
Modèles stochastiques évolutionnaires pour la gestion de tournées de véhicules avec fenêtres de temps souples et demandes floues

THESE

Présentée et soutenue publiquement le 18 décembre 2007
en vue de l'obtention du

Doctorat de l'Université d'Artois

(Spécialité : Génie Informatique)

Par

XU Jian

Composition du jury

<i>Président:</i>	Rémy DUPAS	Professeur de l'Université Bordeaux 1
<i>Rapporteurs :</i>	Aziz MOUKRIM	Professeur à l'Université de Technologie de Compiègne
	Abdelhakim ARTIBA	Professeur à l'Institut Supérieur de Mécanique de Paris
<i>Examineurs :</i>	Daniel JOLLY	Professeur à l'Université d'Artois
	Gilles GONCALVES	Professeur à l'Université d'Artois (Directeur de thèse)
	Tienté HSU	Maître de Conférences à l'Université d'Artois (Co-directeur de thèse)

Remerciements

Cette thèse n'aurait vu le jour sans la confiance, la patience et la générosité de mon directeur de thèse, monsieur **Gilles GONCALVES**, que je veux vivement remercier. Je voudrais aussi le remercier pour le temps et la patience qu'il m'a accordés tout au long de ces années. Les conseils et les aides qu'il m'a divulgués tout au long de la rédaction, ont permis la production de cette thèse.

Je remercie monsieur le professeur **Aziz MOUKRIM** à l'Université de Technologie de Compiègne et monsieur le professeur **Abdelhakim ARTIBA** à l'Institut Supérieur de Mécanique de Paris, de me faire l'honneur d'être rapporteurs de ce mémoire.

Je remercie particulièrement monsieur **Daniel JOLLY** directeur du LGI2A, de m'avoir accueilli dans son laboratoire de recherche.

Je tiens également à remercier monsieur le professeur **Rémy DUPAS** à l'Université Bordeaux 1, pour son intuition et ses nombreux conseils.

Mes plus sincères remerciements vont également à monsieur **Tienté HSU**, qui en agissant au titre de co-directeur a fortement enrichi ma formation. Ses conseils et ses commentaires auront été fort utiles.

Je souhaite également remercier tous les membres de LGI2A.

Merci enfin à toute ma famille, en particulier **Mes parents** et **Mon oncle**, de m'avoir supportée et aidée pendant ces dernières années.

Table des matières

Introduction générale	1
Chapitre1 Le problème de la gestion de tournées de véhicules	5
1.1 Les variantes du VRP	6
1.1.1 Les caractéristiques du VRP	6
1.1.2 Les différentes variantes du VRP	8
1.2 Le problème CVRP	10
1.2.1 Description	10
1.2.2 Formulation	11
1.3 Le problème VRPTW	13
1.3.1 Description	13
1.3.2 Formulation	14
1.4 Complexité.....	15
1.5 Les méthodes de résolution	16
1.5.1 Les méthodes exactes.....	18
1.5.1.1 Programmation dynamique	18
1.5.1.2 Programmation Linéaire en nombres entiers (Branch & Cut)	18
1.5.1.3 La méthode de recherche arborescente (Branch & Bound)	19
1.5.2 Les heuristiques classiques.....	20
1.5.2.1 Méthodes constructives.....	20
1.5.2.2 Méthodes à deux phases.....	21
1.5.2.3 Méthodes d'améliorations	22
1.5.3 Les métaheuristiques.....	22
1.5.3.1 Le recuit simulé	23
1.5.3.2 La recherche Tabou.....	26
1.5.3.3 GRASP et VNS.....	28
1.5.3.4 Algorithmes évolutionnaires et Algorithmes génétiques	28
1.5.3.5 L'algorithme des colonies de fourmis.....	31
1.5.3.6 Métaheuristiques pour le VRP.....	33
1.6 Cadre de travail	34

Chapitre 2 Résolution du problème de la gestion de tournées de véhicules

avec fenêtres de temps et demandes floues 36

2.1 Le VRP avec demande stochastique.....	37
2.1.1 Cadre de modélisation.....	38
2.1.2 Politiques de recours.....	38
2.1.3 Travaux existants.....	39
2.2 Le VRP avec fenêtres de temps et demandes floues.....	40
2.3 Modélisation du problème.....	42
2.3.1 Notions de logique floue.....	43
2.3.1.1 Système à inférences floues.....	43
2.3.1.2 Théorie des possibilités.....	44
2.3.2 Contrainte de capacité floue.....	46
2.3.3 Modèle CCP.....	50
2.3.4 Modèle SPR.....	51
2.3.5 Politique de recours utilisée.....	52
2.3.6 Méthode d'estimation des défauts de tournées.....	53
2.3.6.1 Scénarios d'apprentissage.....	53
2.3.6.2. Procédure « rejouer » les tournées.....	54
2.4 Algorithme génétique pour le VRPTWFD.....	56
2.4.1 Introduction.....	56
2.4.2 Codage de la solution.....	58
2.4.3 Evaluation d'un chromosome.....	60
2.4.3.1 Phase de décodage d'un chromosome.....	60
2.4.3.2 Variantes autour de l'AG.....	64
2.4.4 Population initiale.....	66
2.4.5 Sélection.....	66
2.4.6 Opérateurs de croisement (Crossover).....	67
2.4.6.1 PMX (Partially Mapped Crossover).....	68
2.4.6.2 MX1T/D :.....	68
2.4.7 Opérateurs de mutation.....	70
2.4.8 Méthode de remplacement par l'élitisme.....	72
2.5 Etudes expérimentales.....	73
2.5.1 Le benchmark utilisé.....	73
2.5.1.1 Benchmark de Solomon pour le VRPTW.....	73
2.5.1.2 Benchmark pour le VRPTWFD.....	75
2.5.2 Simulation.....	77
2.5.2.1 Les paramètres.....	77
2.5.2.2 Réglage des seuils α et β du modèle CCP.....	78
2.5.2.3 Réglage des seuils α et β du modèle SPR.....	79
2.5.2.4 Solutions du benchmark VRPTWFD.....	81
2.5.3 Evaluation des solutions dans un environnement réel.....	83
2.5.4 Comparaison avec la version déterministe.....	83

2.5.5 L'influence du taux d'incertitude.....	86
2.5.6 Zoom sur le modèle SPR.....	88
2.5.6.1 Réglage du nombre de scénarios d'apprentissage.....	88
2.5.6.2 Evaluation de la solution du modèle SPR.....	90
2.6 Conclusion.....	92
Chapitre 3 Résolution « en ligne » du VRPTWFD dynamique	94
3.1 Le VRP dynamique.....	95
3.1.1 Description.....	96
3.1.2 Le degré de dynamisme.....	97
3.1.3 Classification du DVRP.....	97
3.1.4 Les méthodologies de résolution du DVRP.....	99
3.1.4.1 Stratégies classiques.....	99
3.1.4.2 Les approches basées sur l'optimisation.....	101
3.2 Plateforme de Simulation pour la résolution du VRPTWFD dynamique.....	103
3.2.1 Principe de résolution.....	103
3.2.2 Architecture de la plateforme.....	104
3.2.2.1. Le composant simulateur du monde réel.....	104
3.2.2.2. Tournées effectuées.....	105
3.2.2.3. Le composant Algorithme Génétique (AG).....	105
3.2.2.4. Le composant Gestion d'Evénements (GE).....	107
3.2.3 La gestion du temps.....	111
3.2.4 Le fonctionnement de la plateforme.....	112
3.3 Etudes expérimentales.....	113
3.3.1 Le benchmark utilisé.....	113
3.3.2 Simulation.....	114
3.3.2.1 Choix des paramètres.....	114
3.3.2.2 Etude comparative des modèles stochastiques.....	116
3.3.3 Comparaison avec la version « Déterministe et Statique ».....	119
3.4 Conclusion.....	122
Conclusions générales et perspectives.....	124
Annexe A Résultats des problèmes de VRPTWFD statique.....	126
A.1 Le réglage des paramètres de l'algorithme génétique.....	126
A.2 Le réglage des seuils α et β du modèle CCP.....	127
A.3 Le réglage des seuils α et β du modèle SPR.....	134
A.4 Comparaison avec la version déterministe.....	142
A.5 Evaluation de la solution du modèle SPR.....	147
Annexe B Résultats des problèmes de VRPTWFD dynamique.....	152

B.1 Etude comparative des modèles stochastiques.....	152
B.2 Comparaison avec la version « Déterministe et Statique ».....	156
Bibliographie	160

Table des figures

Figure 1. 1	Le problème VRP.....	6
Figure 1. 2	Compromis entre temps de calcul et qualité de la solution.....	16
Figure 1. 3	Les méthodes de résolution du VRP.....	17
Figure 1. 4	La recherche de la solution optimale peut détériorer la solution courante.....	27
Figure 1. 4	Comment les fourmis trouvent le plus court chemin.....	31
Figure 2. 1	Le problème VRPSD.....	39
Figure 2. 2	La modélisation d'une demande floue.....	41
Figure 2. 3	Le système à inférences floues.....	43
Figure 2. 4	La satisfaction de la contrainte de capacité floue : possibilité et nécessité.....	47
Figure 2. 5	Le recours du défaut de tournées.....	52
Figure 2. 6	Un exemple de la procédure « rejouer ».....	55
Figure 2. 7	L'algorithme génétique.....	57
Figure 2. 8	Chromosome à codage direct.....	58
Figure 2. 9	Chromosome à codage indirect.....	59
Figure 2. 10	Le codage du chromosome.....	59
Figure 2. 11	Tournées planifiées obtenues après la phase de décodage.....	60
Figure 2. 12	Décodage d'un chromosome à l'aide de l'heuristique HIT.....	61
Figure 2. 13	Une étape du décodage d'un chromosome.....	62
Figure 2. 14	Les tournées planifiées par la phase de décodage.....	64
Figure 2. 15	L'évaluation d'un chromosome dans « FAG_SPR ».....	66
Figure 2. 16	Principe du croisement.....	67
Figure 2. 17	Principe de la mutation.....	70
Figure 2. 18	L'opérateur de mutation échange.....	71
Figure 2. 19	L'opérateur de mutation λ -échange.....	72
Figure 2. 20	L'évolution générale de la population.....	73
Figure 2. 21	La distribution des clients dans les 6 types de problèmes.....	74
Figure 2. 22	L'influence du taux d'incertitude.....	88
Figure 2. 23	L'influence du nombre de scénarios d'apprentissage.....	90
Figure 3. 1	Le problème DVRP.....	96
Figure 3. 2	Classification du DVRP.....	98
Figure 3. 3	heuristique d'insertion.....	101
Figure 3. 4	Plateforme de simulation.....	104
Figure 3. 5	L'ensemble des tournées effectuées.....	105
Figure 3. 6	Chromosome à l'instant t.....	106
Figure 3. 7	Le décodage d'un chromosome à l'instant t.....	107

Figure 3. 8	Heuristique d'Insertion Chromosomique (HIC) dans la procédure d'évènement nouveau client.....	108
Figure 3. 9	Modification du vecteur « tournées effectuées » en cas de « défaut de tournée ».....	109
Figure 3. 10	Procédure d'évènement fin de service	110
Figure 3. 11	La simulation de la journée de service	112
Figure 3. 12	Valeur moyenne t^{SD} en fonction du degré de dynamisme	122

Liste des tableaux

Tableau 2. 1	Un exemple du scénario d'apprentissage de demandes réelles	54
Tableau 2. 2	La construction des demandes floues.....	75
Tableau 2. 3	L'ensemble des solutions du modèle CCP correspond aux combinaisons des seuils pour le problème SRC101	78
Tableau 2. 4	L'ensemble des solutions du modèle SPR correspond aux combinaisons des seuils pour le problème SRC101	80
Tableau 2. 5	Solutions trouvées par trois modèles pour les problèmes à traiter	82
Tableau 2. 6	Comparaison des solutions fournies par les différents modèles du SRC101 avec la version déterministe.....	84
Tableau 2. 7	Comparaison avec la version déterministe.....	85
Tableau 2. 8	Solutions des trois modèles pour les extensions du problème SRC101correspondantes à des taux d'incertitude différents.....	86
Tableau 2. 9	Comparaison les solutions des trois modèles pour les extensions du problème SRC101 correspondantes aux taux d'incertitude avec la version déterministe	87
Tableau 2. 10	Solutions du modèle SPR correspondantes au nombre de scénarios d'apprentissage pour le problème SRC101	89
Tableau 2. 11	Comparaison des solutions du modèle SPR correspondantes au nombre de scénarios d'apprentissage pour le problème SRC101 avec la version déterministe.....	89
Tableau 2. 12	L'évaluation de la solution du modèle SPR pour le problème SRC101.....	91
Tableau 2. 13	Synthèse de l'évaluation des solutions trouvées par le modèle SPR.....	92
Tableau 3. 1	Valeurs des paramètres de réglage correspondants les problèmes dynamiques	115
Tableau 3. 2	Comparaison des solutions fournies par trois modèles avec la version « D&D» pour le problème DSRC101	117
Tableau 3. 3	Comparaison des solutions fournies par trois modèles pour le VRPTWFD dynamique avec la solution de la version « D&D»	118
Tableau 3. 4	Comparaison les meilleures solutions pour le problème DSRC101 avec celles de la version « D&S » correspondante	120
Tableau 3. 5	Comparaison des meilleures solutions obtenues pour les problèmes du VRPTWFD dynamique avec celles de la version « D&S »	121

Introduction générale

Ces dernières années les systèmes de transport utilisés pour le ramassage et la distribution de biens ou de services ont fait l'objet de nombreuses études dans la communauté scientifique Recherche Opérationnelle (RO). Cet intérêt est partiellement dû au nombre élevé de fusions de compagnies qui laissent les planificateurs de flotte de véhicules avec des problèmes encore plus grands et plus complexes, à résoudre. Un fait qui complique la planification est la prise en compte du temps qui est devenu une ressource extrêmement critique pour les entreprises. De nos jours, la plupart des systèmes de transport doivent pouvoir fonctionner en respectant des contraintes temporelles strictes et ceci en s'adaptant aux aléas du problème. En effet, les clients ou partenaires d'une entreprise exigent de celle-ci une qualité de service garantie (délais à respecter). De plus, dans certaines applications réelles, on ne connaît pas avec certitude les données. Cela a suscité un intérêt croissant sur l'aspect « stochastique » de certaines variables. L'environnement dans lequel une entreprise doit évoluer est bien souvent incertain et donc sa réactivité est également un atout important. Ceci a conduit à définir des modèles de pilotage des systèmes de transport dits « dynamiques » dans lesquels une partie des données est considérée comme dépendante du temps.

Le domaine dans lequel s'inscrivent nos travaux, concerne la gestion des tournées de véhicules (VRP : Vehicle Routing Problem). Le problème classique de la gestion des tournées de véhicules (VRP) consiste à trouver des chemins avec un coût minimum pour que les véhicules puissent visiter exactement une fois chaque client géographiquement distribué. Le VRP est un sous problème important dans le domaine des systèmes de distribution et beaucoup d'efforts ont été consacrés en recherche sur divers aspects du VRP. Cependant, une grande partie de ces travaux a essentiellement porté sur le contexte déterministe et statique qui suppose que les données du problème sont d'une part parfaitement et exhaustivement connues et d'autre part ne sont pas sujettes à des changements pendant l'horizon de planification.

Dans beaucoup de situations pratiques, les données du VRP ne sont pas complètement connues au moment de la résolution. Depuis une dizaine d'années, les problèmes de gestion dynamique de tournées de véhicules (DVRP : Dynamic Vehicle Routing Problem) et les problèmes stochastiques de tournées de véhicules (SVRP : Stochastic Vehicle Routing Problem) ont commencé à intéresser aussi bien les chercheurs que les industriels. Le problème VRP est dit stochastique lorsque certains paramètres du problème ne peuvent pas

être connus avec certitude. Ce pourrait être les demandes (quantité demandée) des clients, le temps ou le coût de transport ou bien l'ensemble des clients à visiter. Dans le VRP dynamique, qui est également connu sous le nom du VRP en temps réel, des données sont connues au cours du temps. Le VRP peut être classé en quatre catégories :

- Dans les problèmes « déterministes et statiques », toutes les données sont connues à l'avance avec certitude.
- Dans les problèmes « déterministes et dynamiques », les données sont connues avec certitude et seuls quelques éléments dépendent du temps.
- Dans le problème « stochastiques et statiques », l'incertitude peut porter sur la quantité de la demande, sur le temps de service chez le client, ou sur le temps de trajet. Le temps n'est pas pris en compte explicitement.
- Dans des problèmes « stochastiques et dynamiques », les données incertaines sont modélisées par des variables stochastiques, en plus, des variables dépendent du temps.

Nous nous intéressons sur les deux dernières catégories. Le travail réalisé dans cette thèse traite le problème de la gestion de tournées de véhicules avec fenêtres de temps et demandes floues (VRPTWFD : Vehicle Routing Problem with Time Windows and Fuzzy Demands). Les clients spécifient leur demande à l'aide d'un nombre flou pour une plus grande souplesse. C'est un problème de nature stochastique qu'il faut résoudre. Dans le VRPTWFD statique, toutes les données sont connues au moment de la planification et juste les demandes sont imprécises. Le VRPTWFD est étudié également dans un cadre dynamique. C'est-à-dire que certaines demandes des clients arrivent au cours de journée.

Les objectifs de cette thèse peuvent être résumés dans les points suivants :

- Proposer des modèles stochastiques pour traiter le problème de la gestion de tournées de véhicules avec fenêtres de temps et demandes floues.
- Proposer un algorithme génétique qui intègre ces modèles pour résoudre le VRPTWFD statique.
- Etendre ces travaux au cas du VRPTWFD dynamique en utilisant un algorithme génétique « en ligne ».
- Vérifier les performances de ces différents modèles en évaluant la qualité des solutions fournies par ces modèles.

Dans le VRPTWFD, les demandes sont modélisées par des nombres flous, les quantités réelles des demandes ne peuvent être connues qu'au moment où les véhicules arrivent chez les clients et commencent les services. Malgré ceci, les tournées des véhicules doivent être construites à l'avance par un algorithme qui doit prendre en compte cette incertitude. A cause de cette incertitude, il arrive quelquefois que le véhicule ne puisse servir un client de la tournée planifiée quand il se présente chez celui-ci : par exemple lorsque la capacité résiduelle du véhicule est inférieure à la demande réelle de ce client. On appelle cette situation un « défaut de tournées ». La théorie de possibilité nous a permis d'exprimer la contrainte de capacité floue en fixant des valeurs de seuils de possibilité et de nécessité. En utilisant cette contrainte de capacité floue, un modèle de programmation sous contraintes probabilistes (CCP : Chance Constrained Programming) et un modèle à deux-étapes de programmation stochastique avec recours (SPR : Stochastic Programming with Recourse) ont été proposés pour traiter le VRPTWFD. Nous pouvons ainsi planifier avec ces deux modèles

un ensemble de tournées a priori dans lesquelles les « défauts de tournées » seront inférieurs à un certain seuil. De plus, le modèle SPR permet de prendre en compte le coût supplémentaire de transport induit par les « défauts des tournées » selon une politique de recours choisie et à l'aide d'une méthode de simulation de tournées basée sur des scénarios d'apprentissages.

H. Housroum a utilisé l'Algorithme Génétique (AG) pour la résolution du problème de la gestion de tournées de véhicules avec fenêtres de temps (VRPTW : Vehicle Routing Problem with Time Windows) dans un cadre déterministe dans son travail de thèse [Housroum 2005]. Notre travail est donc une extension de ceux-ci dans un cadre stochastique. Nous reprenons l'algorithme génétique en intégrant les deux modèles précédents pour la résolution du VRPTWFD. Dans cet AG, le codage indirect permet d'obtenir des chromosomes représentant des solutions pertinentes après chaque application d'opérateur génétique. Ce choix initial a permis de traiter facilement et efficacement le VRPTW en mode dynamique. Le décodage utilisé pour construire les tournées de véhicules est basé sur une heuristique d'insertion constructive autorisant ainsi un décodage rapide. C'est au sein de cette phase de décodage que nous avons dû intégrer la prise en compte des demandes floues. En modifiant la version de l'AG proposé par H. Housroum pour la résolution du VRPTW déterministe, nous avons obtenu deux versions de l'AG intégrant les deux modèles du VRPTWFD.

Dans un VRPTWFD dynamique, les clients à servir dans la journée ne sont pas complètement connus au moment où débute la planification. Nous résolvons le problème dynamique « en ligne ». La solution du problème est construite au fur et à mesure de l'arrivée des clients. Une plateforme de simulation, qui ayant la capacité de simuler la journée de service, permet de construire et réaliser les tournées au fur et à mesure de l'avancement du temps.

Afin de vérifier la qualité de nos deux modèles en mode statique, nous avons créé un benchmark pour le VRPTWFD en modifiant le jeu de problèmes fournis par [Solomon 1987] pour le VRPTW. Nous avons également créé un benchmark pour le VRPTWFD dynamique selon la modification introduite par Gendreau [Gendreau et al. 1999] pour le VRPTW dynamique. Nous avons comparé les performances relatives de deux modèles avec un modèle de référence où toutes les informations sont connues avec certitude et à l'avance.

Le présent document est structuré de la façon suivante :

- Le premier chapitre est une revue de l'état de l'art des VRP et des problématiques associées. Nous rappelons les différents éléments qui composent ce problème, ainsi que les contraintes à satisfaire et l'objectif à optimiser. La formulation mathématique de ce problème est ensuite rappelée. Nous détaillons la plupart des variantes de ce problème trouvées dans la littérature, puis nous présentons un rapide état de l'art sur les méthodes qui sont utilisées pour la résolution de ce problème.
- Le deuxième chapitre est consacré à la version statique du VRPTWFD. Dans ce chapitre, nous commençons par une présentation du VRP avec demande stochastique. Puis deux modèles, utilisant la théorie des possibilités, sont proposés pour traiter le VRPTWFD. Un algorithme génétique comme méthode de résolution métaheuristique est utilisé dans ce qui suit. La dernière partie de ce chapitre évalue les performances de ces modèles en résolvant les instances du benchmark proposé pour le VRPTWFD statique.

- Le troisième chapitre se focalise sur l'approche utilisée pour la résolution du VRPTWFD dynamique. Un état de l'art sur les VRP dynamiques est rappelé dans la première partie. Ensuite, nous présentons la plateforme de simulation qui nous a permis de résoudre le VRPTWFD dynamique avec un AG « en ligne ». L'évaluation de performances de nos modèles précédents dans un contexte dynamique est faite à la fin de ce chapitre.

Nous terminons le rapport par une conclusion des travaux réalisés, ainsi que les perspectives d'amélioration et d'extension.

Chapitre1

Le problème de la gestion de tournées de véhicules

Sommaire

1.1 Les variantes du VRP	6
1.1.1 Les caractéristiques du VRP.....	6
1.1.2 Les différentes variantes du VRP	8
1.2 Le problème CVRP	10
1.2.1 Description	10
1.2.2 Formulation	11
1.3 Le problème VRPTW.....	13
1.3.1 Description	13
1.3.2 Formulation	14
1.4 Complexité	15
1.5 Les méthodes de résolution	16
1.5.1 Les méthodes exactes	18
1.5.2 Les heuristiques classiques.....	20
1.5.3 Les métaheuristiques	22
1.6 Cadre de travail	34

Le problème de la gestion de tournées de véhicules, ou *vehicle routing problem (VRP)* en anglais, est l'un des problèmes combinatoires difficiles les plus connus dans la communauté scientifique RO (Recherche Opérationnelle). Ce problème consiste à déterminer un ensemble optimal de tournées de distribution ou de ramassage à moyen d'une flotte de véhicules localisés sur un ou plusieurs dépôts afin de desservir un ensemble de clients répartis géographiquement en respectant des contraintes (voir figure 1.1).

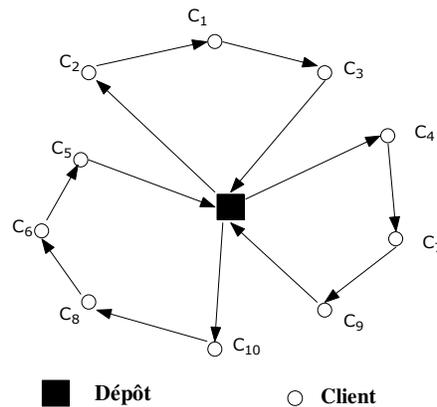


Figure 1. 1 Le problème VRP

Nous allons présenter dans la section 1.1, les variantes du VRP les plus connues par rapport à leurs caractéristiques principales. Les variantes du VRP directement en lien avec notre problème à traiter sont expliquées plus en détail dans les sections 1.2 et 1.3. En montrant la complexité du VRP dans la section 1.4, nous allons exposer les méthodes de résolution pour le VRP ainsi que leurs applications dans la section 1.5. Pour conclure, le cadre général de notre travail sera présenté dans la section 1.6

1.1 Les variantes du VRP

Le problème de tournées de véhicules (VRP) est appliqué dans plusieurs domaines économiques. La livraison de colis, la collecte du lait, la collecte d'argent (distributeurs), le ravitaillement de stations essence sont des exemples parmi les plus connus. Il apparaît aussi dans la distribution de services : visites à domicile de médecins, de représentants de commerce, d'agents de maintenance.

1.1.1 Les caractéristiques du VRP

Par rapport aux problèmes réels, certains auteurs ont tenté de définir une classification des problèmes de tournées à partir des caractéristiques décrites ci dessous. Leur combinaison permet de mieux approcher les problèmes réels [Bodin et al. 1981], [Christofides 1985], [Desrochers et al. 1990]:

1. La distance entre les clients
 - (a). ces distances sont ou ne sont pas euclidiennes ;
 - (b). la distance pour aller d'un client A à un autre client B est la même que pour aller

- du client B au client A (Problème Symétrique) ;
 - (c). ces distances sont différentes (Problème Asymétrique) ;
2. Le nombre de dépôts :
 - (a). un dépôt (VRP Classique) ;
 - (b). plusieurs dépôts ;
 3. Le nombre de véhicules disponibles :
 - (a). un véhicule ;
 - (b). plusieurs véhicules (VRP Classique) ;
 4. La capacité des véhicules :
 - (a). identique pour tous les véhicules (homogène);
 - (b). différente pour au moins un véhicule (hétérogène);
 - (c). pas de restriction de capacité (capacité infinie) ;
 5. La nature de la demande des clients :
 - (a). statique (connue en avance) ;
 - (b). dynamique (apparaît au cours de temps) ;
 - (c). déterministe (connue avec certitude) ;
 - (d). stochastique (connue avec incertitude) ;
 6. Les horaires de service d'un client :
 - (a). non spécifiques (VRP Classique) ;
 - (b). spécifiques par intervalle de temps (notion de fenêtres de temps) ;
 7. Le type de services proposés :
 - (a). un type unique;
 - (b). plusieurs types mais un véhicule ne fournit qu'un seul type de service ;
 - (c). plusieurs types pour un même véhicule ;
 8. La période (ou horizon) considérée :
 - (a). une journée ;
 - (b). une semaine ;
 - (c). périodique ;
 9. La longueur maximale d'une tournée :
 - (a). donnée et identique pour toutes les tournées ;
 - (b). donnée et non identique pour toutes les tournées ;
 - (c). non donnée;
 10. Le temps maximal d'une tournée :
 - (a). donné et identique pour toutes les tournées ;
 - (b). donné et non identique pour toutes les tournées ;
 - (c). non donné ;

De même, l'objectif considéré peut différer. La minimisation de la distance totale parcourue ou la minimisation du temps parcouru sont les objectifs les plus souvent considérés. Mais on peut rencontrer également les objectifs suivants :

- minimisation du nombre de véhicules utilisés pour servir l'ensemble des clients ;
- minimisation de la violation de contraintes, en particulier pour les contraintes de fenêtres

- de temps dites « souples »;
- maximisation des gains lorsque la visite de la totalité de la clientèle est impossible, un gain étant lié à la visite d'un client.

Enfin, plus récemment, les recherches s'intéressent à des versions multi-objectifs rencontrées dans des cas pratiques[Jozefowicz et al. 22002].

1.1.2 Les différentes variantes du VRP

Ces différentes possibilités induisent autant de variantes du problème VRP et c'est ce qui en fait sa richesse et son intérêt pour la communauté scientifique. Récemment, le livre de Toth et Vigo [Toth et al. 2002] a permis de faire un survol complet des recherches menées sur le PTV jusqu'à la fin de ce vingtième siècle. Nous allons présenter les principales variantes du VRP :

1. CVRP (Capacitated Vehicle Routing Problem) : Le problème de tournées de véhicules avec contrainte de capacité est une version standard du problème VRP[Dantzig et al. 1959]. Une flotte homogène de véhicules de capacité finie, partant d'un dépôt unique, doit desservir un ensemble de clients, qui formulent une demande connue (ramassage ou livraison, mais pas les deux à la fois), puis retourner au dépôt à la fin de tournée. Pour chaque véhicule, la demande totale ne peut pas dépasser sa capacité. Le but est de minimiser la distance totale parcourue. Le CVRP est le problème le plus étudié de la littérature. Il est à la base de la définition des extensions du VRP. Nous allons préciser la définition et le modèle dans la section suivante.
2. VRPTW (Vehicle Routing Problem with Time Windows) : Une seconde variante fréquemment considérée du VRP est le problème de tournées de véhicules avec fenêtres de temps. C'est un problème de VRP classique auquel on a ajouté une contrainte supplémentaire sur l'horaire de service: chaque client doit fournir une limite de temps inférieure et une limite supérieure entre lesquelles le service doit démarrer [Solomon 1987] [Cordeau et al. 2001].
3. DVRP (Dynamic Vehicle Routing Problem) : Dans le VRP classique, toutes les demandes émanant des clients sont connues a priori. Cependant, de nombreux problèmes réels incluent au moins un (ou plusieurs) caractère dynamique comme par exemple l'apparition d'un nouveau client en cours de journée. Dans ce cas, le décideur doit changer la planification des visites des véhicules en réponse aux nouvelles demandes [Kilby et al. 1998a] qui arrivent au cours du temps.
4. SVRP (Stochastic Vehicle Routing Problem) : Le problème VRP est dit stochastique lorsque au moins un élément du problème est aléatoire. Autrement dit, un élément du problème ne peut être connu avec certitude. Ce pourrait être la demande du client (quantité à livrer ou à ramasser), le temps de parcours ou de service, le coût du transport, ou même l'ensemble des clients à visiter. Le problème avec demandes stochastiques est l'un des plus étudiés [Gendreau et al.1996a].
5. 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 plusieurs produits sur un horizon de temps donné [Baptista et al. 2002]. Dans ce problème, chaque client détient un stock de produits dont il consomme une certaine quantité par jour. La tâche principale est de planifier les jours de livraison pour chaque client sur un horizon de temps prédéterminé et ensuite d'organiser les tournées de la flotte de véhicules afin d'effectuer les livraisons nécessaires. Cette planification doit respecter les contraintes sur les niveaux de stocks à maintenir chez les clients et en même temps elle doit minimiser le coût total de livraison sur l'ensemble de l'horizon considéré. Le problème ainsi présenté est une extension du problème de tournées de véhicules classique par l'ajout de la notion de période et par la nécessité d'intégrer des éléments de gestion de stocks dans la planification des transports.

6. MDVRP (Multi-Depot Vehicle Routing Problem) : Il y a plusieurs dépôts disponibles qui sont distribués géographiquement dans ce type de problème. Chaque véhicule part de et revient vers son dépôt initial [Fischetti et al. 1999].
7. TSP (Traveling Salesman Problem) : Le problème du voyageur de commerce est un cas particulier du problème de VRP sans contrainte de capacité. Il consiste en la détermination du parcours de coût minimal (distance, temps, etc.) d'un seul véhicule (i.e. représentant de commerce) partant d'une localité, visitant toutes les autres localités et revenant à son point de départ [Rego et al. 1994].
8. TRP (Traveling Repairman Problem) : le problème de la tournée du dépanneur (TRP) est une variante du voyageur de commerce (TSP). L'objectif est de minimiser la latence, autrement dit, la somme pondérée des temps de service des clients, celle-ci est la somme des temps d'attente des clients avant que ceux-ci ne soient atteints [Krumke et al. 2003].
9. VRPPD (Vehicle Routing Problem with Pick-up and Delivery) : Dans le problème de ramassage et de livraison (VRPPD), les clients demandent deux types de service (le ramassage et la livraison du produit) à la fois. Chaque client doit fournir deux positions géographiques différentes : la première indique le lieu de ramassage du produit et la seconde indique le lieu de livraison du produit. Ce type de problème introduit naturellement une contrainte de précédence : pour chaque tournée, l'opération de ramassage chez un client doit précéder l'opération de livraison [Nanry et al. 2000].
10. VRPB (Vehicle Routing Problem with Backhauls) : C'est une prolongation du VRP classique qui inclut un ensemble de clients à qui des produits doivent être livrés et un ensemble de fournisseurs dont les marchandises doivent être amenées au dépôt. En outre, sur chaque tournée, toutes les livraisons doivent être effectuées avant que toutes les marchandises soient ramassées pour éviter de réarranger les charges sur le véhicule [Jacobs-Blecha et al. 1993].
11. OVRP (Open Vehicle Routing Problem) : la caractéristique importante de ce problème, qui le distingue du VRP de base, est que les véhicules ne sont pas obligés de retourner au dépôt ou lorsqu'ils y sont obligés, ils revisitent les mêmes clients dans l'ordre inverse. En conséquence, les itinéraires des véhicules ne sont pas des chemins fermés mais peuvent être ouverts [Eglese et al. 2005].

12. VRPHF (Vehicle Routing Problem with Heterogeneous Fleet) : ce type de problème cherche à satisfaire les objectifs du VRP mais en utilisant une flotte de véhicules de types différents. En fait, nous pouvons distinguer ces véhicules par leurs capacités ou par leurs coûts de déplacement [Prins 2002] et [Taillard 1999].
13. VRPSD (Vehicle Routing Problem with Split Delivery) : chaque client peut être visité plus d'une fois si cela est nécessaire. Autrement dit, la demande de client peut être divisée. Contrairement à ce qui est habituellement supposé dans le problème classique (VRP) la demande de chaque client peut être plus grande que la capacité des véhicules [Archetti et al. 2006].
14. m-VRP (Vehicle Routing Problem with Limited Number of Vehicles) : dans ce problème, nous cherchons à résoudre le problème VRP en utilisant seulement un nombre limité de véhicules [Lau et al. 2003].

Nos travaux s'intéressent plus particulièrement aux extensions CVRP, VRPTW, SVRP et DVRP. Nous allons détailler ces différentes variantes dans ce qui suit.

1.2 Le problème CVRP

1.2.1 Description

La version standard du VRP est le problème de tournées de véhicules avec contraintes de capacité (CVRP) qui a été décrite pour la première fois en 1959 par [Dantzig et al. 1959]. Dans le CVRP, tous les clients demandent le même type de service : ramassage ou livraison. Les quantités de service sont connues avec certitude. Une flotte homogène de véhicules de capacité finie, partent d'un unique dépôt, va desservir l'ensemble des clients, avant de retourner au dépôt. L'objectif est de minimiser le coût total des tournées (la distance totale parcourue ou d'autres mesures telles que le temps total) en respectant la contrainte de capacité des véhicules.

Le CVRP est représenté sous la forme d'un graphe complet $G = (S, A)$, où:

- $S = \{0, 1, \dots, n\}$ représente l'ensemble des sommets;
- $A = \{(i, j) \text{ tel que } i \neq j, i, j \in S\}$ représente l'ensemble des arcs entre les sommets.

Chaque sommet est associé à une position de coordonnées. Le sommet 0 représente le dépôt et les autres sommets représentent l'ensemble des clients, noté $C = \{1, \dots, n\}$, qui désirent obtenir une livraison de marchandise provenant du dépôt. Donc on a $S = C \cup \{0\}$.

Une demande positive de marchandise d_i est associée à chaque client i appartenant à C .

Chaque arc $(i, j) \in A$ est associé à une valeur non négative c_{ij} qui correspond le coût pour aller du sommet i au sommet j . Il est courant de prendre la distance Euclidienne de la position du sommet i à la position du sommet j pour déterminer le coût de l'arc. Nous supposons dans la suite que le graphe G est non orienté. Donc la matrice des coûts est

symétrique, $c_{ij} = c_{ji}$. Elle satisfait l'inéquation triangulaire suivante: $c_{ij} \leq c_{ik} + c_{kj}, \forall i, j, k \in S$.

Une flotte de véhicules $V = \{1, \dots, m\}$ est stationnée au dépôt pour desservir les clients. Tous les véhicules possèdent la même capacité Q (flotte homogène) qui est supérieure à la plus grande demande des clients, $Q \geq \max_i d_i, \forall i \in S$.

L'objectif du problème est de minimiser le coût total des tournées tout en respectant les contraintes suivantes :

- Chaque client ne peut être servi qu'une et une seule fois par un seul véhicule.
- Toutes les tournées des véhicules commencent et se terminent par le dépôt.
- La demande totale des clients à servir de chaque tournée ne doit pas dépasser la capacité Q du véhicule.

1.2.2 Formulation

Il existe de nombreux modèles pour le VRP dans la littérature et nous pouvons trouver plusieurs méthodes de résolution pour un même modèle. Inversement, une même méthode de résolution peut être appliquée à plusieurs modèles. Toth et Vigo classent les différents modèles en trois catégories [Toth et al. 2002] :

- Les modèles de flot de commodités ;
- Les modèles de partitionnement d'ensemble ;
- Les modèles de flot de véhicules;

Les modèles de flot de commodités permettent d'associer aux arcs une demande qui circule sur ceux-ci. Ils sont intéressants de par leur adaptabilité à des problèmes réels comme la distribution d'essence par exemple [Garvin et al. 1957].

Dans les modèles de partitionnement d'ensemble, un ensemble de tournées est généré, et à chaque tournée est associé un coût. Ce type de modèle est très général puisque toutes les contraintes sur la constitution des tournées n'apparaissent pas explicitement. La difficulté principale est la génération de l'ensemble des tournées. Dès lors, suivant que l'on détermine un sous-ensemble de tournées a priori ou que l'on génère les tournées au cours de la résolution, cela donne naissance à des heuristiques ou à des méthodes exactes parmi les plus performantes notamment pour le VRPTW.

Nous présentons ici le modèle de flot de véhicules à trois indices qui a été développé par [Fisher et al. 1978] pour le CVRP, en supposant que la taille de la flotte est fixée à m . Tout d'abord, nous listons les indices et les constantes, puis nous définissons les variables de décision comme suit :

Indices et constantes connues :

- $i, j \in S = \{0, 1, \dots, n\}$: le dépôt ou les clients ;
- $v \in V = \{1, \dots, m\}$: les véhicules ;
- d_i : la demande du client $i, i \in C$
- Q : la capacité de chaque véhicule $v \in V$;

- C_{ij} : le coût (la distance) du sommet i au sommet j , $i, j \in S$

Variables de décision :

- x_{ij}^v : la variable binaire, égale 1 si le véhicule v visite le sommet j après le sommet i .

La formulation mathématique du CVRP s'écrit :

$$\text{Minimiser } \sum_{v \in V} \sum_{i \in S} \sum_{j \in S} c_{ij} x_{ij}^v \quad (1.1)$$

Sujette aux:

$$\sum_{v \in V} \sum_{j \in S} x_{ij}^v = 1, \forall i \in C \quad (1.2)$$

$$\sum_{j \in S} x_{ij}^v - \sum_{j \in S} x_{ji}^v = 0, \forall i \in C, \forall v \in V \quad (1.3)$$

$$\sum_{j \in C} x_{0j}^v = 1, \forall v \in V \quad (1.4)$$

$$\sum_{j \in C} x_{j0}^v = 1, \forall v \in V \quad (1.5)$$

$$\sum_{i \in C} \sum_{j \in S} d_i x_{ij}^v \leq Q, \forall v \in V \quad (1.6)$$

$$x_{ij}^v \in \{0, 1\}, \forall i, j \in S, v \in V \quad (1.7)$$

Dans cette formulation, 1.1 représente l'objectif de minimiser la somme des coûts de parcours.

Les équations 1.2-1.7 constituent les contraintes à satisfaire :

- L'équation 1.2 impose que chaque client soit desservi une et une seule fois par un seul véhicule.
- L'équation 1.3 assure que le véhicule qui arrive chez un client est le même que celui qui part de ce client.
- L'équation 1.4 assure que chaque véhicule ne sort qu'une seule fois du dépôt.
- L'équation 1.5 assure le retour unique au dépôt pour chaque véhicule.
- L'inéquation 1.6 définit la contrainte de capacité qui impose que la demande totale des clients de chaque tournée ne doit pas dépasser la capacité Q du véhicule.
- 1.7 est la définition des types de variables utilisés (i.e. contraintes d'intégrité).

Ce type de formulation a souvent été utilisé comme modèle mathématique pour de nouvelles variantes du VRP. En effet, ayant un indice par véhicule, on peut aisément spécifier l'ensemble des contraintes du problème et il est en général facile d'y ajouter de nouvelles contraintes. Une formulation du VRPTW, qui en découle en introduisant de nouvelles variables et en ajoutant des contraintes est présentée dans la section suivante.

1.3 Le problème VRPTW

Le problème de la gestion de tournées de véhicules avec fenêtres de temps (Vehicle Routing Problem with Time Windows ou VRPTW) considéré par [Solomon 1987], [Duhamel 2001], constitue une extension du VRP dans la mesure où l'on ajoute une contrainte temporelle sur le service demandé. Chaque client dispose d'une fenêtre de temps à l'intérieur de laquelle il désire être servi. Le dépôt central possède également une fenêtre de temps que nous désignons couramment comme horizon de service ou temps d'ouverture de la journée. Son rôle est de fixer une plage horaire durant laquelle les véhicules peuvent effectuer leur tournée. Ces contraintes temporelles vont rendre nécessaire l'utilisation de plusieurs véhicules pour satisfaire l'ensemble des clients sur l'horizon de service. On peut vouloir borner le nombre de véhicules à utiliser et dans ce cas des clients risquent de ne pas être servis.

En dépit du caractère fortement restrictif de sa définition, le problème VRPTW conserve un pouvoir descriptif important. Il permet de modéliser un panel étendu d'applications réelles [Duhamel 2001] :

- Service postal : il s'agit d'établir des tournées pour les facteurs de manière à distribuer le courrier le plus efficacement possible, mais on peut aussi réaliser des tournées pour effectuer le ramassage de courrier dans les boîtes aux lettres publiques.
- Coopératives agricoles : il s'agit de mettre en place des tournées de véhicules pour collecter la production de différents exploitants agricoles. Ceci peut concerner le lait, les produits céréaliers, les produits maraîchers ou encore le ramassage d'animaux par exemple.
- Service bancaire : il s'agit de déterminer des routes pour les convoyeurs de fond de manière à réaliser la collecte des recettes des commerçants et des grandes surfaces.
- Ramassage scolaire : il s'agit d'établir les itinéraires de bus pour amener les écoliers des points de collecte vers l'école le matin et inversement le soir.
- ... etc.

Ces quelques exemples permettent d'illustrer la profonde implication de ce type de problème dans la vie courante.

1.3.1 Description

Basée sur la description du CVRP dans la section précédente, le VRPTW est représenté en y ajoutant des paramètres supplémentaires.

Chaque sommet $i \in S$ dispose d'une fenêtre de temps $[e_i, l_i]$, ou $e_i \leq l_i$. Autrement dit, le client désire être desservi par un véhicule à l'intérieur de cet intervalle du temps. Le dépôt central possède lui aussi une fenêtre de temps qui fournit l'horizon de service. Les véhicules ne peuvent pas quitter le dépôt avant e_0 et ils doivent y revenir avant l_0 .

A chaque sommet $i \in S$ est ajouté un temps de service s_i . En général, cette valeur est

identique pour tous les clients et fixée à 0 pour le dépôt.

A chaque arc $(i, j) \in A$ est ajoutée une valeur non négative t_{ij} représentant le temps nécessaire pour se rendre du sommet i au sommet j . Il est courant d'utiliser un temps de trajet proportionnel à la distance Euclidienne.

La présence de fenêtres de temps complique notablement le problème de base [Solomon 1987] et induit deux types de contraintes pour le véhicule :

- La borne supérieure de la fenêtre de temps l_i engendre une contrainte dure. Le véhicule doit impérativement desservir le client i avant l_i .
- La borne inférieure de la fenêtre de temps e_i génère une contrainte sur le temps de début de service chez le client i . Le véhicule est autorisé à arriver avant e_i . Mais dans ce cas, il doit attendre jusqu'à l'ouverture e_i pour pouvoir effectivement desservir le client i .

1.3.2 Formulation

Nous présentons la formulation du problème VRPTW en utilisant le modèle de flot de véhicules à trois indices pour le CVRP déjà exprimé et en y ajoutant les variables, les constantes et les équations suivantes [Solomon 1987] [LeBouthillier 2000]:

Constantes connues :

- e_i : la borne inférieure de la fenêtre de temps du sommet $i \in S$
- l_i : la borne supérieure de la fenêtre de temps du sommet $i \in S$
- t_{ij} : le temps de parcours du sommet i au sommet j , $i, j \in S$.
- s_i : le temps de service chez le client $i \in C$.

Variables à déterminer :

- a_i : instant d'arrivée chez le client $i \in C$.
- b_i : instant de début de service chez le client $i \in C$.
- b_0^v : instant où le véhicule v quitte le dépôt.
- b_{n+1}^v : instant où le véhicule v retourne au dépôt. L'application de l'indice $n+1$ distingue le retour au dépôt de son aller.

- w_i : temps d'attente chez le client $i \in C$.

Supposons qu'un véhicule part vers le client j après avoir servi le client i ($i, j \in C$), et il arrive trop tôt chez le client j , autrement dit, avant e_j .

L'instant où le service débute chez le client j se définit comme étant $b_j = \max\{e_j, a_j\}$ où $a_j = \{b_i + s_i + t_{ij}\}$ et le temps d'attente chez le client j comme étant $w_j = b_j - a_j$.

Il est possible d'écrire les contraintes supplémentaires à la formulation du CVRP pour formuler le problème VRPTW comme ci-dessous :

$$x_{ij}^v = 1 \Rightarrow b_i + s_i + t_{ij} \leq b_j, \forall i, j \in C, v \in V \quad (1.8)$$

$$x_{0j}^v = 1 \Rightarrow b_0^v + t_{0j} \leq b_j, \forall j \in C, v \in V \quad (1.9)$$

$$x_{i0}^v = 1 \Rightarrow b_i + s_i + t_{i0} \leq b_{n+1}^v, \forall i, j \in C, v \in V \quad (1.10)$$

$$e_0 \leq b_0^v \leq l_0, \forall v \in V \quad (1.11)$$

$$e_0 \leq b_{n+1}^v \leq l_0, \forall v \in V \quad (1.12)$$

$$e_i \leq b_i \leq l_i, \forall i \in C \quad (1.13)$$

Ce sont des contraintes temporelles :

- L'inéquation 1.8 représente la relation entre les instants de début de service chez deux clients consécutifs.
- L'inéquation 1.9 et l'inéquation 1.10 sont les contreparties de l'inéquation 1.8 pour les cas du départ au dépôt et du retour au dépôt.
- Les inéquations 1.11 et 1.12 représentent les contraintes d'ouverture liées au dépôt.
- L'inéquation 1.13 indique que le début de service chez le client doit se faire à l'intérieur de l'intervalle défini par fenêtre du temps du client.

1.4 Complexité

Il a été montré que le problème VRP classique est NP-difficile [Len 81]. C'est-à-dire qu'il n'existe pas à ce jour d'algorithme déterministe pouvant résoudre ce problème en temps polynomial. Ce résultat peut être étendu au CVRP, au VRPTW et aux autres extensions du VRP.

Ainsi, s'il est tout à fait possible de déterminer avec une méthode exacte, une solution optimale pour des instances de petite taille, elle devient rapidement inefficace pour résoudre des instances de moyenne ou de grande taille. Or la plupart des problèmes de la vie réelle se

situent justement dans cette seconde catégorie. Lorsque l'on considère des cas plus complexes faisant intervenir des contraintes multiples ou que l'on doit résoudre des problèmes de plus grandes tailles, des méthodes approchées sont alors nécessaires pour fournir des solutions de bonne qualité. Ce sont ces méthodes que nous allons présenter dans la section suivante.

1.5 Les méthodes de résolution

Rappelons que le VRP est un problème d'optimisation combinatoire. Les méthodes de résolution des problèmes d'optimisation combinatoire peuvent être classées en deux grands groupes :

- les méthodes exactes
- les méthodes approchées

Les méthodes exactes fournissent une solution optimale au prix d'un effort de calcul souvent important. Les méthodes approchées cherchent à approcher une solution optimale, autrement dit, elles se contentent d'obtenir des solutions aussi bonnes que possible dans un laps de temps raisonnable mais ne garantissent pas leur optimalité. Les méthodes approchées basées sur les heuristiques sont classées en deux branches : les heuristiques classiques et les métaheuristiques.

Les propriétés de chacune illustrent bien l'un des problèmes essentiels en optimisation combinatoire : l'obligation d'établir un compromis entre la qualité de solution souhaitée et le temps de calcul nécessaire pour l'obtenir (voir la figure 1.2).

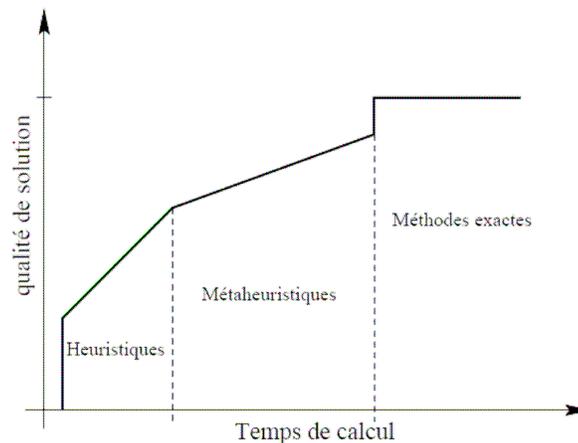


Figure 1. 2 Compromis entre temps de calcul et qualité de la solution

Le but de cette section n'est pas d'exposer toutes les méthodes de résolution des problèmes d'optimisation combinatoire mais de présenter de manière générale les méthodes les plus usitées pour résoudre le VRP.

Depuis l'introduction du VRP formulé par [Dantzig et al. 1959], de nombreuses méthodes ont été proposées pour la résolution de ce type de problème. Il existe des synthèses de ces méthodes dans [Laporte et al. 1987], [Laporte et al. 2000], [Toth et al. 2002]. Les

différentes méthodes (Voir la figure 1.3) et leurs applications pour la résolution du VRP sont présentées dans la suite.



Figure 1.3 Les méthodes de résolution du VRP

1.5.1 Les méthodes exactes

Le problème du VRP étant NP-difficile, les méthodes exactes qui résolvent des instances de taille moyenne avec une durée de temps de calcul raisonnable sont difficiles à développer. Néanmoins plusieurs méthodes ont été développées, pour résoudre efficacement des problèmes avec 50 clients, et même des problèmes comportant 100 clients [Ralphs 2003]. Les méthodes exactes pour le VRP peuvent être classées en trois catégories [Laporte et al. 1987]:

- la programmation dynamique
- la programmation linéaire en nombres entiers
- les méthodes de recherche arborescente (Branch & Bound).

1.5.1.1 Programmation dynamique

La programmation dynamique a été initialement proposée par [Kolen et al. 1987] en reprenant le principe de la méthode de relaxation mise au point par [Christofides et al. 1981] pour le VRP.

