66, 68, 69, 70, 72, 73, 74, 75, 79, 82, 88,
89, 93, 94, 95, 97, 115, 116, 120, 121, 122, 123,
124, 127, 130, 133, 134, 136, 159
contrôle.....34, 111, 112
croisement 19, 37, 46, 48, 49, 53, 64, 77, 78, 79, 88,
89, 90, 95, 115, 124, 125, 127, 130, 138
Croisement..... 49, 53, 77, 78, 89, 124, 125
CURRENT109, 110

D

degré de dynamisme.....98, 101, 103, 126
déterministe29
DF112, 113
direct.....36, 47, 98
dispersion.....135
distance.. 22, 23, 40, 67, 76, 79, 82, 83, 84, 99, 104,
117, 119, 125, 130, 134, 139, 141, 142, 143,
145, 147, 148, 152
DPDPWTW97, 98, 112, 114, 115, 116, 125, 126, 127,
158
DTRP..... 106, 107, 108, 110
DTSPTW108, 109
DVRP 26, 98, 101, 103, 104, 105, 106, 110, 112,
113, 126

DVRPTW.....26, 110, 111, 113, 114
DW 113
dynamique..1, 17, 18, 19, 26, 30, 38, 42, 43, 44, 64,
95, 98, 99, 100, 101, 103, 105, 108, 110, 111,
115, 116, 117, 121, 122, 124, 126, 131, 132,
146, 147, 151, 155, 157, 158

E

effet35, 36, 46, 48, 52, 54, 63, 98, 101
évolutionniste.18, 19, 66, 69, 70, 81, 82, 86, 88, 89,
90, 91, 115, 116, 117, 118, 126, 131, 137, 139,
143, 145, 148, 157

F

FCFS 106, 107
fin de service 112
fonction objectif ...25, 33, 35, 36, 43, 55, 61, 63, 66,
81, 87, 94
formulation.18, 23, 39, 42, 44, 66, 67, 98, 105, 116,
126

G

GEN 107, 108
génération.....37, 38, 43, 44, 47, 51, 76, 77, 88, 94,
107, 115, 118, 119, 127, 130, 134, 137

H

HI-REQ 109

I

indirect 47
insertion31, 42, 79, 80, 89, 106, 110, 111, 112, 114,
116, 124, 125
insertion séquentielle 31

M

MDVRP28
Mutation50, 54, 79, 125
MYOPT 110

N

NEAREST..... 109
NN..... 106
NS..... 110

O

OPTUN 110
OVRP28

P

paramètres43, 85, 130, 136
PART 107

Index

PDPTW ... 17, 18, 19, 22, 28, 38, 39, 42, 43, 44, 64,
66, 98, 130, 131, 132, 134, 157, 158
PMX 48
population 37, 46, 51, 52, 53, 54, 72, 76, 77, 88, 90,
94, 118, 119, 120, 127, 143, 148
PVRP 27

R

Recherche Tabou 35, 55
REOPT 110
résultats.... 13, 18, 19, 30, 31, 32, 33, 43, 54, 58, 63,
66, 95, 102, 115, 127, 130, 134, 139, 140, 141,
142, 143, 144, 146, 147, 148, 150, 151, 152,
155, 158
roulette 51, 52, 80

S

SE110
sélection 32, 37, 46, 51, 52, 53, 64, 80
simulation 112, 130, 136, 153, 158
solution 22, 30, 33, 34, 35, 36, 37, 44, 46, 47, 48,
51, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 68, 70,
71, 79, 88, 89, 91, 92, 93, 94, 95, 100, 102, 109,
112, 116, 118, 124, 126, 137, 139, 142, 143,
145, 147, 148, 149, 150, 181, 182
SQM 106

STATIC 110
statique1, 17, 18, 19, 43, 59, 64, 94, 95, 98, 99, 101,
103, 110, 113, 115, 131, 132, 139, 146, 147,
149, 150, 151, 154, 157, 158
stochastique 27
SVRP 27

T

tournée 17, 22, 23, 25, 26, 27, 28, 33, 39, 42, 58, 70,
98, 99, 110, 113, 116, 126, 157
tournoi 52
TRP 28
TSP 27, 107

V

Vehicle Routing Problem 22, 26, 27, 28, 157
VRP 17, 18, 22, 25, 26, 27, 28, 29, 31, 33, 34, 36,
38, 39, 44, 98, 105, 113, 157
VRPB 26
VRPPD 27
VRPTW 22, 26, 28, 43, 44

W

WF 112, 113