Elle est basée sur le principe de Bellman qui dit que : “toute solution du problème initial de taille N contient la solution optimale du sous-problème de taille N-1”. Dans la pratique, on commence à résoudre une famille de problèmes de taille 1, puis on passe à l’étape suivante pour résoudre une famille de problèmes de taille 2. Après un certain nombre d’étapes, on retrouve le problème initial de taille N. À chaque étape, des états intermédiaires doivent être considérés et correspondent à une famille de problèmes à résoudre. Pour que l’approche soit faisable, le nombre d’états intermédiaires et le nombre d’étapes doivent être les plus petits possible. Il est parfaitement connu que seul un très petit nombre d’instances de problèmes d’Optimisation Combinatoire peuvent être résolus par la programmation dynamique (10 à 25 clients dans [Rego et al. 1994]). Notons que c’est l’un des travaux les plus anciens sur le VRPTW.

1.5.1.2 Programmation Linéaire en nombres entiers (Branch & Cut)

Basées sur le modèle de flot de véhicules à trois indices présenté dans la section précédente, les méthodes de type Branch & Cut constituent une des approches les plus performantes pour résoudre le VRP de façon exacte [Toth et al. 2002]. De manière générale, une méthode de Branch & Cut consiste, à partir d’une modélisation en nombres entiers du problème, à considérer une relaxation du modèle dans laquelle les restrictions sur l’intégrité des variables, ainsi qu’éventuellement sur certaines contraintes, sont relâchées.

Etant donnée la solution optimale de la relaxation, deux cas sont considérés: soit la solution est une solution admissible du problème traité et dans ce cas la méthode s’achève, soit il s’agit d’une solution non admissible. A l’aide d’une procédure de séparation, on identifie alors des contraintes valides du problème qui sont violées par la solution courante. Ces contraintes sont alors ajoutées au modèle relâché et le processus est réitéré. Si aucune coupe ne peut être générée ou si la borne induite par la résolution du modèle relâché ne varie plus de façon significative, un processus de séparation évaluation est déclenché. A chaque

noeud de l'arbre la borne est calculée selon le procédé expliqué précédemment.

Dans [Augerat et al 1998] les auteurs ont implémenté une méthode impliquant plusieurs classes de contraintes valides. Ils ont obtenu les solutions optimales d'instances comportant jusqu'à 135 sommets et 7 véhicules en des temps atteignant au maximum 20570 secondes.

L'implémentation en parallèle de leur méthode n'incluant que des routines de séparation pour les contraintes de capacité a conduit à résoudre des instances comportant jusqu'à 100 sommets et 8 véhicules en des temps de calcul pouvant atteindre environ 2 millions de secondes [Ralphs 2003]. [Lysgaard 2004] inclut comme celle de [Augerat et al 1998] plusieurs classes de coupes dont certaines plus fortes que celles considérées précédemment.

1.5.1.3 La méthode de recherche arborescente (Branch & Bound)

C'est une méthode qui utilise un arbre de recherche représentant l'espace des solutions et élague des parties de cet arbre [Aronson 1995] pour accélérer la recherche. Elle peut être employée pour trouver les solutions exactes du VRP, mais en général la résolution nécessite un temps exponentiel en fonction de la taille du problème. Elle construit de manière incrémentale les routes arc par arc. Le problème est transformé sous la forme d'un TSP en créant des dépôts artificiels (égal au nombre de véhicules disponibles) et en les ajoutant au problème initial. Un branchement dans l'arbre est créé en incluant ou en excluant un arc d'une route. La difficulté de ces méthodes est l'obtention d'une borne de bonne qualité pour améliorer la phase d'élagage.

La relaxation Lagrangienne donne de meilleures bornes que les techniques de relaxation classiques (contrainte de capacité, contrainte de connectivité) par [Miller 1995] [Laporte et al. 1986]. Elle a été utilisée par [Kohl et al. 1997] ainsi que par [Fisher et al. 1997] pour le VRPTW. Cette approche permet de résoudre quelques problèmes à 100 clients.

La décomposition Lagrangienne repose sur un schéma de partage des variables. Ceci permet d'obtenir des bornes inférieures pour la solution, que l'on peut ensuite intégrer dans une méthode de Branch & Bound. [Jörnsten et al. 1986], [Madsen 1990] et [Fisher et al. 1997] proposent dans leurs travaux une manière différente pour réaliser la décomposition Lagrangienne pour le VRPTW. Des instances d'une centaine de clients ont ainsi pu être résolues.

[Desrochers et al. 1992] proposent l'application de la technique de génération de colonnes sur la relaxation du VRPTW exprimé comme problème de couverture (set covering problem). Elle repose sur la formulation du VRPTW en utilisant un modèle de partitionnement d'ensemble. Cette approche, très performante permet la résolution d'instances comprenant jusqu'à 100 clients.

Comme nous venons de le voir, les méthodes exactes pour résoudre les problèmes NP-difficiles nécessitent un nombre de calculs qui augmente exponentiellement avec la taille du problème, et ne permettent de résoudre que des instances de petite ou moyenne taille. La plupart des travaux réalisés sur le VRP utilise des méthodes approchées basées sur les heuristiques.

1.5.2 Les heuristiques classiques

Le mot « heuristique » vient du grec « heuriskein » qui signifie « trouver ». Il qualifie tout ce qui sert à la découverte, à l'invention et à la recherche. Depuis les années soixante, de nombreuses heuristiques classiques ont été proposées par les chercheurs. Les récents développements de la théorie de l'optimisation combinatoire ont grandement enrichi les heuristiques, ce qui s'est concrètement traduit par un grand nombre d'applications réussies.

En général, les heuristiques qui ont été développées pour le VRP sont fondées sur des heuristiques initialement développées pour le TSP. Dans cette section nous présentons un survol des heuristiques classiques proposées pour le VRP. Les heuristiques classiques peuvent être réparties en trois classes [Toth et al. 2002] :

- les méthodes constructives
- les méthodes à deux phases
- les méthodes d'améliorations

1.5.2.1 Méthodes constructives

Les méthodes de ce type construisent progressivement des tournées de véhicules par insertion à chaque étape d'un client selon plusieurs critères de mesure de gain. Les plus connues sont la méthode des économies (savings) de Clarke et Wright [Clarke et al. 64] et les méthodes d'insertion.

1.5.2.1.1 La méthode des économies (savings)

Dans cette méthode, le nombre de véhicules est une variable de décision. Il nécessite le calcul du gain possible pour toute paire de sommets $(i, j) \in A$, et $i, j \in C$ liée à la fusion de routes. Partant d'une solution initiale où chaque client est desservi par une tournée différente, on regarde le gain réalisé par des fusions possibles de routes. Pour calculer le gain lié à une fusion de routes, on évalue $S_{ij} = c_{i0} + c_{0j} - c_{ij}$. On trie ensuite les gains possibles par ordre décroissant et on choisit la fusion réalisant le gain le plus grand. Pour les étapes suivantes, on cherchera à fusionner des routes deux à deux soit au début soit en fin de tournée. Par exemple si on considère une route $(0, i, \dots, j, 0)$ on va chercher à déterminer le meilleur gain S_{ki} ou S_{jl} parmi toutes les fusions possibles qui peuvent être réalisées en formant une route réalisable (i.e. respectant les contraintes du problème) par la fusion de la route courante et d'une autre route se terminant par $(k, 0)$ ou commençant par $(0, l)$. La méthode s'arrête dès qu'il n'est plus possible de trouver une fusion améliorante.

1.5.2.1.2 Les méthodes d'insertion

Les méthodes d'insertion sont constituées de deux phases, la première sélectionne le prochain nœud à insérer et la deuxième réalise l'opération d'insertion.

La méthode d'insertion proposée par Mole et Jameson [Mol 76] construit les tournées de manière séquentielle. Partant de la tournée constituée par le sommet non desservi le plus éloigné du dépôt, elle consiste à insérer les sommets non inclus sur une tournée existante. [Christofides et al. 1979] construisent alternativement en séquentiel et en parallèle les différentes tournées. Suite à une insertion, l'heuristique 3-Opt [Lin 1965] est appliquée pour

améliorer le coût de la tournée courante dans les deux méthodes.

Solomon décrit quelques méthodes de construction de tournées reposant sur l'insertion séquentielle des clients [Solomon 1987]. [Potvin et al. 1993], puis [Russell 1995] proposent des méthodes de construction de tournées en parallèle.

1.5.2.2 Méthodes à deux phases

Les méthodes à deux phases se répartissent en deux classes d'algorithmes suivant l'ordre dans lequel les deux phases sont effectuées :

- les algorithmes de partition d'abord puis de parcours
- les algorithmes de parcours d'abord puis de partition.

Dans la première catégorie se trouvent des méthodes très connues telles que l'algorithme de balayage et les algorithmes de pétales.

1.5.2.2.1 L'algorithme de balayage

L'algorithme de balayage (sweep) [Gillett et al. 1974] repose sur l'hypothèse qu'à chaque sommet du graphe sont associées des coordonnées polaires définies en prenant le dépôt pour origine et un sommet choisi de façon quelconque. Les sommets sont ensuite classés par coordonnée angulaire croissante. Partant du sommet de la plus petite coordonnée angulaire non affectée, les sommets sont inclus successivement dans le groupe courant tant que les contraintes sont vérifiées. Lorsqu'une des contraintes est violée, la constitution du groupe est stoppée et l'on passe à l'élaboration du groupe suivant. Le processus est répété jusqu'à ce que tous les sommets soient inclus dans un groupe. Les groupes de sommets étant formés, un TSP est ensuite résolu pour chacun d'eux.

Si la satisfaction des contraintes de capacité peut se faire aisément lors de l'inclusion d'un sommet dans un groupe, celle portant sur la longueur de la tournée ou sur les fenêtres de temps nécessite la résolution d'un problème du voyageur de commerce (TSP) lors de chaque insertion.

1.5.2.2.2 Les algorithmes de pétales

Les algorithmes de pétales sont les extensions de l'algorithme de balayage. Ils consistent à générer un sous-ensemble de T tournées (appelées pétales) a priori intéressantes puis à résoudre le modèle de partition d'ensemble restreint à ces tournées. La meilleure méthode de ce type pour le VRP est la méthode 2-pétales proposée par [Renaud et al.1996]. Elle consiste à générer non seulement des tournées individuelles mais également des blocs de deux tournées qui, soit s'entrecroisent, soit sont incluses l'une dans l'autre.

1.5.2.2.3 L'algorithme de parcours d'abord puis de partition

Parmi les algorithmes de parcours d'abord puis de partition, l'approche proposée par [Beasley 1983] pour le VRP consiste à résoudre dans un premier temps un tour TSP en omettant les contraintes, puis à déterminer une partition optimale de ce tour en y intégrant les contraintes. Cette dernière est obtenue en résolvant un problème de plus court chemin dans un graphe sans cycle.

1.5.2.3 Méthodes d'améliorations

Deux classes de méthodes d'amélioration ou de recherche locale pour le VRP peuvent être distinguées. La première comprend les méthodes qui essaient d'améliorer individuellement chacune des tournées. Il s'agit d'algorithmes exacts ou heuristiques pour le problème du voyageur de commerce avec ou sans fenêtre de temps (TSPTW). Plusieurs heuristiques classiques pour le TSP telles que les algorithmes de type k-opt proposés par Lin [Lin 1965], la méthode Or-Opt [Or 1976], ou encore la procédure GENIUS [Gendreau et al.1992] ont été adaptés au TSPTW [Gendreau et al.1998a].

La seconde inclut des heuristiques d'amélioration de la solution courante en modifiant plusieurs tournées simultanément. Elles sont typiquement basées sur la définition d'un mécanisme d'échange d'arêtes ou d'arcs entre tournées. Trois types de mécanisme d'échange, que l'on retrouve dans des approches basées sur les métaheuristiques, sont considérés dans [Breedam 1995]. Le premier, appelé relocalisation de chaîne, consiste à déplacer au plus k clients consécutifs d'une tournée vers une autre ; le second, nommé échange de chaîne, vise à échanger k clients consécutifs d'une tournée avec ℓ clients consécutifs d'une seconde tournée ; le troisième, appelé échange mixte de chaîne, combine ces deux opérations en examinant laquelle engendre la meilleure amélioration.

Un type de modification s'apparentant à la seconde classe a été proposé pour le VRPTW par [Thompson et al. 93]. [Russell 1995] propose une approche constructive parallèle partant du nombre minimal de tournées.

A cet ensemble de méthodes se sont adjointes depuis la fin des années quatre-vingts des approches basées sur les métaheuristiques [Teghem et al. 2003].

1.5.3 Les métaheuristiques

Le mot « métaheuristique », présenté la première fois par [Glover 1986], dérive de la composition de deux mots grecs : Heuristique, dérivé du verbe « heuriskein » qui signifie «trouver », et le suffixe « méta » qui signifie « au delà » ou «dans un niveau supérieur ». Avant que ce terme ne soit adopté, les métaheuristiques étaient souvent appelées heuristiques modernes. Plusieurs définitions ont été proposées pour expliquer clairement ce qu'est une métaheuristique [Osman 1996] [Voß et al. 1998] [Stützle 1999]. Aucune de ces définitions n'est universellement reconnue.

Soit S un ensemble de solutions à un problème d'optimisation et soit f une fonction qui mesure la valeur $f(s)$ de chaque solution s dans S . On peut également définir S comme un ensemble de solutions ne satisfaisant pas nécessairement toutes les contraintes du problème considéré; les violations de contraintes sont alors également prises en compte par la fonction f qui pénalise ces violations.

L'objectif d'une méthode d'optimisation est de déterminer une solution dans S qui minimise la fonction f . En d'autres termes, il s'agit de déterminer $s^* \in S$ tel que

$$f(s^*) = \underset{s \in S}{\text{Min}} f(s)$$

Une structure de voisinage (ou tout simplement un voisinage) est une fonction V qui associe un sous-ensemble de S à chaque solution $s \in S$. La solution $s' \in V(s)$ est dite voisine de s .

Une solution $s \in S$ est un minimum local relativement à la structure de voisinage V si $f(s) \leq f(s')$ pour tout $s' \in V(s)$.

Une solution $s \in S$ est un minimum global si $f(s) \leq f(s')$ pour tout $s' \in S$.

Les métaheuristiques sacrifient la garantie d'optimalité ou d'approximation avec en contrepartie l'espoir de trouver très rapidement de bonnes solutions dans S .

Les métaheuristiques sont classées en deux groupes selon le nombre de solutions manipulées :

- Les métaheuristiques à solution unique sont basées sur une recherche de voisinage qui commence avec une solution initiale puis l'améliore pas à pas en choisissant une nouvelle solution dans le voisinage de la solution courante. Les méthodes de descente, le recuit simulé et la recherche tabou sont des exemples typiques de métaheuristique à solution unique.
- Les métaheuristiques à population de solutions améliorent une population de solutions au fur et à mesure des itérations. L'intérêt de ces méthodes est d'utiliser la population comme facteur de diversité. Les métaheuristiques à population de solutions les plus connues sont l'algorithme génétique et l'algorithme des colonies de fourmis.

1.5.3.1 Le recuit simulé

Les premières utilisations de la méthode du recuit simulé remontent au début des années 80 par [Kirkpatrick et al. 1983] et [Cerny 1985]. L'idée fondamentale de la méthode est de permettre des mouvements ayant pour résultats des solutions de plus mauvaises qualités que la solution courante (mouvements défavorables) dans le but d'échapper au minimum local.

La méthode s'inspire d'un processus utilisé en métallurgie dans les années 50 [Metropolis 1953]. En Métallurgie, l'obtention d'un cristal parfait se fait grâce à la méthode du recuit. On porte un métal à une température suffisamment élevée pour qu'il soit dans l'état liquide. Puis, à partir de cet état, on abaisse la température, et de ce fait les atomes se réorganisent en une autre nouvelle structure.

Une même structure initiale peut donner différentes structures finales selon la façon dont on baisse la température. Si celle-ci baisse trop brutalement, on risque d'atteindre un état métastable qui ne correspond pas à l'état fondamental, minimum absolu d'énergie interne, mais correspond à un minimum local de l'énergie. Il est donc essentiel que l'abaissement de la température se fasse très lentement et très régulièrement. Pour faire disparaître d'éventuels défauts dans la structure du cristal, on utilise la technique du recuit. On réchauffe un peu le

métal, afin que les atomes aient plus de liberté de mouvement. En chauffant suffisamment, on donne assez d'énergie pour qu'ils sortent de l'optimum local. En abaissant à nouveau la température régulièrement, ils pourront atteindre l'optimum global.

En s'inspirant de ce procédé, la méthode du recuit simulé a été développée (voir l'Algorithme 1.1 du recuit simulé). Elle consiste à définir une fonction f à minimiser comme fonction d'énergie. Une solution peut être considérée comme un état donné de la matière.

Algorithme 1.1 Pseudo-Code de l'algorithme du recuit simulé

```

générer une solution initiale  $s_0$ 
générer une température initiale  $T$ 
initialiser la meilleure solution  $s^*=s_0$ 
 $f_{\min} = f(s^*)$ 
REPETER
  REPETER
    Solution courante  $s$ 
    générer un voisinage  $V(s)$ 
    sélectionner aléatoirement une voisine  $s' \in V(s)$ 
    calculer  $\Delta = f(s') - f(s)$ 
    SI  $\Delta < 0$ 
       $s = s'$ 
       $f(s') = f(s)$ 
    SINON
       $p = \exp^{-\frac{\Delta}{T}}$ 
      Générer un nombre réel aléatoire  $r$  dans  $[0,1]$ 
      SI  $r \leq p$ 
         $s = s'$ 
         $f(s') = f(s)$ 
      FIN SI
    FIN SI
  SI  $f(s) < f_{\min}$ 
     $s^*=s$ 
     $f_{\min} = f(s)$ 
  FIN SI
JUSQU'À équilibre thermodynamique atteint
décroître température  $T$ 
JUSQU'À conditions d'arrêt satisfaites

```

La méthode commence par la génération d'une solution initiale soit aléatoirement soit par une heuristique et l'initialisation du paramètre de température T . Ensuite, à chaque itération, un voisinage $V(s)$ est généré pour la solution courante s , une solution voisine

$s' \in V(s)$ est sélectionnée au hasard. Soit alors $\Delta = f(s') - f(s)$ la variation dans la qualité induite par le mouvement de s à s' . Si cette variation est négative, ce mouvement apporte alors une amélioration et s' est automatiquement acceptée. En revanche, si la variation est positive ou nulle, le mouvement est accepté avec la probabilité qui est généralement calculée selon la distribution de Boltzmann $p = \exp^{-\frac{\Delta}{T}}$, où T désigne la température courante.

L'efficacité du recuit simulé dépend fortement du choix des paramètres de contrôle, dont le réglage reste lui aussi très empirique. Les principaux paramètres de contrôle sont les suivants :

- la valeur initiale de la température,
- la fonction de décroissance de la température,
- le critère de changement de palier de température,
- les critères d'arrêt.

La température joue un rôle important dans la méthode du recuit simulé, qui influe sur la probabilité d'acceptation d'une solution défavorable. En général, on choisit une température initiale suffisamment élevée qui donne une plus grande liberté pour l'exploration de l'espace de recherche. Puis, petit à petit, la température décroît jusqu'à atteindre une valeur proche de 0, ce qui signifie que la méthode n'acceptera plus de détériorer une solution.

Une forte décroissance de température risque de piéger la recherche dans un minimum local, alors qu'une faible décroissance au début du processus entraîne une convergence très lente de l'algorithme. Un compromis pour adapter la décroissance de la température à l'évolution du processus consiste à utiliser une variation logarithmique. La loi logarithmique de décroissance de la température, qui assure la convergence théorique du recuit simulé, est la

suivante : $T_k = \frac{\mu}{\text{Log}(1+k)}$. Où k est le nombre de paliers de température effectués, et μ est

une constante positive. En pratique, on adopte souvent une décroissance géométrique $T_{k+1} = \alpha \cdot T_k$, avec ($0 < \alpha < 1$), car la loi précédente induit un temps de calcul prohibitif.

Enfin, le critère d'arrêt du processus varie suivant les approches. Une possibilité consiste à fixer le nombre d'itérations sans amélioration de la meilleure solution trouvée. Mais le plus souvent, on choisit de stopper au bout d'un certain nombre de paliers de température, ce qui revient aussi à arrêter le processus d'optimisation dès que la température est jugée suffisamment basse. On peut encore l'arrêter lorsque la baisse de température n'est plus significative.

Il a été prouvé que l'algorithme peut converger vers la solution optimale. Mais ceci peut prendre un temps de calcul important dans certains cas.

Les premières applications du recuit simulé pour résoudre le VRP ont été proposées par [Robuste et al. 1990] et [Alfa et al 1991]. Dans la première, le voisinage est défini par

plusieurs mécanismes (renversements, mouvements de parties de la tournée, etc.). La seconde utilise une heuristique de parcours d'abord puis partition pour construire une première solution. Le voisinage est défini par l'opérateur de recherche 3-opt. L'une des meilleures adaptations au VRP est due à [Osm 1993]. [Breedam 1995] a proposé trois adaptations du recuit simulé au VRP qui se distinguent par le schéma de génération des solutions voisines de la solution courante. Bien que cette méthode produise généralement de bons résultats, elle donne parfois des solutions de mauvaise qualité et trouve rarement la meilleure solution.

1.5.3.2 La recherche Tabou

La recherche Tabou (RT) est une métaheuristique originalement développée par Glover [Glover 1977]. Elle est basée sur des idées simples, mais elle est néanmoins très efficace. Cette méthode combine une procédure de recherche locale avec un certain nombre de règles et de mécanismes permettant à celle-ci de surmonter l'obstacle des optima locaux, tout en évitant les problèmes de cycle. Elle a été appliquée avec succès pour résoudre de nombreux problèmes difficiles d'optimisation combinatoire dont le problème de routage de véhicules (VRP).

Dans une première phase, la méthode de recherche tabou part d'une solution quelconque s appartenant à l'ensemble de solutions S , puis on se déplace vers une solution dans le voisinage $V(s)$ de s . L'algorithme va explorer itérativement un sous-ensemble de l'espace de solutions S . Afin de choisir la meilleure voisine dans le voisinage $V(s)$, l'algorithme évalue la fonction objectif f en chaque $s' \in V(s)$, et retient la meilleure voisine qui améliore la valeur de la fonction objectif f ou si ce n'est pas possible celle qui la dégrade le moins. L'algorithme Tabou classique est présenté dans l'algorithme 1.2.

Algorithme 1.2 Pseudo-code de la méthode de recherche tabou

```

générer une solution initiale  $s_0$ 
initialiser la liste tabou  $T \in \Phi$ 
initialiser la meilleure solution  $s^*=s_0$ 
 $f_{\min} = f(s^*)$ 
REPETER
    Solution courante  $s$ 
    générer un voisinage  $V(s)$ 

    sélectionner la meilleure voisine  $s' \in V(s)$  tq  $s' \notin T$ 

    insérer  $s'$  dans la liste tabou  $T$ 
    supprimer le mouvement le plus ancien dans  $T$ 
     $s = s'$ 
    SI  $f(s) < f_{\min}$ 
         $s^*=s$ 
         $f_{\min} = f(s)$ 
    FIN DE SI
JUSQU' A conditions d'arrêt satisfaites

```

L'originalité de la méthode de recherche Tabou par rapport aux méthodes de recherches locales réside dans le fait que l'on retient la meilleure voisine, même si celle-ci est plus mauvaise que la solution courante d'où l'on vient. Ce caractère autorisant les dégradations de la fonction objectif évite à l'algorithme d'être piégé dans un minimum local (voir la Figure 1.4).

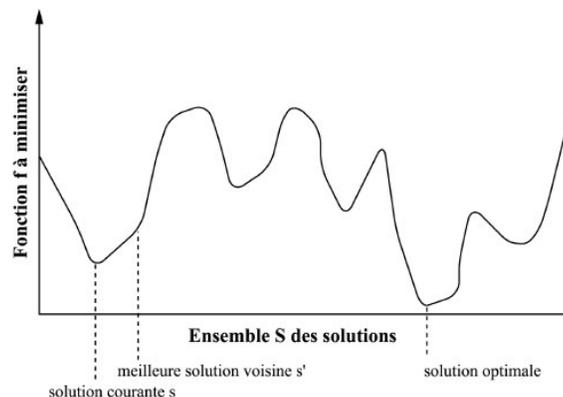


Figure 1. 4 La recherche de la solution optimale peut détériorer la solution courante

Cependant il induit un risque de création de cycles. En effet, lorsque l'algorithme quitte un minimum local par acceptation de la dégradation de la fonction objectif, il peut revenir sur ses pas, à l'itération suivante. Pour régler ce problème, l'algorithme a besoin d'une mémoire pour conserver pendant un moment la trace des dernières meilleures solutions déjà visitées. Ces solutions sont déclarées tabous, 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 la formation de cycles de longueur inférieure ou égale à la longueur de la liste tabou. Il dirige l'exploration de la méthode vers des régions du domaine de solutions non encore visitées.

La liste tabou est généralement gérée comme une liste "circulaire" : on élimine à chaque itération la solution tabou la plus ancienne, en la remplaçant par la nouvelle solution retenue. La longueur L de la liste tabou contrôle la mémoire du processus de recherche. Avec de petites valeurs de L , la recherche se concentre sur de petites régions dans l'espace de recherche. À l'opposé, une grande longueur de L force le processus de recherche à explorer des régions larges, car il empêche de revisiter un grand nombre de solutions. La taille de la liste tabou peut varier pendant la recherche, ce qui mène à des algorithmes plus robustes.

La recherche Tabou se base sur les notions du voisinage et de mémoire tabou. L'élaboration des structures de voisinage et l'identification de la taille de la liste tabou sont parfois difficiles à déterminer.

Comme pour le recuit simulé, il est nécessaire de définir le critère d'arrêt. La différence est que l'on ne peut pas se baser sur un paramètre de contrôle global. Dès lors, le critère d'arrêt peut être un certain nombre d'itérations, ou un certain nombre d'itérations sans amélioration, ou bien une certaine durée d'exécution.

Cette méthode métaheuristique est largement utilisée pour le VRP. Les premières

implémentations ([Willard 1989] et [Pureza et al. 1991]) n'offraient pas des résultats impressionnants mais les travaux suivants ont permis de les rendre de plus en plus efficaces. [Taillard 1993], [Tarantilis et al. 02] et [Tot 03] utilisent le voisinage obtenu par la stratégie de λ -échanges proposée par [Osm 1993]. Pour résoudre le VRPTW en appliquant la recherche Tabou, [Chiang et al. 1997] a utilisé la stratégie de λ -échanges. [Rochat et al. 1995] ainsi que [Cordeau et al. 2001] considèrent le voisinage obtenu par le déplacement d'un sommet d'une tournée à une autre en réalisant une insertion simple. [Taillard et al. 1997] ont recours à un mouvement plus complexe appelé cross-exchange qui consiste à échanger deux chaînes de sommets consécutifs entre deux tournées. Nous trouvons aussi une version parallèle dans [Gendreau et al. 1999]. Les méthodes Tabou sont actuellement les plus performantes pour traiter efficacement le VRPTW.

1.5.3.3 GRASP et VNS

Le recuit simulé et la recherche tabou ainsi que la méthode de descente sont probablement les métaheuristiques à solution unique les plus connues et les plus utilisées. Il existe cependant d'autres méthodes ayant également leur cote de popularité.

GRASP (Greedy Randomized Adaptive Search Procedure) est apparue au milieu des années 90 [Feo 1995]. Elle apporte une réponse simple et élégante aux problèmes soulevés par l'exploration globale de l'espace des solutions pour les procédures de recherche locale. L'idée centrale consiste à encapsuler le processus de recherche locale dans un mécanisme plus général de génération contrôlée de solutions. Cette méthode se compose de deux phases : la phase de construction de solution réalisable et la phase d'amélioration de cette solution.

La recherche à voisinage variable (Variable Neighborhood Search ou VNS) est une métaheuristique d'inspiration récente sur la mise en place et l'exploitation de plusieurs structures de voisinages durant l'exploration de l'espace des solutions [Mladenovic et al. 1997]. Leur idée consiste à déterminer quelle structure de voisinage activer en fonction d'événements spécifiques pouvant survenir durant la recherche.

1.5.3.4 Algorithmes évolutionnaires et Algorithmes génétiques

Les Algorithmes évolutionnaires (AE) sont inspirés du fonctionnement de l'évolution naturelle. Selon Darwin, les mécanismes à l'origine de l'évolution des êtres vivants reposent sur la compétition qui sélectionne les individus les plus adaptés à leur environnement en leur assurant une descendance, ainsi que sur la transmission aux enfants des caractéristiques utiles qui ont permis la survie des parents. Ce mécanisme d'héritage se fonde notamment sur une forme de coopération mise en oeuvre par la reproduction.

Quelques chercheurs des années 1950 ont tenté d'adapter ce principe à l'ingénierie. Mais la faible puissance des machines de l'époque et des connaissances de la génétique naturelle n'a pas permis d'avoir des résultats concluants. Plus tard, dans les années 1960 et 1970, grâce à l'augmentation significative de la puissance des calculateurs, de nombreuses tentatives de modélisation de l'évolution ont été entreprises. Parmi celles-ci, trois écoles utilisant le principe de l'évolution de façon différente apparaissent :

- Les stratégies d'évolution (ES). La méthode fut initialement proposée par Ingo Rechenberg, en 1965 [Rechenberg 1965]. La méthode fut ensuite développée durant la fin des années 1960, principalement par les travaux de Ingo Rechenberg, P. Bienert et Hans-Paul Schwefel sur la conception de profils aérodynamiques [Beyer 2001]. Dans sa version de base, la méthode manipule itérativement un ensemble de vecteurs de variables réelles, à l'aide d'opérateurs de mutation et de sélection. Un énorme progrès a été apporté par les techniques adaptatives d'ajustement des paramètres de mutation, et ce sont sans contestation les meilleurs algorithmes pour les problèmes purement numériques.
- La programmation évolutionnaire (EP), qui a été proposée dans la moitié des années 1960 par L.J.Fogel [Fogel et al. 1966], visait à faire découvrir des automates à états finis par une succession de sélections et mutations pour l'approximation de séries temporelles. Elle a rapidement été adaptée pour des espaces de recherche très variés.
- Les algorithmes génétiques (GA), qui ont été développés en 1975 par J.H.Holland [Holland 1975], ont pour objectif de comprendre les mécanismes sous-jacents de systèmes capables de s'auto adapter à leur environnement.

Dans le monde des algorithmes évolutionnaires, les individus soumis à l'évolution sont des solutions, plus ou moins performantes, du problème posé. Ces solutions appartiennent à l'espace de recherche du problème d'optimisation. La fonction fitness $f(x)$, qui évalue le degré d'adaptation d'un individu x à l'environnement, est construite à partir d'une fonction objectif du problème.

L'ensemble des individus traités simultanément par l'algorithme évolutionnaire constitue une population. Elle évolue durant une succession d'itérations appelées génération jusqu'à ce qu'un critère d'arrêt, qui prend en compte a priori la qualité des solutions obtenues, soit vérifié.

Durant chaque génération, une succession d'opérateurs est appliquée aux individus d'une population pour engendrer la nouvelle population à la génération suivante. Lorsqu'un ou plusieurs individus sont utilisés par un opérateur, ils sont considérés comme des parents. Les individus résultats de l'application de l'opérateur sont alors appelés enfants.

Algorithme 1.3 Pseudo-code de l'algorithme évolutionnaire

```

Initialiser la population POP en générant N individus
Evaluation des individus de PO (i.e. calcul des valeurs de fitness pour tous
les individus) ;
REPETER
    Sélection
    Reproduction (croisement et mutation)
    Evaluation des enfants
    Remplacement
JUSQU' A conditions d'arrêt satisfaites

```

L'algorithme 1.3 montre le Pseudo-code de l'algorithme évolutionnaire. Il part d'une population initiale de N individus (chromosomes) qui représentent les solutions du problème. Chaque individu est évalué par le fitness. A chaque génération, les opérateurs suivants sont utilisés pour construire la nouvelle population :

- Sélection
La sélection consiste à choisir les individus qui vont participer à la reproduction de la population future. L'opérateur de sélection choisit, de façon déterministe ou selon une méthode probabiliste (roulette, tournoi, etc.), un individu pour qu'il participe à l'étape de reproduction pour former la nouvelle population.
- Croisement
La reproduction exploite essentiellement les opérateurs de croisement et de mutation sur les individus sélectionnés précédemment.
L'opérateur de croisement engendre un ou plusieurs enfants à partir des combinaisons de deux ou plus de deux parents avec une certaine probabilité appelée taux ou probabilité de croisement. Cet opérateur permet de bien exploiter le domaine de variation des individus, et de diriger la recherche vers des régions intéressantes de l'espace d'étude en utilisant la connaissance déjà présente dans la population courante.
- Mutation
L'opérateur de mutation modifie un individu pour en former un autre. Il s'applique avec une certaine probabilité, appelée taux ou probabilité de mutation.
L'opération de mutation protège les algorithmes génétiques des pertes prématurées d'informations pertinentes. Elle permet d'introduire une certaine information dans la population, qui a pu être perdue lors de l'opération de croisement. Ainsi elle participe au maintien de la diversité, utile à une bonne exploration du domaine de recherche.
- Remplacement
Le remplacement détermine quels individus devront disparaître de la population à chaque génération. La stratégie de remplacement peut être mise en oeuvre sous plusieurs formes, tels que les stratégies d'évolution « (μ, λ) -ES », la stratégies stationnaire, la stratégies élitisme, etc.

L'évolution stoppe quand le niveau de performance souhaité est atteint, ou qu'un nombre fixé de générations s'est écoulé sans amélioration de l'individu le plus performant.

Les différentes algorithmes évolutionnaires ne diffèrent que sur les implantations des opérateurs et sur les procédures de sélection et remplacement de la population. Bien que leur but soit différent à l'origine, ils sont maintenant surtout utilisés pour résoudre des problèmes d'optimisation.

Les algorithmes génétiques sont les plus connus des algorithmes évolutionnaires. L'efficacité de l'algorithme génétique dépend fortement de la représentation choisie pour le codage des solutions et des opérateurs de reproduction utilisés. Elle dépend aussi fortement du choix de ces paramètres de contrôle suivants, dont le réglage reste très empirique :

- la taille de la population N ,
- le type et le taux de croisement,
- le type et le taux de la mutation,
- le type de la sélection,
- le taux d'élitisme.

Il existe beaucoup de travaux basés sur l'AG pour résoudre le VRP et ses extensions. Les travaux de [Rego et al. 1994] présentent plusieurs méthodes de résolution (heuristiques et métaheuristiques) du problème VRP et TSP. Une première étude de comparaison entre un

algorithme génétique, un recuit simulé et une recherche tabou a été faite par [Breedam 1996]. [Pereira et al. 2002] proposent un algorithme génétique avec une nouvelle représentation chromosomique du VRP. Christian Prins [Prins 2001], [Prins 2004] et Baker [Baker et al. 2003] ont utilisé l'algorithme génétique comme méthode de résolution avec aussi une nouvelle représentation des solutions. [Mester et al. 2005] et [Mester et al. 2007] utilisent les stratégies d'évolution pour résoudre le VRP et son extension VRPTW.

1.5.3.5 L'algorithme des colonies de fourmis

Les algorithmes de fourmis sont issus des travaux de Coloni, Dorigo et Maniezzo [Coloni et al. 1992] au début des années 90. Ils reposent sur une analogie avec le comportement collectif des fourmis à s'organiser pour la recherche de nourriture.

Les fourmis ont la particularité d'employer, pour communiquer, des substances volatiles appelées phéromones. Les fourmis explorent leur environnement immédiat en laissant derrière elles de la phéromone. Elles se servent de cette phéromone pour se guider et tendent naturellement à suivre les traces existantes. En l'absence de phéromone, leur exploration devient complètement aléatoire. Elles possèdent donc une vision très limitée de leur environnement. Une fois la nourriture trouvée, elles se servent des traces qu'elles viennent de déposer pour retrouver le chemin du retour vers le nid. Durant le trajet, elles laissent de nouveau sur leur passage de la phéromone, en quantité proportionnelle à l'intérêt de la source de nourriture (pondération). Le chemin est donc fortement imprégné de phéromones et constitue une piste de choix pour la colonie de fourmis. Plus la source de nourriture a été jugée intéressante, plus les fourmis auront tendance à suivre ce chemin. Ainsi, peu à peu, les traces vers les sources de nourriture seront de plus en plus marquées.

Il est intéressant de noter que les fourmis ont la remarquable propriété de suivre naturellement le chemin le plus court vers la source de nourriture. Nous reprenons le scénario de la figure 1.4 présenté dans [Dorigo et al. 1996] pour bien expliquer.

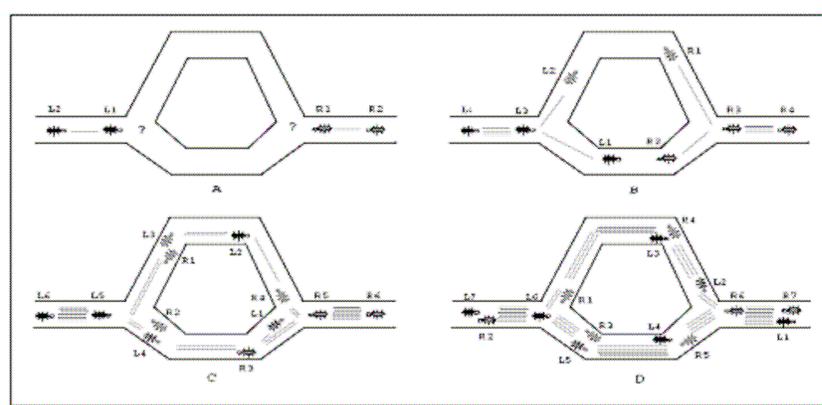


Figure 1.5 Comment les fourmis trouvent le plus court chemin

Dans la figure A, les fourmis arrivent à un point de décision dans lequel elles doivent décider de tourner à gauche ou à droite. Elles n'ont pas une idée sur le choix qui mène vers le chemin le plus court, elles choisissent donc aléatoirement. Les figures B et C montrent que le

nombre de lignes tracées est proportionnel à la quantité de phéromones déposée sur les pistes. Après une période donnée, les fourmis qui ont choisi d'emprunter le chemin à droite reviennent plus vite que les autres, ce qui permet d'accumuler une quantité de phéromones sur ce chemin (figure D) plus importante. La piste courte va alors être renforcée et, à terme, sera choisie par la grande majorité des fourmis.

Colomi, Dorigo et Maniezzo [Colomi et al. 1992] ont adapté ce principe au domaine de l'Optimisation Combinatoire. Pour ce faire, ils ont associé le voisinage du nid à l'espace des solutions. Chaque solution s'apparente à une source de nourriture dont la qualité est fournie par la fonction d'évaluation. Chaque fourmi est assimilée à un processus répétitif de construction de solutions. La construction est guidée par un ensemble de données globales "la phéromone". Cet ensemble est typiquement une mémoire sur les attributs de solution, régulièrement mise à jour par les processus de construction (en fonction de la qualité de la solution) et par un mécanisme simulant l'évaporation de la phéromone. Le problème majeur de ce modèle réside dans sa grande sensibilité vis-à-vis du principe de la phéromone. En fait, la qualité des résultats fournis par la métaheuristique dépend très fortement de l'importance accordée à la phéromone. Lorsque la pondération est trop élevée, le mécanisme de construction tend à favoriser la génération de solutions trouvées dans les itérations précédentes. A l'inverse, lorsqu'il est trop faible, il tend à produire des solutions aléatoires. Le pseudo-code de l'algorithme de la colonie de fourmis est présenté dans l'algorithme 1.4 :

Algorithme 1.4 Pseudo-Code de la méthode de colonie de fourmis

1. Initialiser les traces,
2. Tant qu'un critère d'arrêt n'est pas satisfait, répéter en parallèle pour chacune des p fourmis:
 - (a) Construire une nouvelle solution à l'aide des informations contenues dans les traces et une fonction d'évaluation partielle.
 - (b) Evaluer la qualité de la solution.
 - (c) Mettre à jour les traces.

Nous pouvons améliorer la performance en tenant compte des modifications suivantes qui touchent pratiquement tous les aspects de la métaheuristique :

- le processus de génération de solutions s'est vu intégrer une phase d'amélioration itérative afin de fournir des solutions de meilleure qualité.
- l'actualisation des traces a évolué pour permettre l'application explicite de stratégies de fouille ou de prospection. Seules les meilleures solutions donnent lieu à une mise à jour des traces.
- la coordination des différentes fourmis a été sensiblement améliorée par l'apparition d'une fourmi particulière, la reine. Cette dernière prend en charge la gestion de la mémoire collective de la colonie (les traces) et prend les décisions stratégiques en fonction de l'évolution globale de la recherche. Elle constitue de fait une sorte d'agent "intelligent".

L'application de ces différents aménagements du modèle initial a permis d'augmenter

sensiblement les performances de la métaheuristique. Ceci est particulièrement vrai pour le problème VRP [Gambardella et al. 2003]. L'algorithme « Ant System » (AS) a été la première fois appliqué pour le problème de voyageur de commerce (TSP) [Dorigo et al.1996] et L'algorithme « Ant Colony System » (ACS), a été introduit pour améliorer les performances de l'algorithme AS sur des problèmes de grandes tailles [Dorigo et al. 1997]. La première utilisation des colonies de fourmis pour le VRP a été proposée par [Kawamura 1998]. La principale caractéristique de la méthode de [Reimann et al. 2004] est l'inclusion des techniques provenant de la méthode des économies. Par la suite, [Doerner et al. 2004] ont proposé une extension de cette méthode pour le VPR exploitant le parallélisme.

1.5.3.6 Métaheuristiques pour le VRP

La complexité du VRP exige des méthodes métaheuristiques pour la plupart des exemples réels. Dans les sections précédentes, nous avons largement examiné l'évolution des méthodologies métaheuristiques pour le problème VRP. La qualité des solutions obtenues avec ces différentes techniques est bien souvent meilleure que les heuristiques traditionnelles. En même temps, les métaheuristiques exigent plus de temps de calcul et sont plus complexes à mettre en application et à calibrer [Bräysy et al. 2001].

Le rapport de recherche du GERAD [Cordeau et al. 2004] compare les 9 meilleures méthodes actuelles pour le VRP. Ces méthodes sont :

1. GTS (Granular Tabu Search) [Toth et al. 2003] : Cette méthode utilise une recherche tabou et élimine les tronçons de taille supérieure à un seuil de granularité dans le graphe représentant la région de service.
2. L'heuristique VRTR (Variable-length neighbourhood list Record-to-Record) [Li et al. 2005] basée sur l'heuristique des gains de [Clarke et al. 1964] et l'opérateur de recherche locale 2-Opt.
3. UTSA (Unified Tabu Search Algorithm) [Cordeau et al. 1997] : Cette méthode utilise principalement la recherche tabou, elle utilise seulement une solution initiale et à partir de cette solution elle construit la liste tabou en utilisant l'opérateur « client réinsertion ».
4. VLNS (Very Large Neighbourhood Search) [Ergun et al. 2003] : Cette méthode est basée sur l'opérateur de recherche locale 2-Opt, appliqué sur plusieurs routes.
5. L'algorithme génétique hybride de [Prins 2004] : Cette méthode hybride utilise un algorithme génétique combiné avec une recherche locale. Le chromosome est représenté par une séquence de clients et un algorithme de plus court chemin est utilisé pour découper de manière optimale cette séquence en tournées. L'opérateur de mutation consiste en une recherche locale (client réinsertion, tronçons réinsertion, 2-Opt, 2-Opt sur plusieurs routes, etc) appliquée sur l'ensemble des tournées.
6. L'algorithme "Bone Route" proposé par [Tarantilis et al. 2002] est basé sur le couplage d'une procédure constructive (la méthode des gains modifiée) pour obtenir une solution initiale, suivie d'une recherche tabou qui utilise l'opérateur 2-Opt pour obtenir les solutions voisines.
7. AGES (Active Guided Evolution Strategies) [Mester et al. 2005] combine des stratégies d'évolution avec l'opérateur de recherche locale 2-Opt.
8. L'algorithme génétique hybride [Berger et al. 2003] qui est basé sur l'utilisation d'un

algorithme génétique couplé avec une recherche locale (2-Opt, etc).

9. L'heuristique D-Ants [Reimann et al. 2004], cette méthode utilise l'heuristique des gains modifiée avec la méthode des colonies de fourmis.

La comparaison entre ces méthodes montre que l'algorithme génétique de [Prins 2004], AGES [Mester et al. 2005] et Bone Route [Tarantilis et al. 2002] sont les meilleurs pour toutes les instances des benchmarks traités (les 14 instances de Christofides et les 20 instances à grande taille de Golden).

1.6 Cadre de travail

Nous avons listé les différentes caractéristiques liés au VRP. La combinaison de ces caractéristiques a donné naissance à un grand nombre de variantes. Nous avons défini les problèmes de tournées de véhicules avec attention plus particulière portée sur les extensions CVRP, VRPTW. Les méthodes de résolution du VRP peuvent être classées dans deux grandes classes : les méthodes exactes et les méthodes approchées. Nous avons détaillé ces méthodes et plus particulièrement les métaheuristiques.

Aujourd'hui, le traitement des données en temps réel est possible grâce aux avancées des technologies de l'information et de la communication : en particulier, les systèmes de géolocalisation par satellite (GPS), les systèmes d'information géographique (GIS), les systèmes autoroutiers intelligents pour véhicules (IVHS) et autres systèmes embarqués, ont augmenté l'intérêt des méthodes et/ou des stratégies pour la résolution du problème d'élaboration de tournées de véhicules dans un contexte "en ligne" où l'échange d'information peut se faire en temps réel avec un outil d'aide à la décision. Cette évolution oblige la prise en compte de l'aspect dynamique ou stochastique dans la résolution du VRP. Depuis une dizaine d'années, les problèmes de gestion dynamique de tournées et les problèmes stochastiques de tournées de véhicules sont devenus un domaine de recherche de plus en plus étudié.

Un problème est dit statique si toutes les données nécessaires à la planification ne dépendent pas du temps, sinon il est dynamique. Le caractère déterministe traduit le fait que toutes les données sont parfaitement connues (i.e. sans incertitude). Le VRP statique peut être déterministe ou stochastique. Le VRP dynamique, également connu sous le nom du VRP en temps réel, peut également être déterministe ou stochastique. Dans la littérature [Ghiani et al. 2003], le VRP a été classé en quatre types :

- Dans le VRP déterministe et statique dit aussi VRP classique, toutes les données sont connues à l'avance avec certitude et ne dépendent pas du temps,
- Dans les problèmes déterministes et dynamiques, les données sont connues avec certitude et seuls quelques éléments dépendent du temps.
- Dans les problèmes stochastiques et statiques, l'incertitude peut porter sur la quantité de la demande, sur le temps de service chez le client, ou sur le temps de trajet. Et le temps n'est pas pris en compte explicitement.
- Dans les problèmes stochastiques et dynamiques, les données incertaines sont représentées par des variables stochastiques, de plus, des éléments dépendent du temps. Par exemple, dans [Bent et al. 2003], une partie des clients arrivent dans la journée, et le

temps d'arrivée des clients suit une certaine distribution de probabilité.

Dans le cadre de notre travail, nous avons proposé des approches à base de métaheuristiques pour résoudre le VRPTW stochastique en mode statique et en mode dynamique. Ceci fait l'objet des deux chapitres suivants.

Chapitre 2

Résolution du problème de la gestion de tournées de véhicules avec fenêtres de temps et demandes floues

Sommaire

2.1 Le VRP avec demande stochastique.....	37
2.1.1 Cadre de modélisation.....	38
2.1.2 Politiques de recours	38
2.1.3 Travaux existants.....	39
2.2 Le VRP avec fenêtres de temps et demandes floues	40
2.3 Modélisation du problème.....	42
2.3.1 Notions de logique floue	43
2.3.2 Contrainte de capacité floue.....	46
2.3.3 Modèle CCP	50
2.3.4 Modèle SPR.....	51
2.3.5 Politique de recours utilisée	52
2.3.6 Méthode d'estimation des défauts de tournées.....	53
2.4 Algorithme génétique pour le VRPTWFD	56
2.4.1 Introduction	56
2.4.2 Codage de la solution	58
2.4.3 Evaluation d'un chromosome.....	60
2.4.4 Population initiale	66
2.4.5 Sélection.....	66
2.4.6 Opérateurs de croisement (Crossover)	67
2.4.7 Opérateurs de mutation	70
2.4.8 Méthode de remplacement par l'élitisme	72
2.5 Etudes expérimentales.....	73
2.5.1 Le benchmark utilisé	73
2.5.2 Simulation	77
2.5.3 Evaluation des solutions dans un environnement réel.....	83
2.5.4 Comparaison avec la version déterministe	83
2.5.5 L'influence du taux d'incertitude	86
2.5.6 Zoom sur le modèle SPR.....	88
2.6 Conclusion.....	92

Dans ce chapitre, nous allons d'abord présenter le VRP stochastique et plus particulièrement celui avec demandes stochastiques dans la section 2.1. Le problème de tournées de véhicules avec fenêtres de temps et demandes floues (VRPTWFD) à traiter dans le cadre de notre travail est présenté dans la section 2.2. Dans la section 2.3, nous proposons deux modèles pour traiter ce type de problème. Un algorithme génétique pour la méthode de résolution en intégrant ces modèles est présenté dans la section 2.4. Les solutions respectives obtenues avec différents modèles pour le problème à traiter sont évaluées et comparées d'un point vu qualitatif dans la section 2.5. Nous terminons ce chapitre par une conclusion dans la section 2.6.

2.1 Le VRP avec demande stochastique

Dans certaines applications pratiques, on ne connaît pas avec certitude les données. Ceci a suscité un grand intérêt de la communauté scientifique qui travaille sur le VRP stochastique (SVRP). Le SVRP est très utile dans le domaine des systèmes de transport et logistique. Le problème VRP est dit stochastique lorsque certains paramètres du problème ne peuvent pas être connus avec certitude. Ce pourrait être les demandes (quantité demandée) des clients, le temps ou le coût de transport, ou bien l'ensemble des clients à visiter.

Selon le paramètre sur lequel porte l'incertitude, le SVRP se classe en trois catégories [Laporte et al.1998] [Gendreau et al. 1996a]:

- Le VRP avec clients stochastiques (VRPSC : Vehicle Routing Problem with Stochastic Customers). Un client se présente avec une certaine probabilité p (absent avec une probabilité de $1-p$).
- Le VRP avec demandes stochastiques (VRPSD : Vehicle Routing Problem with Stochastic Demand). Les demandes (quantités) des clients sont des variables aléatoires.
- Le VRP avec temps stochastiques (VRPST : Vehicle Routing Problem with Stochastic Travel Time). Les temps de déplacement et/ou les temps de service sont des variables aléatoires.

Le VRP avec demandes stochastiques est le cas le plus étudié. Nous nous intéressons à cette catégorie dans nos travaux.

Dans le VRPSD, les demandes étant aléatoires, les quantités réelles des demandes ne peuvent être connues qu'au moment où les véhicules arrivent chez les clients. Malgré ceci, les tournées des véhicules doivent être planifiées à l'avance par un algorithme qui doit prendre en compte cette incertitude. A cause de cette incertitude, il arrive quelquefois que le véhicule ne puisse servir un client de la tournée planifiée quand il se présente chez celui-ci : par exemple lorsque la capacité résiduelle du véhicule est inférieure à la demande réelle de ce client. On appelle cette situation un « défaut de tournées » [Dror et al.1986]. Cette situation se produit surtout pour les clients situés en fin de tournée. C'est une caractéristique importante du VRPSD.

2.1.1 Cadre de modélisation

Le VRPSD peut être résolu dans le cadre de programmation stochastique de deux façons différentes. Puisque les données sont incertaines, il devient impossible de satisfaire toutes les contraintes avec certitude. Les contraintes sont satisfaites avec une probabilité minimale donnée ou taux de réussite minimal donné. L'espérance mathématique est utilisée dans la fonction objectif à optimiser. C'est le modèle de programmation sous contraintes probabilistes, ou *Chance Constrained Programming (CCP)* en anglais.

Le VRPSD est résolu par un modèle CCP dans les études de [Golden et al.1979] [Stewart et al. 1983] et [Werners et al. 2003]. On élabore un ensemble de tournées dans lesquelles les « défauts de tournées » seront inférieurs à un certain seuil, tout en minimisant les coûts. Dans le modèle CCP, le coût lié aux « défauts de tournées » est ignoré et donc non évalué. Pour le VRPSD ou le VRPST, le modèle CCP permet de transformer le problème stochastique en son duplicata déterministe.

Le VRPSD peut être traité également par le modèle de programmation stochastique avec recours, ou *Stochastic Programming with Recourse (SPR)* en anglais [Gendreau et al.1995] [Tan et al. 2007]. Avec le modèle SPR, le problème est résolu en deux étapes : dans la première étape une solution a priori est planifiée sur la base des données incertaines, puis un recours ou une action corrective est appliqué dans la deuxième étape pour traiter les situations de « défauts de tournées ». Dans le modèle SPR, l'objectif est de chercher à minimiser le coût prévisionnel en tenant compte du coût supplémentaire généré par les recours de la deuxième étape. Une estimation des coûts supplémentaires peut être obtenue à partir de scénarios de simulation.

Bien que le modèle CCP est plus simple par rapport au modèle SPR pour résoudre le SVRP, la plupart des travaux sur le SVRP utilisent le modèle SPR car la fonction objectif est plus significative puisqu'elle prend en compte le coût induit par les « défauts de tournée » [Gendreau et al. 1996a].

2.1.2 Politiques de recours

La fonction objectif du modèle SPR est plus significative car elle intègre le coût lié aux recours dus aux « défauts de tournées ». Dans ce modèle, un recours (i.e. action corrective) doit être utilisé pour servir les clients mis en défaut. Il existe trois politiques communes de recours :

- Dans la plus simple ([Teodorovic et al. 1992] [Teodorovic et al. 1996] [Teodorovic et al. 2003] [Laporte et al.1998] [Laporte et al.1993] [Gendreau et al.1995] [Gendreau et al.1996b]), le véhicule rentre au dépôt pour décharger ou se réapprovisionner lorsqu'un défaut de capacité se produit chez le client. Puis il retourne chez le client mis en défaut pour le servir et il poursuit sa tournée initialement planifiée. Par exemple, dans la figure 2.1, l'exécution de la solution a priori, peut entraîner un recours provoqué par un défaut de capacité du véhicule chez le client 6 et conduit à un retour prématuré au dépôt pour traiter celui-ci.

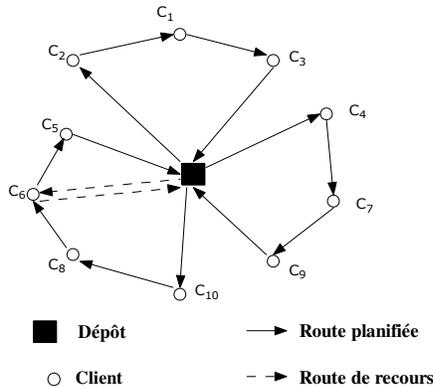


Figure 2. 1 Le problème VRPSD

- Avec la politique de réapprovisionnement préventive ([Bianchi et al. 2004], [Yang et al.2000], [Bertsimas et al.1995]), le véhicule anticipe le retour au dépôt quand il arrive chez un client proche du dépôt et que sa capacité résiduelle est bientôt épuisée. On planifie de cette façon un réapprovisionnement préventif aux points stratégiques, au lieu d'attendre l'occurrence « défaut de tournée ».
- Avec la politique de re-optimisation [Secomandi 2001] et [Secomandi 2000] améliorent la portion restante des tournées initialement planifiées dès que la demande réelle du client est connue ou dès que le « défaut de tournée » se produit.

2.1.3 Travaux existants

Le premier algorithme pour SVRP proposé par [Tillman 1969] est basé sur la méthode des économies de Clark et Wright [Clark et al. 1964].

[Bertsimas et al. 1992] propose une méthode heuristique cyclique pour le VRPSD. Puis [Bertsimas et al. 1995] améliorent cette heuristique en appliquant la programmation dynamique et en y intégrant des retours au dépôt dans les tournées a priori. C'est une approche similaire à la stratégie de réapprovisionnement préventive.

[Gendreau et al.1995] présente une méthode exacte basée sur la programmation stochastique en nombres entiers pour traiter le VRPSCD (clients stochastiques et demandes stochastiques). Avec la méthode de L-shaped en nombres entiers, [Laporte et al.1993] résolvent des instances avec 2 véhicules et 46 et 70 clients, pour respectivement le VRPSCD et le VRPSD. [Gendreau et al.1996b] ont développé une méthode de recherche tabou appelée TABUSTOCH pour résoudre le même problème mais sur des instances de grande taille.

Récemment, [Christiansen et al 2007] propose un algorithme exact avec application de la technique de génération de colonnes sur la relaxation du VRPSD exprimé comme un problème de partition. Cette approche permet la résolution d'instances comprenant jusqu'à 60 clients.

[Teodorovic et al. 1992] proposent une méthode de recuit simulé pour le VRPSD avec

plusieurs véhicules. [Teodorovic et al. 1996] et [Teodorovic et al. 2000] utilisent un système à inférences floues pour déterminer une préférence qui indique la satisfaction de la contrainte de capacité dans une méthode heuristique de type balayage.

[Secomandi 2000] et [Secomandi 2000] appliquent la programmation dynamique neuronale avec une politique de re-optimisation pour le VRPSD.

[Yang et al.2000] étudient le VRPSD avec un seul véhicule et plusieurs véhicules par la programmation dynamique. Ils ont adapté l'opérateur de recherche locale Or-Opt pour le cas stochastique.

[Bianchi et al. 2004] [Bianchi et al. 2006] ont comparé 5 métaheuristiques : le recuit simulé, la recherche tabou, la recherche locale réitérée, l'optimisation par colonies de fourmis et l'algorithme évolutionnaire pour résoudre le VRPSD avec un seul véhicule.

[Tan et al. 2007] introduisent un algorithme évolutionnaire pour le VRPSD avec fenêtres de temps. [Guo et al.2004] ont proposé un algorithme génétique pour le VRPSCD avec fenêtres de temps également. Dans leur approche, la contrainte temporelle peut être violée avec une certaine pénalité.

2.2 Le VRP avec fenêtres de temps et demandes floues

Dans cette thèse, nous considérons un problème VRPTW où les demandes des clients ne sont pas connues avec certitude.

Nous rappelons ici dans les grandes lignes du VRPTW : un nombre limité de véhicules disponibles de capacité identique partent d'un unique dépôt, desservent un ensemble de clients et retournent au dépôt. Chaque client possède une quantité de demande à satisfaire et une fenêtre de temps dans laquelle il préfère être servi. Un client ne peut être servi que par un et un seul véhicule. L'objectif est de minimiser la distance totale parcourue en respectant la contrainte de capacité des véhicules et les contraintes temporelles. Dans cette thèse, nous nous intéressons au problème VRPTW où les demandes des clients sont incertaines.

Dans le VRPSD, les demandes (i.e. quantités à livrer ou à ramasser) des clients ne sont pas connues avec certitude, à l'avance c'est-à-dire au moment de la planification. Les demandes suivent une loi de distribution connue telle que la distribution normale dans la plupart des études [Teodorovic et al. 2000] [Tan et al. 2007]. Dans certaines études [Teodorovic et al. 1996] et [Werners et al. 2003], les demandes sont représentées à l'aide de nombres flous (VRPFD : Vehicle Routing Problem with Fuzzy Demand).

Dans le problème VRPTWFD que nous traitons, l'incertitude de demandes des clients sera également représentée par des nombres flous. Nous pensons que ce choix facilite l'expression de la demande par le client et permet ainsi une plus grande souplesse.

1. Demandes floues

La demande de chaque client $i \in C$ est modélisée par un nombre flou noté \tilde{d}_i . Ce nombre est composé d'un triplet de nombres réels $(\underline{d}_i, \hat{d}_i, \bar{d}_i)$ avec $\underline{d}_i < \hat{d}_i < \bar{d}_i$. Une

fonction d'appartenance μ est associée à chaque nombre flou, elle a une représentation triangulaire (voir la figure 2.2). Les termes \bar{d}_i et \underline{d}_i sont respectivement la borne inférieure et supérieure, leurs degrés d'appartenance correspondants sont égal à 0. Le terme \hat{d}_i est la normalisation qui correspond au degré d'appartenance maximal égal à 1.

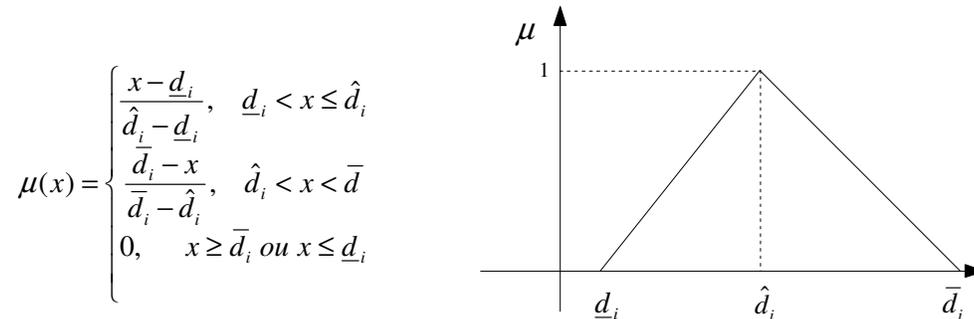


Figure 2. 2 La modélisation d'une demande floue

2. Objectifs du problème

Dans notre problème, la contrainte de fenêtres de temps est supposée « souple » (soft time windows). Autrement dit, lorsqu'un véhicule arrive en dehors de la fenêtre de temps d'un client, c'est-à-dire au-delà de la borne supérieure de la fenêtre de temps, il peut le servir dans certains cas, mais avec une pénalité proportionnelle au retard induit chez le client. Le retard chez le client est défini par la différence entre l'instant de début de service et la borne supérieure de la fenêtre de temps du client considéré.

Notre problème est de nature multi-objectifs, puisque les objectifs à minimiser sont :

- La distance totale parcourue (f_D) ;
- La somme des retards chez les clients qui représente la pénalité sur la violation de la contrainte de fenêtres de temps (f_R).

Les approches utilisées pour la résolution du problème multi-objectifs (PMO) peuvent être classées en trois catégories [Talbi 2001] :

- Approches basées sur la transformation du problème en un problème uni-objectif : Parmi ces méthodes, nous avons les méthodes d'agrégation [Ishibuchi et al. 1998], ϵ -contrainte [Hertz et al. 1994] et les méthodes de programmation par but [Sandgren 1994].
- Approches non-Pareto : Ces approches ne transforment pas le PMO en un problème uni-objectif. Elles utilisent des opérateurs de recherche qui traitent séparément les différents objectifs. Il existe plusieurs approches non-Pareto qui sont : la sélection parallèle dans les algorithmes évolutionnistes [Schaffer 1985], et la sélection lexicographique [Coello 1998].
- Approches Pareto : Ces approches utilisent directement la notion de dominance dans la

sélection des solutions générées. Contrairement aux autres approches déjà vues qui traitent séparément les différents objectifs, celles-ci doivent tenir compte de deux contraintes [Talbi 2001] :

Converger vers la frontière Pareto optimale.

Trouver des solutions diversifiées dans cette frontière Pareto.

Il y a plusieurs méthodes qui visent à résoudre le VRPTW multi-objectifs. La méthode d'agrégation consiste à affecter des poids (représentant des préférences) aux différents objectifs du problème. Dans [Sushil et al. 1999] les auteurs ont défini une fonction d'objectif comme étant la somme pondérée de tous les objectifs de ce problème :

$$F = w_1 \cdot D + w_2 \cdot T + w_3 \cdot R1 + w_4 \cdot R2 + w_5 \cdot O$$

Où :

- D : est la distance totale parcourue par les véhicules.
- T : est le temps total de trajet de l'ensemble des véhicules.
- R1 : est le retard total chez les clients.
- R2 : est le retard total des véhicules au niveau du dépôt.
- O : est la surcharge totale de capacité des véhicules.
- w_i : sont des poids (préférences) associés à chaque critère.

Ils sont déterminés en général par un expert du domaine.

Cette fonction a été réduite par [Gendreau et al. 1999] de la manière suivante pour la résolution du problème dynamique (DVRPTW) :

$$F = w_1 \cdot D + w_2 \cdot R1$$

H. Housroum a utilisé cette fonction simplifiée avec $w_1=1$ et $w_2=1$ dans sa thèse [Housroum 2005] pour pouvoir comparer son approche avec celle de Gendreau [Gendreau et al. 1999]. Notre travail étant une extension de ces travaux, nous avons utilisé la même fonction d'agrégation. La fonction objectif à minimiser de notre problème est donc une agrégation de la distance totale parcourue (f_D) et de la somme des retards chez les clients

(f_R) :

$$f_{DR} = f_D + f_R$$

Nous appellerons dans ce qui suit l'objectif f_{DR} « coût de transport » pour simplifier.

2.3 Modélisation du problème

Nous présentons dans ce qui suit la modélisation du problème lorsque l'imprécision des demandes des clients est exprimée à l'aide de nombres flous.

2.3.1 Notions de logique floue

Nous avons choisi dans notre approche de modéliser l'incertitude des quantités demandées des clients par des nombres flous. Au moment de la planification des routes, la contrainte de capacité des véhicules doit donc prendre en compte cette formulation d'incertitude. Nous rappelons dans cette section quelques notions sur logique floue qui nous ont permis de construire le modèle flou de notre approche.

2.3.1.1 Système à inférences floues

Comme nous pouvons le voir sur la figure 2.3, un système à inférences floues [Mamdani et al.1975] est composé de quatre parties :

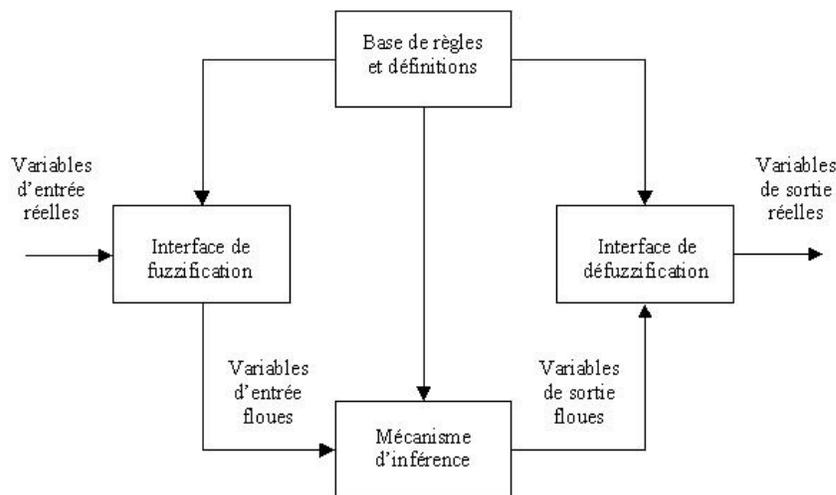


Figure 2.3 Le système à inférences floues

1. Fuzzification

Le composant de la fuzzification lit les variables d'entrées réelles, les traduit en labels de sous-ensembles flous ou en variables linguistiques associées tel que « grand », « petit ».

2. Base de règles et définitions

Une des premières tâches qui incombe aux concepteurs de systèmes flous est de déterminer les variables liées au système à contrôler, puis ensuite de définir les variables linguistiques associées. Ainsi, pour chaque variable, il s'agit de discrétiser le domaine de variation en sous-ensembles flous.

Un sous-ensemble flou F est défini sur un ensemble de valeurs, le référentiel U . Il est caractérisé par une fonction d'appartenance : $\mu : x \in U \rightarrow \mu(x) \in [0,1]$, qui quantifie le degré d'appartenance de chaque élément de U à l'ensemble flou F .

Règles floues :

La forme générale des règles floues est « SI prémisse ALORS conclusion », où les

prémisses portent sur les entrées du contrôleur et les conclusions sur les sorties. Ces règles sont en fait des propositions floues incluant un opérateur d'implication.

3. Inférence

Le mécanisme d'inférence des règles consiste à déterminer un ensemble de règles floues à activer. Au niveau de chacune de ces règles, un degré d'activation est calculé en utilisant les fonctions d'appartenance précédentes. Ce degré va permettre de déterminer une valeur floue pour la (les) variable(s) de sortie apparaissant dans la partie « conclusion ».

4. Défuzzification

La défuzzification est la partie duale de la fuzzification, elle consiste à combiner les sorties des règles calculées à l'étape précédente et à traduire le résultat intermédiaire flou en un résultat final réel dans le but de déterminer la commande effective à appliquer sur le système.

2.3.1.2 Théorie des possibilités

En liaison avec la théorie des sous-ensembles flous, la théorie des possibilités a été introduite par Zadeh [Zadeh 1978] et développée par Dubois et Prade en [Dubois et al. 1988a] et [Dubois et al. 1988b]. La première permet de traiter des données imprécises alors que la seconde offre un moyen de gérer des connaissances entachées d'incertitudes.

Considérons un ensemble de références X. La croyance en un évènement défini sur X est parfaitement déterminée par la connaissance des degrés de nécessité et de possibilité.

1. Mesure de Possibilité

Une mesure de possibilité Pos est une fonction définie sur l'ensemble des parties de X, $P(X)$, à valeurs dans $[0, 1]$ telle que :

- $Pos(\emptyset) = 0$
- $Pos(X) = 1$
- $\forall A_i \in P(X), i = 1, 2, \dots, Pos(\cup_{i=1, 2, \dots} A_i) = \sup_{i=1, 2, \dots} Pos(A_i)$

Plus la possibilité d'un évènement est proche de 1, plus la réalisation de cet évènement est possible. Une possibilité nulle indique qu'un évènement ne se produira jamais. La deuxième propriété est la particularité fondamentale de la mesure de possibilité. Elle implique notamment que si l'on étudie un évènement A de X, et son contraire \bar{A} , on sait qu'au moins un des deux est tout à fait possible, autrement dit : $Pos(A) = 1$ ou $Pos(\bar{A}) = 1$.

La mesure de possibilité n'est pas suffisante pour savoir si un évènement A va être réalisé puisque l'on peut avoir à la fois $Pos(A) = 1$ et $Pos(\bar{A}) = 1$, caractérisant une situation de totale indétermination quant à la réalisation ou non de A. Lever cette indétermination

nécessite une information supplémentaire sur la possibilité de l'événement contraire. Les mesures de nécessité, mesures duales des mesures de possibilité, ont été définies dans ce but.

2. Mesure de Nécessité

La nécessité d'un événement est définie comme la mesure de l'impossibilité de l'évènement contraire. Ainsi :

- $Nec(\emptyset) = 0$
- $Nec(X) = 1$
- $\forall A_i \in P(X), i = 1, 2, \dots, Pos(\bigcap_{i=1, 2, \dots} A_i) = \inf_{i=1, 2, \dots} Nec(A_i)$

La valeur de nécessité indique le degré de certitude avec lequel un événement sera réalisé. Plus ce degré est proche de 1, plus il est certain que cet événement sera réalisé.

On peut identifier également la correspondance entre la dualité des mesures de possibilité et de nécessité : $\forall A \in P(X), Nec(A) = 1 - Pos(\bar{A})$.

Un événement A est certain si l'événement contraire est impossible. Plus la nécessité de A est grande, moins \bar{A} est possible et donc plus on est certain que A sera réalisé.

3. Propriétés des mesures de nécessité et de possibilité

Propriété 1 : $\forall A \in P(X), \max(Pos(A), Pos(\bar{A})) = 1$

Si un événement n'est pas tout à fait possible, alors son événement contraire l'est.

Propriété 2 : $\forall A \in P(X), \min(Nec(A), Nec(\bar{A})) = 0$

Si un événement est nécessaire, alors son événement contraire ne l'est pas.

Propriété 3 : $\forall A \in P(X), Pos(A) < 1 \Rightarrow Nec(A) = 0$

Propriété 4 : $\forall A \in P(X), Nec(A) > 0 \Rightarrow Pos(A) = 1$

Propriétés 3 et 4 représentent la cohérence sémantique des degrés de nécessité et possibilité. Ceci nous amène à remarquer que

Propriété 5 : $\forall A \in P(X), Nec(A) \leq Pos(A)$

Pour chaque événement A, les degrés de nécessité et possibilité ont toujours l'une des formes suivantes :

- $Nec(A) = 1$ et $Pos(A) = 1$: Il est certain que l'événement A soit vérifié ;

- $Nec(A) > 0$ et $Pos(A) = 1$: Il est possible que l'événement A soit vérifié, et l'événement A est plus certainement vérifié que son événement contraire;
- $Nec(A) = 0$ et $Pos(A) = 1$: Il est possible que l'événement A soit vérifié, mais il est également possible que son événement contraire soit vérifié.
- $Nec(A) = 0$ et $Pos(A) < 1$: Il est possible que l'événement A soit vérifié, mais son événement contraire est plus certainement vérifié;
- $Nec(A) = 0$ et $Pos(A) = 0$: Il est certain que l'événement contraire de A soit vérifié .

4. Distribution de Possibilité

On peut utiliser une fonction qui attribue un degré de possibilité entre 0 et 1 à tout élément de X. On peut définir ainsi une distribution de possibilité π sur X à valeur dans $[0,1]$ vérifiant la propriété de normalisation, c'est-à-dire $\sup_{x \in X} \pi(x) = 1$. La fonction Pos définie pour toute partie A de X par $Pos(A) = \sup_{x \in A} \pi(x)$ est une mesure de possibilité.

En liaison avec la théorie des sous-ensembles flous, La fonction Pos est définie par la fonction d'appartenance : $Pos(A) = \sup_{x \in A} \mu_E(x)$.

2.3.2 Contrainte de capacité floue

Dans notre problème, la contrainte de capacité du véhicule a besoin d'être reformulée différemment pour traiter des demandes floues. La contrainte de capacité indique que la demande totale des clients de chaque tournée ne doit pas dépasser la capacité Q du véhicule réalisant la tournée.

Pour le VRP avec demande déterministe, la contrainte de capacité est formulée par l'inéquation 1.6 $\sum_{i \in C} \sum_{j \in S} d_i x_{ij}^v \leq Q, \forall v \in V$ (voir section 1.2.2 du chapitre 1).

Nous adaptons cette inéquation puisque dans notre cas, les demandes sont représentées par des nombres flous[Xu et al. 2006]. Lors de la planification, la demande totale des clients servis par le véhicule v, noté \tilde{D}_v est modélisée par un nombre flou, qui est défini par :

$$\tilde{D}_v = \sum_{i \in C} \sum_{j \in S} \tilde{d}_i x_{ij}^v = \left(\sum_{i \in C} \sum_{j \in S} \underline{d}_i x_{ij}^v, \sum_{i \in C} \sum_{j \in S} \hat{d}_i x_{ij}^v, \sum_{i \in C} \sum_{j \in S} \bar{d}_i x_{ij}^v \right) = (\underline{D}_v, \hat{D}_v, \bar{D}_v)$$

La contrainte de capacité peut alors être exprimée par l'inéquation floue suivante:

$$\tilde{D}_v \preceq Q \tag{2.1}$$

L'inéquation 2.1 exprime une contrainte de capacité floue, dans laquelle le symbole

« $\tilde{\leq}$ » est utilisé pour désigner la comparaison entre un nombre flou \tilde{D}_v et un nombre ordinaire Q .

On ne peut vérifier directement si la contrainte de capacité floue est satisfaite ou non. Par exemple, si $\tilde{D}_v = (90, 100, 110)$ et $Q = 100$, comment vérifie-t-on cette contrainte (Eq 2.1) ?

Un système à inférences floues basé sur les règles floues est employé dans [Teodorovic 1996] pour vérifier un tel type de contrainte de capacité floue. Il construit un ensemble de règles floues par rapport à la capacité résiduelle du véhicule et la demande floue du client suivant à servir. Le système à inférences floues permet d'obtenir un indicateur p_k correspondant à un taux de préférence d'insertion du client suivant dans la tournée du véhicule k . Si l'indicateur de préférence p_k est plus grand ou égal à une référence donnée, alors le client suivant est inclus dans cette tournée, sinon il est inséré dans une nouvelle tournée.

Nous exprimons la contrainte de capacité floue à l'aide de la théorie des possibilités. La mesure de possibilité et la mesure de nécessité de l'événement, correspondant à la satisfaction de la contrainte de capacité floue, sont définies de la manière suivante :

$$Pos(\tilde{D}_v \leq Q) = \sup_{x \leq Q} \mu_{\tilde{D}_v}(x) = \begin{cases} 1, \hat{D}_v \leq Q \\ \frac{Q - \underline{D}_v}{\hat{D}_v - \underline{D}_v}, \underline{D}_v \leq Q < \hat{D}_v \\ 0, Q \leq \underline{D}_v \end{cases} \quad (2.2)$$

$$Nec(\tilde{D}_v \leq Q) = 1 - Pos(\tilde{D}_v > Q) = \begin{cases} 1, \bar{D}_v \leq Q \\ \frac{Q - \hat{D}_v}{\bar{D}_v - \hat{D}_v}, \hat{D}_v \leq Q < \bar{D}_v \\ 0, Q \leq \hat{D}_v \end{cases} \quad (2.3)$$

Un exemple de mesures de possibilité et de nécessité pour la satisfaction de la contrainte de capacité floue est représenté dans la figure 2.4 pour une capacité Q et une demande floue données.

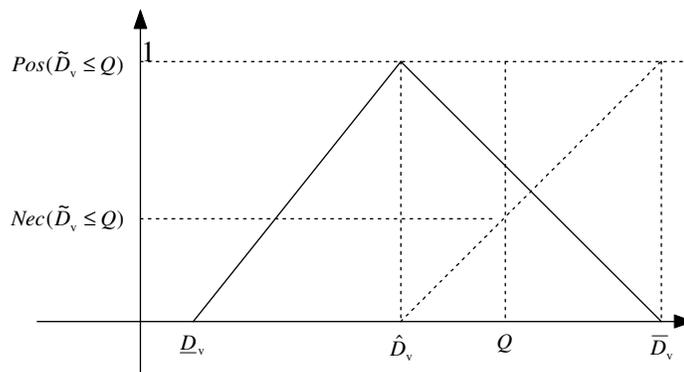


Figure 2.4 La satisfaction de la contrainte de capacité floue : possibilité et nécessité

La contrainte de capacité floue peut alors être remplacée par les deux inéquations ci-dessous :

$$Pos(\tilde{D}_v \leq Q) \geq \alpha \quad (2.4)$$

$$Nec(\tilde{D}_v \leq Q) \geq \beta \quad (2.5)$$

L'inéquation 2.4 indique que la mesure de possibilité liée à l'évènement satisfaction de la contrainte de capacité doit être supérieure à un seuil α .

L'inéquation 2.5 indique que la mesure de nécessité liée à l'évènement satisfaction de la contrainte de capacité doit être supérieure à un seuil β .

Les seuils sont choisis entre 0 et 1. Plus ils sont grands, plus stricte est la contrainte de capacité floue et moins il y aura de risque de défaut de capacité, mais plus grand sera alors le coût engendré.

L'expression de la contrainte de capacité va donc dépendre du choix des valeurs fixées pour les seuils α et β , Nous distinguons 5 cas possibles :

1. $\alpha=0$, et $\beta=0$

Dans cette configuration, les inéquations 2.4 et 2.5 deviennent :

$$Pos(\tilde{D}_v \leq Q) \geq 0$$

$$Nec(\tilde{D}_v \leq Q) \geq 0$$

Etant donnée que les mesures de possibilité et de nécessité sont comprises entre 0 et 1, ces 2 inéquations sont toujours vérifiées quelles que soient les valeurs de ces mesures. En d'autres termes la contrainte de capacité floue est toujours vérifiée quelle que soit la valeur de la demande \tilde{D}_v .

Cette combinaison de seuils représente donc la situation la moins contraignante mais la plus risquée.

2. $0 < \alpha < 1$, et $\beta = 0$

Il faut que l'évènement contrainte de capacité satisfaite soit possible et que sa mesure de possibilité soit non nulle et supérieure au seuil α . Nous déduisons l'inéquation 2.4 :

$$Pos(\tilde{D}_v \leq Q) \geq \alpha \Rightarrow \frac{Q - \underline{D}_v}{\hat{D}_v - \underline{D}_v} \geq \alpha$$

L'inéquation 2.4 peut être remplacée par

$$\alpha \hat{D}_v + (1 - \alpha) \underline{D}_v \leq Q \quad (2.6)$$

D'après la définition de la nécessité, l'inéquation 2.5 est toujours vérifiée avec $\beta=0$

($Nec(\tilde{D}_v \leq Q) \geq 0$). En d'autres termes, il faut que la capacité du véhicule Q soit entre la borne inférieure (\underline{D}_v) et la normalisation (\hat{D}_v) de la somme des demandes floues.

Avec cette combinaison de seuils α et β , la contrainte de capacité floue devient plus contraignante.

3. $\alpha=1$, et $\beta=0$

Avec $\alpha = 1$, l'inéquation 2.4 devient

$$Pos(\tilde{D}_v \leq Q) \geq \alpha \Rightarrow Pos(\tilde{D}_v \leq Q) = 1 \Rightarrow \hat{D}_v \leq Q$$

Cette nouvelle inégalité est un cas particulier de l'inéquation 2.6 en remplaçant α par 1.

Selon la définition de la nécessité, l'inéquation 2.5 est toujours vérifiée. La contrainte de capacité floue est vérifiée lorsque la capacité du véhicule Q est supérieure à \hat{D}_v , qui est la valeur normalisée de la somme des demandes floues.

4. $\alpha=1$, et $0 < \beta < 1$

Il faut que la nécessité de l'événement de satisfaction de la contrainte de capacité soit non nulle et supérieure au seuil β . Nous déduisons l'inéquation 2.5 :

$$Nec(\tilde{D}_v \leq Q) \geq \beta \Rightarrow \frac{Q - \hat{D}_v}{\underline{D}_v - \hat{D}_v} \geq \beta$$

Donc l'inéquation 2.5 peut être remplacée par

$$\beta \bar{D}_v + (1 - \beta) \hat{D}_v \leq Q \quad (2.7)$$

Du fait de propriété 4 ($\forall A \in P(X), Nec(A) > 0 \Rightarrow Pos(A) = 1$), l'inéquation 2.7 implique que l'inéquation 2.4 est toujours vérifiée.

En d'autres termes, il faut que la capacité du véhicule Q soit entre la normalisation (\hat{D}_v) et la borne supérieure (\bar{D}_v) de la somme des demandes floues.

Avec cette combinaison de seuils, la contrainte de capacité floue est plus difficile à satisfaire que celles correspondantes aux combinaisons précédentes.

5. $\alpha=1$, et $\beta=1$

L'expression de la nécessité peut être simplifiée en posant $\beta = 1$ dans l'inéquation 2.7:

$$\bar{D}_v \leq Q$$

Si cette inéquation est satisfaite, alors d'après la propriété 4, elle implique que

l'inéquation 2.4 est également satisfaite.

Avec cette combinaison de seuils, la contrainte de capacité floue est satisfaite lorsque la capacité du véhicule Q est supérieure ou égale à la borne supérieure (\overline{D}_v) de la somme des demandes floues. C'est le cas le plus contraignant.

En concluant la discussion ci-dessus, la contrainte de capacité floue peut être défuzzifiée selon la combinaison des seuils choisis :

- Si $0 < \alpha \leq 1$, et $\beta = 0$, la contrainte de capacité floue est défuzzifiée par l'inéquation

$$2.6 (\alpha \hat{D}_v + (1 - \alpha) \underline{D}_v \leq Q);$$

- Si $\alpha = 1$, et $0 < \beta \leq 1$, la contrainte de capacité floue est défuzzifiée par l'inéquation

$$2.7 (\beta \overline{D}_v + (1 - \beta) \hat{D}_v \leq Q);$$

Basé sur la contrainte de capacité floue et en utilisant la théorie de possibilité, nous proposons dans ce qui suit deux modèles issus du cadre de la programmation stochastique pour traiter le VRPTWFD.

2.3.3 Modèle CCP

En raison de l'incertitude sur les demandes des clients, certaines tournées planifiées peuvent au moment de leur réalisation provoquer des violations de contrainte de capacité ou « défauts de tournées », c'est-à-dire que la capacité résiduelle du véhicule est insuffisante pour satisfaire la demande du client courant. L'idée d'un modèle de programmation sous contraintes probabilistes (CCP) consiste à imposer que la violation des contraintes ne se produise qu'avec une probabilité fixée.

[Kaufmann et Gupta 1985] indiquent que la théorie des possibilités joue dans la théorie des sous-ensembles flous un rôle similaire à celui joué par la mesure de la probabilité dans la théorie des probabilités. Nous avons donc analysé la contrainte de capacité floue en fonction des seuils des mesures de possibilité et de nécessité qu'elle doit satisfaire. Plus les seuils sont grands, plus la contrainte de capacité floue sera contraignante et moins il y aura de risque de défauts de capacité. Autrement dit, en choisissant à propos les seuils α et β , nous pouvons planifier un ensemble de tournées a priori dans lesquelles on pourra limiter le nombre de « défauts de tournées ».

En utilisant la théorie des possibilités, nous proposons un modèle CCP pour traiter le VRPTWFD. Basée sur la formulation du VRPTW (voir les sections 1.2 et 1.3 du chapitre 1), ce modèle est exprimé comme suite :

$$\text{Minimiser } f_{DR}^{Plan} = \sum_{v \in V} \sum_{i \in S} \sum_{j \in S} c_{ij} x_{ij}^v + \sum_{i \in C} p_i \quad (2.8)$$

Sujette aux contraintes 1.2-1.5, 1.7-1.12, et 2.4-2.5

L'équation 2.8 représente l'objectif du VRPTWFD à minimiser. Le terme f_{DR}^{Plan} représente le coût planifié de transport. C'est une agrégation de la distance totale parcourue

$(\sum_{v \in V} \sum_{i \in S} \sum_{j \in S} c_{ij} x_{ij}^v)$ et de la somme des retards chez les clients $(\sum_{\forall i \in C} p_i)$. Le terme c_{ij} est la distance du sommet i au sommet j , avec i et $j \in S$. Le terme x_{ij}^v est à 1 si le véhicule v part du sommet i au sommet j , 0 sinon.

Nous rappelons que dans notre cas, lorsqu'un véhicule arrive chez un client au-delà de la borne supérieure de la fenêtre de temps de ce client, il peut le servir avec une pénalité proportionnelle au retard induit chez le client. Donc la contrainte 1.13 est remplacée par le terme du retard chez le client :

$$p_i = \max\{b_i - l_i, 0\}$$

Elle représente la violation de la fenêtre de temps du client i , avec b_i est l'instant où le service débute chez le client i , et l_i est la borne supérieure de la fenêtre de temps du client i .

Il faut noter que les inéquations 2.4-2.5 représentent la contrainte de capacité floue, elles peuvent être défuzzifiées selon la combinaison de valeurs choisies pour les seuils α et β par :

$$\text{l'inéquation 2.6 } (\alpha \hat{D}_v + (1-\alpha) \underline{D}_v \leq Q)$$

$$\text{ou l'inéquation 2.7 } (\beta \bar{D}_v + (1-\beta) \hat{D}_v \leq Q)$$

Ce choix peut être du ressort d'un expert du domaine ou être réglé par la simulation.

2.3.4 Modèle SPR

Néanmoins, le coût induit par le recours ou action corrective lié aux « défauts des tournées » n'est pas pris en compte dans le modèle CCP [Gendreau et al. 1996a]. Nous reformulons ci-dessous le traitement du problème VRPTWFD à l'aide cette fois-ci d'un modèle de programmation stochastique à deux étapes avec recours ou modèle SPR pour simplifier [Xu et al. 2007c].

Nous planifions dans la première étape un ensemble de tournées a priori basé sur les demandes floues en utilisant la contrainte de capacité floue. Puis les « défauts des tournées » sont pris en compte dans la deuxième étape. Comme tous les modèles de type SPR, l'objectif de notre modèle à deux étapes est de chercher à minimiser le coût prévisionnel de transport (noté f_{DR}^{Prev}), qui est défini comme étant la somme du coût planifié de transport (f_{DR}^{Plan}) et du coût supplémentaire de transport (noté f_{DR}^{Supp}). Le terme f_{DR}^{Plan} représente le coût de transport des tournées planifiées lors de la première étape et le terme f_{DR}^{Supp} est le coût de transport des recours dus aux « défauts des tournées » lorsqu'on réalise les tournées planifiées. Ce coût est estimé lors de la deuxième étape. La nouvelle fonction objectif de

notre modèle devient donc :

$$\text{Minimiser } f_{DR}^{Prev} = f_{DR}^{Plan} + f_{DR}^{Supp} \quad (2.9)$$

Les contraintes de ce modèle restent les mêmes que le modèle CCP précédent.

Il est à noter que le coût supplémentaire de transport (f_{DR}^{Supp}) induit par ces « défauts des tournées » est lié à la politique de recours choisie. Il dépend également du moment où se produit le défaut de capacité lors de la réalisation de la tournée. En général, il ne peut pas être décrit à l'aide d'une fonction linéaire sur les termes x_{ij}^v [Laporte et al.1998]. Le calcul de ce coût supplémentaire de transport est une des difficultés principales du VRPSD [Tan et al. 2007].

Nous présentons dans ce qui suit la politique de recours utilisée dans notre approche et la méthode de simulation de tournées qui permet d'estimer le coût supplémentaire de transport.

2.3.5 Politique de recours utilisée

Nous utilisons pour simplifier notre approche, la politique dans laquelle le véhicule rentre au dépôt pour décharger ou se réapprovisionner lorsqu'un « défaut de tournées » se produit. Le coût supplémentaire de transport induit par les actions correctives se compose de la distance supplémentaire à parcourir et du retard engendré. Nous supposons également pour simplifier que le temps de service au dépôt pour décharger ou se réapprovisionner est négligeable (=0). Nous distinguons 2 cas possibles (voir la figure 2.5):

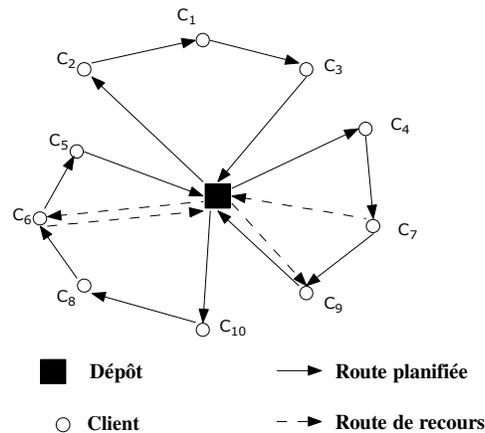


Figure 2. 5 Le recours du défaut de tournées

- La capacité résiduelle du véhicule est inférieure à la demande du client courant (i.e. le client 6). En supposant que la demande du client ne peut pas se diviser, le véhicule doit retourner au dépôt pour décharger ou se réapprovisionner, puis revenir chez ce client pour assurer le service et poursuivre sa tournée initiale. La distance supplémentaire est deux fois la distance entre ce client et le dépôt (i.e. $2 \cdot c_{60}$). L'instant où le service débutera chez

ce client est donc reporté. Il y aura retard si cet instant est au-delà de la fenêtre de temps de ce client, sinon, le retard chez ce client est nul. De plus, ce détour peut engendrer d'autres retards chez les clients dans la suite de la tournée.

- La capacité résiduelle du véhicule est exactement égale à la demande du client courant (i.e. le client 7). Le véhicule rentre au dépôt après avoir servi ce client, puis il peut repartir directement vers le client suivant (i.e. le client 9) pour poursuivre sa tournée. La distance supplémentaire est égale à $c_{70}+c_{90}-c_{79}$ pour cet exemple. Où c_{ij} est la distance du sommet i au sommet j . Ce détour peut également engendrer retards chez les clients dans la suite de la tournée.

Nous devons prendre en compte le coût supplémentaire de transport induit par cette politique de recours des « défauts des tournées » dans la deuxième étape du modèle SPR. Cet aspect est traité dans la section suivante.

2.3.6 Méthode d'estimation des défauts de tournées

Nous proposons une méthode de simulation de tournées pour estimer le coût supplémentaire de transport ainsi que le coût prévisionnel de transport [Xu et al. 2007a]. L'idée générale de notre méthode d'estimation est de « rejouer » les tournées planifiées en simulant des situations réelles à l'aide de scénarios d'apprentissage. Nous présentons ci-dessous les scénarios d'apprentissage et la procédure « rejouer » que nous avons utilisés.

2.3.6.1 Scénarios d'apprentissage

Un scénario d'apprentissage contient un ensemble de données qui a pour but de décrire l'une des nombreuses situations futures possibles. Ces scénarios pourraient être construits en analysant des bases d'historiques ou à l'aide de tirages aléatoires conjugués à des lois de distribution. Nous pouvons trouver une première utilisation de scénarios pour le problème stochastique dans [Jagannathan 1985]. [Hvattum et al. 2003] ont généré des scénarios afin de simuler les clients d'un VRP dont les clients sont stochastiques.

Dans notre cas, les demandes des clients sont représentées par des nombres flous. Chaque scénario d'apprentissage contient les demandes réelles (exactes) des clients, ces valeurs sont connues au moment où les véhicules commenceront à servir les clients.

[Teodorovic et al. 1996] ont présenté une méthode de tirage des demandes réelles à partir de demandes floues. Pour une demande floue \tilde{d}_i représentée par un triplet de nombres réels $(\underline{d}_i, \hat{d}_i, \bar{d}_i)$, la demande réelle est générée de la façon suivante :

- On tire aléatoirement un nombre d suivant la loi uniforme, avec $d \in [\underline{d}_i, \bar{d}_i]$. On calcule la possibilité ($Pos(d = \tilde{d}_i)$) que la valeur d soit la demande réelle de ce client par la formule 2.10 :

$$Pos(d = \tilde{d}_i) = \mu(d) = \begin{cases} \frac{d - \underline{d}_i}{\hat{d}_i - \underline{d}_i}, & \underline{d}_i \leq d \leq \hat{d}_i \\ \frac{\hat{d}_i - d}{\hat{d}_i - \bar{d}_i}, & \hat{d}_i < d \leq \bar{d}_i \end{cases} \quad (2.10)$$

(b). On tire aléatoirement un nombre $r \in [0,1]$, suivant la loi uniforme. Si

$r \leq Pos(d = \tilde{d}_i)$, le nombre d est pris comme étant la demande réelle, sinon on recommence le tirage (retour au point (a)).

Nous générons les demandes réelles pour tous les clients en utilisant cette méthode de tirage afin de construire un scénario d'apprentissage. La troisième colonne du tableau 2.1 représente un scénario d'apprentissage de demandes réelles issues des demandes floues de la deuxième colonne.

Client	Demande floue	Demande réelle
1	(36, 40, 44)	42
2	(17, 20, 23)	22
3	(27, 30, 33)	31
4	(28, 40, 52)	51
5	(9, 10, 11)	10
6	(7, 10, 13)	12
7	(30, 40, 50)	49
8	(21, 30, 39)	38
9	(8, 10, 12)	11
10	(28, 40, 52)	51

Tableau 2. 1 Un exemple du scénario d'apprentissage de demandes réelles

Pour tenir compte de la variabilité de demandes, nous avons créé un ensemble de scénarios d'apprentissage de demandes réelles pour chacun des problèmes à traiter.

2.3.6.2. Procédure « rejouer » les tournées

Afin de simuler la réalisation des tournées pour déterminer les coûts associés, nous « rejouons » les tournées planifiées sur un scénario d'apprentissage de demandes réelles de la façon suivante :

Chaque véhicule part pour accomplir la tournée qui lui a été affectée. L'ordre de visite des clients respecte l'ordre de leur apparition dans la tournée. Chaque véhicule cumule au fur

et à mesure du parcours, les demandes réelles des clients issues de ce scénario d'apprentissage. Lorsque la somme des demandes dépasse la capacité du véhicule, c'est que le véhicule ne peut plus servir le client courant, on a alors un « défaut de tournée ». Le véhicule doit réinitialiser sa capacité en rentrant au dépôt pour décharger ou se réapprovisionner, puis revenir chez le client pour poursuivre sa tournée. Une fois que chaque véhicule a terminé sa tournée respective, nous obtenons un coût de transport simulé qui intègre le coût supplémentaire de transport dû aux « défauts de tournées » correspondant à ce scénario d'apprentissage.

Nous donnons un exemple pour illustrer la procédure « rejouer » des tournées (voir la figure 2.6). Soit trois véhicules ayant chacun une capacité de 100. Le problème ici consiste à livrer des produits à 10 clients. Les flèches en trait continu indiquent les tournées planifiées lors de la résolution. Les tournées sont « rejouées » avec un scénario d'apprentissage. Les valeurs mises entre parenthèses représentent les demandes réelles des clients issues de ce scénario.

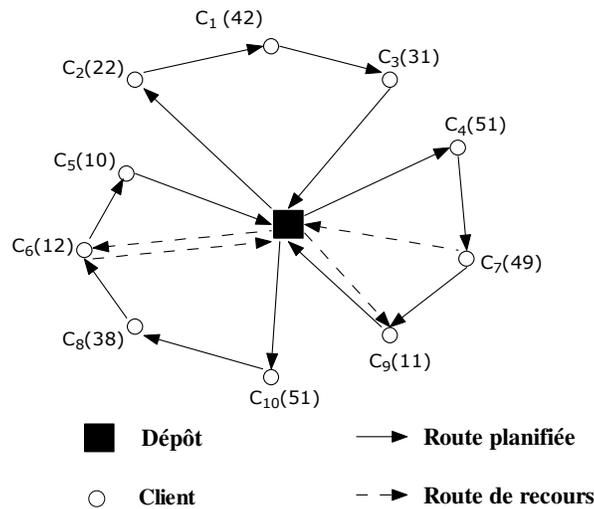


Figure 2. 6 Un exemple de la procédure « rejouer »

- Le véhicule 1 dessert les clients 2, 1 puis 3 pour une demande respective de 22, 42 et 31. Le véhicule 1 réalise sa tournée sans défaut puisque la demande totale (95) n'excède pas la capacité du véhicule (100).
- Le véhicule 2 dessert les clients 4, 7 puis 9. Après avoir servi le client 4 (quantité demandée : 51), puis le client 7 (49), le véhicule a atteint sa capacité maximale(100) : il doit donc faire un détour par le dépôt pour se re-alimenter, avant de reprendre sa tournée vers le client 9. Le bilan de la simulation est que le trajet du client 7 vers le client 9 est remplacé par le trajet du client 7 vers dépôt et du dépôt vers client 9. Ce détour engendre une distance supplémentaire à parcourir ainsi que la possibilité d'augmenter le retard chez client 9.
- Le véhicule 3 doit desservir les clients 10, 8, 6 puis 5. Après avoir servi les clients 10 (51)

et 8 (38), la capacité résiduelle du véhicule n'est plus que de 11 au moment où il arrive chez le client 6. Etant donné que la quantité nécessaire est de 12, le véhicule est obligé de faire un aller-retour au dépôt. Le bilan de la simulation est que le véhicule 3 parcourt un trajet supplémentaire (un aller-retour client 6 et dépôt) par rapport à la tournée planifiée. Ce défaut de tournée engendre donc une distance supplémentaire et peut retarder les services chez les clients restants (6 et 5).

De cette façon, la procédure « rejouer », simule le coût de transport qui comprend le coût initial planifié et le coût supplémentaire de transport (distance + retard) provoqué par les «défauts de tournées» correspondant à ce scénario de demandes réelles.

Pour estimer le coût de transport supplémentaire, nous rejouons l'ensemble de tournées planifiées a priori de la première étape sur chacun de ces scénarios d'apprentissage générés et nous calculons les coûts de transports simulés correspondants. Nous prenons alors la valeur moyenne des coûts de transports simulés correspondants à ces scénarios comme étant le coût prévisionnel de transport (f_{DR}^{Prev}). La différence entre le coût prévisionnel et le coût planifié de transport représente le coût supplémentaire de transport estimé dû aux « défauts de tournées » (f_{DR}^{Supp}).

2.4 Algorithme génétique pour le VRPTWFD

Pour résoudre les modèles précédents, nous avons utilisé une approche de type métaheuristique basée sur les algorithmes génétiques (AG). Pour ce faire nos travaux sur la prise en compte de demandes floues ont été intégrés dans la cadre d'une plateforme Java évolutionnaire développée initialement dans la thèse de Haiyan Housroum [Housroum 2005]. Nous reprenons ci-dessous les éléments qui ont été impactés par cette intégration ainsi que les éléments nécessaires à la compréhension générale du modèle. Il est à noter que nous aurions pu intégrer les modèles précédents dans toute autre approche de type métaheuristique.

2.4.1 Introduction

C'est John Holland qui a formulé le premier, le principe des Algorithmes Génétiques en 1975 [Holland 1975]. David Goldberg apporta également beaucoup à la création de ces algorithmes [Goldberg 1989]. Les algorithmes génétiques font partie des algorithmes évolutionnistes de la classe des métaheuristicques (voir section 1.5.3.4). Ces algorithmes sont des méthodes d'optimisation stochastiques qui peuvent opérer dans des espaces de recherche extrêmement vastes. Le terme évolutionniste fait référence aux techniques basées sur les mécanismes de l'évolution naturelle.

Le cycle principal d'un algorithme génétique présenté en Figure 2.7 consiste à sélectionner des individus parents à partir d'une population de départ, qui en se reproduisant par des opérations de "croisement" et "mutation" engendrent de nouveaux individus. Une nouvelle population est ainsi constituée par "remplacement" de l'ensemble ou d'une partie de la population de départ, par ces nouveaux individus. La répétition de ce cycle de base

engendre donc une succession de générations de solutions, jusqu'à satisfaction d'un critère donné de fin de cycle. La "sélection" est basée sur une évaluation de chaque individu qui détermine le degré d'adaptation de cet individu dans son environnement.

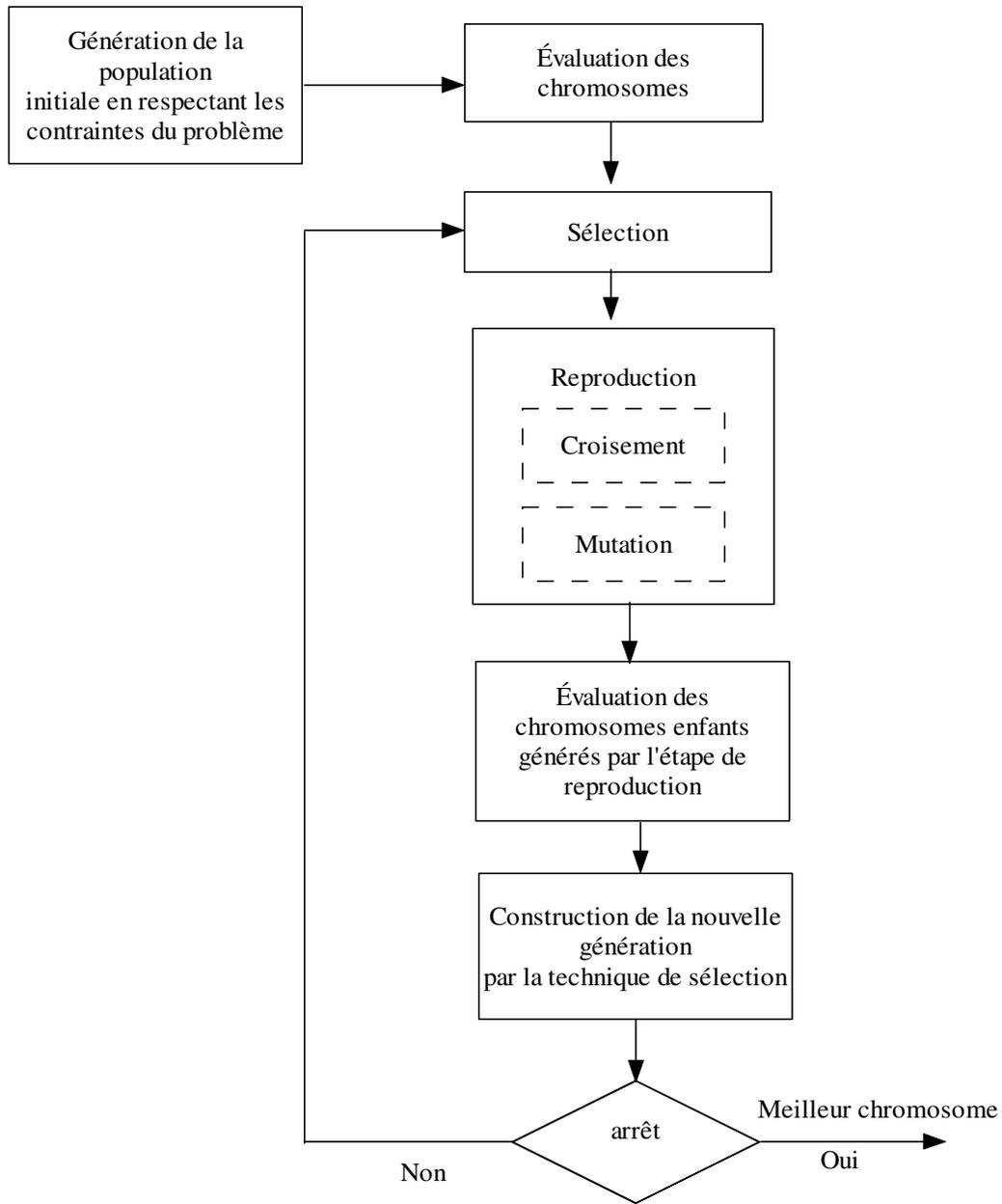


Figure 2. 7 L'algorithme génétique

Par analogie avec la biologie, la terminologie courante désigne la représentation d'un individu par le terme "génotype" et désigne le comportement de celui-ci dans un environnement donné par le terme "phénotype". Le modèle général des algorithmes génétiques doit être adapté au problème traité. Il est nécessaire de définir en particulier la représentation des solutions ou génotypes, la fonction d'évaluation d'une solution ainsi que

les opérateurs génétiques appropriés à la représentation choisie. Les opérations de sélection et de remplacement sont, quant à elles, indépendantes des choix de représentation.

2.4.2 Codage de la solution

Chaque individu de la population représente une solution au problème à optimiser. Un individu est représenté par un chromosome. Ce chromosome est constitué de gènes qui peuvent prendre des valeurs appelées “allèles”. La position du gène dans le chromosome se nomme “locus”. Il existe plusieurs manières de coder ces individus. Chacune de ces manières dépend du problème à traiter. Un allèle peut être représenté soit par un nombre binaire, soit par un nombre entier ou réel, soit par une chaîne de caractères, etc.

Chaque individu est associé à une évaluation (fitness) qui mesure la qualité de la solution. L'évaluation représente la performance de l'individu vis-à-vis du problème. Le “génotype” représente l'ensemble des allèles du chromosome alors que le “phénotype” représente la solution réelle qui en découle. De la pertinence du codage va dépendre le choix des opérateurs de reproduction et l'efficacité globale de l'algorithme. Il existe deux cas de figures pour coder un problème. Le premier utilise un codage “direct” dans lequel toute l'information de la solution est présente dans le chromosome. Le deuxième reporte la difficulté de représentation d'une solution du problème à l'extérieur du codage, en utilisant un codage “indirect”.

- Direct : Dans un codage direct, le gène doit représenter une solution complète. Ce dernier doit contenir toutes les informations utiles à la solution [Tavares et al. 2003]. La représentation chromosomique du problème VRPTW qui utilise ce type de codage peut être tout simplement un ensemble de vecteurs de tailles différentes. Chaque vecteur de cet ensemble représente une tournée d'un véhicule et contient une suite de clients à visiter par ce véhicule. Ces vecteurs ont des tailles différentes selon le nombre de clients à visiter pour chaque tournée (voir la figure 2.8).

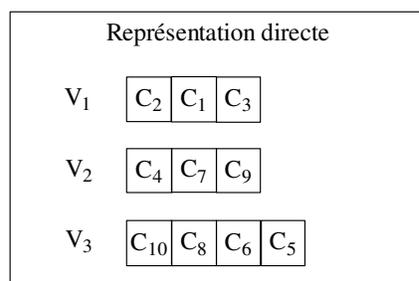


Figure 2. 8 Chromosome à codage direct

- Indirect : Dans ce type de codage, le génotype ne représente pas directement la solution. Il est nécessaire de passer par une étape complémentaire pour obtenir la solution [Sushil et al. 1999], [Prins 2004]. En utilisant ce codage pour représenter la solution du problème VRPTW, le chromosome vecteur va contenir une suite ordonnée de clients à servir. Dans cette représentation, nous ne pouvons pas distinguer les tournées de véhicules avant que le chromosome ne passe par une étape de décodage qui va construire les tournées en

appliquant par exemple une heuristique du type PFIH (Push Forward Insertion Heuristic) [Solomon 1987] (voir la figure 2.9).

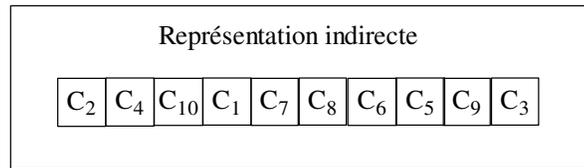


Figure 2. 9 Chromosome à codage indirect

Chaque type de codage présente des avantages et des inconvénients :

- Dans le codage direct on n’a pas besoin d’une étape supplémentaire pour construire les tournées. Cela signifie qu’avec ce codage nous gagnerons du temps de calcul.
- Il est plus simple d’appliquer les opérateurs génétiques sur un chromosome représenté par un codage indirect. Certaines contraintes du problème peuvent en effet être reportées dans la phase de décodage.

Dans notre plateforme, nous avons opté pour un codage indirect des solutions de notre problème. Ce choix initial a permis de traiter facilement et efficacement le VRPTW en mode dynamique. Il nous a contraints d’utiliser une heuristique d’insertion rapide pour réaliser la phase de décodage. C’est au sein de cette phase de décodage que nous avons dû intégrer la prise en compte des demandes floues.

Le codage indirect que nous avons choisi, repose sur l’utilisation de deux structures de données :

- Un vecteur “chromosome” qui représente une liste ordonnée de clients à servir (voir la figure 2.10). La taille du chromosome correspond au nombre de clients à servir. Dans le cas dynamique, cette taille varie en fonction du nombre de clients restants à servir. L’ordre des clients dans le chromosome a une importance car il correspond à l’ordre dans lequel la phase de décodage va insérer les clients dans les tournées en cours d’élaboration.

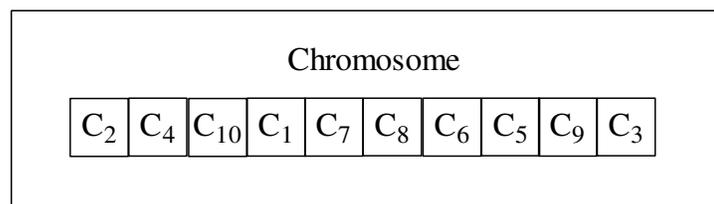


Figure 2. 10 Le codage du chromosome

- Un ensemble de vecteurs “tournées planifiées”, qui représente les tournées obtenues après la phase de décodage du chromosome (voir la figure 2.11). Chaque vecteur de cet ensemble représente la tournée d’un véhicule, et contient l’ensemble clients rangés dans l’ordre où ils seront servis par ce véhicule.

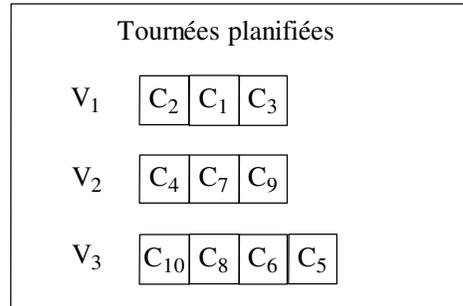


Figure 2. 11 Tournées planifiées obtenues après la phase de décodage

Les tournées ainsi obtenues après la phase de décodage peuvent être évaluées pour définir ainsi le fitness du chromosome ou la qualité de la solution correspondante.

2.4.3 Evaluation d'un chromosome

Chaque chromosome représente une solution au problème. La fonction d'évaluation permet de quantifier l'aptitude d'un chromosome à survivre. Cette quantité est appelée aussi fitness. Cette fonction dépend directement du type de problème à résoudre. Dans le cas d'un problème d'optimisation, celle-ci est fortement liée à la fonction d'objectif. C'est au sein de cette partie que nous avons dû intégrer la prise en compte des demandes floues. Comme nous avons utilisé un codage indirect, l'évaluation d'un chromosome doit passer une phase de décodage pour obtenir l'ensemble de tournées correspondant au codage.

2.4.3.1 Phase de décodage d'un chromosome

L'algorithme de décodage utilisé est basé sur une heuristique d'insertion simple. Le principe de ce décodage est le suivant :

Un chromosome est représenté par un vecteur de clients. Une heuristique d'insertion de tournées(HIT) est utilisée pour construire les tournées correspondantes au chromosome. L'heuristique d'insertion HIT traite les clients dans l'ordre de leur apparition dans le chromosome. Le but étant de construire un ensemble de m tournées de véhicules, avec m fixé à l'avance.

L'algorithme démarre en affectant les m premiers clients aux m véhicules disponibles, chaque tournée contient alors un et un seul client. Dans la figure 2 .12, les 3 premiers clients du chromosome C₂, C₄, C₁₀ sont affectés respectivement aux 3 véhicules V₁, V₂ et V₃.

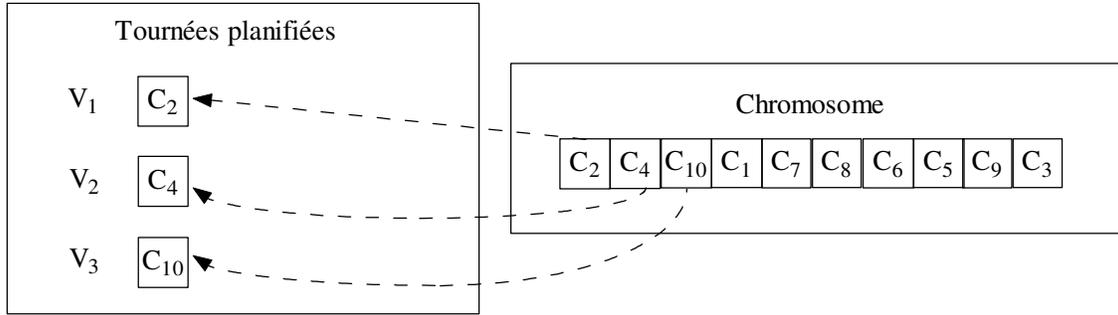


Figure 2. 12 Décodage d'un chromosome à l'aide de l'heuristique HIT

Ensuite, l'heuristique d'insertion traite les clients restants un par un suivant leur ordre d'apparition dans le chromosome. Un client i , placé avant un autre client j dans le chromosome et affecté au même véhicule sera toujours servi en premier dans la tournée correspondante. De ce fait, l'insertion d'un client se fait toujours en fin d'une tournée existante.

Pour chaque client à insérer, on n'essaie uniquement que les m positions qui se trouvent aux fins de m tournées existantes. Par exemple, dans la figure 2.13, le client à insérer est C_1 . Les positions d'insertion possibles sont : dans la tournée du véhicule v_1 après le client C_2 , dans la tournée du véhicule V_2 après le client C_4 , ou dans la tournée véhicule V_3 après le client C_{10} .

La contrainte de capacité doit être examinée pour chaque position d'insertion. Dans notre modification d'AG pour la résolution du VRPTWFD (noté FAG : FuzzyAG), la contrainte de capacité est formulée par l'inéquation 2.6 ($\alpha \hat{D}_v + (1 - \alpha) \underline{D}_v \leq Q$) ou l'inéquation 2.7 ($\beta \bar{D}_v + (1 - \beta) \hat{D}_v \leq Q$) selon la combinaison choisie de seuils α et β .

Si la contrainte de capacité est satisfaite à une position d'insertion appartenant à une tournée, alors on calcule son surcoût de transport par rapport à la partie déjà planifiée de la tournée, sinon l'insertion est impossible, on passe à la position suivante (tournée suivante). Dans certain cas lorsque la contrainte de capacité est trop stricte, c'est-à-dire que les valeurs des seuils α et β sont grandes, certains clients peuvent ne pas être insérés dans les tournées existantes (pas de position d'insertion possible).

On affecte le client à la position qui introduit le surcoût de transport minimal (i.e. le client C_1 est inséré sur la tournée du véhicule V_1 après le client C_2 dans la figure 2.13.).

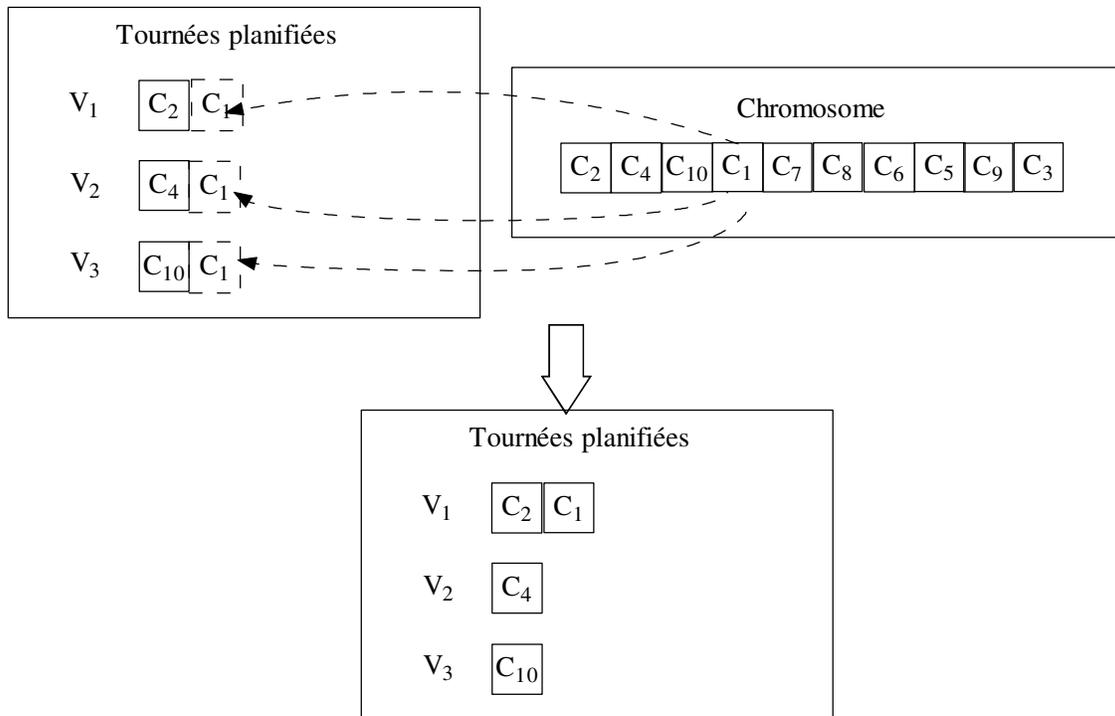


Figure 2. 13 Une étape du décodage d'un chromosome

Il faut noter qu'on ne calcule que le surcoût de transport pour chaque position d'insertion potentielle afin de réduire le temps de calcul. Le surcoût de transport est la somme de la distance parcourue supplémentaire et du retard chez le client à insérer. Par exemple, pour l'insertion de client C_1 dans la tournée du véhicule v_1 après le client C_2 , la distance parcourue supplémentaire est la distance entre les clients C_1 et C_2 , le retard chez le client C_1 dépend également la distance parcourue par le véhicule V_1 .

Cette heuristique d'insertion est décrite ci-dessous (algorithme 2.1) :

Algorithme 2.1 Algorithme d'heuristique d'insertion HIT pour le décodage d'un chromosome

Entrée :

Chrom[N] : le chromosome de taille N qui représente la liste des clients à servir
 NbVehicules : le nombre de véhicules

Sortie :

Tournées[NbVehicules] : L'ensemble de tournées planifiées
 Cout[NbVehicules] : le coût de transport sur chaque véhicule
 Cout : le coût planifié de transport de l'ensemble des tournées planifiées

A suivre =>

Algorithme 2.1 Algorithme d'heuristique d'insertion HIT pour le décodage d'un chromosome

Début :

Indice = 1

Client = Chrom[Indice]

// Etape 1 : on affecte le premier client de chaque tournée

TantQue (Indice<=N et Indice<= NbVehicules) Faire

Affecter Client à la Tournées [Indice]

Indice ++

FinTantQue

// Etape 2 : on affecte le reste des clients aux tournées en cours de construction

TantQue (Indice<=N) faire

MinSurcout = 0

NumVehCourant = 1

// On cherche la première position d'insertion du Client courant pour initialiser MinSurcout

TantQue (MinSurcout = 0 et NumVehCourant <= NbVehicules) Faire

tester l'insertion du Client à la fin de la Tournées [NumVehCourant]

Si (la contrainte de capacité est satisfaite) Alors

MinSurcout = le surcoût de transport correspondant

Sinon NumVehCourant++

FinSi

FinTantQue

// On cherche dans les autres positions d'insertion, la meilleure position donnant le meilleur Surcout

NumVehCourant++

TantQue (NumVehCourant <= NbVehicules) Faire

tester l'insertion du Client à la fin de la Tournées [NumVehCourant]

Si (la contrainte de capacité est satisfaite) Alors

Surcout = le surcoût de transport correspondant

Si Surcout < MinSurcout alors

MinSurcout = Surcout ; MinNum = NumVehCourant

FinSi

FinSi

NumVehCourant++

FinTantQue

Si (MinSurcout>0) Alors //Sinon Client ne peut pas être inséré dans les tournées

Affecter Client[Indice] à la fin de la Tournée[MinNum]

FinSi

Indice ++

FinTantQue

// Etape 3 : Calcul du coût planifié de transport des tournées

calculer le Cout

Fin.

On peut déduire les tournées à réaliser par chaque véhicule à la fin de la phase de décodage (voir la figure 2.14). Le coût planifié de transport associé peut aussi être évalué.

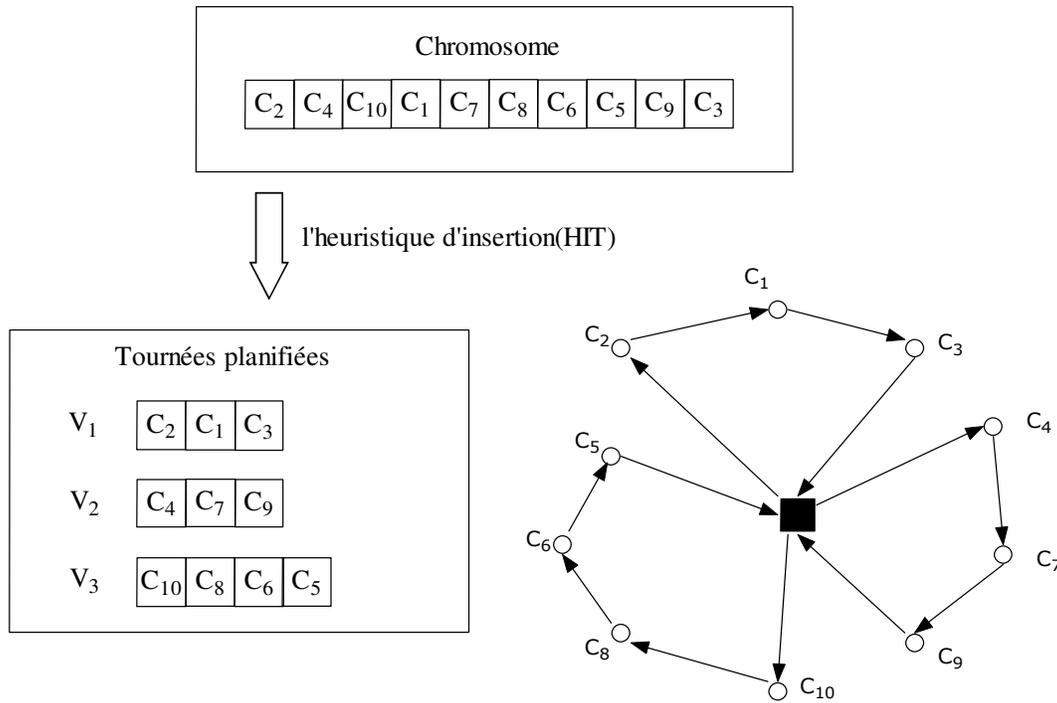


Figure 2. 14 Les tournées planifiées par la phase de décodage

La complexité de cet algorithme est en $O(n*m)$, où n est le nombre de clients à servir et m est le nombre de véhicules, ce qui la rend efficace quelque soit la taille de l'instance à traiter.

En fait, en utilisant ce codage indirect nous pouvons traiter d'autres variantes de VRP (par exemple : le problème du voyageur de commerce TSP, le problème du tournée du réparateur TRP, le problème de ramassage et de livraison de colis PDP, .. etc.). Par ailleurs, c'est là un des avantages de notre codage, l'algorithme de décodage étant adapté pour chaque type de problème.

2.4.3.2 Variantes autour de l'AG

L'évaluation du chromosome va dépendre du type de modèle à résoudre et elle est liée à la fonction d'objectif utilisée. De ce fait, nous distinguerons dans la suite de ce document, trois versions de notre AG correspondantes aux résolutions du VRPTWFD par deux modèles stochastiques de la section 2.3 et la version initiale existante pour la résolution du VRPTW avec demandes connues (ou déterministes).

1. Résolution du VRPTW déterministe

Dans la version de l'AG proposée dans les travaux de thèse [Housroum 2005] pour la résolution du VRPTW déterministe (notée « DAG »), la contrainte de capacité dans la phase

de décodage est vérifiée par une inéquation sur des nombres ordinaires $D_v \leq Q$, où D_v est la somme des demandes des clients servis par le véhicule v .

Le fitness du chromosome correspond au coût de transport des tournées planifiées en fin de la phase de décodage :

$$\text{fitness (chromosome dans la version « DAG »)} = f_{DR}$$

2. Résolution basée sur le modèle CCP du VRPTWFD

Dans la version d'AG intégrant le modèle CCP du VRPTWFD (notée « FAG_CCP »), la contrainte de capacité est vérifiée à la phase de décodage à l'aide de l'inéquation 2.6 ($\alpha \hat{D}_v + (1-\alpha) \underline{D}_v \leq Q$) ou l'inéquation 2.7 ($\beta \bar{D}_v + (1-\beta) \hat{D}_v \leq Q$), selon les valeurs des seuils α et β .

L'objectif ici est de minimiser le coût planifié de transport en maintenant le risque de « défauts des tournées » inférieur à un certain seuil. De ce fait, dans cette version de l'AG, le chromosome peut être évalué en passant la phase du décodage. Le fitness du chromosome est défini comme étant le coût planifié de transport des tournées planifiées en fin de la phase de décodage.

$$\text{fitness (chromosome dans la version « FAG_CCP »)} = f_{DR}^{Plan}$$

3. Résolution basée sur le modèle SPR du VRPTWFD

Nous avons intégré aussi un modèle SPR dans notre AG pour résoudre le VRPTWFD. L'objectif de ce modèle SPR est de chercher à minimiser le coût prévisionnel de transport. Ce coût est la somme du coût de transport des tournées planifiées de la première étape et le coût supplémentaire de transport induit par les recours dus aux « défauts des tournées » de la deuxième étape.

Cette version d'AG sera notée « FAG_SPR » dans la suite du document. Les tournées planifiées ainsi que le coût planifié de transport sont obtenus après la phase de décodage. Le fitness du chromosome est donné par le coût prévisionnel, que nous obtenons à la fin de la phase de simulation des tournées grâce à la procédure « rejouer » décrite plus haut [Xu et al. 2007b].

Dans cette phase de simulation, les tournées planifiées lors de la phase de décodage sont « rejouées » avec un ensemble de scénarios d'apprentissage de demandes réelles (cf section 2.3.6.1). Nous calculons la valeur moyenne des coûts de transports correspondants à ces scénarios d'apprentissage. Cette valeur moyenne correspond au coût prévisionnel (f_{DR}^{Prev}) qui est la somme du coût planifié et du coût supplémentaire de transport induit par les recours aux « défauts des tournées ». Le fitness du chromosome dans cette version est égal à ce coût prévisionnel :

$$\text{fitness (chromosome dans la version « FAG_SPR »)} = f_{DR}^{Prev} = f_{DR}^{Plan} + f_{DR}^{Supp}$$

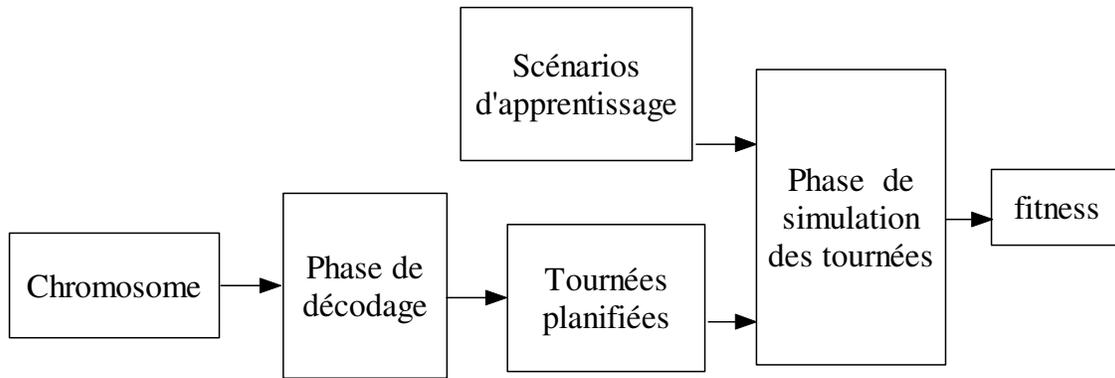


Figure 2. 15 L'évaluation d'un chromosome dans « FAG_SPR »

Chaque chromosome de la version « FAG_SPR » fournit après la phase de décodage et la phase de simulation de tournées une solution potentielle du problème (voir Figure 2.15).

2.4.4 Population initiale

La construction d'une population initiale représente l'étape de départ de l'algorithme génétique. En général, cette population est créée de façon aléatoire. Il est possible de placer les individus initiaux de façon uniforme dans l'espace de recherche. Une partie de ces individus peut être construite en utilisant des méthodes heuristiques simples pour obtenir des individus de bonne qualité. Par exemple l'heuristique de PFIH (Push Forward Insertion Heuristic) a été utilisée par [Solomon 1987] ou l'heuristique de sweep a été utilisée par [Baker et al. 2003]. Cette dernière méthode permet d'accroître la vitesse de convergence, mais empêche l'algorithme de se focaliser sur une autre partie de l'espace de recherche qui pourrait contenir l'optimum. Dans notre approche nous avons choisi de générer aléatoirement la population initiale.

2.4.5 Sélection

A chaque génération, l'opérateur de sélection sert à former une population intermédiaire à la population de la génération courante. Ces individus seront ensuite croisés et mutés pour former la population de la génération suivante. Les chromosomes sélectionnés représentent les individus les mieux adaptés. Il existe de nombreuses techniques de sélection, nous allons présenter les plus connues :

- La plus simple, connue sous le nom de "ranking" consiste à classer les n chromosomes de la population par ordre croissant de leur évaluation respective (ou décroissant selon l'objectif). Les m premiers individus sont ensuite sélectionnés. Ainsi, seuls les meilleurs individus sont conservés.
- La sélection par roulette ou "roulette wheel selection" : elle consiste à associer à chaque chromosome un segment dont la longueur est proportionnelle à son fitness. Ces segments sont ensuite concaténés sur un axe gradué que l'on normalise entre 0 et 1. On tire alors un nombre aléatoire de distribution uniforme entre 0 et 1, puis on identifie le segment

sélectionné et le chromosome correspondant. Avec cette technique, les bons chromosomes seront plus souvent sélectionnés que les mauvais, et un même chromosome pourra avec cette méthode être sélectionné plusieurs fois. Néanmoins, sur des populations de petite taille, il est difficile d’obtenir exactement l’espérance mathématique de sélection à cause du faible nombre de tirages. On aura donc un biais de sélection plus ou moins fort suivant la dimension de la population.

- La sélection par tournoi : cette technique tire au hasard deux ou plusieurs individus de la population et le plus fort est sélectionné, c’est-à-dire celui ayant le fitness le plus intéressant.
- La sélection aléatoire : comme son nom l’indique, ce type de sélection choisit le chromosome selon une distribution uniforme.

Dans notre plateforme, nous avons choisi la sélection par tournoi qui est utilisée par de nombreux auteurs [Goldberg et al. 1992], [Deb et al. 1999].

2.4.6 Opérateurs de croisement (Crossover)

Le croisement permet d’accoupler deux parents pour former deux enfants. L’idée est que les enfants garderont les meilleures caractéristiques de leurs parents. Prenons l’exemple de la fonction $f(x)$ décrite par la figure 2.16. Le principe du croisement consiste à recombinaison les gènes faisant partie des chromosomes des deux parents P1 et P2 pour engendrer deux enfants E1 et E2 de meilleure qualité. La difficulté réside dans le fait que la bonne partie du chromosome n’est pas connue et qu’il est alors possible de créer des enfants de plus mauvaise qualité que celles des parents initiaux. L’enfant E3 est un exemple de cette difficulté dans le cas d’une minimisation de la fonction $f(x)$. Cette difficulté est d’autant plus importante lorsqu’il s’agit d’un codage de type indirect puisque les bonnes parties des solutions parentes sont “masquées” par celui-ci.

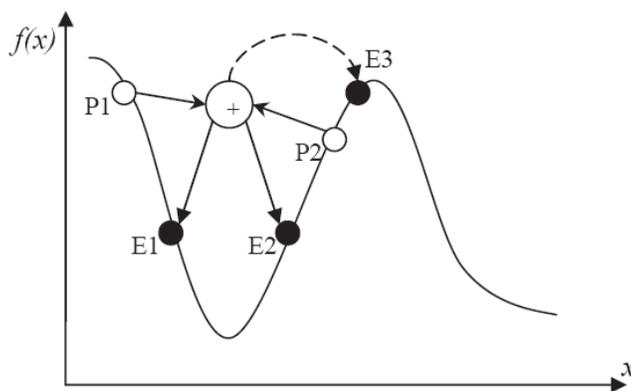


Figure 2. 16 Principe du croisement

Il existe de nombreux opérateurs de croisement, nous allons présenter ceux utilisés pour la résolution du problème VRPTW.

2.4.6.1 PMX (Partially Mapped Crossover)

Le PMX fut proposé par [Goldberg et al. 1985] dans le cadre de problèmes d'ordonnancement. Le but était de construire un enfant par le choix de sous séquences ordonnancées de l'un des parents et de préserver l'ordre et la position d'autant de sous séquences que possible des autres parents. La sous-séquence de l'ordonnancement est sélectionnée par le choix de deux points de coupure aléatoire, lesquels servent de frontière pour l'opération de substitution. Considérons par exemple les deux parents :

$$P1 = 123 | 4567 | 89$$

$$P2 = 452 | 1876 | 93$$

- Etape 1 : ces deux parents vont produire deux enfants. Dans cette première étape, les segments compris entre les points de coupures sont échangés :

$$E1 = xxx | 1876 | xx$$

$$E2 = xxx | 4567 | xx$$

Notons qu'en vue de procéder à l'étape 3 de résolution des conflits, il est possible d'établir entre les deux points de coupures une table de correspondances entre les allèles; cette table est définie comme suit dans notre exemple :

$$1 \leftrightarrow 4; 8 \leftrightarrow 5; 7 \leftrightarrow 6; 6 \leftrightarrow 7$$

- Etape 2 : les chromosomes "Enfants" sont complétés en transmettant les allèles non conflictuels issus du second parent.

$$E1 = x23 | 1876 | x9$$

$$E2 = xx2 | 4567 | 93$$

- Etape 3 : cette dernière étape consiste à terminer l'élaboration des descendants en résolvant les conflits à l'aide de la table de correspondance :

$$E1 = \mathbf{4}23 | 1876 | \mathbf{5}9$$

$$E2 = \mathbf{1}82 | 4567 | 93$$

2.4.6.2 MX1T/D :

Ce croisement a été proposé par [Sushil et al. 1999]. Il est composé de deux parties: le croisement heuristique (MX1D) et le croisement de fusion (MX1T). Le premier (MX1D) est basé sur les distances entre les clients et le deuxième (MX1T) est basé sur la borne inférieure des fenêtres de temps des clients. Dans ce croisement, deux parents produisent un seul descendant avec le croisement heuristique. Le deuxième descendant est obtenu à partir des mêmes parents avec le croisement de fusion.

1. MX1T :

Nous allons expliquer ce type de croisement en utilisant l'exemple suivant :

$$P1 = 2\ 5\ 6\ 1\ 0\ 7\ 3\ 8\ 4\ 9$$

$$P2 = 4\ 1\ 6\ 9\ 3\ 8\ 2\ 0\ 5\ 7$$

Cet opérateur utilise un vecteur global de précédence. A titre d'exemple, considérons le vecteur suivant :

0 1 2 3 4 5 6 7 8 9

Ce vecteur est basé sur la borne inférieure de la fenêtre de temps (e_i) de chaque client. Les clients dans ce vecteur, sont triés en ordre croissant selon leur borne (e_i). Dans ce vecteur, le client 1 peut être servi avant les clients 2 et 3, et ainsi de suite.

- Le premier client de P1 (client 2) précède le premier client de P2 (client 4) dans le vecteur global de précédence. Ce client est placé comme étant le premier client dans l'enfant E1. Pour maintenir la faisabilité de la suite de l'élaboration du chromosome, le premier client de P2 est permuté. C'est le client 4 qui est permuté avec le client 2:

P1 = 2 5 6 1 0 7 3 8 4 9

P2 = 4 1 6 9 3 8 2 0 5 7 \Rightarrow P2 = 2 1 6 9 3 8 4 0 5 7

E1 = 2 x x x x x x x x x

- Puis on compare le deuxième client de chaque parent (en gras) pour déterminer le deuxième client de l'enfant. Le client 1 de P2 apparaît dans le vecteur de précédence avant le client 5 de P1, ainsi nous le plaçons en deuxième position dans E1. Nous permutons le deuxième client de P1 pour maintenir la faisabilité. C'est le client 5 qui est permuté avec le client 1:

P1 = 2 **5** 6 1 0 7 3 8 4 9 \Rightarrow P1 = 2 1 6 **5** 0 7 3 8 4 9

P2 = 2 1 6 9 3 8 4 0 5 7

E1 = 2 **1** x x x x x x x x x

- Nous continuons de cette façon jusqu'à la fin des chromosomes. Finalement, l'enfant obtenu est le suivant :

E1 = 2 1 6 5 0 7 3 4 8 9

2. MX1D :

Cet opérateur exploite les distances entre les clients. Par exemple, nous avons les deux parents suivants :

P1 = 2 5 6 1 0 7 3 8 4 9

P2 = 4 1 6 9 3 8 2 0 5 7

- D'abord, nous choisissons aléatoirement l'un des deux premiers clients des parents (par exemple, le client 4 de P2), et nous initialisons l'enfant E2 avec ce client. Dans P1 nous permettons le client 4 avec le client 2, de manière à débiter la tournée avec le même client.

P1 = 4 5 6 1 0 7 3 8 2 9

P2 = 4 1 6 9 3 8 2 0 5 7

E2 = 4 x x x x x x x x x

- Maintenant nous comparons la distance entre le client 4 et le client 5 (c_{45}) à la distance entre le client 4 et le client 1 (c_{41}). On suppose que $c_{45} > c_{41}$, nous choisissons alors 1 pour

être le prochain client de E2. Pour les mêmes raisons que précédemment (faisabilité de la poursuite), on doit permuter le client 1 et le client 5 dans P1.

$$P1 = 4 \mathbf{5} 6 \mathbf{1} 0 7 3 8 2 9 \Rightarrow P1 = 4 \mathbf{1} 6 \mathbf{5} 0 7 3 8 2 9$$

$$P2 = 4 \mathbf{1} 6 9 3 8 2 0 5 7$$

$$E2 = 4 \mathbf{1} x x x x x x x x$$

- On continue de la même manière jusqu'à ce qu'un nouveau chromosome E2 soit entièrement produit.

$$E2 = 4 1 6 5 3 7 2 8 0 9$$

Les parents choisis par l'opérateur de sélection ne seront pas tous croisés. Une probabilité permet de décider si les parents seront utilisés pour le croisement ou s'ils seront tout simplement recopiés. Le croisement est davantage basé sur l'exploitation du voisinage des parents que sur l'exploration de l'espace de recherche.

2.4.7 Opérateurs de mutation

La mutation permet d'explorer l'espace de recherche en évitant à l'algorithme de converger trop rapidement vers un optimum local. La figure 2.17 présente un exemple de fonction $f(x)$ à minimiser. Les jetons noirs représentent les chromosomes de la population qui convergent vers un optimum local. Le principe de la mutation est donc de prendre un de ces chromosomes et de le modifier afin d'explorer une autre partie de l'espace de recherche que l'on espère plus intéressante.

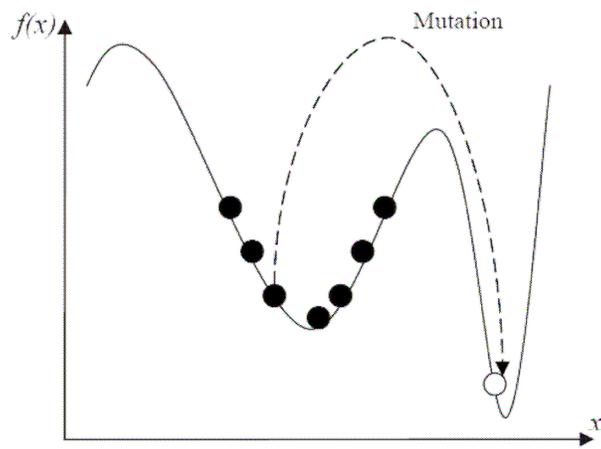


Figure 2. 17 Principe de la mutation

La mutation est traditionnellement considérée comme un opérateur marginal bien qu'elle confère en quelque sorte aux algorithmes génétiques la propriété d'ergodicité (i.e. tous les points de l'espace de recherche peuvent être atteints). Cet opérateur a pour but de permettre à une recherche de « s'extraire » de l'optimum local afin d'explorer d'autres régions.

On appelle voisinage d'une solution s , un ensemble de solutions qui sont obtenues en appliquant à la solution s une fonction de transformation. Par exemple une fonction de

permutation de 2 clients, est un opérateur de voisinage, et le voisinage d'une solution $s = (1, 2, 3, 4)$ est un ensemble de solutions $\{(2, 1, 3, 4), (3, 2, 1, 4), (4, 2, 3, 1), (1, 3, 2, 4), (1, 4, 3, 2), (1, 2, 4, 3)\}$. Le voisinage d'une solution dépend directement de son opérateur.

Un opérateur de mutation peut être assimilé à un opérateur de voisinage. Plusieurs classifications des techniques de voisinage existent dans littérature [Funke et al. 2005] [Bräysy 2005]. Elles sont classées en deux types d'après l'entité manipulée:

- Arc-échange voisinages : la famille des opérateurs λ -Opt ($\lambda = 2, 3, 4, \dots$) est le mécanisme le plus utilisé dans ce groupe. Il s'agit de remplacer λ arcs existants (un arc est une liaison entre deux clients consécutifs dans une tournée) par λ nouveaux arcs.
- Nœud-échange voisinages. Dans les opérateurs de ce type, l'entité considérée est la chaîne de nœuds, et non plus l'arc ou l'arrêt. C'est-à-dire les chaînes manipulées sont constituées d'un ou plusieurs clients.

Dans notre cas, il est impossible d'utiliser directement les arc-échange voisinages parce que le chromosome ne permet pas de distinguer clairement les tournées dans le codage indirect que nous utilisons.

L'opérateur de mutation sur un chromosome dans notre cas est basé sur les nœud-échange voisinages :

1. Echange (nommé aussi swap) :

Cet opérateur est inspiré de la technique de mutation utilisée pour la résolution de problèmes de voyageur de commerce (TSP) [Grefenstette et al. 1985]. Cet opérateur est le plus répandu. Il consiste à prendre au hasard deux gènes (2 clients) du chromosome, D et B dans l'exemple (voir la figure 2.18) et à les inverser.

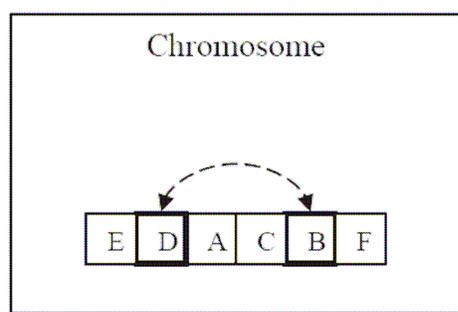


Figure 2. 18 L'opérateur de mutation échange

2. λ -échange

Cet opérateur, proposé par [Osman 1993], choisit aléatoirement deux suites de gènes (clients), puis les échange. Il se distingue avec l'opérateur précédent par la taille des suites choisies, qui peut varier entre 0 et λ (nous utilisons $\lambda=3$). Cet opérateur échange deux suites ou il déplace une suite vers une autre position si la taille de l'une des suites est égale à zéro (voir la figure 2.19).

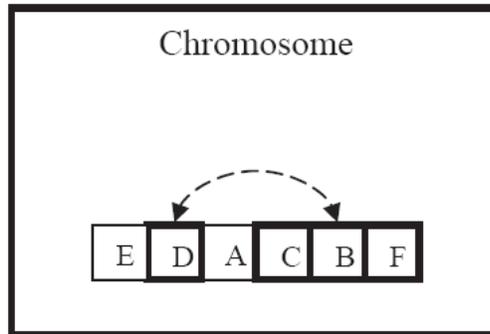


Figure 2. 19 L'opérateur de mutation λ -échange

L'opérateur de mutation ne s'applique pas à tous les chromosomes. Chacun a une probabilité, d'être muté. Cette probabilité est en général très faible.

Une fois que les chromosomes enfants sont générés par la reproduction en utilisant les opérateurs de croisement et de mutation, il faut les évaluer afin de pouvoir sélectionner la prochaine population pour l'itération suivante. Cette évaluation se réalise à l'aide de la fonction d'évaluation, c'est-à-dire avec la fonction d'objectif du problème traité.

2.4.8 Méthode de remplacement par l'élitisme

A la création d'une nouvelle population, il y a de grandes chances pour que les meilleurs chromosomes soient perdus après les opérations de croisement et de mutation. Pour éviter cela, on utilise une méthode d'élitisme. Cette méthode améliore en général considérablement les algorithmes génétiques, car elle permet de ne pas perdre les meilleures solutions trouvées au cours de l'évolution [Michalewicz 1994].

Dans notre approche, nous utilisons une population temporaire de la taille double de la population pour l'étape de remplacement (voir la figure 2.20). Nous copions les chromosomes de la population courante dans cette population temporaire, et y mettons aussi les chromosomes enfants générés par les opérateurs génétiques. Puis, nous trions les chromosomes de la population temporaire par ordre croissant de leur fitness. Ensuite, nous construisons la nouvelle population pour la prochaine génération à l'aide du mécanisme de l'élitisme. Nous remplissons un pourcentage de la nouvelle population avec les meilleurs chromosomes de la population temporaire triée tandis que l'autre partie de la nouvelle population est remplie par des chromosomes générés aléatoirement afin d'introduire de la diversité. Avec cette représentation le chromosome ayant le meilleur fitness est toujours rangé en position 1.

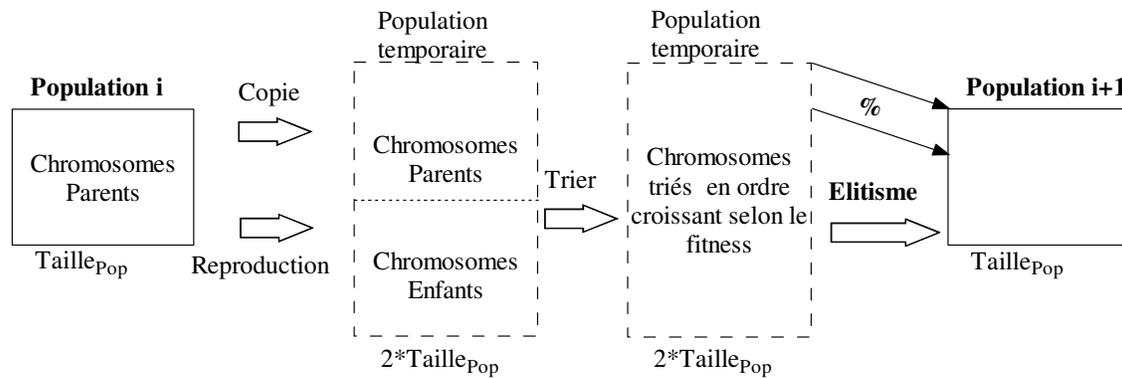


Figure 2. 20 L'évolution générale de la population

A ce stade, nous avons présenté les différentes étapes de notre algorithme génétique. Nous présentons dans ce qui suit les résultats obtenus lors de la résolution du VRPTWFD par les différentes variantes de notre AG.

2.5 Etudes expérimentales

2.5.1 Le benchmark utilisé

Il n'existe pas de benchmark commun pour le problème SVRP dans la littérature [Tan et al. 2007]. Pour évaluer notre modèle de traitement des demandes incertaines, nous avons créé un benchmark pour le VRPTWFD en modifiant le jeu de problèmes fournis par [Solomon 1987] pour le VRPTW.

2.5.1.1 Benchmark de Solomon pour le VRPTW

Le benchmark 100 de Solomon est une base de référence pour l'étalonnage des performances de la plupart des méthodes proposées pour le VRPTW. Il comprend 56 problèmes au total. Chaque problème contient 100 clients distribués géographiquement dans un carré de 100 sur 100. Le dépôt est situé au centre du carré. On suppose que tous les véhicules sont identiques et se déplacent tous à une vitesse constante et égale à 1 (unité de distance/unité de temps). De ce fait, le temps de trajet entre deux clients est égal à la distance euclidienne entre ces deux clients. Les problèmes de ce benchmark sont regroupés en 6 catégories (C1, C2, R1, R2, RC1, RC2).

- Les lettres des noms des six types de problèmes indiquent la loi de distribution des clients dans l'espace (voir la figure 2.21).
 - (a). le type R correspond à une distribution uniforme des clients dans l'espace.
 - (b). le type C correspond à une distribution des clients dans des clusters deux à deux disjoints.
 - (c). le type RC est une combinaison des deux modes de répartition précédents. Une partie des clients est distribuée uniformément dans le carré tandis que l'autre partie est distribuée dans des regroupements (clusters).

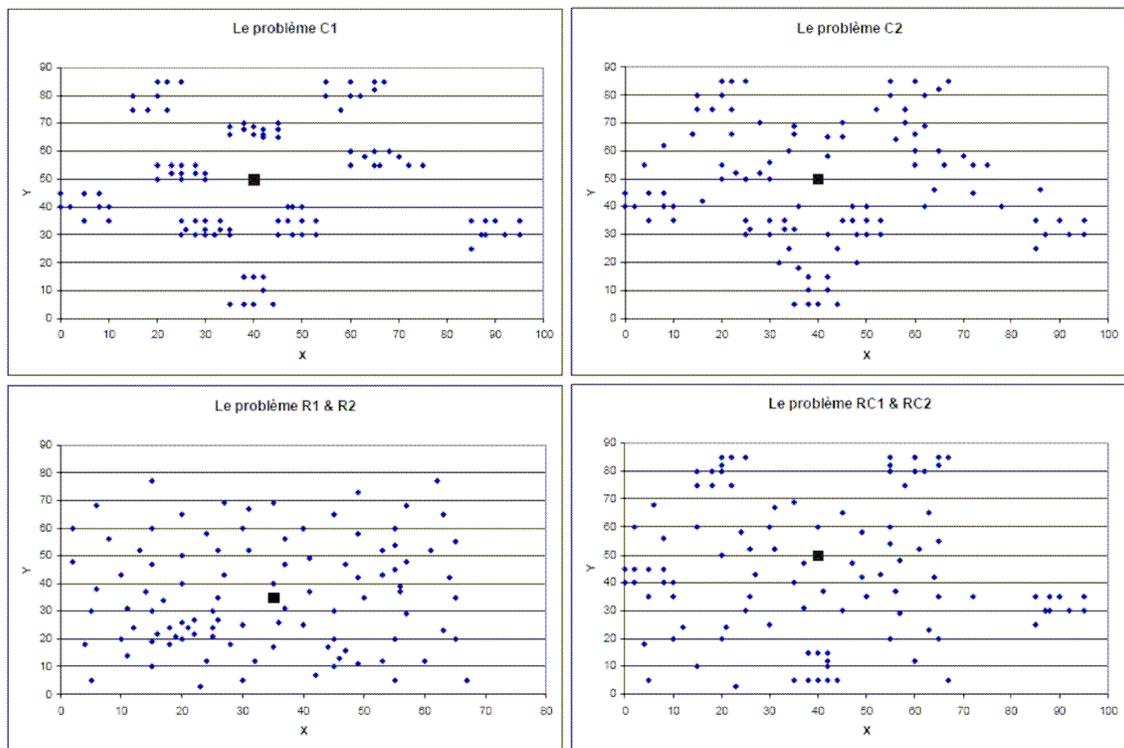


Figure 2. 21 La distribution des clients dans les 6 types de problèmes

- Les chiffres 1 ou 2 indiquent la taille de la fenêtre de temps.
 - (a). le type 1 met en place une fenêtre de temps étroite sur le dépôt central. Ceci implique un horizon de service court ainsi que des tournées comportant peu de clients. La capacité des véhicules est également réduite.
 - (b). le type 2 met en place des fenêtres de temps larges sur le dépôt central. L'horizon de service étant plus étendu, les tournées comprendront en principe plus de clients. En plus la capacité des véhicules est assez large.
- Chaque catégorie contient des problèmes qui se distinguent par des propriétés de fenêtres de temps :
 - (a). La densité de fenêtres de temps, qui est le pourcentage des clients possédant les fenêtres de temps, étant 25, 50, 75 et 100 %.
 - (b). Les positions des fenêtres de temps.
 - (c). la restriction des fenêtres de temps.

Par exemple, pour le problème R104, premier problème de type R1, les clients sont uniformément répartis dans l'espace, avec un horizon de service court et une capacité des véhicules faible. 25% des clients possèdent une fenêtre de temps de la taille identique. Dans le problème RC205, les clients sont répartis de façon mixte (uniforme et clusters), avec un horizon de service étendu et une capacité de véhicules importante. La densité de fenêtres de temps du RC205 est 100%, autrement dit, chaque client possède une fenêtre de temps.

Il convient de rappeler que ce benchmark a été conçu pour le VRPTW. Dans la thèse d'H. Housroum [Housroum 2005], celui-ci avait été utilisé pour mesurer les performances de son

approche génétique et mettre au point le réglage des différents paramètres génétiques.

2.5.1.2 Benchmark pour le VRPTWFD

Nous modifions le benchmark de Solomon pour l'adapter au VRPTW avec demande floue. La modification du benchmark concerne les demandes des clients et la capacité des véhicules :

1. Demande floue

Les demandes des clients sont des nombres flous $\tilde{d}_i = (\underline{d}_i, \hat{d}_i, \bar{d}_i)$ à la forme triangulaire symétrique, c'est à dire que $\hat{d}_i - \underline{d}_i = \bar{d}_i - \hat{d}_i$. Pour cela, nous ajoutons un nouvel attribut à chaque client pour modéliser la demande floue du client. Cet attribut est appelé le taux d'incertitude, et noté tn_i . La valeur du taux d'incertitude du client est tirée aléatoirement suivant une loi uniforme entre 0 et un tiers (1/3). Les trois composants de la demande floue \tilde{d}_i du client i sont déterminés de la manière suivante (voir Tableau 2.2) :

- La normalisation (\hat{d}_i) égale à la demande initiale de ce client dans le benchmark de Solomon.
- La borne inférieure $\underline{d}_i = \hat{d}_i \times (1 - tn_i)$
- La borne supérieure $\bar{d}_i = \hat{d}_i \times (1 + tn_i)$

Client	Demande dans C109	tn_i	Demande floue (\tilde{d}_i)
1	30	0.25	(22.5, 30, 37.5)
2	10	0.17	(8.3, 10, 11.7)
3	20	0.31	(13.8, 20, 26.2)
4	40	0.02	(39.2, 40, 40.8)
...			

Tableau 2. 2 La construction des demandes floues

2. La capacité des véhicules

Rappelons que dans notre problème, le nombre de véhicules utilisés n'est pas un objectif à minimiser. En fait, il a été fixé par [Gendreau et al. 1999] comme étant le nombre optimal de véhicules obtenu en résolvant la version déterministe et statique du VRPTW correspondant dans [Taillard et al. 1997].

Il est à noter que dans le benchmark de Solomon, la capacité des véhicules est fixée à une valeur assez grande. La contrainte de capacité jouant un rôle important dans le domaine du

VRP avec demande stochastique, l'étude du VRP avec demande stochastique n'est justifiée que si la capacité des véhicules est une contrainte forte du problème. De ce fait, nous avons dû réduire la capacité des véhicules. Il est à noter que l'absence de benchmark reconnu pour le problème SVRP nous a conduits à adapter le benchmark de Solomon :

- Pour les problèmes de type 1, la capacité des véhicules est définie par :

$$Q = \left\lceil \frac{\text{Demande totale des clients} \times (1 + 10\%)}{\text{Nombre de véhicules}} \right\rceil$$

La capacité totale des véhicules est égale à la demande totale des clients plus 10%. Prenons l'exemple du problème R101, le nombre de véhicules est fixé à 19. La demande totale des clients du R101 est égale à 1458. D'après l'équation ci-dessus, la capacité par véhicule est égale à $(1458 * 1,1) / 19 = 84.41$. Donc, en arrondissant ce nombre à l'entier supérieur nous obtenons la capacité des véhicules (85) pour R101.

- Pour les problèmes de type 2, dans lesquels il y a très peu de véhicules, la capacité des véhicules ne sera pas assez contraignante si elle est définie comme pour les problèmes de type 1. C'est pourquoi, nous réduisons la capacité des véhicules par la définition suivante:

$$Q = \left\lceil \frac{\text{Demande totale des clients}}{\text{Nombre de véhicules}} + \frac{\text{Demande totale de clients}}{\text{Nombre de clients}} \right\rceil$$

A titre d'exemple, pour le problème C201, la demande totale des 100 clients est égale à 1810. Pour ce problème, le nombre de véhicules est fixé à 3. Ce nombre est le nombre optimal de véhicules pour C201. Le calcul de la capacité par véhicule nous donne $1810/3 + 1810/100 = 621.43$, une valeur approchée 620 est définie pour la capacité des véhicules.

Nous ajoutons un préfixe « S » sur les noms des instances du benchmark modifié pour le VRPTWFD. Nous obtenons 6 types (SC1, SC2, SR1, SR2, SRC1, SRC2) de problèmes à traiter. Chaque problème est représenté par un fichier qui contient trois parties de données :

- La partie qui concerne les attributs des clients :
 - (a). (x; y) : est la position de client.
 - (b). d : est la normalisation de la demande floue du client.
 - (c). tIn : le taux d'incertitude du client qui est un nouvel attribut pour le VRPTWFD. On modélise la demande floue du client en utilisant la valeur de la normalisation d et le taux d'incertitude tIn.
 - (d). (e; l) : est la fenêtre de temps du client, dans laquelle il souhaite être servi.
 - (e). s : est la durée de service chez le client. Notons que le temps de service auprès de chaque client est fixé à 10 unités de temps pour les problèmes de type SR et SRC. Il est de 90 unités de temps pour les problèmes de type SC.
- La partie qui concerne le dépôt :
 - (a). (x; y) : détermine sa position.
 - (b). (e; l) : sont les horaires d'ouverture et de fermeture.
- La dernière partie concerne les véhicules disponibles :
 - (a). La capacité du véhicule.
 - (b). Le nombre des véhicules. Nous le fixons au nombre optimal de véhicules obtenu en

résolvant la version déterministe et statique du VRPTW.

Nous avons sélectionné pour chaque type de problèmes quatre instances à résoudre. Les problèmes choisis présentent des caractéristiques différentes (nombre de véhicules, la densité de fenêtres de temps, la largeur moyenne des fenêtres de temps, ...).

2.5.2 Simulation

Dans cette partie, nous allons résoudre chaque problème choisi par le modèle CCP et le modèle SPR en exécutant les versions « FAG_CCP » et « FAG_SPR » de notre AG. Avant cela, il convient de définir quels sont les réglages que nous avons effectués pour l'AG et pour les modèles CCP et SPR.

2.5.2.1 Les paramètres

L'efficacité de l'algorithme génétique dépend fortement de la valeur des paramètres suivants :

- la taille de la population(Taille).
- le type de croisement(Cr).
- la probabilité de croisement(T2Cr).
- le type de mutation (Mt).
- la probabilité de mutation(T2Mt).
- le taux d'élitisme(T2Elt).

Le réglage des paramètres reste généralement très empirique. H. Housroum, dans ses travaux de thèse [Housroum 2005] a utilisé les plans d'expériences pour régler les paramètres de l'algorithme génétique dédié à la résolution du problème VRPTW. Ses travaux ont montré que d'une part l'opérateur de croisement de MX1T/D est très influent sur la plupart des problèmes, et d'autre part que l'opérateur de mutation λ -échange est meilleur que les autres dans la plupart des combinaisons.

Pour chaque problème à traiter, nous avons adopté les réglages optimaux obtenus par H. Housroum pour déterminer les combinaisons de paramètres qui aide l'algorithme génétique à converger rapidement vers une bonne solution. Les réglages utilisés pour chacun des problèmes se trouvent dans l'Annexe 2.1.

Nous avons cité dans la section 2.3.2 l'influence des seuils α et β pour la possibilité et la nécessité de la contrainte de capacité floue. Le réglage des seuils α et β a une influence importante sur le coût de transport de tournées. Nous déterminons le réglage des seuils α et β en simulant le problème à résoudre. Autrement dit, nous cherchons plusieurs solutions du problème correspondantes aux différentes combinaisons des seuils α et β en exécutant l'AG, et choisissons la combinaison qui donne la meilleure solution du problème à résoudre.

Afin de réduire le nombre de simulations, nous créons un ensemble de combinaisons des seuils α et β de la façon suivante : nous augmentons la valeur du seuil α de 0 à 1 par pas de 0.2 avec la valeur du seuil β fixé à 0. Puis, lorsque α est égal à 1, nous augmentons la valeur de β de 0 à 1 par pas de 0.1. Les combinaisons des seuils α et β représentent une évolution de

plus en plus stricte de la contrainte de capacité floue.

Nous résolvons chaque problème par les modèles CCP et SPR en exécutant la version de l'algorithme génétique correspondante, et nous déterminons le réglage des seuils correspondants. Compte tenu du caractère stochastique des algorithmes génétiques, nous exécutons l'algorithme génétique cinq fois pour chaque essai (i.e. combinaison) et nous calculons la moyenne de chaque réponse. Le nombre maximal de générations a été fixé à 10000.

2.5.2.2 Réglage des seuils α et β du modèle CCP

Pour chaque combinaison de seuils α et β , nous recherchons une bonne solution du problème en exécutant l'algorithme génétique « FAG_CCP ».

A titre d'exemple pour le problème SRC101, les solutions de modèle CCP correspondants aux différentes combinaisons de seuils α et β , sont données dans le tableau 2.3.

Combinaison de seuils		Solution du modèle CCP		Réalisation de la solution	
α	β	f_{DR}^{Plan}	Nombre de clients planifiés	f_{DR}^M	f_{DR}^S
0	0	1727.23	100	2765.71	1038.48
0.2	0	1743.38	100	2757.41	1014.03
0.4	0	1779.29	100	2839.64	1060.35
0.6	0	1799.04	100	2539.97	740.93
0.8	0	1810.16	100	2179.38	369.22
1	0	1870.03	100	2042.77	172.74
1	0.1	1901.24	100	1981.97	80.73
1	0.2	1968.46	100	2020.49	52.04
1	0.3	2019.64	100	2029.75	10.11
1	0.4	1986.55	99	1988.48	1.92

Tableau 2.3 L'ensemble des solutions du modèle CCP correspond aux combinaisons des seuils pour le problème SRC101

Le coût de transport (f_{DR}^{Plan}) de la solution obtenue correspondant à chaque combinaison des seuils α et β est présenté dans la troisième colonne du tableau 2.3. La

quatrième colonne indique le nombre de clients dans les tournées planifiées. Nous avons déjà mentionné dans la section 2.2.2.1 que certains clients peuvent ne pas être insérés dans les tournées si les valeurs des seuils α et β sont trop grandes. C'est ce que nous constatons pour les valeurs des seuils $\alpha=1$ et $\beta=0.4$, le nombre de clients servis est de 99 au lieu de 100. Notre objectif ici est d'élaborer des tournées qui servent tous les clients, la solution obtenue avec $\alpha=1$ et $\beta=0.4$ ne sera pas valable même si le coût supplémentaire de transport dû aux « défauts des tournées » est nul. Dans la suite, nous ne retenons comme combinaison possible que les combinaisons avec $\alpha \leq 1$ et $\beta \leq 0.3$.

Nous utilisons une phase de simulation pour évaluer ces solutions par rapport au réglage des seuils α et β . Nous avons généré un ensemble de scénarios d'apprentissage. Nous « rejouons » avec la procédure « rejouer » précédente, les tournées planifiées sur chacun de ces scénarios pour déterminer son coût réel de transport. Les moyennes de ces coûts de transport (f_{DR}^M) sont reportées dans la colonne 5. Nous reportons dans la colonne 6 les moyennes des coûts supplémentaires (f_{DR}^S) dus aux « défauts de tournées », ce coût représente la différence entre le coût planifié et de coût réel :

$$f_{DR}^S = f_{DR}^M - f_{DR}^P$$

L'idée du modèle CCP est d'élaborer un ensemble de tournées dans lesquelles les « défauts de tournées » seront inférieurs à certain seuil tout en minimisant les coûts. La meilleure combinaison retenue pour le modèle CCP pour SRC101 est obtenue, lorsque $\alpha=1$ et $\beta=0.1$. La solution correspondant à cette combinaison est considérée comme étant la solution du modèle CCP pour le problème SRC101.

Nous constatons que plus les valeurs des seuils α et β sont grandes, plus la contrainte de capacité floue est stricte, plus grand sera le coût engendré, et moins il y aura de risque de défauts dans les tournées planifiées. La troisième colonne du tableau montre que le coût planifié de transport augmente avec le durcissement de la contrainte de capacité floue. Nous vérifions également avec la dernière colonne du tableau que le coût supplémentaire de transport dû aux « défauts de tournées » diminue lorsque la contrainte de capacité floue se durcit.

2.5.2.3 Réglage des seuils α et β du modèle SPR

Pour chaque combinaison de seuils α et β , nous recherchons une bonne solution du problème en exécutant l'algorithme génétique « FAG_SPR ». Le nombre de scénarios d'apprentissage est un nouveau paramètre, il a été fixé à 20. Nous analyserons son influence sur la résolution du problème dans la section 2.5.6.1.

A titre d'exemple, nous résolvons le problème SRC101 par le modèle SPR et nous présentons les solutions correspondantes aux différentes combinaisons de seuils α et β dans le tableau 2.4. Le coût planifié de transport (f_{DR}^{Plan}), le coût prévisionnel de transport (f_{DR}^{Prev}),

le coût supplémentaire de transport dû aux « défauts des tournées » (f_{DR}^{Supp}) et le nombre de clients dans les tournées planifiées de chaque solution y sont indiqués.

Combinaison de seuils		Solution du modèle SPR			
α	β	f_{DR}^{Plan}	f_{DR}^{Prev}	f_{DR}^{Supp}	Nombre de clients planifiés
0	0	2011.51	2068.21	56.70	100
0.2	0	1968.34	2052.23	83.89	100
0.4	0	2008.19	2079.04	70.85	100
0.6	0	1988.66	2029.72	41.05	100
0.8	0	1934.65	1973.38	38.73	100
1	0	1903.82	1932.58	28.76	100
1	0.1	1912.59	1936.41	23.82	100
1	0.2	1909.33	1929.02	19.68	100
1	0.3	1943.43	1947.83	4.41	100
1	0.4	1877.38	1877.38	0.00	99

Tableau 2. 4 L'ensemble des solutions du modèle SPR correspond aux combinaisons des seuils pour le problème SRC101

Nous voyons qu'il faut avoir une combinaison $\alpha \leq 1$ et $\beta \leq 0.3$ pour pouvoir servir tous les clients. Au fur et à mesure du durcissement de la contrainte de capacité, le coût prévisionnel de transport a tendance à diminuer jusqu'à la combinaison $\alpha=1$ et $\beta=0.2$. Le coût supplémentaire de transport est minimal pour $\beta=0.3$. L'objectif du problème étant de minimiser le coût prévisionnel de transport, la meilleure combinaison retenue pour le SRC101 est $\alpha=1$ et $\beta=0.2$. La solution correspondante à cette combinaison est donc considérée comme étant la solution de référence du problème SRC101.

Nous constatons que le coût supplémentaire de transport dû aux nombres de « défauts de tournées » diminue lorsque nous durcissons la contrainte de capacité floue.

A travers les tableaux 2.3 et 2.4, nous remarquons que le coût supplémentaire de transport (f_{DR}^{Supp}) de la solution du modèle SPR est inférieur au coût supplémentaire (f_{DR}^S) obtenu après la phase de simulation de la solution du modèle CCP pour la même combinaison de seuils. Par exemple, pour la combinaison de $\alpha=1$ et $\beta=0$, le coût supplémentaire de transport est égal 28.76 pour le modèle SPR, il est égal à 172.24 pour le modèle CCP. L'explication vient du fait que le coût supplémentaire de transport dû aux « défauts des tournées » est pris en compte à la deuxième étape du modèle SPR en estimant celui-ci à l'aide de scénarios d'apprentissage.

Autrement dit, d'une part, l'utilisation de la théorie des possibilités pour modéliser la contrainte de capacité floue nous aide à maintenir les « défauts de tournées » en dessous d'un certain seuil en choisissant une bonne combinaison de α et β . La solution obtenue comporte dans ce cas moins de risque de défauts. D'autre part, le modèle SPR à l'aide de ces scénarios d'apprentissage, nous permet de minimiser le coût prévisionnel de transport.

2.5.2.4 Solutions du benchmark VRPTWFD

Nous avons résolu les problèmes sélectionnés précédemment pour le benchmark VRPTWFD à l'aide des deux versions AG (FAG_CCP et FAG_SPR) et en utilisant les meilleures combinaisons de seuils α et β .

L'ensemble des solutions par rapport aux différentes combinaisons des seuils α et β sont présentées dans l'Annexes A.2 et A.3 . Nous avons reporté les meilleures solutions obtenues en fonction des valeurs de seuils α et β pour chaque problème du benchmark dans le tableau 2.5.

Les valeurs des seuils α et β pour les modèles de CCP et SPR dépendent des caractéristiques du problème à traiter, comme par exemple la distribution géographique des clients, les demandes des clients, le nombre de véhicules, ou la capacité du véhicule, etc.

A titre de comparaison, nous résolvons aussi le VRPTWFD comme un VRPTW déterministe en utilisant les valeurs normalisation des demandes floues des clients(modèle Nor). La version AG « DAG » qui intègre le modèle Nor nous permet de trouver les solutions présentées dans la dernière colonne du tableau 2.5.

Les coûts planifiés de transport des solutions fournies par les 3 modèles sont identiques pour le problème SC101. Le coût planifié de transport du modèle SPR est égal à celui du modèle CCP pour trois problèmes du type SC2 : SRC201, SC205, SC208.

Dans la section suivante, nous évaluons la qualité des solutions obtenues dans ces 3 modèles.

Problème	Solution du modèle SPR					Solution du modèle CCP			Solution du modèle Nor
	α	β	f_{DR}^{Plan}	f_{DR}^{Prev}	f_{DR}^{Supp}	α	β	f_{DR}^{Plan}	f_{DR}^{Plan}
SC101	1	0	828.94	986.06	157.12	1	0	828.94	828.94
SC105	1	0	864.61	917.50	52.89	1	0.1	906.63	828.94
SC108	1	0	887.89	912.64	24.76	1	0.1	916.58	828.94
SC109	1	0	901.65	916.70	15.05	1	0.1	896.70	828.94
SC201	1	0.05	636.72	656.56	19.84	1	0.05	636.72	603.88
SC205	1	0.05	630.12	633.62	3.5	1	0.05	630.12	601.2
SC207	1	0.05	630.32	637.62	7.3	1	0.05	630.32	621.85
SC208	1	0.05	629.95	629.95	0	1	0.05	630.08	600.65
SR101	1	0.2	1756.94	1765.25	8.31	1	0.2	1774.39	1747
SR103	1	0.0	1326.63	1328.76	2.13	1	0.1	1316.30	1334.97
SR104	1	0.1	1213.58	1217.33	3.75	1	0.1	1223.91	1180.43
SR109	1	0.1	1347.51	1355.4	7.89	1	0.1	1368.05	1364.55
SR205	0.6	0	1160.71	1165.42	4.7	0.2	0	1134.91	1151.72
SR206	1	0.1	1055.67	1059.29	3.62	0	0.8	1041.40	1056.52
SR207	0.2	0	983.24	983.81	0.57	0.2	0	939.50	990.19
SR208	1	0	910.18	911.46	1.28	0.6	0	914.54	953.13
SRC101	1	0.2	1909.33	1929.02	19.68	1	0.1	1901.24	1830.88
SRC104	1	0.1	1383.95	1394.86	10.91	1	0.1	1362.77	1298.10
SRC106	1	0.1	1631.95	1645.64	13.68	1	0.1	1627	1628.72
SRC107	1	0.1	1425.94	1435.24	9.29	1	0.1	1446.67	1491.59
SRC202	0.6	0	1316.77	1318.79	2.02	1	0.1	1352.32	1374.27
SRC204	0.8	0	1093.92	1099.25	5.33	0.8	0	1047.03	1103.53
SRC205	0.8	0	1486	1490.05	4.05	1	0	1509.06	1486.18
SRC206	0.4	0	1382.68	1387.12	4.44	0.8	0	1410.52	1480.41

Tableau 2. 5 Solutions trouvées par trois modèles pour les problèmes à traiter

2.5.3 Evaluation des solutions dans un environnement réel

Nous ne pouvons pas comparer nos solutions avec celles présentées dans la littérature par manque de benchmark commun. De ce fait, nous évaluons la qualité des solutions obtenues par les modèles CCP, SPR et Nor en simulant les situations réelles pour chaque problème. Ces résultats sont présentés dans le tableau 2.5.

Dû à l'incertitude des demandes des clients, le coût réel de transport (f_{DR}^{Reel}) de la solution trouvée ne peut être connue qu'au moment où les tournées planifiées sont réellement effectuées par les véhicules.

Nous générons pour chaque problème quatre scénarios « test » comportant les demandes réelles qui représentent les valeurs connues lorsque les véhicules arrivent chez les clients. La génération des demandes réelles a été présentée dans la section 2.3.6.1. Chaque scénario « tests » correspond à une réalisation possible des variables floues au moment où les tournées planifiées sont réellement effectuées par les véhicules. Ils sont différents des scénarios d'apprentissage utilisé par le modèle SPR.

Pour une solution donnée, on aura donc un coût réel de transport qui sera différent en fonction du scénario « test » utilisé. Ce coût réel de transport est obtenu en utilisant toujours la procédure de « rejouer » présentée dans la section 2.3.6.2.

2.5.4 Comparaison avec la version déterministe

Pour évaluer la qualité des solutions obtenues par les trois modèles précédents, nous comparons leur performance avec celle d'un modèle déterministe où la demande des clients est connue avec certitude, donc en situation favorable pour chacun des scénarios « test » du problème. La solution de ce modèle de référence est obtenue en exécutant la version « DAG » de notre algorithme génétique. Cette solution nous donne le coût de transport déterministe (f_{DR}^{Det}) de référence.

Pour chaque scénario « test », nous calculons les valeurs t^{SPR} , t^{CCP} , t^{Nor} , qui représentent les pourcentages d'augmentation du coût réel de transport des solutions obtenues par les trois modèles par rapport au coût de transport déterministe obtenu par le modèle de référence.

$$t^{SPR} = \frac{f_{DR}^{Reel} \text{ de la solution du modèle de SPR} - f_{DR}^{Det}}{f_{DR}^{Det}} \times 100\%$$

$$t^{CCP} = \frac{f_{DR}^{Reel} \text{ de la solution du modèle de CCP} - f_{DR}^{Det}}{f_{DR}^{Det}} \times 100\%$$

$$t^{Nor} = \frac{f_{DR}^{Reel} \text{ de la solution du modèle de Nor} - f_{DR}^{Det}}{f_{DR}^{Det}} \times 100\%$$

Le tableau 2.6 représente les résultats de ce comparatif pour le problème SRC101. La troisième colonne (f_{DR}^{Det}) est le coût de transport déterministe correspondant à chaque scénario « test ». Pour chacune des solutions fournies par les différents modèles, nous avons le coût réel de transport (f_{DR}^{Reel}). Sur la dernière ligne, nous indiquons la valeur moyenne correspondante aux quatre scénarios pour les pourcentages t^{SPR} , t^{CCP} et t^{Nor} .

SRC101		VRPTW	VRPTWFD					
		Réel de référence	Solution du SPR ($f_{DR}^{Plan} = 1909.33$)		Solution du CCP ($f_{DR}^{Plan} = 1901.24$)		Solution du Nor ($f_{DR}^{Plan} = 1830.88$)	
	scénario « test »	f_{DR}^{Det}	f_{DR}^{Reel}	t^{SPR}	f_{DR}^{Reel}	t^{CCP}	f_{DR}^{Reel}	t^{Nor}
	test 1	1862.12	1909.33	2.54%	1901.24	2.10%	1840.02	-1.19%
	test 2	1854.52	1909.33	2.96%	1951.22	5.21%	1912.04	3.10%
	test 3	1896.37	1933.89	1.98%	1901.24	0.26%	1951.59	2.91%
	test 4	1866.32	1909.33	2.30%	2071.47	10.99%	1985.13	6.37%
	Moyenne			2.45%		4.64%		2.80%

Tableau 2. 6 Comparaison des solutions fournies par les différents modèles du SRC101 avec la version déterministe

Nous remarquons que la valeur moyenne des t^{SPR} est plus petite que les valeurs moyennes de t^{CCP} et t^{Nor} . Cela signifie que la solution du modèle SPR est plus proche de la solution déterministe de référence qui est obtenue en supposant que toutes les informations sont connues parfaitement à l'avance, que les solutions trouvées par les modèles CCP et Nor.

Nous reportons dans l'Annexe A.4 le détail de ce comparatif pour chaque problème du benchmark VRPTWFD. Nous présentons ici dans le tableau 2.7 la synthèse des valeurs moyennes t^{SPR} , t^{CCP} , t^{Nor} correspondantes aux quatre scénarios « test » pour chaque problème. Nous calculons également les valeurs moyennes correspondantes aux 6 types ainsi que pour l'ensemble des 24 problèmes de benchmark.

Type	Problème	VRPTWFD		
		Moy(t^{SPR})	Moy(t^{CCP})	Moy(t^{Nor})
SC1	SC101	2.06%	2.06%	2.06%
	SC105	2.09%	4.78%	6.41%
	SC108	1.22%	2.60%	10.51%
	SC109	4.93%	5.27%	5.11%
	Moyenne	2.58%	3.68%	6.02%
SC2	SC201	4.81%	4.81%	8.49%
	SC205	4.50%	4.50%	7.70%
	SC207	5.19%	5.19%	3.77%
	SC208	3.64%	3.73%	6.85%
	Moyenne	4.54%	4.56%	6.70%
SR1	SR101	1.93%	5.22%	10.21%
	SR103	-0.01%	0.28%	4.36%
	SR104	0.01%	4.57%	5.06%
	SR109	0.31%	3.74%	5.84%
	Moyenne	0.56%	3.45%	6.37%
SR2	SR205	0.78%	2.30%	1.58%
	SR206	0.41%	2.42%	0.42%
	SR207	1.14%	1.50%	2.42%
	SR208	2.46%	12.30%	9.07%
	Moyenne	1.20%	4.63%	3.37%
SRC1	SRC101	2.45%	4.64%	2.80%
	SRC104	4.26%	4.40%	14.45%
	SRC106	1.58%	2.07%	6.74%
	SRC107	1.96%	4.40%	7.46%
	Moyenne	2.56%	3.88%	7.86%
SRC2	SRC202	0.90%	1.43%	5.69%
	SRC204	2.46%	3.48%	3.70%
	SRC205	0.44%	4.74%	2.10%
	SRC206	-0.28%	5.16%	6.74%
	Moyenne	0.88%	3.70%	4.56%
Tous	Moyenne	2.05%	3.98%	5.79%

Tableau 2. 7 Comparaison avec la version déterministe

Parmi les valeurs moyennes t^{SPR} , t^{CCP} et t^{Nor} correspondantes aux scénarios « test », la valeur moyenne t^{Nor} est la plus grande pour la plupart des problèmes. Autrement dit, la formulation du VRPTW déterministe en normalisant les demandes floues des clients globalement n'arrive pas à résoudre le VRPTWFD avec une aussi bonne qualité que les deux autres modèles qui utilisent la théorie de possibilité.

Les valeurs moyennes t^{CCP} et t^{SPR} correspondantes aux scénarios « test » sont identiques pour les problèmes SC201, SC205 et SC207, et sont assez proches pour le SC208. Cela signifie que la qualité de la solution fournie par le modèle CCP est aussi bonne que celle de la solution fournie par le modèle SPR pour les problèmes du type SC2.

A travers ce tableau, nous voyons que pour chaque problème résolu (à part le SRC207), la valeur moyenne t^{SPR} correspondante aux scénarios « test » est la plus petite des trois. Ceci montre que la solution trouvée par le modèle SPR est de meilleure qualité et elle est la plus proche de la solution trouvée en connaissant toutes les informations avec certitude à l'avance.

2.5.5 L'influence du taux d'incertitude

Pour illustrer l'influence du taux d'incertitude (t_{In}) des données sur la qualité des solutions obtenues par les trois modèles, nous reprenons les problèmes SRC101. Nous considérons quatre extensions de ce problème : SRC101_0.05, SRC101_0.1, SRC101_0.2 et SRC101_0.3. Tous les clients du SRC101_0.05 ont un taux d'incertitude identique de 0.05. Respectivement, tous les clients du SRC101_0.1, du SRC101_0.2 et du SRC101_0.3 ont un taux d'incertitude identique de 0.1, 0.2 et 0.3. Plus le taux d'incertitude est grand, plus les demandes des clients sont incertaines.

Nous résolvons chacune de ces extensions par les 3 modèles SPR, CCP et Nor. Nous avons reporté les solutions obtenues en fonction des combinaisons de seuils α et β correspondantes dans le tableau 2.8. Les solutions trouvées par le modèle Nor pour les quatre extensions sont identiques car les normalisations des demandes des clients sont les mêmes quelque soit la valeur du taux d'incertitude.

Problème	Solution du modèle SPR					Solution du modèle CCP			Solution du modèle Nor	
	α	β	f_{DR}^{Plan}	f_{DR}^{Prev}	f_{DR}^{Supp}	α	β	f_{DR}^{Plan}	f_{DR}^{Plan}	
SRC101_0.05	1	0.2	1818.07	1818.07	0	1	0.2	1846.88	1830.88	
SRC101_0.1	1	0.2	1867.96	1871.81	3.85	1	0.1	1858.12	1830.88	
SRC101_0.2	1	0.1	1905.52	1924.96	19.44	1	0.1	1890.64	1830.88	
SRC101_0.3	1	0.1	1950.1	2000.73	50.63	1	0.2	1952.44	1830.88	

Tableau 2. 8 Solutions des trois modèles pour les extensions du problème SRC101 correspondantes à des taux d'incertitude différents

La lecture du tableau montre que plus l'incertitude des demandes est importante, plus le coût de transports et le risque de défaut de tournées sont grands.

Pour chaque taux d'incertitude, nous générons quatre scénarios « test » et déterminons les solutions déterministes correspondantes. Nous évaluons les solutions trouvées par les trois modèles en les comparant avec les solutions du modèle déterministe de référence. Nous calculons les valeurs moyennes t^{SPR} , t^{CCP} et t^{Nor} correspondant aux quatre scénarios « test » pour chacune des extensions du problème dans le tableau 2.9. Nous illustrons les variations de ces valeurs moyennes t^{SPR} , t^{CCP} et t^{Nor} par rapport au taux d'incertitude dans la figure 2.22.

		VRPTW	VRPTWFD					
		Réel de référence	Solution du SPR		Solution du CCP		Solution du Nor	
	scénario « test »	f_{DR}^{Det}	f_{DR}^{Reel}	t^{SPR}	f_{DR}^{Reel}	t^{CCP}	f_{DR}^{Reel}	t^{Nor}
SRC101_0.05	test 1	1814.19	1818.07	0.21%	1846.88	1.80%	1834.29	1.11%
	test 2	1796.36	1818.07	1.21%	1846.88	2.81%	1856.82	3.37%
	test 3	1823.56	1818.07	-0.30%	1846.88	1.28%	1894.21	3.87%
	test 4	1765.11	1818.07	3.00%	1846.88	4.63%	1879.78	6.50%
	Moyenne			1.03%		2.63%		3.71%
SRC101_0.1	test 1	1831.78	1867.96	1.98%	1858.12	1.44%	1852.04	1.11%
	test 2	1867.72	1910.03	2.27%	1962.86	5.09%	2303.24	23.32%
	test 3	1812.96	1867.96	3.03%	1858.12	2.49%	1899.13	4.75%
	test 4	1844.09	1867.96	1.29%	1914.72	3.83%	2009.6	8.98%
	Moyenne			2.14%		3.21%		9.54%
SRC101_0.2	test 1	1828.00	1909.76	4.47%	1974.22	8.00%	2046.98	11.98%
	test 2	1805.88	1905.52	5.52%	1975.62	9.40%	2064.54	14.32%
	test 3	1825.58	1909.76	4.61%	1961.97	7.47%	2169.66	18.85%
	test 4	1834.24	1905.52	3.89%	1932.56	5.36%	2032.52	10.81%
	Moyenne			4.62%		7.56%		13.99%
SRC101_0.3	test 1	1865.10	1950.1	4.56%	2001.88	7.33%	2187.94	17.31%
	test 2	1861.66	2003.5	7.62%	2028.74	8.97%	2146.34	15.29%
	test 3	1885.61	1983.44	5.19%	2044.42	8.42%	2083.55	10.50%
	test 4	1950.01	2099.28	7.66%	2086.07	6.98%	2098.98	7.64%
	Moyenne			6.26%		7.93%		12.68%

Tableau 2.9 Comparaison les solutions des trois modèles pour les extensions du problème SRC101 correspondantes aux taux d'incertitude avec la version déterministe

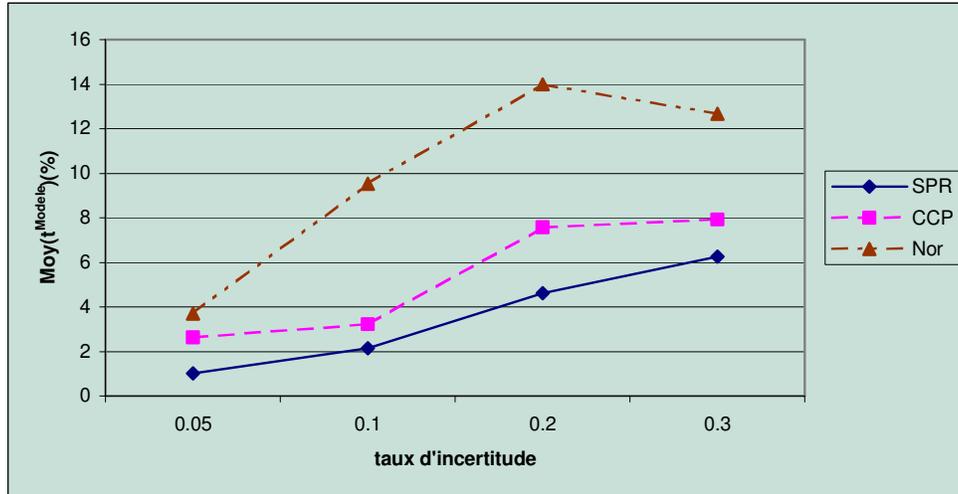


Figure 2. 22 L'influence du taux d'incertitude

A travers le tableau 2.9 et la figure 2.22, nous voyons que les valeurs moyennes t^{SPR} , t^{CCP} et t^{Nor} correspondantes aux quatre scénarios « test » augmentent avec l'augmentation du taux d'incertitude. Plus l'incertitude des demandes est grande, plus la solution trouvée par un modèle est éloignée de la solution trouvée par le modèle de référence c'est-à-dire en connaissant parfaitement toutes les données.

Pour chacune des extensions, la solution trouvée par le modèle SPR a une valeur moyenne t^{SPR} plus petite par rapport aux modèles CCP et Nor. Autrement dit, les meilleures solutions sont déterminées par le modèle SPR lors de la résolution du VRPTWFD.

2.5.6 Zoom sur le modèle SPR

Nous avons vu que le modèle SPR peut donner une meilleure solution par rapport aux modèles CCP et Nor. Nous étudions ci-dessous l'influence du nombre de scénarios d'apprentissage sur la qualité de la planification du modèle SPR.

2.5.6.1 Réglage du nombre de scénarios d'apprentissage

Le nombre de scénarios d'apprentissage dans le modèle du SPR est un nouveau paramètre à considérer. Nous étudions l'influence de ce paramètre sur la qualité de la solution trouvée.

Nous résolvons le problème SRC101 par le modèle SPR en utilisant un ensemble de p scénarios d'apprentissage. Nous faisons varier la valeur de p avec les valeurs 1, 10, 20, 30, 40. Nous avons reporté les meilleures solutions et les combinaisons de seuils α et β correspondantes aux différentes valeurs de p dans le tableau 2.10.

p	Solution du modèle SPR				
	α	β	f_{DR}^{Plan}	f_{DR}^{Prev}	f_{DR}^{Supp}
1	1	0.2	1894.77	1906.12	11.35
10	1	0.2	1924.61	1942.77	18.16
20	1	0.2	1909.33	1929.02	19.68
30	1	0.2	1903.82	1932.58	28.76
40	1	0.2	1900.4	1923.29	22.89

Tableau 2. 10 Solutions du modèle SPR correspondantes au nombre de scénarios d'apprentissage pour le problème SRC101

Pour chacune des solutions obtenues, nous calculons le coût réel de transport des tournées correspondant à chaque scénario « test » en utilisant la procédure « rejouer ». Nous comparons l'écart entre ce coût réel de transport et le coût de transport obtenu avec la solution déterministe de référence. Cette comparaison est présentée dans le tableau 2.11. Nous calculons les valeurs moyennes t^{SPR} correspondantes à chaque valeur p. Nous illustrons les valeurs moyennes t^{SPR} par rapport au nombre de scénarios d'apprentissage p dans la figure 2.23.

	VRPTW	Solution du modèle SPR pour VRPTWFD									
	Réel de référence	p=1		p=10		p=20		p=30		p=40	
scénario « test »	f_{DR}^{Det}	f_{DR}^{Reel}	t^{SPR}	f_{DR}^{Reel}	t^{SPR}	f_{DR}^{Reel}	t^{SPR}	f_{DR}^{Reel}	t^{SPR}	f_{DR}^{Reel}	t^{SPR}
test 1	1862.12	2014.38	8.18%	1956.25	5.05%	1909.33	2.54%	1935.61	4.10%	1919.5	3.08%
test 2	1896.37	1978.06	4.31%	1924.61	1.49%	1933.89	1.98%	1935.61	0.39%	1921.4	1.32%
test 3	1854.52	2021.98	9.03%	1924.61	3.78%	1909.33	2.96%	1935.61	3.61%	1904.47	2.69%
test 4	1866.32	1978.04	5.99%	1924.61	3.12%	1909.33	2.30%	1935.61	3.71%	1900.4	1.83%
Moyenne	1869.83		6.88%		3.36%		2.45%		2.96%		2.23%

Tableau 2. 11 Comparaison des solutions du modèle SPR correspondantes au nombre de scénarios d'apprentissage pour le problème SRC101 avec la version déterministe

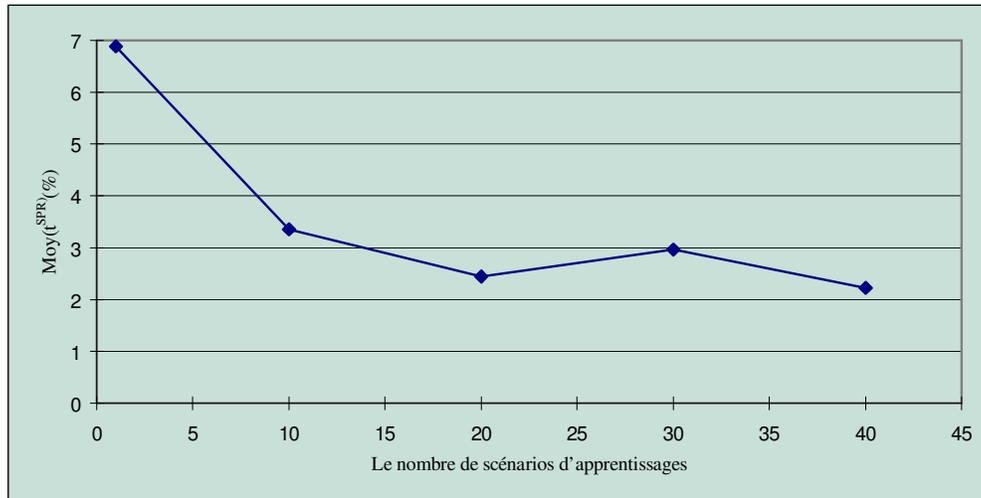


Figure 2. 23 L'influence du nombre de scénarios d'apprentissage

Nous voyons que la valeur moyenne t^{SPR} est grande en fixant le nombre de scénarios d'apprentissage à une petite valeur (i.e. $p=1$). La variation des valeurs moyennes t^{SPR} n'est pas significative lors que le nombre de scénarios d'apprentissage est supérieur à 20. Par contre, plus le nombre de scénarios d'apprentissage est grand, plus le temps de l'exécution de l'algorithme génétique de la version « FAG_SPR » sera grand. De ce fait, nous avons utilisé la valeur 20 comme étant le nombre de scénarios d'apprentissage au modèle SPR.

Nous allons évaluer la qualité de la planification de la solution trouvée par le modèle SPR.

2.5.6.2 Evaluation de la solution du modèle SPR

La solution du modèle SPR nous donne le coût planifié de transport et le coût prévisionnel de transport des tournées planifiées. Nous avons calculé le coût réel de transport correspondant à chaque scénario « test » réellement effectué par les véhicules.

La différence entre le coût réel de transport correspondant au scénario « test » et le coût planifié de transport est le coût de transport dû aux « défauts des tournées » lors de la réalisation, c'est le coût supplémentaire réel de transport (noté f_{DR}^{SR}). Le t^{SR} est défini comme étant le pourcentage du coût supplémentaire réel de transport sur le coût planifié de transport :

$$t^{SR} = \frac{f_{DR}^{Reel} - f_{DR}^{Plan}}{f_{DR}^{Reel}} \times 100\% = \frac{f_{DR}^{SR}}{f_{DR}^{Reel}} \times 100\%$$

Cette mesure t^{SR} donne une indication sur la fiabilité de tournées planifiées. Plus cette valeur est faible, moins il y a de « défauts des tournées » et plus fiable sont les tournées planifiées.

Nous calculons également le pourcentage de la déviation entre le coût réel de transport correspondant au scénario « test » et le coût prévisionnel de transport par rapport au coût réel de transport (noté t^{RP}).

$$t^{RP} = \frac{f_{DR}^{Reel} - f_{DR}^{Prev}}{f_{DR}^{Reel}} \times 100\%$$

Cette mesure t^{RP} évalue l'aptitude de la solution à estimer les situations réelles. Plus cette valeur sera proche de 0, plus la solution sera exacte c'est-à-dire proche de la réalité.

Au titre d'un exemple, nous évaluons la solution trouvée par le modèle SPR avec la combinaison de $\alpha=1$ et $\beta=0.2$ pour le problème SRC101 dans le tableau 2.12. Le coût prévisionnel de transport des tournées planifiées est égal à 1929.02 compris le coût planifié de transport (1909.33). Les coûts réels de transport correspondant à 4 scénarios « test » sont présentés dans la colonne f_{DR}^{Reel} . Nous calculons les valeurs t^{SR} et t^{RP} , ainsi les valeurs moyennes respectives t^{SR} et t^{RP} correspondantes aux quatre scénarios « test »

Solution du modèle SPR pour VRPTWFD			Evaluation de solutions					
f_{DR}^{Plan}	f_{DR}^{Prev}	f_{DR}^{Supp}	scénario « test »	f_{DR}^{Reel}	f_{DR}^{SR}	t^{SR}	D^{RP}	t^{RP}
1909.33	1929.02	19.68	test 1	1909.33	0	0.00%	-19.68	-1.03%
			test 2	1909.33	0	0.00%	-19.68	-1.03%
			test 3	1933.89	24.56	1.27%	4.88	0.25%
			test 4	1909.33	0	0.00%	-19.68	-1.03%
			Moyenne			0.32%		-0.71%

Tableau 2. 12 L'évaluation de la solution du modèle SPR pour le problème SRC101

Nous voyons que la mesure t^{SR} est égale à 0 pour trois scénarios « test ». Autrement dit, le coût de transport réel correspondant au scénario « test » est égal au coût planifié de transport. Cela signifie qu'il n'y a pas eu de « défauts des tournées » dans les réalisations des tournées correspondantes à ces scénarios « test ». La valeur moyenne des t^{SR} correspondantes aux quatre scénarios « test » est égale à 0.32%. Les tournées planifiées sont donc fiables. Les mesures t^{RP} sont négatives pour trois scénarios « test ». La valeur moyenne des t^{RP} correspondantes aux quatre scénarios « test » est égale à -0.71%. Ceci montre que la solution fournie par le modèle SPR a bien anticipé les situations réelles.

Nous reportons le détail de cette analyse pour tous les problèmes du benchmark dans l'Annexe A.5.

Nous faisons ici dans le tableau 2.13 une synthèse des valeurs moyennes t^{SR} , t^{RP} correspondantes aux quatre scénarios « test » pour les solutions fournies par le modèle SPR. Nous calculons les valeurs moyennes correspondantes aux 6 types de problèmes ainsi que pour l'ensemble des 24 problèmes.

Type	Problème	Moy(t^{SR})	Moy(t^{RP})	Type	Problème	Moy(t^{SR})	Moy(t^{RP})
SC1	SC101	13.75%	-2.59%	SC2	SC201	3.63%	0.62%
	SC105	5.31%	-0.48%		SC205	1.32%	0.76%
	SC108	2.16%	-0.57%		SC207	0.00%	0.00%
	SC109	3.69%	2.09%		SC208	0.00%	0.00%
	Moyenne	6.23%	-0.39%		Moyenne	1.24%	0.35%
SR1	SR101	0.22%	-0.25%	SR2	SR205	0.73%	0.33%
	SR103	0.45%	0.29%		SR206	0.48%	0.14%
	SR104	0.96%	0.65%		SR207	1.29%	1.23%
	SR109	0.37%	-0.22%		SR208	0.84%	0.70%
	Moyenne	0.50%	0.12%		Moyenne	0.83%	0.60%
SRC1	SRC101	0.32%	-0.71%	SRC2	SRC202	2.20%	2.05%
	SRC104	0.89%	0.11%		SRC204	0.18%	-0.32%
	SRC106	1.16%	0.33%		SRC205	0.24%	-0.03%
	SRC107	1.93%	1.29%		SRC206	0.31%	-0.01%
	Moyenne	1.07%	0.25%		Moyenne	0.73%	0.42%
Tous	Moyenne	1.77%	0.23%				

Tableau 2. 13 Synthèse de l'évaluation des solutions trouvées par le modèle SPR

Mis à part les problèmes de type SC1, les valeurs moyennes t^{SR} correspondantes aux quatre scénarios « test » sont inférieures à 5% et la plupart de celles-ci sont à moins 1%. Nous pouvons vérifier que les solutions trouvées par le modèle SPR nécessitent peu de coût supplémentaire de transport dû aux « défauts des tournées » lors de la réalisation effective sur les scénarios « test ».

A partir de la synthèse des valeurs moyennes t^{RP} correspondantes aux quatre scénarios « test », nous remarquons que les valeurs absolues ne sont pas supérieures à 3% et que la plupart de celles-ci sont à moins 1%. Autrement dit, la déviation entre le coût réel de transport correspondant à chaque scénario « test » et le coût prévisionnel de transport est faible. L'aptitude de la solution à estimer les situations réelles est assez bonne.

Nous pouvons conclure que les solutions obtenues par le modèle SPR sont fiables et elles ont bien prévu les situations réelles.

2.6 Conclusion

Dans ce chapitre nous avons résolu le VRPTWFD dans lequel les clients spécifient leur demande à l'aide d'un nombre flou pour une plus grande souplesse d'utilisation.

La théorie de possibilité nous a permis de réécrire la contrainte de capacité en fixant les valeurs des seuils. En utilisant cette contrainte de capacité floue, un modèle de programmation sous contraintes probabilistes (CCP) et un modèle à deux-étapes de type programmation stochastique avec recours (SPR) ont été proposés pour la résolution du VRPTWFD. Nous pouvons planifier avec ces deux modèles un ensemble de tournées a priori pour lesquelles les « défauts de tournées » seront inférieurs à un certain seuil. De plus, le modèle SPR nous a permis d'estimer le calcul du coût supplémentaire de transport induit par les « défauts des tournées » selon la politique de recours choisie à l'aide d'une méthode de simulation de tournées basée sur des scénarios d'apprentissage.

Une plateforme basée sur algorithme génétique a été utilisée pour la recherche de bonnes solutions. Deux versions de l'AG ont été proposées pour intégrer ces deux modèles et évaluer leur performance respective.

Nous avons construit un benchmark pour le VRPTWFD en modifiant le jeu de problèmes fourni par Solomon pour le VRPTW. Nous avons comparé les deux modèles précédents (CCP et SPR) avec un modèle déterministe (modèle Nor).

Nous avons évalué les qualités des solutions fournies par les trois modèles dans un environnement réel en simulant les situations réelles à l'aide de scénarios « test ». Les solutions déterministes correspondantes aux scénarios « test » ont été trouvées en connaissant toute l'information à l'avance et avec certitude. Nous avons vu que les solutions obtenues par le modèle SPR sont robustes par rapport à la réalisation effective des variables stochastiques et que leurs qualités sont proches de celle des solutions déterministes.

Chapitre 3

Résolution « en ligne » du VRPTWFD dynamique

Sommaire

3.1 Le VRP dynamique	95
3.1.1 Description	96
3.1.2 Le degré de dynamisme.....	97
3.1.3 Classification du DVRP	97
3.1.4 Les méthodologies de résolution du DVRP	99
3.2 Plateforme de Simulation pour la résolution du VRPTWFD dynamique	103
3.2.1 Principe de résolution.....	103
3.2.2 Architecture de la plateforme	104
3.2.3 La gestion du temps.....	111
3.2.4 Le fonctionnement de la plateforme.....	112
3.3 Etudes expérimentales.....	113
3.3.1 Le benchmark utilisé	113
3.3.2 Simulation	114
3.3.3 Comparaison avec la version « Déterministe et Statique »	119
3.4 Conclusion.....	122

Jusqu'ici nous avons traité le VRPTWFD dans un contexte statique où nous supposons que tous les clients sont connus à l'avance mais avec des quantités demandées qui sont représentées par des nombres flous. Les quantités ne sont connues avec certitude qu'au moment où les véhicules commencent à les servir.

Dans ce chapitre, on se place dans un contexte dynamique pour traiter le VRPTWFD . Nous étendons les travaux précédents dans ce cadre pour les adapter à notre plateforme de simulation JAVA-AG. Dans cette plateforme, un simulateur permet de simuler des événements dynamiques comme l'arrivée d'une nouvelle requête du client. La connaissance d'une quantité demandée avec certitude va déclencher un événement qui va modifier les tournées existantes. Avant de présenter notre approche de résolution du VRPTWFD dans un contexte dynamique, nous présentons dans ce qui suit un état de l'art sur les VRP dynamiques.

3.1 Le VRP dynamique

Dans beaucoup de situations pratiques, les données du VRP ne sont pas complètement connues au moment de la résolution. Ces dernières années, nous remarquons que le problème de la gestion de tournées de véhicules dynamique (DVRP) est devenu un domaine de recherche important. Le DVRP fait partie de la famille des problèmes de transports dynamiques (DTP)

Psaraftis [Psaraftis 1995] définit un problème de gestion de tournées de véhicules comme étant statique si le résultat d'une certaine formulation est un ensemble de tournées pré-planifiées qui ne sont pas re-optimisées et qui sont calculées à partir de données n'évoluant pas au cours du temps. (*if the output of a certain formulation is a set of preplanned routes that are not re-optimized and are computed from inputs that do not evolve in real-time.*)

Il définit un problème comme étant dynamique si le résultat n'est pas un ensemble de tournées, mais plutôt une politique qui décrit comment les tournées devront être modifiées en fonction des données qui évoluent en au cours du temps. (*the output is not a set of routes, but rather a policy that prescribes how the routes should evolve as a function of those inputs that evolve in real-time.*)

Bianchi [Bianchi 2000] définit deux caractéristiques qui rendent un problème d'optimisation comme étant dynamique :

- Des données du problème dépendent du temps. Elles sont connues ou mises à jour au cours du temps.
- Des solutions doivent être trouvées au fur et à mesure avec les informations entrantes. Ceci signifie qu'aucune solution a priori ne peut être trouvée. La seule chose qu'on peut faire a priori (répétition) est de définir une « stratégie », ou une « politique », qui indique quelles actions devraient être réalisées en fonction de l'état du système.

Les informations qui dépendent du temps peuvent concerner par exemple les clients à servir dans la journée. C'est le cas de notre problème VRPTWFD dynamique, où le nombre de clients à servir évolue au cours du temps : de nouvelles demandes de service apparaissent

au fil du temps alors que les véhicules réalisent déjà les tournées préalablement établies.

3.1.1 Description

On définit un horizon de service comme une période de temps $[0 .. T]$ durant laquelle un ensemble de clients va être servi. Dans le VRP Dynamique, seul un sous ensemble de clients est connu avant le début de l'horizon de service. Ces clients sont appelés clients statiques.

Un client est dit dynamique, lorsque sa requête n'est connue (ou n'apparaît) que pendant l'horizon de service. A chaque client dynamique i est associée une valeur tr_i avec $0 < tr_i < T$, qui représente l'instant d'apparition de sa requête.

Dans le VRPTW, la fenêtre de temps du dépôt fournit l'horizon de service ($T = l_0$) et chaque client i est caractérisé par une fenêtre de temps (e_i, l_i) . La valeur de tr_i associée n'est valide que si sa requête dynamique apparaît dans la fenêtre de temps du dépôt ($0 < tr_i < l_0$) et avant sa propre fenêtre de temps ($tr_i < e_i$). Ceci signifie qu'il peut y avoir en cours de journée de nouveaux clients qui demandent un service après que la planification ait été effectuée et alors que les véhicules sont déjà en route pour servir les clients planifiés. Ces nouveaux clients doivent alors pouvoir être insérés dans la solution courante.

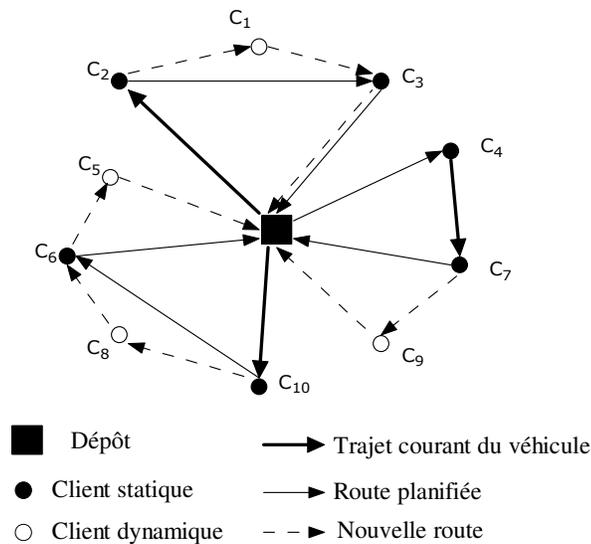


Figure 3. 1 Le problème DVRP

Dans la figure 3.1, nous avons un exemple simple de problème dynamique où trois véhicules doivent servir un ensemble de clients. Les clients statiques sont représentés par des nœuds noirs, alors que les clients dynamiques sont représentés par les nœuds blancs. Les arcs en trait fin représentent les routes planifiées pour les clients statiques. Les trois arcs en trait plein épais indiquent les trajets en cours des véhicules lorsque les requêtes dynamiques sont

reçues. Les traits pointillés représentent les nouvelles routes pour les clients non encore servis en tenant compte de l'insertion des nouveaux clients.

3.1.2 Le degré de dynamisme

Contrairement à un VRP statique nous supposons que la performance dépend non seulement du nombre de clients et de la distribution spatiale de ces derniers, mais également du nombre d'évènements dynamiques et des instants où ces évènements ont lieu réellement. Par conséquent, une mesure pour décrire le dynamisme du système est d'une grande utilité pour examiner la performance d'un algorithme spécifique dans des conditions variables.

Pour cela, le degré de dynamisme (dod : Degree Of Dynamism) a été défini comme ci-dessous [Lund et al. 1996] :

$$dod = \frac{n_{Dyn}}{n_{Tot}}$$

où $n_{Tot} = n_{Sta} + n_{Dyn}$, n_{Sta} est le nombre de clients statiques, n_{Dyn} est le nombre de clients dynamiques .

3.1.3 Classification du DVRP

Nous présentons la classification du DVRP donnée par [Larsen 2001]. Cette classification est faite en employant trois échelons de dynamisme. On distingue les problèmes faiblement, modérément, et fortement dynamiques (voir la figure 3.2). Cette figure montre que pour chaque catégorie de problèmes, l'objectif principal à atteindre peut varier de la minimisation du coût de routage qui correspond à la distance totale parcourue, à la minimisation du temps de réponse qui représente l'intervalle de temps séparant l'apparition de la requête et la prise en charge de celle-ci.

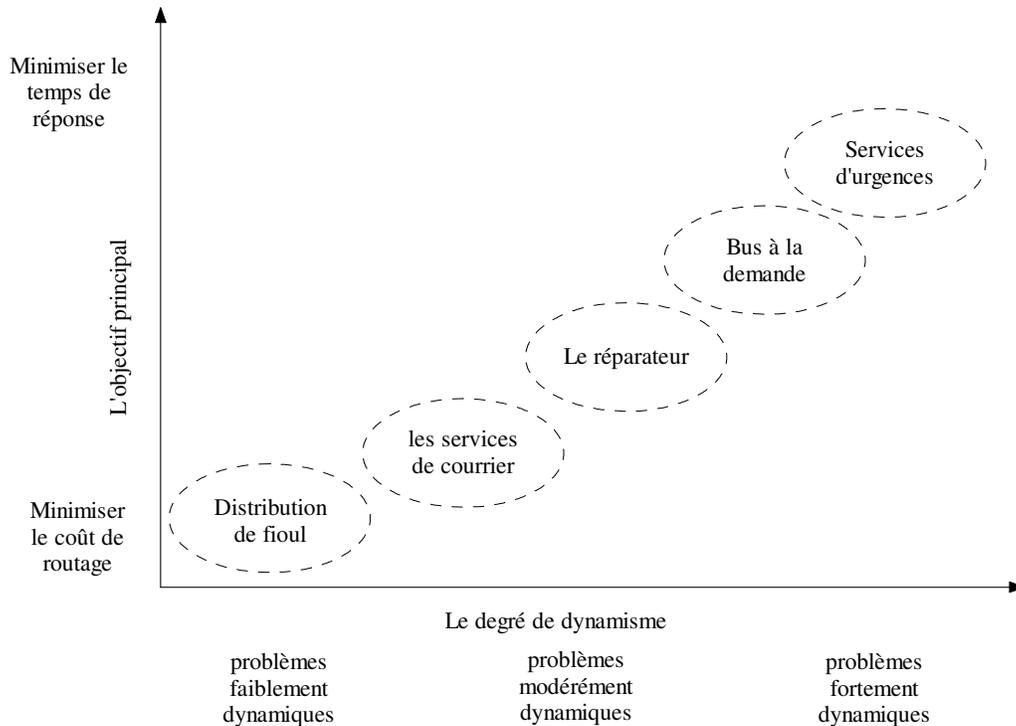


Figure 3. 2 Classification du DVRP

1. Problèmes faiblement dynamiques

Les problèmes de VRP avec un degré faible de dynamisme incluent la distribution de mazout ou de gaz liquide chez les particuliers. Dans cet exemple, la plupart des clients (plus de 80 %) sont connus à l'heure de la planification des routes. D'autres exemples incluent des services d'installations chez les particuliers, tels que la télévision par câble et le téléphone. Le transport de personnes âgées ou handicapées peut être également classé comme un problème faiblement dynamique. L'objectif à atteindre est de minimiser un coût de routage tel que la distance totale parcourue par les véhicules.

L'approche traditionnelle pour résoudre de tels problèmes est basée sur l'adaptation des procédures statiques. Un problème statique d'élaboration de tournées de véhicules est résolu chaque fois qu'un événement se produit et change l'environnement du problème.

2. Problèmes modérément dynamiques

En comparant ces problèmes avec ceux ayant un degré de dynamisme faible, le nombre de requêtes immédiates forment une partie substantielle du nombre total de clients. Des exemples pratiques de ces problèmes incluent les services de courrier ou les services de dépannage chez les particuliers.

3. Problèmes fortement dynamiques

Les services de secours, tels que la police, les pompiers et les ambulances montrent un comportement fortement dynamique [Brotcorne et al. 2003]. Toute requête est forcément non prévisible et donc dynamique. Un autre exemple est celui du service des taxis ou des bus à la

demande pour lesquels seul un nombre négligeable de clients a commandé une course à l'avance. En général, le but principal à satisfaire est de minimiser le temps de réponse pour satisfaire la demande du client.

3.1.4 Les méthodologies de résolution du DVRP

Précédemment nous avons dit que la solution du DVRP doit être trouvée au cours du temps avec les informations nouvelles entrantes. Une stratégie indique généralement comment les actions sont prises en fonction des informations qui évoluent en temps réel. Une stratégie peut également préciser la manière dont on décompose un problème dynamique en plusieurs sous-problèmes statiques. Par exemple, un nouveau sous-problème peut être considéré chaque fois qu'un nouveau client apparaît ou alors après une certaine période de temps pour avoir un ensemble de nouveaux clients à traiter. Plusieurs algorithmes d'optimisation peuvent être appliqués aux sous-problèmes statiques [Bianchi 2000].

Dans la littérature, on distingue deux types de travaux : ceux dont le but est plutôt l'étude de stratégies adaptées, et ceux qui utilisent des méthodes d'optimisation pour résoudre les problèmes de façon optimale ou quasi optimale. Nous présentons dans ce qui suit les stratégies classiques et les algorithmes d'optimisation les plus utilisés.

3.1.4.1 Stratégies classiques

[Bianchi 2000] a donné un survol des stratégies classiques pour résoudre le problème dynamique de voyage de réparateur (DTRP):

1. FCFS (First Come First Serve) : Les clients sont servis dans l'ordre où ils ont été reçus par le planificateur.
2. SQM (Stochastic Queue Median) : C'est une variante de la précédente. Les véhicules sont initialement placés à la médiane de la région et y retournent une fois les services clients terminés par la stratégie FCFS.
3. TSP (Traveling Salesman Problem strategy) :
 - (a). Au fur et à mesure que les clients arrivent, on les place dans une file d'attente.
 - (b). On optimise à l'aide d'un algorithme inspiré du TSP.
4. mod TSP (modified TSP) :
 - (a). La région de service est découpée en k sous-régions.
 - (b). Nous appliquons une résolution du type TSP sur chaque sous-région.
 - (c). Une file d'attente permet d'indiquer l'ordre dans lequel les sous-régions vont être servies
5. NN (Nearest Neighbor) :
 - (a). Les m véhicules servent les m premiers clients les plus proches.
 - (b). Après avoir servi un client, le véhicule se dirige vers le client le plus proche pas encore servi.
6. PART (PARTitioning policy) : La région de service est divisée en sous-régions dans

lesquelles les clients sont servis en utilisant la stratégie FCFS.

7. GEN (GENeration strategy) : Cette stratégie a été appliquée au problème DTRP avec un seul véhicule.
 - (a). Initialement, on met le véhicule dans un point d'attente de la région de service.
 - (b). Dès qu'un client arrive, le véhicule se déplace directement pour servir ce client (le premier client forme la première génération).
 - (c). Si aucun client supplémentaire n'apparaît quand tous les clients de la génération précédente ont été servis, le véhicule retourne au point d'attente (a). Dans le cas contraire, on applique une optimisation type TSP sur les clients apparus. Une nouvelle génération de clients à servir est ainsi générée.

[Yang et al. 2004] ont proposé plusieurs stratégies pour la résolution d'une variante de VRPTW dynamique où les véhicules considérés sont des camions affrétés pour une entreprise. Dans cette variante, un camion ne sert qu'un client à la fois. Celui-ci doit transporter directement les produits depuis un lieu de ramassage vers un lieu de livraison. De plus, le planificateur a le droit d'accepter ou de refuser la requête.

Les travaux de Gendreau ([Gendreau et al. 1998b] et [Gendreau et al. 1999]) résolvent le VRPTW dynamique (DVRPTW) et le DVRPTW avec ramassage et livraison (DVRPTWP&D) respectivement au moyen d'une stratégie simple : le problème dynamique est transformé en une série de problèmes statiques. Chaque problème statique est défini à chaque nouvel événement comme :

- L'arrivée d'un nouveau client.
- La fin de service.

L'algorithme d'optimisation résout le nouveau problème statique entre 2 « événements » consécutifs et s'arrête dès qu'un nouvel événement se produit.

Dans le cas de l'arrivée d'un nouveau client, le problème statique à résoudre est une mise à jour en ajoutant le nouveau client à l'ensemble de clients à servir. Puis l'optimisation est remise en marche pour rechercher une nouvelle solution.

Dans le cas de la fin de service chez un client, le prochain client à visiter est choisi d'après la meilleure solution courante obtenue. Une fois ce client affecté, l'algorithme est redémarré sur l'ensemble de clients restants.

Pour éviter de relancer systématiquement la phase de « re-optimisation » à chaque nouvel événement, [Larsen 2001] a proposé dans le cadre du problème VRPTW dynamique, deux stratégies de type « batch » pour différer le démarrage de cette phase

- Une stratégie de regroupement basée sur la taille, dans laquelle les clients dynamiques sont rassemblés par un ensemble de taille fixe n (par exemple $n=3$). L'insertion heuristique s'exécute dès qu'un ensemble complet a été formé.
- Une stratégie de regroupement basée sur le temps, où la re-optimisation redémarre à des intervalles de temps fixes (par exemple toutes les 10 minutes).

Il donne une conclusion basée sur une analyse empirique : globalement les stratégies n'ont pas influencé la performance de l'heuristique. Dans quelques cas, la stratégie basée sur la taille a amélioré la distance au prix d'une augmentation considérable de la somme des

retards clients.

Dans les travaux de [Kilby et al. 1998a] et [Montamenni et al. 2002], la journée de service est divisée en tranches de temps. Pour chacune d'elles, les requêtes reçues précédemment et les anciennes requêtes non encore servies sont regroupées. Chacun de ces groupes sert d'entrée à un problème statique qu'ils tentent de résoudre pendant une tranche de temps. Les nouvelles requêtes reçues pendant cette tranche de temps seront traitées dans la tranche suivante. C'est une stratégie de regroupement basée sur le temps.

3.1.4.2 Les approches basées sur l'optimisation

Nous allons donner une courte description des méthodes d'optimisations les plus utilisés pour le VRP dynamique. On cherche dans ce cas à optimiser une fonction de coût.

1. Heuristique d'insertion

Une heuristique d'insertion simple décrite dans [Roy et al. 1984] est largement utilisée pour la résolution de problème VRP dynamique. Elle cherche à insérer le nouveau client dans la meilleure position des tournées courantes.

Dans la figure 3.3, le client (I) a demandé un service. Nous devons rechercher la meilleure position d'insertion possible dans la matrice des tournées. Cette matrice contient tous les clients à servir, et ne contient pas les clients déjà servis. Pour chaque tournée (par exemple v1), nous vérifions toutes les positions d'insertion possibles pour ce client (dans notre exemple, deux positions). Puis nous pouvons simplement décider dans quelle position le client doit être inséré. Une telle heuristique est simple, elle peut être très rapide et donc être utilisable pour des problèmes de nature dynamique.

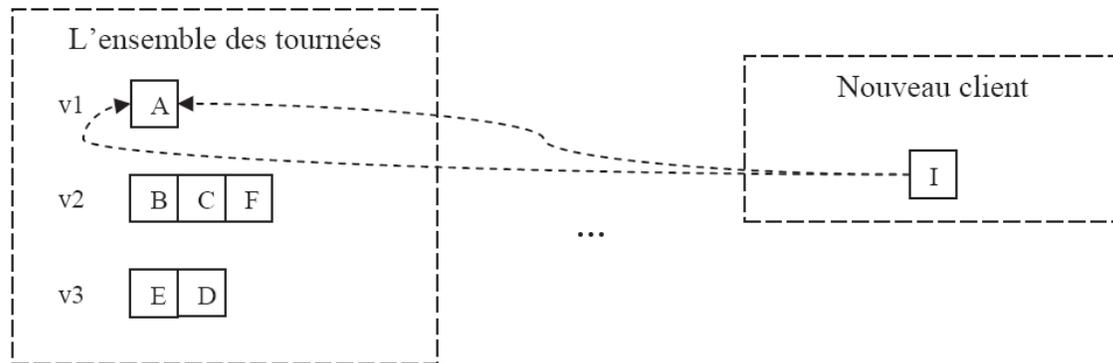


Figure 3.3 heuristique d'insertion

[Larsen 2001] a utilisé cette heuristique d'insertion pour résoudre le problème VRPTW dynamique. Il a réalisé un simulateur qui se compose d'un module de contrôle et d'un module de routage. Le module de contrôle traite les événements (l'arrivée d'un nouveau client et la fin de service) qui peuvent intervenir au cours du temps et changent alors l'état du système. Le module de routage prend la décision de réarranger les tournées existantes selon le nouvel état. Il démarre la méthode heuristique d'insertion pour répondre à l'évènement d'une nouvelle requête.

Une autre étude qui utilise cette méthode heuristique d'insertion est réalisée par Kilby [Kilby et al. 1998a]. Dans ses travaux, le temps de simulation (le temps de la journée de service) est divisé en tranches (50 au total). Durant chaque tranche, le planificateur utilise la méthode heuristique d'insertion pour trouver les meilleures positions pour les nouvelles requêtes reçues au cours de la tranche précédente. Dans le même temps, ce planificateur essaie d'améliorer la solution obtenue en appliquant un opérateur de recherche locale 2-Opt proposé par [Lin 1965].

2. Les métaheuristiques

Récemment, sont apparus quelques articles sur l'utilisation des métaheuristiques pour la résolution du problème VRP dynamique.

Parmi eux, nous avons les travaux de Montamenni [Montamenni et al. 2002], qui implémentent une métaheuristique basée sur les colonies de fourmis pour résoudre le DVRP. La journée de service est divisée en tranches de temps. Pour chacune d'elles, un problème VRP statique, prenant en compte toutes les requêtes déjà reçues, est créé puis résolu.

Dans les articles de Gendreau [Gendreau et al. 1998b] et [Gendreau et al. 1999], on trouve la résolution de deux problèmes dynamiques (DPDPTW) et (DVRPTW). Dans ces deux articles, l'algorithme d'optimisation principal est une recherche tabou adaptative parallèle. En fait, cet algorithme conçu pour une utilisation en parallèle, a été testé sur une machine multi-processeurs où chaque processeur lance un algorithme de recherche tabou. Une mémoire adaptative et des opérateurs de croisement permettent la coopération entre les différents tabous.

Par ailleurs, les auteurs ont comparé cette méthode avec des méthodes conçues spécialement pour résoudre ces problèmes dynamiques. Ces méthodes sont :

1. Méthode par insertion classique.
2. Méthode par insertion avec recherche locale: cette méthode est une amélioration de la méthode précédente en intégrant une procédure d'amélioration de la solution courante après chaque insertion d'un nouveau client.
3. Méthode par construction : cette méthode reconstruit la solution, en utilisant la méthode d'insertion, après chaque événement correspondant à un nouveau client.
4. Méthode par construction avec recherche locale (construction+) : cette méthode est identique à la précédente ; elle applique en plus une procédure d'amélioration locale après chaque étape de construction.
5. Méthode par descente adaptative : Celle-ci est une version modifiée de la méthode tabou utilisée. Dans ce cas-ci, les processus tabous sont arrêtés dès qu'un optimum local est trouvé. Les résultats ont montré que la méthode parallèle avec la recherche tabou est bien meilleure que les autres.

Par ailleurs, nous pouvons citer les travaux d'H. Housroum [Housroum 2005], qui utilisent un algorithme génétique pour résoudre le problème VRPTW dynamique. Ses travaux ont montré que l'algorithme génétique est aussi performant et parfois meilleur que les méthodes ci-dessus.

3.2 Plateforme de Simulation pour la résolution du VRPTWFD dynamique

Dans un VRPTWFD dynamique, les clients à servir dans la journée ne sont pas complètement connus au moment où débute la planification. Certains clients vont arriver au cours du temps. De plus comme dans le modèle statique, les demandes des clients ne sont connues avec certitude qu'au moment où les véhicules commencent les services.

3.2.1 Principe de résolution

Nous résolvons le problème dynamique « en ligne », la solution du problème est construite au fur et à mesure de l'arrivée des clients. Le conducteur d'un véhicule n'a que la connaissance du prochain client à servir. Pendant les temps de service ou les temps de trajet, l'algorithme d'optimisation continue d'améliorer en « arrière plan » la solution courante.

Dans notre approche, dès qu'une information est mise à jour, un nouveau VRPTWFD statique est créé puis résolu. Autrement dit, nous traitons le VRPTWFD dynamique par une série de VRPTWFD statiques, qui dépend directement des événements apparaissant au cours du temps. En fait pour éviter de démarrer à chaque événement un nouveau problème statique, on vient perturber légèrement le problème en cours pour intégrer celui-ci. L'algorithme génétique utilisé au sein de notre plateforme de simulation est bien adapté en cela.

Trois événements vont perturber la planification en cours :

- Arrivée d'un nouveau client. Dans notre problème dynamique, un sous-ensemble de clients dynamiques apparaît au cours du temps. Sur l'arrivée d'un nouveau client, le problème statique associé est mis à jour en ajoutant le nouveau client à l'ensemble des clients à servir.
- Connaissance de la demande avec certitude. Les demandes des clients sont représentées par des nombres flous au moment de la planification. Quand un véhicule commence à servir un client, il obtient la valeur exacte de sa demande. Si sa capacité résiduelle est supérieure à cette valeur, le client est servi. sinon, c'est un « défaut de tournée ». Dans ce dernier cas, le véhicule rentre au dépôt pour décharger ou se réapprovisionner puis il retourne chez ce client pour le servir. On peut constater que le problème devient de moins en moins « stochastique » à chaque fois qu'un client est servi, l'incertitude sur la quantité résiduelle du véhicule diminue avec la connaissance exacte de la demande du client.
- Fin de service. Lorsqu'un véhicule finit le service d'un client, il faut lui fournir le prochain client à visiter. On l'obtient à partir de la meilleure solution obtenue. Puis ce client est supprimé de l'ensemble des clients à servir. Un nouveau problème statique est alors à résoudre.

Chaque fois qu'un des événements se produit, un nouveau problème VRPTWFD statique est créé par une modification de l'ancien problème par exemple en ajoutant ou en supprimant des clients de l'ensemble courant des clients à servir. Les méthodes de résolution itératives comme les AG vont pouvoir naturellement absorber cette modification et s'adapter à celle-ci sans devoir démarrer un nouveau problème.

Pour intégrer notre stratégie de résolution du VRPTWFD dynamique, nous avons modifié une plateforme de simulation ayant la capacité de simuler la journée de service [Goncalves et al. 2007]. Elle a été initiée dans les travaux de H. Housroum [Housroum 2005].

3.2.2 Architecture de la plateforme

La plateforme s'appuie sur trois composants principaux qui coopèrent (voir la figure 3.4). Un composant simulateur du monde réel qui simule le monde réel en générant les événements significatifs d'une journée. Un composant gestion d'évènements (GE) qui attend l'arrivée d'un événement et déclenche les actions appropriées à cet événement. Un composant méthode de résolution qui cherche à résoudre les VRPTWFD statiques correspondant aux événements qui interviennent au cours du temps. Dans notre cas le composant méthode de résolution est ici un algorithme génétique (AG) mais d'autres méthodes de résolution pourraient y être intégrées. Les tournées sont planifiées et réalisées par simulation au fur et à mesure de l'avancement du temps.

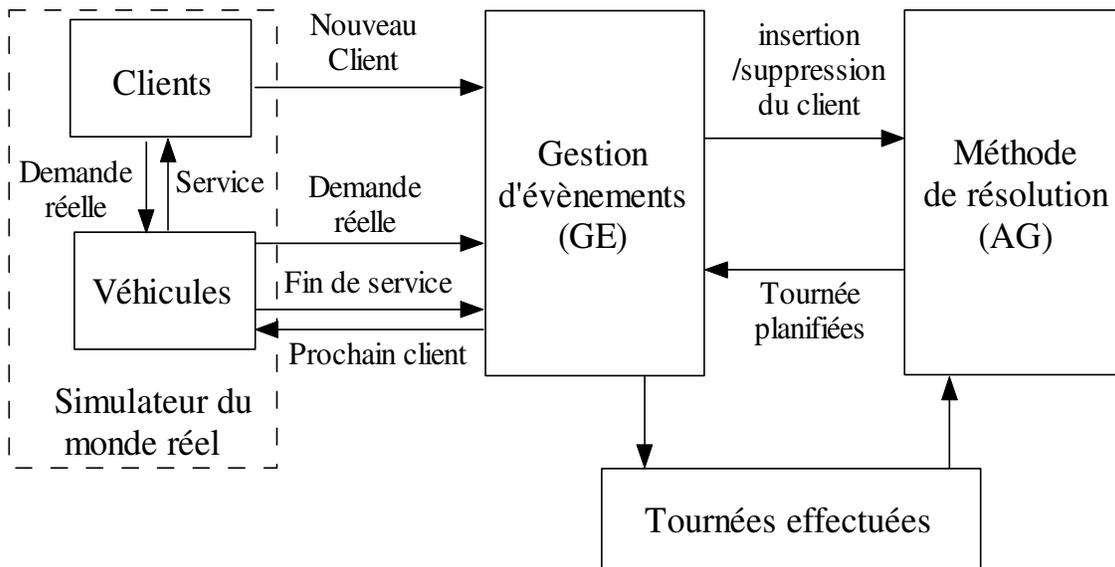


Figure 3.4 Plateforme de simulation

3.2.2.1. Le composant simulateur du monde réel

Dans la version opérationnelle de notre plateforme, trois types d'évènements sont actuellement pris en compte. L'arrivée d'un nouveau client, la connaissance de la demande avec certitude et la fin de service chez un client.

Pour générer ces événements, d'une part ce composant s'appuie sur un fichier benchmark qui décrit la structure du problème que l'on veut traiter : le nombre de véhicules, le nombre de clients, leur position géographique et l'instant d'arrivée de la requête du client. D'autre part, ce composant va simuler au fur et à mesure l'état d'avancement des tournées en cours de réalisation.

L'événement « arrivée d'un nouveau client » vient en fait du fichier Benchmark (fichier de données) qui décrit le problème dynamique à résoudre.

Les temps où se produisent l'événement « la connaissance de la demande avec certitude » et l'événement « fin de service client » sont calculés par le simulateur sur la base de l'instant de départ du véhicule, de la distance à parcourir et de la vitesse moyenne du véhicule.

3.2.2.2. Tournées effectuées

Nous utilisons un ensemble de vecteurs « tournées effectuées », afin de mémoriser les clients servis ou sur le point de l'être par les véhicules disponibles (voir la figure 3.5). Chaque vecteur de cet ensemble représente les clients rangés dans l'ordre où ils ont été servis par le véhicule correspondant.

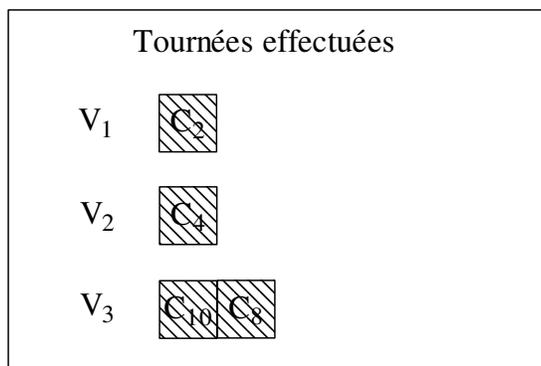


Figure 3. 5 L'ensemble des tournées effectuées

Le dernier client de chaque vecteur correspond à la destination courante du véhicule associé. Le véhicule est soit en train de le servir, soit en attente d'ouverture de la fenêtre de temps, soit se trouve sur le parcours qui le conduit vers ce client.

A la fin de la journée (ou à la fin de la simulation), l'ensemble des tournées effectuées correspond à la solution définitive de notre problème.

3.2.2.3. Le composant Algorithme Génétique (AG)

A chaque fois qu'un changement se produit sur les hypothèses initiales d'un problème d'optimisation dynamique, la solution optimale du problème initial peut se trouver ou non affectée par cette perturbation. Si c'est le cas, une adaptation de la solution courante est donc nécessaire.

L'approche classique utilisée pour traiter ce problème dynamique est de considérer que chaque changement engendre un nouveau problème d'optimisation qui doit être de nouveau résolu à partir de zéro. Cependant bien que cette approche soit simple, elle est souvent inapplicable dans un contexte à contraintes de temps serrées. En effet, la résolution d'un problème de type NP-difficile prend généralement trop de temps pour trouver des solutions de bonne qualité. Bien souvent, la solution du nouveau problème ne différera pas trop de la

solution de l'ancien problème lorsque l'environnement est faiblement perturbé, ce qui est en général le cas dans notre application. Puisque les algorithmes évolutionnistes ont en commun avec le processus d'évolution naturelle leur caractère adaptatif, ils sont de bons candidats appropriés aux environnements dynamiques [Branke 2001]. Nous utilisons donc les anciennes solutions pour fournir de nouvelles solutions qui tiennent compte de la perturbation [Housroum 2005].

Rappelons que notre AG utilise un codage indirect pour résoudre le VRPTWFD statique. Ce type de codage indirect est bien adapté pour traiter les événements du VRPTWFD dynamique. Dans la version dynamique, chaque chromosome représente une liste ordonnée de clients connus restant à servir (voir la figure 3.6). Ce chromosome possède donc une taille variable au cours du temps.

Il faut conjointement utiliser une phase de décodage pour construire les tournées des véhicules au moyen d'une heuristique d'insertion rapide. Un ensemble de vecteurs "tournées planifiées" représente les tournées planifiées.

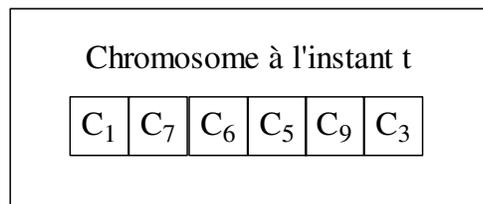


Figure 3.6 Chromosome à l'instant t

Il faut noter que les clients servis sont mémorisés dans l'ensemble de vecteurs "tournées effectuées". De ce fait, dans la première étape de la phase de décodage, il faut initialiser les "tournées planifiées" avec les clients servis issus des "tournées effectuées" en respectant la structure des tournées. Puis l'heuristique d'insertion simple est utilisée pour décoder le chromosome afin de compléter les "tournées planifiées". On obtient à la fin de l'insertion une solution au problème courant que l'on peut évaluer.

La figure 3.7 représente la phase de décodage d'un chromosome à l'instant t. Les nœuds grisés (C₂, C₄, C₁₀, C₈) représentent les clients servis ou sur le point de l'être par les véhicules. Les arcs en trait plein épais indiquent les trajets en cours ou les trajets effectués des véhicules.

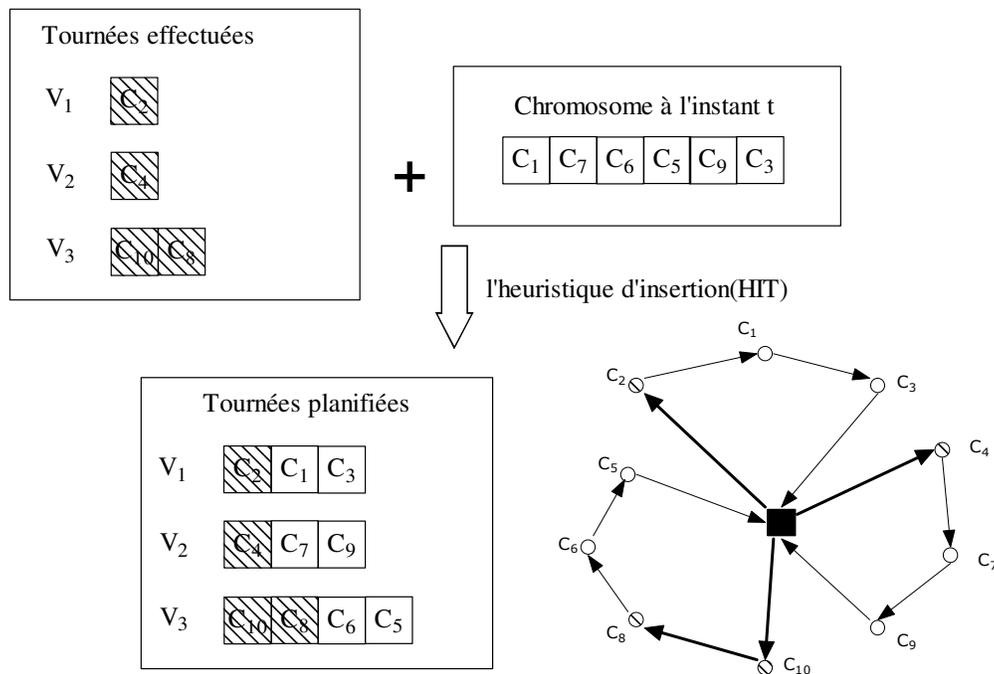


Figure 3. 7 Le décodage d'un chromosome à l'instant t

Comme nous l'avons déjà vu dans le chapitre 2, la phase d'évaluation d'un chromosome va dépendre du modèle stochastique retenu pour traiter le VRPTWFD : « DAG », « FAG_CCP » et « FAG_SPR ». Nous présentons dans ce qui suit le traitement des événements.

3.2.2.4. Le composant Gestion d'Événements (GE)

Ce composant traite les trois types d'événements « arrivée d'un nouveau client », « connaissance de la demande avec certitude » et « fin de service d'un client » générés par le simulateur du monde réel. En conséquence nous devons décrire les trois procédures répondant aux trois types d'événements à traiter.

1. L'évènement "arrivée d'un nouveau client"

Une procédure est utilisée pour traiter chaque événement "arrivée d'un nouveau client". Pour chaque chromosome de la population, elle ajoute le nouveau client dans la meilleure position du chromosome. Elle utilise pour cela une seconde heuristique d'insertion « HIC » (Heuristique d'Insertion Chromosomique) pour choisir la position d'insertion d'un nouveau client dans le chromosome. La Figure 3.8 montre un exemple de cette heuristique d'insertion au moment où le client 11 demande un service. L'opération d'insertion au niveau d'un chromosome consiste à tester l'insertion du client 11 dans toutes les positions possibles du chromosome considéré. A chaque essai, nous évaluons le fitness du chromosome ainsi obtenu, puis nous pouvons choisir la position d'insertion du client 11 qui donne le meilleur fitness (i.e. la position d'après le client 6 dans notre exemple). Pour évaluer le fitness d'un chromosome nous devons au préalable le décoder au moyen de l'heuristique d'insertion

décrite dans le chapitre deux. Rappelons que nous avons utilisé une heuristique simple « HIT » (Heuristique d’Insertion de Tournées) dans la phase de décodage pour réduire le temps de calcul.

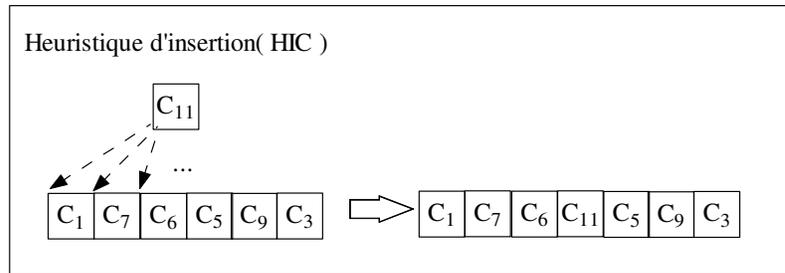


Figure 3. 8 Heuristique d’Insertion Chromosomique (HIC) dans la procédure d’évènement nouveau client

Algorithme 3.1 - Algorithme de traitement de l’évènement “nouveau client”

Entrée :
 Client : le nouveau client
 Population : La population courante qui contient Taille_{Pop} chromosomes de taille N

Sortie :
 Population : La population courante qui contient Taille_{Pop} chromosome de taille N+1.

Début :
 i = 0
 // Chrom[i] : Le chromosome i de la population.
TantQue (i < Taille_{Pop}) **Répéter**
 pos = 0
 p_{min} = pos
 ajouter un gène supplémentaire à Chrom[i] // sa taille passe de N à N+1
 f_{min} = le fitness du Chrom[i] en inserant le client Client à la position pos
 pos ++
 TantQue (pos ≤ la taille de Chrom[i]) **Répéter**
 f = le fitness de Chrom[i] en inserant le client Client à la position pos
 Si (f < f_{min})
 f_{min} = f
 p_{min} = pos
 FinSi
 pos ++
FinTantQue
 Insérer le client Client à la position p_{min} dans Chrom[i]
 i ++
FinTantQue
Fin.

La procédure cherche à insérer le nouveau client dans la meilleure position de tous les chromosomes de la population courante (algorithme3.1). Cette procédure est simple et sa complexité est en $O(\text{TaillePop} * m * n^2)$ où n représente le nombre de clients connus restant à servir est m représente le nombre de véhicules.

2. L'évènement « connaissance de la demande avec certitude »

La demande d'un client est connue avec certitude lorsque le véhicule commence à le servir. Au niveau du problème à résoudre, la demande initialement floue de ce client est alors remplacée par cette valeur. La procédure associée à cet évènement doit alors vérifier si la capacité résiduelle du véhicule est supérieure à cette valeur. En cas de « défaut de tournée » dû à une insuffisance de la capacité résiduelle, le véhicule doit rentrer au dépôt pour décharger ou se réapprovisionner puis il revient chez ce client pour le servir. Le coût supplémentaire de transport dû au « défaut de tournée » doit être pris en compte. L'ensemble de vecteurs « tournées effectuées » devra être modifié.

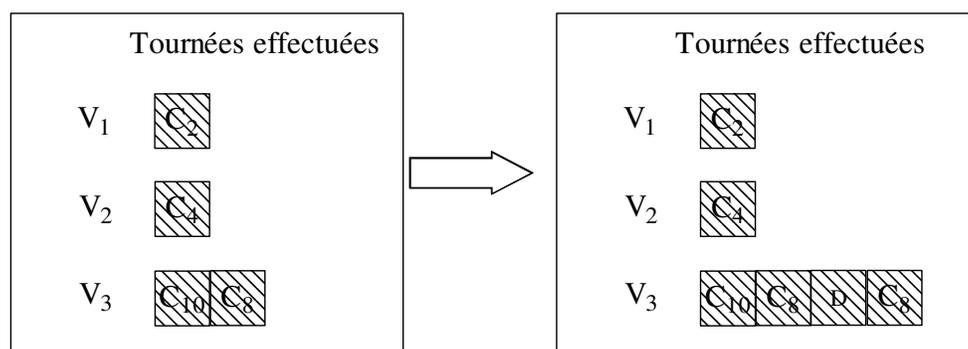


Figure 3.9 Modification du vecteur « tournées effectuées » en cas de « défaut de tournée »

La Figure 3.9 montre un exemple de modification des vecteurs « tournées effectuées » en cas de « défaut de tournée » au moment où le véhicule commence à servir le client 8. Les nœuds D et C_8 sont ajoutés au vecteur V_3 pour représenter les trajets de recours.

3. L'évènement « fin de service »

Une procédure est conçue pour traiter l'évènement fin de service d'un client (Algorithme 3.2). Le but est de fournir au conducteur du véhicule correspondant, le prochain client à servir. Dans le cas où il ne reste plus de clients à servir, le véhicule rentre au dépôt. De plus, les chromosomes de la population d'AG sont modifiés par cette procédure pour tenir compte du prochain client à servir.

Algorithme 3.2 - Algorithme de traitement de l'évènement "fin de service"

Entrée :

Véhicule : Le véhicule qui a généré l'évènement.

Tournées effectuées: L'ensemble des tournées contenant les clients déjà servis.

Population : La population courante qui contient Taille_{Pop} chromosomes de taille N

Sortie :

Tournées effectuées: L'ensemble des tournées contenant les clients déjà servis en ajoutant le client associé au véhicule qui a généré l'évènement.

Population : La population courante qui contient Taille_{Pop} chromosome de taille N-1.

Début :

Chrom[0] : Meilleur chromosome dans la population.

// Solution : L'ensemble des tournées prévisionnelles

Solution = Décoder le chromosome Chrom[0].

Ajouter le premier client Client non encore servi dans la tournée du Véhicule, à la tournée correspondante dans les Tournées effectuées.

Supprimer le client Client dans tous les chromosomes de la population.

Fin.

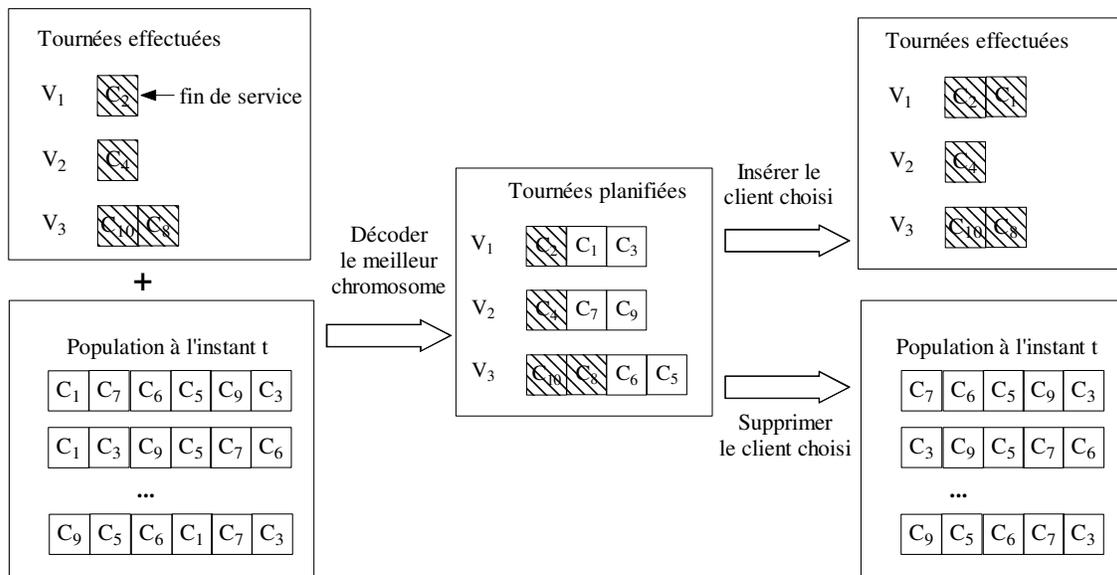


Figure 3. 10 Procédure d'évènement fin de service

La figure 3.10 illustre le déroulement de cette procédure. Rappelons qu'à chaque génération de l'AG la nouvelle population est obtenue en utilisant une méthode d'élitisme. Donc au moment où le véhicule V_1 a fini de servir le client 2, la procédure commence par récupérer le meilleur chromosome de la population courante (meilleur fitness) qui est mémorisé en première position. Elle le décode ensuite pour obtenir les tournées correspondantes. Puis, le planificateur peut simplement choisir le prochain client à servir (i.e. le client 1 dans notre exemple). Finalement, on élimine le client choisi de l'ensemble des chromosomes (vecteurs des clients à servir) de la population courante et on ajoute celui-ci dans le vecteur des tournées effectuées.

Il faut rappeler que les procédures réagissent directement après l'apparition des événements correspondants. Ainsi l'AG continue de faire évoluer la population en arrière plan tant qu'il n'y a pas de nouveaux événements.

3.2.3 La gestion du temps

Une des particularités de notre plateforme est la gestion de son horloge interne pour simuler le temps logique écoulé (t_{sim}). Toute simulation se fait à durée de temps de simulation constante (T_{sim}) quel que soit le problème traité. L'intervalle d'ouverture d'une journée de service $[0, T]$ est simulé à l'aide d'une durée de simulation constante T_{sim} . Le temps logique écoulé (t_{sim}) est calculé à l'aide du ratio suivant :

$$t_{sim} = t_{cpu} \cdot \frac{T}{T_{Sim}}$$

où t_{cpu} représente la durée de temps CPU mesurée depuis le début de la simulation. T est la durée d'ouverture du dépôt. T_{sim} représente la durée de simulation d'une journée de service. Dans nos expériences, nous avons pris $T_{sim} = 30$ mn. Ces trente minutes de temps CPU sont utilisées pour faire progresser le simulateur et dérouler l'AG. Dans une implémentation réelle sur site, on peut espérer obtenir des solutions de meilleure qualité si le temps d'ouverture du dépôt est de 8 heures par exemple et laisser l'AG tourner en « arrière plan » pendant tout ce laps de temps.

La seconde particularité de notre simulateur concerne les instants d'observation des événements qui permettent de faire évoluer dans le temps la dynamique du système étudié. Dans notre cas, le simulateur est cadencé par le déroulement de l'algorithme génétique. La figure 3.11 montre la situation quand les nouveaux événements (i.e. $E(C_i)$, $E(D_i)$, $E(F_i)$, $E(\dots)$) apparaissent au cours de temps. La prise en compte des nouveaux événements se fait à la fin de chaque génération de l'AG. A chaque fin de génération de l'AG, le temps CPU est relevé pour calculer avec la formule précédente le temps logique écoulé. Connaissant celui-ci, on peut en déduire les événements survenus durant la dernière période écoulée. En considérant que la durée (noté G_i) nécessaire pour exécuter une nouvelle génération est faible, les événements sont traités quasi-immédiatement dès leur apparition.

En l'absence d'événement, l'algorithme génétique entame la génération suivante pour continuer à faire évoluer la population courante. Dans le cas contraire, les nouveaux événements, apparus durant la génération qui vient de se terminer, sont à prendre en compte

par les procédures correspondantes.

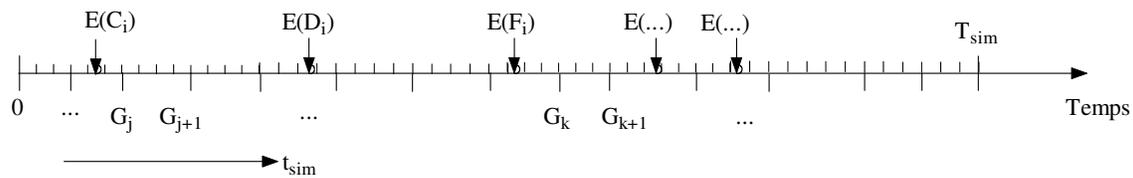


Figure 3. 11 La simulation de la journée de service

Ainsi, le temps de logique écoulé (t_{sim}) avance de façon variable selon la durée prise pour exécuter une nouvelle génération de la population courante. Pour une génération fixée, tous les chromosomes de la population courante ont une taille identique. Cette durée dépend dans notre cas de la taille des chromosomes qui correspond aux clients restant à servir.

Quand un nouveau client demande le service, l'application des opérateurs génétiques prendra plus de temps puisque la taille de chaque chromosome de la population courante augmente d'un client. Par conséquent, la durée de traitement va augmenter.

Par exemple, dans la figure 3.11, considérons initialement que la durée G_j prise par l'AG pour l'évolution d'une génération vaut trois unités de temps CPU. La prise en compte d'un nouveau client C_i pendant la durée G_j entraîne une augmentation du temps nécessaire à l'AG pour produire la prochaine population G_{j+1} , et la durée nécessite quatre unités de temps CPU. Le même phénomène peut se répéter après l'apparition d'autres clients.

Par contre, quand un évènement de fin de service apparaît (par exemple : $E(F_i)$ pendant la durée G_k), un client est choisi et fourni au conducteur du véhicule correspondant. La durée nécessaire pour exécuter la génération (G_{k+1}) est plus courte puisque ce client est supprimé de tous les chromosomes de la population courante.

3.2.4 Le fonctionnement de la plateforme

La plateforme fonctionne en deux étapes. Dans la première étape, le planificateur utilise l'algorithme génétique pour résoudre le VRPTWFD statique qui ne contient que les clients statiques, ceux-ci représentent les clients connus à l'avance (i.e. avant le début de la journée). Généralement ces clients sont connus la veille au soir et leur planification ne pose donc pas de problème de temps. Dans cette étape nous cherchons à trouver les meilleures solutions pour les clients statiques. A la fin de cette étape nous obtenons une population initiale de solutions pour traiter le problème dynamique.

La deuxième étape de notre plateforme traite de la partie dynamique du problème VRPTWFD dynamique. La solution pour tous les clients y compris les clients dynamiques est trouvée « en ligne » au cours de la journée. Cette étape fonctionne comme suit :

Démarrer la simulation d'une journée : initialiser t_{sim}

Tantque non fin de journée (i.e. $t_{sim} < T_{sim}$) **faire**

Faire évoluer la population d'une génération

Avancer t_{sim} de la durée de la génération

Traiter tous les événements « connaissance de la demande avec certitude » arrivés au cours de l'évolution courante

Traiter tous les événements « fin de service » arrivés au cours de l'évolution courante

Traiter tous les événements « nouveau client » arrivés au cours de l'évolution courante

Fin tantque

Comme nous l'avons énoncé, au cours de la simulation, la taille des chromosomes varie (i.e. elle correspond aux clients restants à servir), mais pour une génération fixée, tous les chromosomes ont une taille identique. En fin de simulation (i.e. fin de la journée de service) si chaque chromosome a une taille nulle, cela signifie que tous les clients du problème ont été servis. Dans le cas contraire (i.e. taille non nulle), il existe des clients en fin de journée qui n'ont pu être servis par les véhicules. Ces clients pourront être pris en compte dans la journée suivante et seront considérés alors comme des clients statiques. On pourrait imaginer donner une priorité haute dans le traitement de ces clients pour tenir compte de ce report.

Tous les clients servis sont mémorisés au fur et à mesure dans le vecteur « Tournées effectuées ». A la fin de la simulation, l'ensemble des tournées effectuées correspond à la solution définitive pour le VRPTWFD dynamique.

3.3 Etudes expérimentales

3.3.1 Le benchmark utilisé

Le benchmark de Salomon a été modifié par Gendreau [Gendreau et al. 1999a] pour l'adapter au problème VRPTW dynamique. Pour cela, il a ajouté un nouvel attribut à chaque client. Cet attribut, noté (tr_i) , est l'heure d'apparition de la requête. Si $(tr_i = 0)$ pour un client, ce client est considéré comme un client statique c'est-à-dire connu à l'avance (avant le début d'ouverture de la journée). Dans le cas contraire, il est considéré comme un client dynamique et la valeur de (tr_i) donne l'heure où sa requête apparaît dans la journée. Ces clients dynamiques sont aléatoirement choisis parmi l'ensemble des clients.

L'ensemble des clients est divisé en deux sous ensembles. Le premier contient à la fois les clients dont les requêtes sont arrivées avant le début de la journée ou la veille après la clôture de la journée et les requêtes qui n'ont pu être honorées la veille. Le deuxième sous ensemble contient les clients dont les requêtes vont apparaître au cours de la journée. A chaque client dynamique i est associée une valeur tr_i qui représente l'instant de réception de

sa requête. Ces valeurs tr_i sont générées aléatoirement suivant une loi uniforme dans l'intervalle $[0, \bar{e}_i]$, où :

$$\bar{e}_i = \frac{T}{l_0 - e_0} \times \min\{e_i, tdep_{i-1}\}$$

Où, $tdep_{i-1}$ est le temps de fin de service chez le client (i-1). Ce client est le prédécesseur du client i dans la tournée de la meilleure solution connue pour le benchmark de Solomon correspondant (problème statique). T est le temps nécessaire à la simulation d'une journée. e_0 et l_0 sont les horaires d'ouverture et de fermeture du dépôt.

Pour construire un benchmark pour le VRPTWFD dynamique, nous avons donc ajouté cet attribut à chaque client du benchmark conçu pour le VRPTWFD, qui a été présenté dans la section 2.5.1.2. Nous ajoutons un préfixe « D » sur les noms du benchmark pour le VRPTWFD dynamique. Nous allons résoudre trois problèmes dynamiques dans chaque catégorie (DSC1, DSC2, DSR1, DSR2, DSRC1 et DSRC2).

3.3.2 Simulation

Dans cette partie, nous allons résoudre en ligne chacun de 18 problèmes choisis par le modèle de CCP, le modèle SPR et le modèle Nor en les intégrant sur la plateforme de simulation.

Nous résolvons le VRPTWFD dynamique « en ligne ». D'une part, nous cherchons à trouver la meilleure solution au fur et à mesure d'après les informations connues au cours du temps. D'autre part, les véhicules servent les clients en suivant l'ordre indiqué dans les tournées issues de la solution. Les valeurs exactes des demandes sont connues au moment où les véhicules commencent à les servir. Nous générons de la même façon que pour le VRPTWFD statique, un scénario « test » comportant les demandes réelles des clients. La solution obtenue à la fin de la simulation est une solution particulière correspondant à ce scénario « test ». Afin de tenir compte de la variabilité de la demande, nous avons créé quatre scénarios « test » pour chaque problème à traiter.

3.3.2.1 Choix des paramètres

Dans son travail de thèse [Housroum 2005], H. Housroum a réglé les paramètres de l'algorithme génétique afin de trouver la meilleure résolution du problème VRPTW dynamique en utilisant la technique des plans d'expériences. Nous réutilisons ses réglages pour la résolution du VRPTWFD dynamique.

Rappelons que dans la version statique du VRPTWFD, les coûts des solutions trouvées par le modèle CCP et le modèle SPR varient en fonction de la combinaison choisie pour les seuils α et β correspondants à la possibilité et à la nécessité de la contrainte de capacité floue.

Il est à noter que dans le VRPTWFD dynamique, la recherche d'une solution se faisant en temps réel par étapes successives, la solution définitive n'est effectivement obtenue qu'en fin de la journée après avoir servi tous les clients. Nous ne pouvons donc pas évaluer plusieurs solutions alternatives en temps réel, revenir sur des situations passées pour régler les seuils α et β à une valeur intéressante. De ce fait pour chaque problème VRPTWFD dynamique, nous avons utilisé la combinaison des seuils α et β qui donnait la meilleure solution dans la version statique du VRPTWFD correspondant. Le nombre de scénarios d'apprentissages de demandes réelles dans le modèle SPR est fixé à 20 comme dans la résolution du VRPTWFD statique.

Nous indiquons dans le tableau 3.1 les valeurs utilisées des paramètres de réglage correspondantes à chaque problème de notre benchmark.

Problème	Paramètres de l'algorithme génétique						Combinaison des seuils			
	Taille	Cr	T2Cr	Mt	T2Mt	T2Elt(%)	SPR		CCP	
α							β	α	β	
DSC101	30	MX1T/D	0.5	λ -échange	0.2	70	1	0	1	0
DSC105	30	MX1T/D	0.5	λ -échange	0.2	70	1	0	1	0.1
DSC108	30	MX1T/D	0.5	λ -échange	0.2	70	1	0	1	0.1
DSC201	50	MX1T/D	0.7	λ -échange	0.2	50	1	0.05	1	0.05
DSC205	50	MX1T/D	0.7	λ -échange	0.2	50	1	0.05	1	0.05
DSC207	50	MX1T/D	0.7	λ -échange	0.2	50	1	0.05	1	0.05
DSR101	50	MX1T/D	0.7	λ -échange	0.5	50	1	0.2	1	0.2
DSR104	30	PMX	0.5	échange	0.2	50	1	0.1	1	0.1
DSR109	50	MX1T/D	0.7	λ -échange	0.5	50	1	0.1	1	0.1
DSR205	50	MX1T/D	0.5	λ -échange	0.5	50	0.6	0	0.2	0
DSR207	50	PMX	0.7	λ -échange	0.5	70	0.2	0	0.2	0
DSR208	50	PMX	0.7	λ -échange	0.5	70	1	0	0.6	0
DSRC101	30	MX1T/D	0.7	λ -échange	0.2	70	1	0.2	1	0.1
DSRC104	50	MX1T/D	0.5	λ -échange	0.5	50	1	0.1	1	0.1
DSRC107	30	MX1T/D	0.7	λ -échange	0.2	70	1	0.1	1	0.1
DSRC202	50	MX1T/D	0.7	échange	0.5	70	0.6	0	1	0.1
DSRC204	30	PMX	0.7	λ -échange	0.5	50	0.8	0	0.8	0
DSRC205	50	MX1T/D	0.7	échange	0.5	70	0.8	0	1	0

Tableau 3. 1 Valeurs des paramètres de réglage correspondants aux problèmes dynamiques

Dans la première étape de simulation, nous cherchons les meilleures solutions pour les clients statiques en fixant à 5000 le nombre maximal de générations d'AG. Nous fixons à 30 min le temps de simulation nécessaire pour simuler une journée d'ouverture (Tsim) dans la deuxième étape. La vitesse des véhicules est égale à 1 (unité de distance parcourue par unité de temps). Compte tenu du caractère stochastique des algorithmes génétiques, nous lançons cinq fois notre AG pour chaque scénario « test » et nous considérons la moyenne des 5 résultats.

3.3.2.2 Etude comparative des modèles stochastiques

Nous considérons ici trois modèles pour résoudre les problèmes du VRPTWFD dynamique et nous comparons la qualité respective des solutions fournies par ceux-ci.

Dans le modèle SPR, les situations futures sont prises en compte en générant des scénarios d'apprentissage. L'algorithme génétique correspondant «FAG_SPR» est le composant méthode de résolution de la plateforme de simulation. Les chromosomes de l'AG sont évalués après la phase de décodage et nécessite une phase de simulation de tournées à base de scénarios d'apprentissages. Les fitness sont donnés par le coût prévisionnel de transport qui intègre le coût dû aux « défauts de tournées ». Les scénarios d'apprentissages sont générés au début de la simulation. Mais la phase de simulation de tournées de l'AG n'utilise que la partie concernant les clients connus à l'instant logique t .

Dans la résolution par le modèle CCP, le risque de « défaut de tournée » est inférieur à un certain seuil en réglant la combinaison de seuils α et β . Cependant le coût dû aux « défauts de tournées » n'est pas prise en compte. Le composant AG méthode de résolution de la plateforme de simulation est nommé «FAG_CCP». Les chromosomes de l'AG sont uniquement évalués par le coût planifié de transport en passant la phase de décodage.

Pour le modèle Nor, le VRPTWFD est résolu comme étant un VRPTW déterministe en utilisant les valeurs normalisées des demandes floues des clients. La version AG utilisée est nommée « DAG ».

Afin de comparer la qualité des solutions fournies par chacun de ces trois modèles, nous résolvons aussi la version « Déterministe et Dynamique » de ces mêmes problèmes. C'est-à-dire que pour chaque scénario « réel », nous supposons que les demandes réelles des clients sont connues au moment de la planification. Ce modèle, noté « D&D », par la suite nous sert donc de référence car il représente une situation favorable pour la planification.

Pour chaque solution trouvée, on donne le coût de transport de la solution (noté f_{DR}), qui est l'agrégation (i.e. somme de pondérée) de la distance totale parcourue et la somme des retards chez les clients. Dans le cas dynamique, des clients peuvent ne pas être servis par les véhicules en fin de la journée. Nous notons C_{NS} le nombre de clients non servis dans la solution retenue.

La valeur t^{SPR} représente le pourcentage d'augmentation du coût de transport de la solution trouvée par le modèle SPR par rapport à la solution de la version « D&D ».

$$t^{SPR} = \frac{f_{DR} \text{ de la solution du SPR} - f_{DR} \text{ de la solution de « D \& D »}}{f_{DR} \text{ de la solution de « D \& D »}} \times 100\%$$

Respectivement, les valeurs t^{CCP} et t^{Nor} représentent les pourcentages d'augmentation du coût de transport de la solution trouvée par le modèle CCP et par le modèle Nor par rapport à la solution de la version « D&D ».

$$t^{CCP} = \frac{f_{DR} \text{ de la solution du CCP} - f_{DR} \text{ de la solution de « D \& D »}}{f_{DR} \text{ de la solution de « D \& D »}} \times 100\%$$

$$t^{Nor} = \frac{f_{DR} \text{ de la solution du Nor} - f_{DR} \text{ de la solution de « D \& D »}}{f_{DR} \text{ de la solution de « D \& D »}} \times 100\%$$

Pour les solutions trouvées par chaque modèle, nous calculons la valeur moyenne des pourcentages d'augmentation du coût de transport (t^{SPR} , ou t^{CCP} , ou t^{Nor}) correspondants à quatre scénarios « test ». Cette valeur moyenne est la déviation des solutions trouvées par le modèle correspondant par rapport à la solution de la version « D&D ». Nous calculons aussi la valeur moyenne du nombre de clients non servis correspondants aux quatre scénarios « test » pour chaque modèle.

Au titre d'un exemple du problème DSRC101, les qualités des solutions fournies par les trois modèles sont évaluées dans le tableau 3.2 pour chaque scénario « test ». Les valeurs moyennes pour l'ensemble des tests sont présentées dans la dernière ligne.

scénario « test »	« D&D »		VRPTWFD dynamique								
			Solution du modèle SPR			Solution du modèle CCP			Solution du modèle Nor		
	f_{DR}	C_{NS}	f_{DR}	C_{NS}	t^{SPR}	f_{DR}	C_{NS}	t^{CCP}	f_{DR}	C_{NS}	t^{Nor}
test 1	2055.04	0.2	2164.28	0	5.32%	2168.17	0.2	5.50%	2152	0	4.72%
test 2	2056.50	0	2189.90	0.6	6.49%	2103.00	0	2.26%	2178.69	0	5.94%
test 3	2070.65	0	2120.84	0.2	2.42%	2171.24	0	4.86%	2179.28	0.2	5.25%
test 4	2149.09	0.4	2160.71	0.8	0.54%	2250.71	0.2	4.73%	2230.07	0.8	3.77%
Moyenne		0.15		0.40	3.69%		0.10	4.34%		0.25	4.92%

Tableau 3. 2 Comparaison des solutions fournies par trois modèles avec la version « D&D » pour le problème DSRC101

Nous remarquons dans le tableau que les coûts de transport des solutions des problèmes du VRPTWFD dynamique sont globalement plus grands que celui de la version « D&D » correspondant. La valeur moyenne t^{SPR} est plus petite que les valeurs moyennes des t^{CCP} et t^{Nor} des modèles correspondants. Cela signifie que pour ce problème stochastique et dynamique, la qualité de la solution trouvée par le modèle SPR est la plus proche de celle obtenue avec le modèle « D&D ». Le modèle Nor donne les résultats les moins bons pour ce problème.

Nous reportons le détail des comparaisons pour chaque problème dans l'Annexe B.1. Nous présentons ici dans le tableau 3.3 la synthèse de ces comparaisons. Nous mettons en gras les plus petites déviations obtenues parmi celles correspondantes à 3 modèles.

Type	Problème	« D&D »	VRPTWFD dynamique					
			Solution du modèle SPR		Solution du modèle CCP		Solution du modèle Nor	
		Moy(C_{NS})	Moy(C_{NS})	Moy(t^{SPR})	Moy(C_{NS})	Moy(t^{CCP})	Moy(C_{NS})	Moy(t^{Nor})
DSC1	DSC101	0	0	0.40%	0	0.00%	0	0.00%
	DSC105	0	0	2.67%	0	0.15%	0	1.54%
	DSC108	0	0	0.61%	0	0.01%	0	1.27%
	Moyenne	0.00	0.00	1.23%	0.00	0.05%	0.00	0.94%
DSC2	DSC201	0	0	1.63%	0	0.07%	0	0.07%
	DSC205	0	0	3.68%	0	0.00%	0	0.00%
	DSC207	0	0	6.00%	0	1.22%	0	1.22%
	Moyenne	0.00	0.00	3.77%	0.00	0.43%	0.00	0.43%
DSR1	DSR101	0.6	1.1	3.11%	0.85	1.34%	0.95	0.97%
	DSR104	0.45	0.4	4.25%	0.35	3.06%	0.25	1.44%
	DSR109	0.05	1.45	5.10%	0.9	2.82%	1.05	1.94%
	Moyenne	0.37	0.98	4.15%	0.7	2.41%	0.75	1.45%
DSR2	DSR205	0.1	0.1	2.93%	0.15	1.09%	0.2	3.17%
	DSR207	0	0.15	5.12%	0.35	2.19%	0.5	5.48%
	DSR208	0.1	0.45	5.42%	0.35	4.29%	0.2	6.10%
	Moyenne	0.07	0.23	4.49%	0.28	2.52%	0.30	4.92%
DSRC1	DSRC101	0.15	0.4	3.69%	0.1	4.34%	0.25	4.92%
	DSRC104	0.2	0.2	3.24%	0.35	4.39%	0.35	3.64%
	DSRC107	0.3	0.45	4.76%	0.35	6.18%	0.35	5.43%
	Moyenne	0.22	0.35	3.90%	0.27	4.97%	0.32	4.66%
DSRC2	DSRC202	0.05	0	4.85%	0.15	7.66%	0	6.25%
	DSRC204	0.05	0.05	1.17%	0.15	3.85%	0.45	3.67%
	DSRC205	0	0	3.85%	0	4.35%	0	4.04%
	Moyenne	0.03	0.02	3.29%	0.10	5.29%	0.15	4.65%
Tous	Moyenne	0.11	0.26	3.47%	0.23	2.61%	0.25	2.84%

Tableau 3.3 Comparaison des solutions fournies par trois modèles pour le VRPTWFD dynamique avec la solution de la version « D&D »

A travers ce tableau, nous voyons que pour les problèmes de type DSC, qui correspondent à une distribution des clients en clusters, en servant tous les clients, les déviations (Moy(t^{SPR})) des solutions trouvées par le modèle SPR sont plus grandes que les

déviations ($\text{Moy}(t^{CCP})$) des solutions trouvées par le modèle CCP. Cela signifie que les solutions trouvées par le modèle CCP sont meilleures que les solutions trouvées par le modèle SPR. Dans ce cas, la simulation de tournées à base de scénarios d'apprentissage n'a pas amélioré la qualité de la solution.

Nous remarquons aussi que les meilleures solutions trouvées par le modèle CCP pour les problèmes du type DSC sont identiques ou très proches de celles de la version « D&D » correspondantes : 0% dans deux problèmes et proches de 1% dans les deux autres problèmes de ce type.

De plus, les déviations trouvées par le modèle Nor et les déviations trouvées par le modèle CCP sont identiques pour 4 problèmes du type DSC. Cela signifie que les solutions trouvées par le modèle Nor sont aussi bonnes que les solutions trouvées par le modèle CCP.

Notre explication est que l'influence des paramètres VRPTW sur les résolutions des problèmes, telle que la distribution géographique et les fenêtres de temps des clients est plus importante que l'influence de l'incertitude des demandes.

Pour les problèmes du type DSRC qui correspondent à une combinaison de distributions uniformes et de clusters, le modèle SPR fournit des solutions de qualités plus proches de celles de la version « Déterministe et Dynamique » par rapport au modèle CCP et modèle Nor.

Le modèle CCP fournit des meilleures solutions pour les problèmes du type DSR2 qui correspondent à une distribution uniforme des clients dans l'espace et une fenêtre de temps larges sur le dépôt central.

Le modèle Nor fournit des meilleures solutions pour les problèmes du type DSR1 qui correspondent à une distribution uniforme des clients dans l'espace et une fenêtre de temps étroite sur le dépôt central.

En résumé pour les 18 problèmes du VRPTWFD dynamique, le modèle CCP fournit de meilleures solutions pour 9 problèmes des 3 catégories (DSC1, DSC2 et DSR2), le modèle SPR fournit de meilleures solutions pour 6 problèmes des 2 catégories (DSRC1 et DSRC2). Pour la plupart de ces problèmes, le modèle flou (SPR ou CCP) en utilisant la théorie de possibilité nous permet de trouver une solution assez proche de la solution trouvée pour la version « stochastique et dynamique » correspondante.

La résolution en ligne du VRPTWFD dynamique se fait en fixant une durée de simulation constante de 30 minutes. Dans la version AG « FAG_SPR », l'évaluation des chromosomes nécessite, après la phase de décodage, une phase de simulation de tournées à base de scénarios d'apprentissages. La durée de cette phase augmente le temps nécessaire pour calculer une nouvelle génération. Elle est plus grande en comparaison avec les versions AG « FAG_CCP » et « DAG ». De ce fait, contrairement au cas VRPTWFD statique, le modèle SPR ne peut pas toujours fournir la meilleure solution pour tous les problèmes du VRPTWFD dynamique.

3.3.3 Comparaison avec la version « Déterministe et Statique »

Nous avons également comparé la qualité des meilleures solutions fournies par le modèle

correspondant avec une version « Déterministe et Statique » (notée « D&S ») de ces problèmes. Dans celle-ci, nous supposons que tous les clients sont connus à l'avance et que les demandes sont les demandes réelles qui figurent dans les scénarios « test » correspondants. A priori on se place dans une situation favorable pour l'AG qui dispose ainsi de toute la connaissance pour faire la planification. Les solutions de la version « D&S » sont obtenues en exécutant la version AG nommée « DAG ».

La valeur t^{SD} représente le pourcentage d'augmentation du coût de transport par rapport à celui de la solution de la version « D&S ».

$$t^{SD} = \frac{f_{DR} \text{ de la meilleure solution} - f_{DR} \text{ de la solution de « D \& S »}}{f_{DR} \text{ de la solution de « D \& S »}} \times 100\%$$

Pour chaque problème, nous calculons les valeurs moyennes des t^{SD} correspondants aux scénarios « test ». Cette déviation par rapport à la solution de la version « D&S » évalue la performance du modèle stochastique choisi pour traiter le VRPTWFD dynamique.

Par exemple pour le DSRC101, nous comparons les meilleures solutions trouvées par le modèle SPR avec celles de la version « D&S » correspondante (voir tableau 3.4). En moyenne, on se trouve à 17% de la solution fournie par la version « D&S », ce qui n'est pas trop dégradé.

scénario « test »	« D&S »	VRPTWFD dynamique		
	f_{DR}	f_{DR}	C_{NS}	t^{SD}
test 1	1831.06	2164.28	0	18.20%
test 2	1859.92	2189.90	0.6	17.74%
test 3	1856.59	2120.84	0.2	14.23%
test 4	1846.73	2160.71	0.8	17.00%
Moyenne			0.4	16.79%

Tableau 3.4 Comparaison des meilleures solutions pour le problème DSRC101 avec celles de la version « D&S » correspondante

En reprenant les résultats obtenus dans la section précédente 3.3.2, nous avons comparé pour chaque type de problème la meilleure solution obtenue par l'un des trois modèles avec la meilleure solution trouvée avec la version « D&S ». Nous donnons le détail de ces comparaisons dans l'Annexe B.2.

Nous présentons ici une synthèse de ces comparaisons sur les valeurs moyennes des C_{NS} et t^{SD} correspondantes aux scénarios « test » dans le tableau 3.5.

Type	Problème	dod	Moyenne(C_{NS})	Moyenne(t^{SD})
DSC1	DSC101	0.40	0	0.00%
	DSC105	0.49	0	5.60%
	DSC108	0.52	0	7.99%
	Moyenne		0.00	4.53%
DSC2	DSC201	0.48	0	0.07%
	DSC205	0.49	0	0.61%
	DSC207	0.56	0	-1.51%
	Moyenne		0.00	-0.28%
DSR1	DSR101	0.37	1.2	4.15%
	DSR104	0.10	0.25	1.27%
	DSR109	0.49	1.25	15.31%
	Moyenne		0.90	6.91%
DSR2	DSR205	0.47	0.1	2.32%
	DSR207	0.29	0.3	1.57%
	DSR208	0.11	0.25	0.73%
	Moyenne		0.22	1.54%
DSRC1	DSRC101	0.47	0.25	16.79%
	DSRC104	0.10	0.2	3.49%
	DSRC107	0.36	0.4	10.05%
	Moyenne		0.28	10.11%
DSRC2	DSRC202	0.36	0.1	4.55%
	DSRC204	0.16	0.05	2.49%
	DSRC205	0.50	0	5.92%
	Moyenne		0.05	4.32%
Tours	Moyenne			4.52%

Tableau 3. 5 Comparaison des meilleures solutions obtenues pour les problèmes du VRPTWFD dynamique avec celles de la version « D&S »

Nous voyons dans ce tableau, pour le problème DSC101, que la valeur moyenne t^{SD} correspondant aux scénarios est égale à 0. C'est-à-dire que la solution de ce problème du VRPTWFD dynamique et la solution de la version « déterministe et statique » correspondante sont identiques ou de même qualité.

Globalement, les solutions des problèmes du VRPTWFD dynamique ont un coût de transport supérieur par rapport aux solutions de la version « D&S » correspondante. La seule exception est le cas du problème DSC207 mais n'oublions pas que nous utilisons des AG dont le comportement est par nature stochastique.

Pour les problèmes de type DSC1, DSR1, DSR2, DSRC1 et DSRC2, les valeurs moyennes t^{SD} varient avec le degré de dynamisme du problème (colonne « dod » du tableau 3.5).

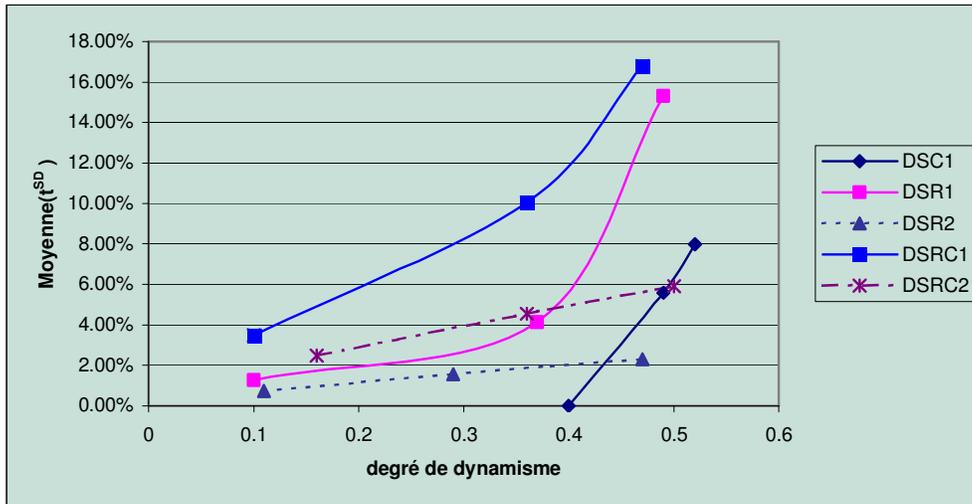


Figure 3.12 Valeur moyenne t^{SD} en fonction du degré de dynamisme

Nous remarquons que dans une même famille, plus le degré de dynamisme du problème est grand, plus la valeur moyenne t^{SD} correspondante est grande (voir la figure 3.12). Cela signifie que plus il y a d'apparitions de clients au cours du temps, plus le coût de transport augmente par rapport à la solution trouvée en connaissant toutes les informations avec certitude à l'avance ce qui est conforme avec l'intuition.

La déviation moyenne des solutions obtenues en ligne pour le VRPTWFD dynamique est de 4.52%. Ceci montre que les solutions trouvées pour le VRPTWFD dynamique et les solutions trouvées en connaissant toutes les informations avec certitude à l'avance sont relativement proches, et que notre approche est assez performante.

3.4 Conclusion

Dans ce chapitre nous avons présenté trois modèles pour la résolution du problème VRPTWFD dynamique « en ligne ». Les trois types d'événements « arrivée d'un nouveau client », « la demande connue avec certitude » et « fin de service d'un client » sont pris en compte dans la résolution dynamique.

Nous avons intégré ces versions dans une plateforme Java capable de simuler une journée de service et qui utilise comme moteur d'optimisation un algorithme génétique. Cette plateforme vérifie l'occurrence des événements après chaque itération de l'AG et répond à chaque événement en utilisant une procédure adaptée. Il est à noter que la taille de nos chromosomes varie selon le nombre de clients restant à servir par les véhicules pendant toute la durée de la simulation.

Nous avons construit un ensemble de benchmarks pour le VRPTWFD dynamique en ajoutant l'attribut de l'instant de réception du client et nous avons résolu ces problèmes en utilisant cette plateforme déclinée sous ces trois modèles.

Nous avons comparé la qualité des solutions fournies par ces modèles avec celle d'une version « Déterministe et Dynamique » où toutes les demandes des clients sont connues de façon exacte et progressive dans le temps. Nous avons vu que la qualité des solutions obtenues par le modèle flou CCP est globalement proche de celle de la version « D&D ».

Nous avons également comparé la qualité de la meilleure solution fournie des trois modèles précédents avec celle d'une version « déterministe et statique » où toutes les informations sont connues avec certitude et à l'avance. Nous avons vu que la qualité des solutions obtenues en ligne est proche de celle de la version « D&S » (environ 5% d'écart). Ceci montre que notre approche pour la résolution VRPTWFD dynamique est relativement performante.

Conclusions générales et perspectives

Etant donné son importance économique, nous avons traité le problème de la gestion des tournées de véhicules avec fenêtres de temps et demandes floues (VRPTWFD) dans un contexte aussi bien statique que dynamique.

En fixant les valeurs des seuils de possibilité et nécessité pour la contrainte de capacité d'un véhicule, nous avons proposé un modèle de programmation sous contraintes probabilistes (CCP) et un modèle à deux-étapes de programmation stochastique avec recours (SPR) pour traiter le VRPTWFD. Dans le modèle SPR, les situations futures sont prises en compte à l'aide de scénarios d'apprentissages. Nous proposons aussi un modèle, nommé Nor, dans lequel le VRPTWFD est résolu comme un VRPTW déterministe en utilisant les valeurs de normalisation des demandes floues.

Trois algorithmes génétiques qui intègrent ces trois modèles, ont été proposés pour la recherche de bonnes solutions. Nous avons construit un benchmark pour le VRPTWFD statique en modifiant le jeu de problèmes fournis par Solomon pour le VRPTW. Nous avons évalué les qualités des solutions fournies par les modèles dans un environnement réel en simulant les situations réelles à l'aide des scénarios « test ». Nous avons vu que la qualité des solutions obtenues par le modèle SPR est proche de celle obtenue par un modèle déterministe et que ces solutions sont robustes par rapport à la réalisation effective des variables stochastiques. Ceci nous permet de dire que :

- Le modèle SPR fournit une solution de meilleure qualité pour le VRPTWFD statique par rapport aux deux autres modèles.
- La prise en compte de situations futures peut améliorer la solution pour le VRPTWFD même si le taux d'incertitude sur la demande n'est pas élevé.
- L'algorithme génétique adapté au problème traité est une méthode alternative, efficace et flexible pour la résolution de ce type de problème.

Pour la résolution « en ligne » du VRPTWFD dynamique, nous avons intégré ces modèles dans une plateforme de simulation capable de simuler une journée de service et qui utilise comme moteur d'optimisation l'algorithme génétique précédent.

Basé sur le benchmark pour le VRPTWFD statique, nous avons construit un ensemble de benchmarks pour le VRPTWFD dynamique en ajoutant l'attribut de l'instant de réception du client et nous avons résolu ces problèmes en utilisant cette plateforme déclinée sous ces trois modèles.

Nous avons comparé la qualité des solutions fournies par ces modèles avec celle d'une version « Déterministe et Dynamique » où toutes les demandes des clients sont connues de façon exacte et progressive dans le temps. Nous avons également comparé la qualité de la

meilleure solution fournie des trois modèles précédents avec celle d'une version « Déterministe et Statique » où toutes les informations sont connues avec certitude et à l'avance.

- Le modèle CCP fournit une solution de meilleure qualité pour le VRPTWFD dynamique par rapport aux deux autres modèles.
- La résolution « en ligne » en utilisant notre plateforme de simulation pour le VRPTWFD dynamique est relativement performante.

Notre étude empirique a montré que l'algorithme génétique, qui intègre les modèles stochastiques, permet de donner les solutions de bonnes qualités pour le VRPTWFD.

Dans le cadre de benchmark pour le VRPTWFD, nous comptons étendre les modifications sur les 56 instances du benchmark de Solomon pour le VRPTW, puis les résoudre toutes.

Comme perspective on se propose d'introduire une recherche locale au sein des opérateurs génétiques comme c'est le cas dans les approches évolutionnistes hybrides (memetic algorithms) pour obtenir des solutions de qualités supérieures.

L'intégration de ces modèles stochastiques avec les autres metaheuristiques, tel que la recherche Tabou est aussi envisageable, afin de doter la recherche globale de mécanisme d'exploration local.

Nous pourrions également étendre notre travail pour tenir compte de temps de trajet variable selon la période de la journée (i.e. période normale ou période de pointe), ou bien selon la région (i.e. la vitesse des véhicules dans les villes diffère de celle dans les villages).

Toutes ces perspectives peuvent faire l'objet de futures améliorations et extensions du travail réalisé, tout en prenant en considération les exigences du marché.

Annexe A

Résultats des problèmes de VRPTWFD statique

A.1 Le réglage des paramètres de l'algorithme génétique

Problème	Taille	Cr	T2Cr	Mt	T2Mt	T2Elt(%)
SC101	50	MX1T/D	0.5	échange	0.5	50
SC105	50	MX1T/D	0.5	échange	0.5	50
SC108	50	MX1T/D	0.5	échange	0.5	50
SC109	50	MX1T/D	0.5	échange	0.5	50
SC201	50	MX1T/D	0.7	λ -échange	0.2	70
SC205	50	MX1T/D	0.7	λ -échange	0.2	70
SC207	50	MX1T/D	0.7	λ -échange	0.2	70
SC208	50	MX1T/D	0.7	λ -échange	0.2	70
SR101	50	MX1T/D	0.7	échange	0.5	50
SR103	50	MX1T/D	0.7	échange	0.5	50
SR104	50	PMX	0.5	λ -échange	0.5	70
SR109	50	MX1T/D	0.7	λ -échange	0.5	50
SR205	50	MX1T/D	0.5	λ -échange	0.2	50
SR206	50	MX1T/D	0.5	λ -échange	0.2	50
SR207	50	MX1T/D	0.5	λ -échange	0.2	50
SR208	50	MX1T/D	0.5	λ -échange	0.2	50
SRC101	50	MX1T/D	0.7	λ -échange	0.2	50
SRC104	50	PMX	0.5	λ -échange	0.5	70
SRC106	50	MX1T/D	0.7	λ -échange	0.2	50
SRC107	50	MX1T/D	0.7	λ -échange	0.2	50
SRC202	50	MX1T/D	0.7	λ -échange	0.5	50
SRC204	50	MX1T/D	0.7	λ -échange	0.5	50
SRC205	50	MX1T/D	0.7	λ -échange	0.5	50
SRC208	50	MX1T/D	0.7	λ -échange	0.5	50

Tableau A. 1 Le réglage des paramètres de l'algorithme génétique

A.2 Le réglage des seuils α et β du modèle CCP

Problème	Combinaison de seuils		Solution du modèle CCP		Réalisation de la solution	
	α	β	f_{DR}^{Plan}	Nombre de clients planifiés	f_{DR}^M	f_{DR}^S
SC101	0	0	828.94	100	949.56	120.62
	0.2	0	828.94	100	949.56	120.62
	0.4	0	828.94	100	949.56	120.62
	0.6	0	828.94	100	949.56	120.62
	0.8	0	828.94	100	949.56	120.62
	1	0	828.94	100	949.56	120.62
	1	0.1	2264.38	100	2274.48	10.10
	1	0.2	3781.00	100	3796.12	15.12
	1	0.3	4300.87	100	4308.91	8.04
	1	0.4	5018.88	99	5020.79	1.90
SC105	0	0	827.12	100	1060.62	233.50
	0.2	0	828.51	100	965.39	136.88
	0.4	0	828.51	100	965.39	136.88
	0.6	0	828.51	100	965.39	136.88
	0.8	0	828.94	100	944.31	115.37
	1	0	828.94	100	944.31	115.37
	1	0.1	906.63	100	921.12	14.49
	1	0.2	906.63	100	921.12	14.49
	1	0.3	1947.64	100	1948.57	0.93
	1	0.4	2385.26	99	2385.26	0.00
SC108	0	0	828.33	100	978.96	150.63
	0.2	0	828.33	100	978.96	150.63
	0.4	0	827.76	100	968.59	140.83
	0.6	0	828.94	100	955.71	126.77
	0.8	0	828.94	100	955.71	126.77
	1	0	828.94	100	955.71	126.77
	1	0.1	916.58	100	927.66	11.08
	1	0.2	948.39	100	959.97	11.58
	1	0.3	1017.36	100	1020.60	3.24
	1	0.4	1162.17	99	1162.17	0.00

SC109	0	0	828.73	100	997.96	169.23
	0.2	0	828.73	100	997.96	169.23
	0.4	0	828.73	100	997.96	169.23
	0.6	0	828.73	100	997.96	169.23
	0.8	0	828.94	100	920.44	91.50
	1	0	828.94	100	920.44	91.50
	1	0.1	896.70	100	912.88	16.18
	1	0.2	946.74	100	960.71	13.97
	1	0.3	947.79	100	952.36	4.57
	1	0.4	976.04	99	980.61	4.57

Tableau A. 2 L'ensemble des solutions du modèle CCP correspond aux combinaisons des seuils pour les problèmes du type SC1

Problème	Combinaison de seuils		Solution du modèle CCP		Réalisation de la solution	
	α	β	f_{DR}^{Plan}	Nombre de clients planifiés	f_{DR}^M	f_{DR}^S
SC201	0	0	591.56	100	662.09	70.53
	0.2	0	591.56	100	662.09	70.53
	0.4	0	591.56	100	662.09	70.53
	0.6	0	591.56	100	662.09	70.53
	0.8	0	591.56	100	662.09	70.53
	1	0	603.88	100	652.84	48.96
	1	0.05	636.72	100	643.87	7.15
	1	0.1	678.52	100	683.79	5.27
	1	0.15	608.11	99	611.03	2.92
SC205	0	0	588.88	100	651.8	62.92
	0.2	0	588.88	100	651.8	62.92
	0.4	0	588.88	100	651.8	62.92
	0.6	0	588.88	100	651.8	62.92
	0.8	0	588.88	100	651.8	62.92
	1	0	601.2	100	644.29	43.09
	1	0.05	630.12	100	633.62	3.5
	1	0.1	630.12	100	633.62	3.5
	1	0.15	605.20	99	605.20	0
SC207	0	0	609.53	100	682.98	73.45

	0.2	0	609.53	100	682.98	73.45
	0.4	0	609.53	100	682.98	73.45
	0.6	0	609.53	100	682.98	73.45
	0.8	0	609.53	100	682.98	73.45
	1	0	621.85	100	672.26	50.41
	1	0.5	630.32	100	630.32	0
	1	0.1	639.94	100	643.57	3.63
	1	0.15	611.46	99	611.46	0
	0	0	588.32	100	674.84	86.52
	0.2	0	588.32	100	674.84	86.52
	0.4	0	588.32	100	674.84	86.52
	0.6	0	588.32	100	674.84	86.52
	0.8	0	588.32	100	674.84	86.52
	1	0	600.65	100	666.75	66.1
	1	0.05	630.08	100	636.8	6.72
	1	0.1	630.08	100	636.8	6.72
	1	0.15	601.14	99	601.48	0.35

Tableau A. 3 L'ensemble des solutions du modèle CCP correspond aux combinaisons des seuils pour les problèmes du type SC2

Problème	Combinaison de seuils		Solution du modèle CCP		Réalisation de la solution	
	α	β	f_{DR}^{Plan}	Nombre de clients planifiés	f_{DR}^M	f_{DR}^S
SR101	0	0	1653.21	100	2450.31	797.10
	0.2	0	1668.98	100	2473.70	804.72
	0.4	0	1671.69	100	2349.30	677.61
	0.6	0	1697.80	100	2131.94	434.14
	0.8	0	1701.54	100	2022.65	321.11
	1	0	1722.15	100	1899.01	176.86
	1	0.1	1726.44	100	1813.09	86.65
	1	0.2	1774.39	100	1801.68	27.30
	1	0.3	1837.87	100	1858.45	20.58
	1	0.4	1855.46	99	1858.74	3.28
SR103	0	0	1272.05	100	1799.90	527.86
	0.2	0	1244.07	100	1695.74	451.67
	0.4	0	1270.63	100	1672.89	402.26

	0.6	0	1327.81	100	1576.50	248.69
	0.8	0	1315.48	100	1521.84	206.36
	1	0	1341.46	100	1419.86	78.40
	1	0.1	1316.30	100	1344.49	28.19
	1	0.2	1345.51	100	1365.10	19.59
	1	0.3	1397.50	100	1399.56	2.06
	1	0.4	1434.08	99	1435.44	1.35
SR104	0	0	1129.25	100	1337.70	208.45
	0.2	0	1111.12	100	1462.29	351.17
	0.4	0	1133.35	100	1414.24	280.89
	0.6	0	1139.06	100	1369.02	229.96
	0.8	0	1142.34	100	1331.07	188.73
	1	0	1198.26	100	1272.21	73.96
	1	0.1	1223.91	100	1258.87	34.96
	1	0.2	1242.84	100	1261.55	18.71
	1	0.3	1409.02	100	1419.74	10.72
	1	0.4	1433.83	99	1436.08	2.25
SR109	0	0	1321.28	100	1797.75	476.47
	0.2	0	1284.83	100	1666.52	381.68
	0.4	0	1329.93	100	1648.27	318.34
	0.6	0	1307.21	100	1614.78	307.57
	0.8	0	1349.92	100	1541.20	191.28
	1	0	1366.27	100	1429.83	63.55
	1	0.1	1368.05	100	1395.26	27.20
	1	0.2	1393.02	100	1404.35	11.33
	1	0.3	1427.25	100	1431.32	4.07
	1	0.4	1489.35	99	1489.35	0.00

Tableau A. 4 L'ensemble des solutions du modèle CCP correspond aux combinaisons des seuils pour les problèmes du type SR1

Problème	Combinaison de seuils		Solution du modèle CCP		Réalisation de la solution	
	α	β	f_{DR}^{Plan}	Nombre de clients planifiés	f_{DR}^M	f_{DR}^S
SR205	0	0	1174.39	100	1226.94	52.55

	0.2	0	1134.91	100	1181.71	46.80
	0.4	0	1160.51	100	1227.38	66.87
	0.6	0	1142.41	100	1204.88	62.47
	0.8	0	1183.41	100	1227.83	44.42
	1	0	1201.52	100	1224.82	23.31
	1	0.1	1215.31	100	1222.97	7.65
	1	0.2	1178.13	99	1179.03	0.90
SR206	0	0	1048.63	100	1117.73	69.10
	0.2	0	1037.81	100	1077.32	39.51
	0.4	0	1052.50	100	1127.56	75.05
	0.6	0	1044.94	100	1089.64	44.70
	0.8	0	1041.40	100	1077.27	35.87
	1	0	1078.90	100	1091.31	12.40
	1	0.1	1095.47	100	1097.55	2.09
	1	0.2	1084.64	99	1084.91	0.27
SR207	0	0	918.92	100	971.83	52.91
	0.2	0	939.50	100	986.36	46.86
	0.4	0	923.68	100	1000.58	76.89
	0.6	0	990.69	100	1057.57	66.88
	0.8	0	1000.16	100	1040.19	40.03
	1	0	973.74	100	990.51	16.76
	1	0.1	1048.25	100	1062.94	14.69
	1	0.2	995.06	99	996.06	1.00
SR208	0	0	1113.52	100	1189.61	76.09
	0.2	0	1043.44	100	1192.90	149.46
	0.4	0	1035.48	100	1087.60	52.12
	0.6	0	914.54	100	1011.17	96.63
	0.8	0	948.84	100	1035.64	86.8
	1	0	992.94	100	1017.03	24.09
	1	0.1	1013.49	100	1016.34	2.85
	1	0.2	1005.85	99	1005.85	0

Tableau A. 5 L'ensemble des solutions du modèle CCP correspond aux combinaisons des seuils pour les problèmes du type SR2

Problème	Combinaison de seuils		Solution du modèle CCP		Réalisation de la solution	
	α	β	f_{DR}^{Plan}	Nombre de clients planifiés	f_{DR}^M	f_{DR}^S
SRC101	0	0	1727.23	100	2765.71	1038.48
	0.2	0	1743.38	100	2757.41	1014.03
	0.4	0	1779.29	100	2839.64	1060.35
	0.6	0	1799.04	100	2539.97	740.93
	0.8	0	1810.16	100	2179.38	369.22
	1	0	1870.03	100	2042.77	172.74
	1	0.1	1901.24	100	1981.97	80.73
	1	0.2	1968.46	100	2020.49	52.04
	1	0.3	2019.64	100	2029.75	10.11
	1	0.4	1986.55	99	1988.48	1.92
SRC104	0	0	1187.48	100	2090.21	902.73
	0.2	0	1218.33	100	1907.14	688.81
	0.4	0	1206.59	100	1911.19	704.6
	0.6	0	1294.59	100	1936.64	642.04
	0.8	0	1343.13	100	1631.44	288.32
	1	0	1338.21	100	1493.33	155.12
	1	0.1	1351.68	100	1440.96	89.28
	1	0.2	1362.77	100	1393.34	30.57
	1	0.3	1535.96	100	1543.38	7.426
	1	0.4	1538.914	99	1538.91	0
SRC106	0	0	1460.28	100	2264.61	804.34
	0.2	0	1518.14	100	2199.77	681.62
	0.4	0	1535.59	100	2167.78	632.19
	0.6	0	1570.53	100	2063.92	493.39
	0.8	0	1557.73	100	1889.82	332.09
	1	0	1628.72	100	1743.79	115.07
	1	0.1	1627	100	1666.54	39.53
	1	0.2	1658.58	100	1685.12	26.53
	1	0.3	1751.75	100	1766.89	15.14
	1	0.4	1759.52	99	1762.17	2.65
SRC107	0	0	1327.01	100	2034.40	707.40
	0.2	0	1413.14	100	1963.77	550.63

	0.4	0	1421.59	100	2082.76	661.17
	0.6	0	1481.46	100	1959.27	477.81
	0.8	0	1459.95	100	1732.80	272.85
	1	0	1447.57	100	1590.79	143.21
	1	0.1	1446.67	100	1481.21	34.54
	1	0.2	1494.08	100	1506.89	12.81
	1	0.3	1552.44	99	1552.44	0

Tableau A. 6 L'ensemble des solutions du modèle CCP correspond aux combinaisons des seuils pour les problèmes du type SRC1

Problème	Combinaison de seuils		Solution du modèle CCP		Réalisation de la solution	
	α	β	f_{DR}^{Plan}	Nombre de clients planifiés	f_{DR}^M	f_{DR}^S
SRC202	0	0	1301.61	100	1518.56	216.96
	0.2	0	1248.17	100	1496.48	248.31
	0.4	0	1263.39	100	1428.60	165.20
	0.6	0	1315.44	100	1463.59	148.14
	0.8	0	1318.78	100	1466.65	147.87
	1	0	1332.50	100	1372.03	39.53
	1	0.1	1352.32	100	1354.60	2.28
	1	0.2	1436.58	99	1436.81	0.23
SRC204	0	0	1005.25	100	1201.10	195.84
	0.2	0	993.55	100	1106.40	112.85
	0.4	0	1026.30	100	1114.72	88.43
	0.6	0	1095.73	100	1230.49	134.76
	0.8	0	1047.03	100	1106.01	58.98
	1	0	1106.65	100	1145.54	38.89
	1	0.1	1158.10	100	1165.35	7.26
	1	0.2	1131.03	99	1131.03	0.00
SRC205	0	0	1448.50	100	1620.20	171.69
	0.2	0	1427.17	100	1618.12	190.95
	0.4	0	1432.16	100	1581.11	148.95
	0.6	0	1419.85	100	1597.92	178.07
	0.8	0	1463.86	100	1559.17	95.31

	1	0	1509.06	100	1556.92	47.87
	1	0.1	1552.40	100	1559.57	7.17
	1	0.2	1714.52	100	1715.09	0.58
SRC206	0	0	1386.87	100	1530.96	144.09
	0.2	0	1380.51	100	1574.57	194.03
	0.4	0	1375.21	100	1587.72	212.51
	0.6	0	1362.03	100	1547.05	185.02
	0.8	0	1410.52	100	1473.70	63.18
	1	0	1488.35	100	1519.92	31.57
	1	0.1	1634.91	100	1643.58	8.67
	1	0.2	1575.35	100	1576.98	1.63

Tableau A. 7 L'ensemble des solutions du modèle CCP correspond aux combinaisons des seuils pour les problèmes du type SRC2

A.3 Le réglage des seuils α et β du modèle SPR

Problème	Combinaison de seuils		Solution du modèle SPR			
	α	β	f_{DR}^{Plan}	f_{DR}^{Pre}	f_{DR}^{St}	Nombre de clients planifiés
SC101	0	0	828.94	986.06	157.12	100
	0.2	0	828.94	986.06	157.12	100
	0.4	0	828.94	986.06	157.12	100
	0.6	0	828.94	986.06	157.12	100
	0.8	0	828.94	986.06	157.12	100
	1	0	828.94	986.06	157.12	100
	1	0.1	1277.17	1308.87	31.70	100
	1	0.2	1313.95	1348.47	34.53	100
	1	0.3	1526.84	1539.07	12.23	100
	1	0.4	2371.70	2371.70	0.00	100
SC105	0	0	865.65	917.96	52.31	100
	0.2	0	865.65	917.96	52.31	100
	0.4	0	865.65	917.96	52.31	100
	0.6	0	865.65	917.96	52.31	100

	0.8	0	864.61	917.50	52.89	100
	1	0	864.61	917.50	52.89	100
	1	0.1	925.89	929.14	3.25	100
	1	0.2	1073.20	1075.78	2.58	100
	1	0.3	1711.38	1711.38	0.00	100
	1	0.4	1461.67	1461.67	0.00	100
SC108	0	0	892.62	926.21	33.59	100
	0.2	0	888.08	926.40	38.32	100
	0.4	0	889.85	928.45	38.61	100
	0.6	0	915.18	926.29	11.11	100
	0.8	0	903.23	929.41	26.18	100
	1	0	887.89	912.64	24.76	100
	1	0.1	925.26	935.30	10.03	100
	1	0.2	949.86	957.38	7.52	100
	1	0.3	950.23	952.71	2.48	100
	1	0.4	980.96	980.96	0.00	100
SC109	0	0	916.67	946.46	29.80	100
	0.2	0	899.26	935.48	36.23	100
	0.4	0	899.26	935.48	36.23	100
	0.6	0	881.20	919.61	38.41	100
	0.8	0	908.23	919.35	11.12	100
	1	0	901.65	916.70	15.05	100
	1	0.1	915.29	921.00	5.71	100
	1	0.2	931.59	938.10	6.51	100
	1	0.3	992.63	996.05	3.43	100
	1	0.4	1042.25	1047.18	4.93	100

Tableau A. 8 L'ensemble des solutions du modèle SPR correspond aux combinaisons des seuils pour les problèmes du type SC1

Problème	Combinaison de seuils		Solution du modèle SPR			
	α	β	f_{DR}^{Plan}	f_{DR}^{Pre}	f_{DR}^{St}	Nombre de clients planifiés
SC201	0	0	621.51	659.46	37.95	100
	0.2	0	621.51	659.46	37.95	100

Problème	Combinaison de seuils		Solution du modèle SPR			
	α	β	f_{DR}^{Plan}	f_{DR}^{Pre}	f_{DR}^{St}	Nombre de clients planifiés
SR101	0	0	1811.68	1859.81	48.13	100
	0.2	0	1873.69	1904.89	31.20	100
	0.4	0	1816.31	1834.28	17.97	100
	0.6	0	1781.35	1796.27	14.92	100
	0.8	0	1771.04	1786.60	15.56	100
	1	0	1778.19	1783.67	5.48	100
	1	0.1	1774.74	1786.10	11.37	100
	1	0.2	1756.94	1765.25	8.31	100
	1	0.3	1776.03	1778.30	2.27	100
	1	0.4	1829.35	1829.35	0.00	99
SR103	0	0	1376.84	1395.74	18.90	100
	0.2	0	1405.57	1406.96	1.39	100
	0.4	0	1384.44	1393.98	9.55	100
	0.6	0	1369.04	1372.23	3.19	100
	0.8	0	1360.60	1366.29	5.69	100
	1	0	1326.63	1328.76	2.13	100
	1	0.1	1369.06	1374.30	5.24	100
	1	0.2	1347.48	1347.84	0.36	100
	1	0.3	1417.53	1418.46	0.93	100
	1	0.4	1519.11	1519.11	0.00	99
SR104	0	0	1233.09	1256.74	23.66	100
	0.2	0	1226.28	1251.59	25.31	100
	0.4	0	1249.57	1257.82	8.25	100
	0.6	0	1253.91	1266.81	12.89	100
	0.8	0	1309.81	1316.83	7.01	100
	1	0	1233.74	1239.04	5.30	100
	1	0.1	1213.58	1217.33	3.75	100
	1	0.2	1215.17	1218.20	3.03	100
	1	0.3	1289.93	1291.41	1.48	100
	1	0.4	1334.40	1334.40	0.00	99
SR109	0	0	1413.45	1443.65	30.21	100
	0.2	0	1394.95	1426.87	31.92	100

	0.4	0	1430.72	1453.90	23.19	100
	0.6	0	1391.33	1405.13	13.80	100
	0.8	0	1367.09	1378.30	11.21	100
	1	0	1351.19	1367.06	15.87	100
	1	0.1	1347.51	1355.40	7.89	100
	1	0.2	1392.64	1395.39	2.74	100
	1	0.3	1426.40	1428.70	2.31	100
	1	0.4	1428.87	1429.43	0.57	99

Tableau A. 10 L'ensemble des solutions du modèle SPR correspond aux combinaisons des seuils pour les problèmes du type SR1

Problème	Combinaison de seuils		Solution du modèle SPR			
	α	β	f_{DR}^{Plan}	f_{DR}^{Pre}	f_{DR}^{Su}	Nombre de clients planifiés
SR205	0	0	1225.71	1235.74	10.03	100
	0.2	0	1191.02	1193.53	2.51	100
	0.4	0	1202.96	1207.22	4.26	100
	0.6	0	1160.71	1165.42	4.70	100
	0.8	0	1176.52	1181.57	5.05	100
	1	0	1205.75	1208.53	2.78	100
	1	0.1	1199.73	1203.88	4.15	100
	1	0.2	1281.30	1281.30	0.00	99
SR206	0	0	1106.52	1124.53	18.01	100
	0.2	0	1049.67	1060.74	11.07	100
	0.4	0	1084.56	1087.39	2.83	100
	0.6	0	1123.36	1127.67	4.30	100
	0.8	0	1081.53	1087.86	6.33	100
	1	0	1118.31	1122.24	3.93	100
	1	0.1	1055.67	1059.29	3.62	100
	1	0.2	1139.14	1139.14	0.00	99
SR207	0	0	1003.51	1003.75	0.24	100
	0.2	0	983.24	983.81	0.57	100
	0.4	0	998.35	999.45	1.10	100
	0.6	0	994.13	994.45	0.32	100

	0.8	0	996.93	997.16	0.24	100
	1	0	995.32	995.90	0.57	100
	1	0.1	1008.05	1008.98	0.93	100
	1	0.2	1015.49	1015.49	0.00	99
SR208	0	0	1024.26	1027.50	3.24	100
	0.2	0	975.00	977.28	2.28	100
	0.4	0	957.78	963.12	5.35	100
	0.6	0	943.45	946.15	2.70	100
	0.8	0	945.37	945.37	0.00	100
	1	0	910.18	911.46	1.28	100
	1	0.1	1041.29	1043.76	2.47	100
	1	0.2	1035.95	1035.95	0.00	99

Tableau A. 11 L'ensemble des solutions du modèle SPR correspond aux combinaisons des seuils pour les problèmes du type SR2

Problème	Combinaison de seuils		Solution du modèle SPR			
	α	β	f_{DR}^{Plan}	f_{DR}^{Pre}	f_{DR}^{St}	Nombre de clients planifiés
SRC101	0	0	2011.51	2068.21	56.70	100
	0.2	0	1968.34	2052.23	83.89	100
	0.4	0	2008.19	2079.04	70.85	100
	0.6	0	1988.66	2029.72	41.05	100
	0.8	0	1934.65	1973.38	38.73	100
	1	0	1903.82	1932.58	28.76	100
	1	0.1	1912.59	1936.41	23.82	100
	1	0.2	1909.33	1929.02	19.68	100
	1	0.3	1943.43	1947.83	4.41	100
	1	0.4	1877.38	1877.38	0.00	99
SRC104	0	0	1453.32	1478.81	25.49	100
	0.2	0	1399.04	1430.52	31.47	100
	0.4	0	1397.49	1438.53	41.03	100
	0.6	0	1429.25	1445.43	16.19	100
	0.8	0	1420.77	1439.94	19.17	100
	1	0	1438.79	1448.43	9.64	100

	1	0.1	1383.95	1394.86	10.91	100
	1	0.2	1411.15	1420.96	9.81	100
	1	0.3	1433.36	1438.87	5.51	100
	1	0.4	1542.22	1542.22	0.00	99
SRC106	0	0	1699.35	1702.40	3.05	100
	0.2	0	1690.18	1712.49	22.31	100
	0.4	0	1687.63	1718.63	31.00	100
	0.6	0	1728.60	1761.64	33.04	100
	0.8	0	1727.33	1733.43	6.10	100
	1	0	1622.86	1663.32	40.46	100
	1	0.1	1631.95	1645.64	13.68	100
	1	0.2	1709.82	1719.96	10.14	100
	1	0.3	1688.97	1688.97	0.00	100
	1	0.4	1736.25	1736.25	0.00	99
SRC107	0	0	1501.67	1519.35	17.68	100
	0.2	0	1514.34	1543.14	28.81	100
	0.4	0	1540.90	1566.16	25.26	100
	0.6	0	1485.57	1502.92	17.36	100
	0.8	0	1480.66	1498.80	18.13	100
	1	0	1450.01	1459.15	9.14	100
	1	0.1	1425.94	1435.24	9.29	100
	1	0.2	1459.31	1465.99	6.68	100
	1	0.3	1481.05	1483.69	2.64	100
	1	0.4	1515.03	1515.03	0.00	99

Tableau A. 12 L'ensemble des solutions du modèle SPR correspond aux combinaisons des seuils pour les problèmes du type SRC1

Problème	Combinaison de seuils		Solution du modèle SPR			
	α	β	f_{DR}^{Plan}	f_{DR}^{Pre}	f_{DR}^{St}	Nombre de clients planifiés
SRC202	0	0	1383.21	1396.61	13.41	100
	0.2	0	1362.23	1366.37	4.14	100
	0.4	0	1419.33	1421.49	2.16	100
	0.6	0	1316.77	1318.79	2.02	100

	0.8	0	1326.43	1328.43	2	100
	1	0	1339.03	1340.91	1.88	100
	1	0.1	1402.57	1403	0.43	100
	1	0.2	1458.33	1460.63	2.30	99
SRC204	0	0	1165.83	1169.94	4.12	100
	0.2	0	1126.34	1132.75	6.40	100
	0.4	0	1096.56	1101.21	4.65	100
	0.6	0	1095.44	1102.65	7.20	100
	0.8	0	1093.92	1099.25	5.33	100
	1	0	1110.62	1115.04	4.42	100
	1	0.1	1157.07	1159.53	2.45	100
	1	0.2	1177.47	1177.89	0.42	99
SRC205	0	0	1497.59	1505.64	8.04	100
	0.2	0	1506.38	1512.56	6.18	100
	0.4	0	1519.20	1521.90	2.69	100
	0.6	0	1493.67	1496.31	2.63	100
	0.8	0	1486.00	1490.05	4.05	100
	1	0	1532.06	1537.26	5.20	100
	1	0.1	1596.28	1596.57	0.28	100
	1	0.2	1610.01	1610.01	0.00	99
SRC206	0	0	1503.14	1508.79	5.65	100
	0.2	0	1489.56	1498.04	8.48	100
	0.4	0	1382.68	1387.12	4.44	100
	0.6	0	1383.47	1389.14	5.67	100
	0.8	0	1423.37	1430.24	6.87	100
	1	0	1488.61	1493.14	4.52	100
	1	0.1	1462.76	1464.89	2.13	100
	1	0.2	1603.53	1603.53	0.00	99

Tableau A. 13 L'ensemble des solutions du modèle SPR correspond aux combinaisons des seuils pour les problèmes du type SRC2

A.4 Comparaison avec la version déterministe

Problème		VRPTW	VRPTWFD					
		Réel de référence	Solution du SPR		Solution du CCP		Solution du Nor	
	scénario « test »	f_{DR}^{Det}	f_{DR}^{Reel}	t^{SPR}	f_{DR}^{Reel}	t^{CCP}	f_{DR}^{Reel}	t^{Nor}
SC101	test 1	1005.89	961.05	-4.46%	961.05	-4.46%	961.05	-4.46%
	test 2	1042.83	1145.48	9.84%	1145.48	9.84%	1145.48	9.84%
	test 3	932.91	959.72	2.87%	959.72	2.87%	959.72	2.87%
	test 4	828.94	828.94	0.00%	828.94	0.00%	828.94	0.00%
	Moyenne			2.06%		2.06%		2.06%
SC105	test 1	885.95	912.04	2.94%	970.27	9.52%	975.15	10.07%
	test 2	906.00	969.78	7.04%	970.27	7.09%	1032.89	14.01%
	test 3	904.96	912.04	0.78%	906.63	0.18%	876.37	-3.16%
	test 4	885.95	864.61	-2.41%	906.63	2.33%	927.72	4.71%
	Moyenne			2.09%		4.78%		6.41%
SC108	test 1	899.54	914.18	1.63%	916.58	1.89%	1031.27	14.64%
	test 2	895.40	914.18	2.10%	916.58	2.37%	1031.27	15.17%
	test 3	850.06	887.89	4.45%	916.58	7.83%	876.37	3.10%
	test 4	945.17	914.18	-3.28%	929.31	-1.68%	1031.27	9.11%
	Moyenne			1.22%		2.60%		10.51%
SC109	test 1	843.39	901.65	6.91%	896.7	6.32%	828.94	-1.71%
	test 2	895.14	901.65	0.73%	910.22	1.68%	919.49	2.72%
	test 3	951.60	1057.93	11.17%	1058.2	11.20%	1130.89	18.84%
	test 4	893.65	901.65	0.90%	910.22	1.85%	898.83	0.58%
	Moyenne			4.93%		5.27%		5.11%

Tableau A. 14 Comparaison des solutions fournies par les différents modèles avec la version déterministe pour les problèmes du type SC1

Problème		VRPTW	VRPTWFD					
		Réel de référence	Solution du SPR			Solution du CCP		Solution du Nor
	scénario « test »	f_{DR}^{Det}	f_{DR}^{Reel}	t^{SPR}	f_{DR}^{Reel}	t^{CCP}	f_{DR}^{Reel}	t^{Nor}
SC201	test 1	591.56	636.72	7.63%	636.72	7.63%	603.88	2.08%
	test 2	633.18	636.72	0.56%	636.72	0.56%	705.55	11.43%
	test 3	636.72	636.72	0.00%	636.72	0.00%	705.77	10.84%
	test 4	670.52	744.7	11.06%	744.7	11.06%	734.78	9.58%
	Moyenne			4.81%		4.81%		8.49%
SC205	test 1	618.586	630.12	1.86%	630.12	1.86%	700.19	13.19%
	test 2	588.88	630.12	7.00%	630.12	7.00%	601.2	2.09%
	test 3	608.89	665.11	9.23%	665.11	9.23%	636.18	4.48%
	test 4	630.7	630.12	-0.09%	630.12	-0.09%	700.19	11.02%
	Moyenne			4.50%		4.50%		7.70%
SC207	test 1	592.38	630.32	6.40%	630.32	6.40%	621.85	4.97%
	test 2	593.4	630.32	6.22%	630.32	6.22%	621.85	4.79%
	test 3	605.72	630.32	4.06%	630.32	4.06%	621.85	2.66%
	test 4	605.72	630.32	4.06%	630.32	4.06%	621.85	2.66%
	Moyenne			5.19%		5.19%		3.77%
SC208	test 1	615.42	629.57	2.30%	630.08	2.38%	699.64	13.68%
	test 2	626.39	629.57	0.51%	630.08	0.59%	699.64	11.69%
	test 3	588.49	629.57	6.98%	630.08	7.07%	600.65	2.07%
	test 4	600.82	629.57	4.79%	630.08	4.87%	600.65	-0.03%
	Moyenne			3.64%		3.73%		6.85%

Tableau A. 15 Comparaison des solutions fournies par les différents modèles avec la version déterministe pour les problèmes du type SC2

Problème		VRPTW	VRPTWFD					
		Réel de référence	Solution du SPR			Solution du CCP		Solution du Nor
	scénario « test »	f_{DR}^{Det}	f_{DR}^{Reel}	t^{SPR}	f_{DR}^{Reel}	t^{CCP}	f_{DR}^{Reel}	t^{Nor}
SR101	test 1	1746.26	1772.71	1.51%	1818.35	4.13%	1961.21	12.31%
	test 2	1728.09	1756.94	1.67%	1796.06	3.93%	1891.07	9.43%
	test 3	1719.27	1756.94	2.19%	1881.91	9.46%	1942.31	12.97%

	test 4	1716.87	1756.94	2.33%	1774.39	3.35%	1821.89	6.12%
	Moyenne			1.93%		5.22%		10.21%
SR103	test 1	1325.64	1326.63	0.07%	1332.07	0.49%	1360.32	2.62%
	test 2	1330.8	1326.63	-0.31%	1316.3	-1.09%	1372.5	3.13%
	test 3	1339.83	1326.63	-0.99%	1355.31	1.16%	1403.21	4.73%
	test 4	1334.97	1350.9	1.19%	1342.35	0.55%	1427.94	6.96%
	Moyenne			-0.01%		0.28%		4.36%
SR104	test 1	1229.18	1229.19	0.00%	1364.59	11.02%	1339.11	8.94%
	test 2	1234.93	1238.48	0.29%	1229.59	-0.43%	1301.3	5.37%
	test 3	1203.26	1213.58	0.86%	1278.26	6.23%	1239.23	2.99%
	test 4	1233.95	1220.37	-1.10%	1252.21	1.48%	1270.25	2.94%
	Moyenne			0.01%		4.57%		5.06%
SR109	test 1	1347.13	1347.51	0.03%	1368.05	1.55%	1364.55	1.29%
	test 2	1357.94	1357.52	-0.03%	1440.5	6.08%	1496.11	10.17%
	test 3	1352.38	1357.52	0.38%	1418.96	4.92%	1429.42	5.70%
	test 4	1335.94	1347.51	0.87%	1368.05	2.40%	1418.93	6.21%
	Moyenne			0.31%		3.74%		5.84%

Tableau A. 16 Comparaison des solutions fournies par les différents modèles avec la version déterministe pour les problèmes du type SR1

Problème	scenario « test »	VRPTW	VRPTWFD					
		Réel de référence	Solution du SPR			Solution du CCP		Solution du Nor
		f_{DR}^{Det}	f_{DR}^{Reel}	t^{SPR}	f_{DR}^{Reel}	t^{CCP}	f_{DR}^{Reel}	t^{Nor}
SR205	test 1	1137.24	1160.71	2.06%	1187.65	4.43%	1174.69	3.29%
	test 2	1149.61	1160.71	0.97%	1177.62	2.44%	1171.1	1.87%
	test 3	1169.81	1174.29	0.38%	1190.3	1.75%	1181.81	1.03%
	test 4	1185.28	1181.56	-0.31%	1192.05	0.57%	1186.67	0.12%
	Moyenne			0.78%		2.30%		1.58%
SR206	test 1	1050.24	1055.67	0.52%	1080.77	2.91%	1056.51	0.60%
	test 2	1054.86	1055.67	0.08%	1072.61	1.68%	1056.51	0.16%
	test 3	1054.44	1062.46	0.76%	1092.78	3.64%	1056.51	0.20%
	test 4	1066.23	1069.25	0.28%	1081.83	1.46%	1073.85	0.71%
	Moyenne			0.41%		2.42%		0.42%

SR207	test 1	978.13	983.24	0.52%	989.2	1.13%	1011.34	3.40%
	test 2	969.65	983.24	1.40%	998.89	3.02%	990.19	2.12%
	test 3	958.92	989.99	3.24%	985.62	2.78%	990.19	3.26%
	test 4	1035.26	1029.2	-0.59%	1025.82	-0.91%	1044.78	0.92%
	Moyenne			1.14%		1.50%		2.42%
SR208	test 1	882.55	910.18	3.13%	1008.65	14.29%	953.13	8.00%
	test 2	882.55	910.18	3.13%	989.27	12.09%	953.13	8.00%
	test 3	883.29	913.93	3.47%	1014.2	14.82%	980.2	10.97%
	test 4	936.73	937.74	0.11%	1011.67	8.00%	1024.01	9.32%
	Moyenne			2.46%		12.30%		9.07%

Tableau A. 17 Comparaison des solutions fournies par les différents modèles avec la version déterministe pour les problèmes du type SR2

Problème	scénario « test »	VRPTW	VRPTWFD					
		Réel de référence	Solution du SPR			Solution du CCP		Solution du Nor
		f_{DR}^{Det}	f_{DR}^{Reel}	t_{SPR}	f_{DR}^{Reel}	t_{CCP}	f_{DR}^{Reel}	t_{Nor}
SRC101	test 1	1862.12	1909.33	2.54%	1901.24	2.10%	1840.02	-1.19%
	test 2	1854.52	1909.33	2.96%	1951.22	5.21%	1912.04	3.10%
	test 3	1896.37	1933.89	1.98%	1901.24	0.26%	1951.59	2.91%
	test 4	1866.32	1909.33	2.30%	2071.47	10.99%	1985.13	6.37%
	Moyenne			2.45%	1956.29	4.64%	1922.20	2.80%
SRC104	test 1	1336.47	1383.95	3.55%	1389.72	3.98%	1406.02	5.20%
	test 2	1328.4	1387.25	4.43%	1362.77	2.59%	1534.15	15.49%
	test 3	1300.08	1387.25	6.70%	1375.22	5.78%	1457.09	12.08%
	test 4	1395.3	1428.12	2.35%	1468.41	5.24%	1744.77	25.05%
	Moyenne			4.26%		4.40%		14.45%
SRC106	test 1	1650.01	1711.01	3.70%	1671.35	1.29%	1779.43	7.84%
	test 2	1611.72	1631.95	1.26%	1627	0.95%	1651.97	2.50%
	test 3	1610.11	1631.95	1.36%	1627	1.05%	1694.31	5.23%
	test 4	1631.76	1631.95	0.01%	1713.1	4.98%	1817.47	11.38%
	Moyenne			1.58%		2.07%		6.74%
SRC107	test 1	1460.75	1477.7	1.16%	1501.57	2.79%	1530.22	4.76%

	test 2	1413.78	1425.94	0.86%	1468.4	3.86%	1544.76	9.26%
	test 3	1411.02	1425.94	1.06%	1468.4	4.07%	1507.86	6.86%
	test 4	1420.93	1488.71	4.77%	1518.38	6.86%	1548.11	8.95%
	Moyenne			1.96%		4.40%		7.46%

Tableau A. 18 Comparaison des solutions fournies par les différents modèles avec la version déterministe pour les problèmes du type SRC1

Problème	scénario « test »	VRPTW	VRPTWFD					
		Réel de référence	Solution du SPR		Solution du CCP		Solution du Nor	
		f_{DR}^{Det}	f_{DR}^{Reel}	t^{SPR}	f_{DR}^{Reel}	t^{CCP}	f_{DR}^{Reel}	t^{Nor}
SRC202	test 1	1346.31	1347.05	0.05%	1356.89	0.79%	1446.77	7.46%
	test 2	1332.33	1341.33	0.68%	1352.32	1.50%	1385.81	4.01%
	test 3	1326.57	1341.33	1.11%	1352.32	1.94%	1374.27	3.60%
	test 4	1332.23	1355.76	1.77%	1352.32	1.51%	1434.52	7.68%
	Moyenne			0.90%		1.43%		5.69%
SRC204	test 1	1089.44	1097.73	0.76%	1126.13	3.37%	1109.53	1.84%
	test 2	1074.96	1093.92	1.76%	1091.85	1.57%	1120.11	4.20%
	test 3	1067.51	1093.92	2.47%	1092.98	2.39%	1103.53	3.37%
	test 4	1047.19	1097.73	4.83%	1116.35	6.60%	1103.53	5.38%
	Moyenne			2.46%		3.48%		3.70%
SRC205	test 1	1484.3	1486	0.11%	1533.28	3.30%	1543.09	3.96%
	test 2	1483.88	1486	0.14%	1588.11	7.02%	1490.85	0.47%
	test 3	1482.79	1486	0.22%	1560.04	5.21%	1512.81	2.02%
	test 4	1481.53	1500.3	1.27%	1532.3	3.43%	1510.24	1.94%
	Moyenne			0.44%		4.74%		2.10%
SRC206	test 1	1439.35	1388.39	-3.54%	1511.07	4.98%	1497.64	4.05%
	test 2	1381.26	1382.68	0.10%	1422.25	2.97%	1480.42	7.18%
	test 3	1374.16	1382.68	0.62%	1497.31	8.96%	1480.41	7.73%
	test 4	1370.97	1394.1	1.69%	1421.94	3.72%	1480.41	7.98%
	Moyenne			-0.28%		5.16%		6.74%

Tableau A. 19 Comparaison des solutions fournies par les différents modèles avec la version déterministe pour les problèmes du type SRC2

A.5 Evaluation de la solution du modèle SPR

Problème	Solution du modèle SPR pour VRPTWFD			scénario « test »	Evaluation de solutions				
	f_{DR}^{Plan}	f_{DR}^{Prev}	f_{DR}^{Supp}		f_{DR}^{Reel}	f_{DR}^{SR}	t^{SR}	D^{RP}	t^{RP}
SC101	828.94	986.06	165.98	test 1	961.05	132.11	13.75%	-25.01	-2.60%
				test 2	1145.48	316.54	27.63%	159.42	13.92%
				test 3	959.72	130.78	13.63%	-26.34	-2.74%
				test 4	828.94	0.00	0.00%	-157.12	-18.95%
				Moyenne			13.75%		-2.59%
SC105	864.61	917.50	52.89	test 1	912.04	47.43	5.20%	-5.46	-0.60%
				test 2	969.78	105.17	10.84%	52.28	5.39%
				test 3	912.04	47.43	5.20%	-5.46	-0.60%
				test 4	864.61	0.00	0.00%	-52.89	-6.12%
				Moyenne			5.31%		-0.48%
SC108	887.89	912.64	24.75	test 1	914.18	26.29	2.88%	1.54	0.17%
				test 2	914.18	26.29	2.88%	1.54	0.17%
				test 3	887.89	0.00	0.00%	-24.75	-2.79%
				test 4	914.18	26.29	2.88%	1.54	0.17%
				Moyenne			2.16%		-0.57%
SC109	901.65	916.7	15.05	test 1	901.65	0.00	0.00%	-15.05	-1.67%
				test 2	901.65	0.00	0.00%	-15.05	-1.67%
				test 3	1057.93	156.28	14.77%	141.23	13.35%
				test 4	901.65	0.00	0.00%	-15.05	-1.67%
				Moyenne			3.69%		2.09%

Tableau A. 20 L'évaluation de la solution du modèle SPR pour les problèmes du type SC1

Problème	Solution du modèle SPR pour VRPTWFD			scénario « test »	Evaluation de solutions				
	f_{DR}^{Plan}	f_{DR}^{Prev}	f_{DR}^{Supp}		f_{DR}^{Reel}	f_{DR}^{SR}	t^{SR}	D^{RP}	t^{RP}
SC201	636.72	656.56	19.84	test 1	636.72	0	0.00%	-19.84	-3.12%
				test 2	636.72	0	0.00%	-19.84	-3.12%

				test 3	636.72	0	0.00%	-19.84	-3.12%
				test 4	744.7	107.98	14.50%	88.14	11.84%
				Moyenne			3.62%		0.62%
SC205	630.12	633.62	3.5	test 1	630.12	0	0.00%	-3.5	-0.56%
				test 2	630.12	0	0.00%	-3.5	-0.56%
				test 3	665.11	34.99	5.26%	31.49	4.73%
				test 4	630.12	0	0.00%	-3.5	-0.56%
				Moyenne			1.32%		0.76%
SC207	630.32	637.62	7.3	test 1	630.32	0	0.00%	0	0.00%
				test 2	630.32	0	0.00%	0	0.00%
				test 3	630.32	0	0.00%	0	0.00%
				test 4	630.32	0	0.00%	0	0.00%
				Moyenne			0.00%		0.00%
SC208	629.57	629.57	0	test 1	629.57	0	0.00%	0	0.00%
				test 2	629.57	0	0.00%	0	0.00%
				test 3	629.57	0	0.00%	0	0.00%
				test 4	629.57	0	0.00%	0	0.00%
				Moyenne			0.00%		0.00%

Tableau A. 21 L'évaluation de la solution du modèle SPR pour les problèmes du type SC2

Problème	Solution du modèle SPR pour VRPTWFD			scénario « test »	Evaluation de solutions				
	f_{DR}^{Plan}	f_{DR}^{Prev}	f_{DR}^{Supp}		f_{DR}^{Reel}	f_{DR}^{SR}	t^{SR}	D^{RP}	t^{RP}
SR101	1756.94	1765.25	8.31	test 1	1772.71	15.77	0.89%	7.47	0.42%
				test 2	1756.94	0	0.00%	-8.31	-0.47%
				test 3	1756.94	0	0.00%	-8.31	-0.47%
				test 4	1756.94	0	0.00%	-8.31	-0.47%
				Moyenne			0.22%		-0.25%
SR103	1326.63	1328.76	2.13	test 1	1326.63	0	0.00%	-2.13	-0.16%
				test 2	1326.63	0	0.00%	-2.13	-0.16%
				test 3	1326.63	0	0.00%	-2.13	-0.16%
				test 4	1350.9	24.27	1.80%	22.14	1.64%
				Moyenne			0.45%		0.29%
SR104	1213.58	1217.33	3.75	test 1	1229.19	15.61	1.27%	11.86	0.96%

				test 2	1238.48	24.9	2.01%	21.15	1.71%
				test 3	1213.58	0	0.00%	-3.75	-0.31%
				test 4	1220.37	6.79	0.56%	3.04	0.25%
				Moyenne			0.96%		0.65%
SR109	1347.51	1355.4	7.89	test 1	1347.51	0	0.00%	-7.89	-0.59%
				test 2	1357.52	10.01	0.74%	2.12	0.16%
				test 3	1357.52	10.01	0.74%	2.12	0.16%
				test 4	1347.51	0	0.00%	-7.89	-0.59%
				Moyenne			0.37%		-0.22%

Tableau A. 22 L'évaluation de la solution du modèle SPR pour les problèmes du type SR1

Problème	Solution du modèle SPR pour VRPTWFD			scénario « test »	Evaluation de solutions				
	f_{DR}^{Plan}	f_{DR}^{Prev}	f_{DR}^{Supp}		f_{DR}^{Reel}	f_{DR}^{SR}	t^{SR}	D^{RP}	t^{RP}
SR205	1160.71	1165.42	4.7	test 1	1160.71	0	0.00%	-4.7	-0.40%
				test 2	1160.71	0	0.00%	-4.7	-0.40%
				test 3	1174.29	13.58	1.16%	8.88	0.76%
				test 4	1181.56	20.85	1.76%	16.15	1.37%
				Moyenne			0.73%		0.33%
SR206	1055.67	1059.29	3.62	test 1	1055.67	0	0.00%	-3.62	-0.34%
				test 2	1055.67	0	0.00%	-3.62	-0.34%
				test 3	1062.46	6.79	0.64%	3.17	0.30%
				test 4	1069.25	13.58	1.27%	9.97	0.93%
				Moyenne			0.48%		0.14%
SR207	983.24	983.81	0.57	test 1	983.24	0	0.00%	-0.57	-0.06%
				test 2	983.24	0	0.00%	-0.57	-0.06%
				test 3	989.99	6.75	0.68%	6.18	0.62%
				test 4	1029.2	45.96	4.47%	45.38	4.41%
				Moyenne			1.29%		1.23%
SR208	910.18	911.46	1.28	test 1	910.18	0	0.00%	-1.28	-0.14%
				test 2	910.18	0	0.00%	-1.28	-0.14%
				test 3	913.93	3.75	0.41%	2.47	0.27%
				test 4	937.74	27.56	2.94%	26.28	2.80%
				Moyenne			0.84%		0.70%

Tableau A. 23 L'évaluation de la solution du modèle SPR pour les problèmes du type SR2

Problème	Solution du modèle SPR pour VRPTWFD			scénario « test »	Evaluation de solutions				
	f_{DR}^{Plan}	f_{DR}^{Prev}	f_{DR}^{Supp}		f_{DR}^{Reel}	f_{DR}^{SR}	t^{SR}	D^{RP}	t^{RP}
SRC101	1909.33	1929.02	19.68	test 1	1909.33	0	0.00%	-19.68	-1.03%
				test 2	1909.33	0	0.00%	-19.68	-1.03%
				test 3	1933.89	24.56	1.27%	4.88	0.25%
				test 4	1909.33	0	0.00%	-19.68	-1.03%
				Moyenne			0.32%		-0.71%
SRC104	1383.95	1394.86	10.91	test 1	1383.95	0	0.00%	-10.91	-0.79%
				test 2	1387.25	3.3	0.24%	-7.61	-0.55%
				test 3	1387.25	3.3	0.24%	-7.61	-0.55%
				test 4	1428.12	44.18	3.09%	33.27	2.33%
				Moyenne			0.89%		0.11%
SRC106	1631.95	1645.64	13.68	test 1	1711.01	79.06	4.62%	65.37	3.82%
				test 2	1631.95	0.00	0.00%	-13.69	-0.84%
				test 3	1631.95	0.00	0.00%	-13.69	-0.84%
				test 4	1631.95	0.00	0.00%	-13.69	-0.84%
				Moyenne			1.16%		0.33%
SRC107	1425.94	1435.24	9.29	test 1	1477.7	51.76	3.50%	42.47	2.87%
				test 2	1425.94	0	0.00%	-9.29	-0.65%
				test 3	1425.94	0	0.00%	-9.29	-0.65%
				test 4	1488.71	62.77	4.22%	53.47	3.59%
				Moyenne			1.93%		1.29%

Tableau A. 24 L'évaluation de la solution du modèle SPR pour les problèmes du type SRC1

Problème	Solution du modèle SPR pour VRPTWFD			scénario « test »	Evaluation de solutions				
	f_{DR}^{Plan}	f_{DR}^{Prev}	f_{DR}^{Supp}		f_{DR}^{Reel}	f_{DR}^{SR}	t^{SR}	D^{RP}	t^{RP}
SRC202	1316.77	1318.79	2.02	test 1	1347.05	30.28	2.25%	28.26	2.10%
				test 2	1341.33	24.56	1.83%	22.54	1.68%
				test 3	1341.33	24.56	1.83%	22.54	1.68%
				test 4	1355.76	38.99	2.88%	36.97	2.73%
				Moyenne			2.20%		2.05%
SRC204	1093.92	1099.25	5.33	test 1	1097.73	3.81	0.35%	-1.52	-0.14%

				test 2	1093.92	0	0.00%	-5.33	-0.49%
				test 3	1093.92	0	0.00%	-5.33	-0.49%
				test 4	1097.73	3.81	0.35%	-1.52	-0.14%
				Moyenne			0.18%		-0.32%
SRC205	1486.00	1490.05	4.05	test 1	1486	0	0.00%	-4.05	-0.27%
				test 2	1486	0	0.00%	-4.05	-0.27%
				test 3	1486	0	0.00%	-4.05	-0.27%
				test 4	1500.3	14.3	0.95%	10.25	0.68%
				Moyenne			0.24%		-0.03%
SRC206	1382.68	1387.12	4.44	test 1	1388.39	5.71	0.41%	1.27	0.09%
				test 2	1382.68	0	0.00%	-4.44	-0.32%
				test 3	1382.68	0	0.00%	-4.44	-0.32%
				test 4	1394.1	11.42	0.82%	6.98	0.50%
				Moyenne			0.31%		-0.01%

Tableau A. 25 L'évaluation de la solution du modèle SPR pour les problèmes du type SRC2

Annexe B

Résultats des problèmes de VRPTWFD dynamique

B.1 Etude comparative des modèles stochastiques

Problème	scénario « test »	« D&D »		VRPTWFD dynamique								
		f_{DR}	C_{NS}	Solution du modèle SPR			Solution du modèle CCP			Solution du modèle Nor		
				f_{DR}	C_{NS}	t^{SPR}	f_{DR}	C_{NS}	t^{CCP}	f_{DR}	C_{NS}	t^{Nor}
DSC101	test 1	1037.45	0	1054.22	0	1.62%	1037.45	0	0.00%	1037.45	0	0.00%
	test 2	907.5	0	907.5	0	0.00%	907.5	0	0.00%	907.5	0	0.00%
	test 3	833.24	0	833.24	0	0.00%	833.24	0	0.00%	833.24	0	0.00%
	test 4	828.94	0	828.94	0	0.00%	828.94	0	0.00%	828.94	0	0.00%
	Moyenne		0		0	0.40%		0	0.00%		0	0.00%
DSC105	test 1	828.94	0	828.94	0	0.00%	828.94	0	0.00%	828.94	0	0.00%
	test 2	906.77	0	970.15	0	6.99%	906.77	0	0.00%	959.56	0	5.82%
	test 3	931.85	0	951.91	0	2.15%	949.61	0	1.91%	947.32	0	1.66%
	test 4	975.12	0	990.27	0	1.55%	962.28	0	-1.32%	962.28	0	-1.32%
	Moyenne		0		0	2.67%		0	0.15%		0	1.54%
DSC108	test 1	930.19	0	935.29	0	0.55%	930.19	0	0.00%	963.46	0	3.58%
	test 2	995.29	0	967.13	0	-2.83%	962.03	0	-3.34%	995.29	0	0.00%
	test 3	991.64	0	1038.23	0	4.70%	1024.98	0	3.36%	1006.23	0	1.47%
	test 4	936.41	0	936.61	0	0.02%	936.61	0	0.02%	936.61	0	0.02%
	Moyenne		0		0	0.61%		0	0.01%		0	1.27%

Tableau B. 1 Comparaison des solutions fournies par trois modèles avec la version « Déterministe et Dynamique » pour les problèmes du type DSC1

Problème	scénario « test »	« D&D »		VRPTWFD dynamique								
		f_{DR}	C_{NS}	Solution du modèle SPR			Solution du modèle CCP			Solution du modèle Nor		
		f_{DR}	C_{NS}	f_{DR}	C_{NS}	t^{SPR}	f_{DR}	C_{NS}	t^{CCP}	f_{DR}	C_{NS}	t^{Nor}
DSC201	test 1	591.56	0	621.51	0	5.06%	591.56	0	0.00%	591.56	0	0.00%
	test 2	633.18	0	637.01	0	0.60%	633.18	0	0.00%	633.18	0	0.00%
	test 3	636.72	0	637.01	0	0.05%	638.48	0	0.28%	638.48	0	0.28%
	test 4	670.52	0	675.95	0	0.81%	670.52	0	0.00%	670.52	0	0.00%
	Moyenne		0		0	1.63%		0	0.07%		0	0.07%
DSC205	test 1	617.61	0	625.31	0	1.25%	617.61	0	0.00%	617.61	0	0.00%
	test 2	608.89	0	644.33	0	5.82%	608.89	0	0.00%	608.89	0	0.00%
	test 3	630.702	0	638.16	0	1.18%	630.73	0	0.00%	630.73	0	0.00%
	test 4	601.2	0	640.04	0	6.46%	601.2	0	0.00%	601.2	0	0.00%
	Moyenne		0		0	3.68%		0	0.00%		0	0.00%
DSC207	test 1	593.4	0	648	0	9.20%	593.4	0	0.00%	593.4	0	0.00%
	test 2	605.72	0	639.56	0	5.59%	605.72	0	0.00%	605.72	0	0.00%
	test 3	605.72	0	626.97	0	3.51%	605.72	0	0.00%	605.72	0	0.00%
	test 4	605.72	0	640.24	0	5.70%	635.28	0	4.88%	635.28	0	4.88%
	Moyenne		0		0	6.00%		0	1.22%		0	1.22%

Tableau B. 2 Comparaison des solutions fournies par trois modèles avec la version « Déterministe et Dynamique » pour les problèmes du type DSC2

Problème	scénario « test »	« D&D »		VRPTWFD dynamique								
		f_{DR}	C_{NS}	Solution du modèle SPR			Solution du modèle CCP			Solution du modèle Nor		
		f_{DR}	C_{NS}	f_{DR}	C_{NS}	t^{SPR}	f_{DR}	C_{NS}	t^{CCP}	f_{DR}	C_{NS}	t^{Nor}
DSR101	test 1	1761.53	1.2	1825.58	0	3.64%	1839.21	0.6	4.41%	1753.4	1.4	-0.46%
	test 2	1768.3	0.6	1841.05	1.4	5.62%	1774.66	1.4	0.36%	1804.12	0.4	2.03%
	test 3	1783.11	0.4	1803.26	2	1.13%	1807.04	0.8	1.34%	1781.74	1	-0.08%
	test 4	1787.32	0.2	1823.92	1	2.05%	1773.77	0.6	-0.76%	1830.37	1	2.41%
	Moyenne		0.6		1.1	3.11%		0.85	1.34%		0.95	0.97%
DSR104	test 1	1151.33	0.4	1195.88	0.2	3.87%	1184.57	0	2.89%	1169.34	0.2	1.56%
	test 2	1153.05	0.2	1174.95	0.2	1.90%	1195.58	0.2	3.69%	1181.32	0.6	2.45%
	test 3	1169.2	0.6	1226.13	0.6	4.87%	1185.29	0.6	1.38%	1171.94	0	0.23%

	test 4	1175.6	0.6	1250.34	0.6	6.36%	1225.83	0.6	4.27%	1193.51	0.2	1.52%
	Moyenne		0.45		0.4	4.25%		0.35	3.06%		0.25	1.44%
DSR109	test 1	1455.31	0	1622.39	2.2	11.48%	1569.85	2.2	7.87%	1564.24	0.2	7.49%
	test 2	1489.62	0	1588.79	1.6	6.66%	1544.58	1.2	3.69%	1485.1	3.6	-0.30%
	test 3	1534.6	0.2	1603.2	0.2	4.47%	1589.86	0.2	3.60%	1518.58	0.4	-1.04%
	test 4	1559.62	0	1524.86	1.8	-2.23%	1499	0	-3.89%	1584.95	0	1.62%
	Moyenne		0.05		1.45	5.10%		0.9	2.82%		1.05	1.94%

Tableau B. 3 Comparaison des solutions fournies par trois modèles avec la version « Déterministe et Dynamique » pour les problèmes du type DSR1

Problème	scénario « test »	« D&D »		VRPTWFD dynamique								
		f_{DR}	C_{NS}	Solution du modèle SPR			Solution du modèle CCP			Solution du modèle Nor		
				f_{DR}	C_{NS}	t^{SPR}	f_{DR}	C_{NS}	t^{CCP}	f_{DR}	C_{NS}	t^{Nor}
DSR205	test 1	1143.17	0	1206.26	0	5.52%	1167.67	0	2.14%	1205.77	0	5.48%
	test 2	1185.71	0	1190.16	0	0.38%	1186.48	0	0.07%	1188.24	0.2	0.21%
	test 3	1127.97	0.4	1189.55	0	5.46%	1167.7	0	3.52%	1196.47	0.2	6.07%
	test 4	1184.93	0	1189.39	0.4	0.38%	1168.86	0.6	-1.36%	1195.66	0.4	0.91%
	Moyenne		0.1		0.1	2.93%		0.15	1.09%		0.2	3.17%
DSR207	test 1	1030.92	0	1024.14	0	-0.66%	1034.49	1	0.35%	1088.9	0.8	5.62%
	test 2	1011.6	0	1024.53	0	1.28%	1031.5	0	1.97%	1056.04	0	4.39%
	test 3	985.02	0	1117.79	0.4	13.48%	996.99	0.2	1.21%	1047.82	1	6.38%
	test 4	979.81	0	1042.19	0.2	6.37%	1031.02	0.2	5.23%	1033.81	0.2	5.51%
	Moyenne		0		0.15	5.12%		0.35	2.19%		0.5	5.48%
DSR208	test 1	860.02	0	888.57	0	3.32%	919.01	0.2	6.86%	988.25	0	14.91%
	test 2	905.54	0.2	921.24	0.6	1.73%	925.96	0.6	2.26%	1005.6	0.6	11.05%
	test 3	899.05	0	916.01	0.2	1.89%	898.72	0	-0.04%	837.65	0	-6.83%
	test 4	874.87	0.2	1003.74	1	14.73%	945.72	0.6	8.10%	920.83	0.2	5.25%
	Moyenne		0.1		0.45	5.42%		0.35	4.29%		0.2	6.10%

Tableau B. 4 Comparaison des solutions fournies par trois modèles avec la version « Déterministe et Dynamique » pour les problèmes du type DSR2

Problème	scénario « test »	« D&D »		VRPTWFD dynamique								
		f_{DR}	C_{NS}	Solution du modèle SPR			Solution du modèle CCP			Solution du modèle Nor		
		f_{DR}	C_{NS}	f_{DR}	C_{NS}	t^{SPR}	f_{DR}	C_{NS}	t^{CCP}	f_{DR}	C_{NS}	t^{Nor}
DSRC101	test 1	2055.04	0.2	2164.28	0	5.32%	2168.17	0.2	5.50%	2152	0	4.72%
	test 2	2056.5	0	2189.9	0.6	6.49%	2103	0	2.26%	2178.69	0	5.94%
	test 3	2070.65	0	2120.84	0.2	2.42%	2171.24	0	4.86%	2179.28	0.2	5.25%
	test 4	2149.09	0.4	2160.71	0.8	0.54%	2250.71	0.2	4.73%	2230.07	0.8	3.77%
	Moyenne		0.15		0.4	3.69%		0.1	4.34%		0.25	4.92%
DSRC104	test 1	1365.43	0	1361.57	0.2	-0.28%	1420.62	0.4	4.04%	1434.87	0.2	5.09%
	test 2	1343.77	0.4	1424.44	0.2	6.00%	1419.78	0.4	5.66%	1393.8	0.8	3.72%
	test 3	1363.91	0	1421.13	0.2	4.20%	1447.75	0.2	6.15%	1422.2	0.2	4.27%
	test 4	1379.16	0.4	1421.21	0.2	3.05%	1402.93	0.4	1.72%	1399.81	0.2	1.50%
	Moyenne		0.2		0.2	3.24%		0.35	4.39%		0.35	3.64%
DSRC107	test 1	1558.83	0.2	1719.26	0.6	10.29%	1580.73	0.2	1.41%	1602.41	0.4	2.80%
	test 2	1567.75	0	1650.05	0.4	5.25%	1709.65	0.6	9.05%	1736.58	0.4	10.77%
	test 3	1565.18	1	1660.24	0.6	6.07%	1660.51	0.2	6.09%	1706.13	0.4	9.01%
	test 4	1636.18	0	1593.72	0.2	-2.60%	1769.87	0.4	8.17%	1622.39	0.2	-0.84%
	Moyenne		0.3		0.45	4.76%		0.35	6.18%		0.35	5.43%

Tableau B. 5 Comparaison des solutions fournies par trois modèles avec la version « Déterministe et Dynamique » pour les problèmes du type DSRC1

Problème	scénario « test »	« D&D »		VRPTWFD dynamique								
		f_{DR}	C_{NS}	Solution du modèle SPR			Solution du modèle CCP			Solution du modèle Nor		
		f_{DR}	C_{NS}	f_{DR}	C_{NS}	t^{SPR}	f_{DR}	C_{NS}	t^{CCP}	f_{DR}	C_{NS}	t^{Nor}
DSRC202	test 1	1354.85	0	1365.72	0	0.80%	1437.25	0	6.08%	1417.16	0	4.60%
	test 2	1355.72	0.2	1456.22	0	7.41%	1443.08	0.2	6.44%	1417.98	0	4.59%
	test 3	1348.53	0	1437.98	0	6.63%	1437.96	0	6.63%	1404.1	0	4.12%
	test 4	1296.36	0	1355.51	0	4.56%	1445.4	0.4	11.50%	1447.97	0	11.69%
	Moyenne		0.05		0	4.85%		0.15	7.66%		0	6.25%
DSRC204	test 1	1067.13	0	1038.43	0.2	-2.69%	1063.3	0	-0.36%	1120.75	0.6	5.02%
	test 2	1034.18	0.2	1074.94	0	3.94%	1051.87	0.2	1.71%	1076.69	0	4.11%
	test 3	1065.62	0	1068.19	0	0.24%	1169.26	0.2	9.73%	1083.94	0.4	1.72%
	test 4	1022.94	0	1055.64	0	3.20%	1067.07	0.2	4.31%	1062.19	0.8	3.84%
	Moyenne		0.05		0.05	1.17%		0.15	3.85%		0.45	3.67%

DSRC205	test 1	1529.22	0	1559.36	0	1.97%	1596.53	0	4.40%	1590.48	0	4.01%
	test 2	1589.75	0	1653.46	0	4.01%	1662.27	0	4.56%	1610.59	0	1.31%
	test 3	1515.71	0	1587.5	0	4.74%	1599.82	0	5.55%	1624.9	0	7.20%
	test 4	1536.1	0	1607.76	0	4.66%	1580.49	0	2.89%	1591.97	0	3.64%
	Moyenne		0		0	3.85%		0	4.35%		0	4.04%

Tableau B. 6 Comparaison des solutions fournies par trois modèles avec la version « Déterministe et Dynamique » pour les problèmes du type DSRC2

B.2 Comparaison avec la version « Déterministe et Statique »

Problème	scénario « test »	« D&S »	VRPTWFD dynamique		
		f_{DR}	f_{DR}	C_{NS}	t^{SD}
DSC101	test 1	1037.45	1037.45	0	0.00%
	test 2	907.5	907.5	0	0.00%
	test 3	833.24	833.24	0	0.00%
	test 4	828.94	828.94	0	0.00%
	Moyenne			0	0.00%
DSC105	test 1	828.94	828.94	0	0.00%
	test 2	870.33	906.77	0	4.19%
	test 3	881.37	949.61	0	7.74%
	test 4	870.94	962.28	0	10.49%
	Moyenne			0	5.60%
DSC108	test 1	866.25	930.19	0	7.38%
	test 2	903.8	962.03	0	6.44%
	test 3	903.37	1024.98	0	13.46%
	test 4	894.69	936.61	0	4.69%
	Moyenne			0	7.99%

Tableau B. 7 Comparaison les meilleures solutions pour les problèmes du type DSC1 avec celles de la version « déterministe et statique » correspondante

Problème	scénario « test »	« D&S »	VRPTWFD dynamique		
		f_{DR}	f_{DR}	C_{NS}	t^{SD}
DSC201	test 1	591.56	591.56	0	0.00%
	test 2	633.18	633.18	0	0.00%
	test 3	636.72	638.48	0	0.28%
	test 4	670.52	670.52	0	0.00%
	Moyenne			0	0.07%
DSC205	test 1	603.48	617.61	0	2.34%
	test 2	608.89	608.89	0	0.00%
	test 3	630.12	630.73	0	0.10%
	test 4	601.2	601.2	0	0.00%
	Moyenne			0	0.61%
DSC207	test 1	610.42	593.4	0	-2.79%
	test 2	622.1	605.72	0	-2.63%
	test 3	622.74	605.72	0	-2.73%
	test 4	622.14	635.28	0	2.11%
	Moyenne			0	-1.51%

Tableau B. 8 Comparaison les meilleures solutions pour les problèmes du type DSC2 avec celles de la version « déterministe et statique » correspondante

Problème	scénario « test »	« D&S »	VRPTWFD dynamique		
		f_{DR}	f_{DR}	C_{NS}	t^{SD}
DSR101	test 1	1701.53	1753.4	1.4	3.05%
	test 2	1728.1	1804.12	0.4	4.40%
	test 3	1729.34	1781.74	1	3.03%
	test 4	1724.97	1830.37	1	6.11%
	Moyenne			0.95	4.15%
DSR104	test 1	1172.39	1169.34	0.2	-0.26%
	test 2	1150.64	1181.32	0.6	2.67%
	test 3	1159.1	1171.94	0	1.11%
	test 4	1174.98	1193.51	0.2	1.58%
	Moyenne			0.25	1.27%
DSR109	test 1	1333.04	1564.24	0.2	17.34%

	test 2	1337.23	1485.1	3.6	11.06%
	test 3	1329.22	1518.58	0.4	14.25%
	test 4	1336.63	1584.95	0	18.58%
	Moyenne			1.05	15.31%

Tableau B. 9 Comparaison les meilleures solutions pour les problèmes du type DSR1 avec celles de la version « déterministe et statique » correspondante

Problème	scénario « test »	« D&S »		VRPTWFD dynamique	
		f_{DR}	f_{DR}	C_{NS}	t^{SD}
DSR205	test 1	1160.09	1167.67	0	0.65%
	test 2	1151.46	1186.48	0	3.04%
	test 3	1121.3	1167.7	0	4.14%
	test 4	1152.39	1168.86	0.6	1.43%
	Moyenne			0.15	2.32%
DSR207	test 1	1012.5	1034.49	1	2.17%
	test 2	1014.5	1031.5	0	1.68%
	test 3	975	996.99	0.2	2.26%
	test 4	1029.06	1031.02	0.2	0.19%
	Moyenne			0.35	1.57%
DSR208	test 1	876.29	919.01	0.2	4.88%
	test 2	901.2	925.96	0.6	2.75%
	test 3	904.57	898.72	0	-0.65%
	test 4	985.83	945.72	0.6	-4.07%
	Moyenne			0.35	0.73%

Tableau B. 10 Comparaison les meilleures solutions pour les problèmes du type DSR2 avec celles de la version « déterministe et statique » correspondante

Problème	scénario « test »	« D&S »		VRPTWFD dynamique	
		f_{DR}	f_{DR}	C_{NS}	t^{SD}
DSRC101	test 1	1831.06	1365.72	0	18.20%
	test 2	1859.92	1456.22	0	17.74%
	test 3	1856.59	1437.98	0	14.23%
	test 4	1846.73	1355.51	0	17.00%
	Moyenne			0	16.79%

DSRC104	test 1	1339.43	1038.43	0.2	1.65%
	test 2	1357.25	1074.94	0	4.95%
	test 3	1352.21	1068.19	0	5.10%
	test 4	1389.7	1055.64	0	2.27%
	Moyenne			0.05	3.49%
DSRC107	test 1	1487.38	1559.36	0	15.59%
	test 2	1528.53	1653.46	0	7.95%
	test 3	1505.41	1587.5	0	10.28%
	test 4	1498.43	1607.76	0	6.36%
	Moyenne			0	10.05%

Tableau B. 11 Comparaison les meilleures solutions pour les problèmes du type DSRC1 avec celles de la version « déterministe et statique » correspondante

Problème	scénario « test »	« D&S »		VRPTWFD dynamique	
		f_{DR}	f_{DR}	C_{NS}	t^{SD}
DSRC202	test 1	1346.28	1365.72	0	1.44%
	test 2	1325.32	1456.22	0	9.88%
	test 3	1363.24	1437.98	0	5.48%
	test 4	1336.92	1355.51	0	1.39%
	Moyenne			0	4.55%
DSRC204	test 1	1067.5	1038.43	0.2	-2.72%
	test 2	1027.45	1074.94	0	4.62%
	test 3	1039.23	1068.19	0	2.79%
	test 4	1002.75	1055.64	0	5.27%
	Moyenne			0.05	2.49%
DSRC205	test 1	1510.34	1559.36	0	3.25%
	test 2	1509.43	1653.46	0	9.54%
	test 3	1508.23	1587.5	0	5.26%
	test 4	1521.77	1607.76	0	5.65%
	Moyenne			0	5.92%

Tableau B. 12 Comparaison les meilleures solutions pour les problèmes du type DSRC2 avec celles de la version « déterministe et statique » correspondante

Bibliographie

- [Alfa et al 1991] A.S. Alfa, S.S. Heragu and M. Chen. « A 3-opt based simulated annealing algorithm for vehicle routing problems ». *Computers & Industrial Engineering* 21, pages 635-639, 1991.
- [Archetti et al. 2006] C. Archetti, A. Hertz and M. G. Speranza. « A Tabu Search Algorithm for the Split Delivery Vehicle Routing Problem ». *Transportation science* 40(1), pages 64-73, 2006.
- [Aronson 1995] L.D. Aronson. « Algorithms for vehicle routing - A survey ». Technical Report, Faculty of Technical Mathematics and Computer Science, Delft University of Technology, 15 November 1995.
- [Augerat 1995] P. Augerat. « Approche polyédrale du problème de tournées de véhicules ». PhD thesis, Institut National Polytechnique de Grenoble, Grenoble, France, 1995.
- [Augerat et al 1998] P. Augerat, J.M. Belenguer, E. Benavent, A. Corberan and D. Naddef. « Separating Capacity Constraints in the CVRP Using Tabu Search ». *European Journal of Operational Research* 106(2), pages 546–557, 1998.
- [Baker et al. 2003] B. M. Baker and M.A. Ayechev. « A genetic algorithm for the vehicle routing problem ». *Computers and Operations Research* 30(5), pages 787–800, 2003.
- [Baptista et al. 2002] S. Baptista, R.C. Oliveira and E. Zúquete. « A Period Vehicle Routing Case Study ». *European Journal of Operational Research* 139, pages 220-229, 2002.
- [Beasley 1983] J.E. Beasley. « Route-first cluster-second methods for vehicle routing ». *Omega* 11(4), pages 403–408, 1983.
- [Bent et al. 2003] R. Bent and P. Van Hentenryck. « Dynamic Vehicle Routing with Stochastic Requests ». *International Joint Conference on Artificial Intelligence*, pages 1362–1363, 2003.
- [Bent et al. 2004] R. Bent and P. V. Hentenryck. « Scenario Based Planning for Partially Dynamic Vehicle Routing Problems with Stochastic Customers ». *Operations Research* 52(6), pages 977-987. 2004.

[Berger et al. 2003] J. Berger and M. Barkaoui. « A new hybrid genetic algorithm for the capacitated vehicle routing problem ». *Journal of the Operational Research Society* 54(12), pages 1254–1262, 2003.

[Bertsimas et al. 1990] D.J. Bertsimas, P. Jaillet, A.R. Odoni. « A priori optimization ». *Operations Research* 38(6), pages 1019–1033. 1990.

[Bertsimas et al. 1992] D. J. Bertsimas. « A vehicle routing problem with stochastic demand ». *Operations Research* 40(3), pages 574–585, 1992.

[Bertsimas et al.1995] D. J. Bertsimas, P. Chervi and M. Peterson. « Computational approaches to stochastic vehicle routing problems ». *Transportation Science* 29, pages 342-352, 1995.

[Bertsimas et al.1996] D. Bertsimas and D. Simchi-Levi. « A New Generation of Vehicle Routing Research: Robust Algorithms, Addressing Uncertainty ». *Operations Research* 44(2), pages 286-304, 1996.

[Beyer 2001] H. G. Beyer. *The Theory of Evolutionary Strategies*. Berlin: Springer, 2001.

[Bianchi 2000] L. Bianchi. « Notes on Dynamic Vehicle Routing - The State of the Art ». Technical Report, IDSIA-05-01, 20 December 2000.

[Bianchi et al. 2004] L. Bianchi, M. Birattari, M. Chiarandini, et al.. « Metaheuristics for the vehicle routing problem with stochastic demands ». In: *Proceedings of the 8th International Conference on Parallel Problem Solving from Nature*, Volume 3242/2004, pages 450-460, 2004

[Bianchi et al. 2006] L. Bianchi, M. Birattari, M. Manfrin, Mastrolilli, et al. « Hybrid metaheuristics for the vehicle routing problem with stochastic demands » *Journal of Mathematical Modelling and Algorithms* 5(1), pages 91-110, 2006.

[Bodin et al. 1981] L. Bodin and B. Golden. « Classification in vehicle routing and scheduling ». *Networks* 11, pages 97-108, 1981.

[Branke 2001] J. Branke. *Evolutionary Optimization in Dynamic Environments (Genetic Algorithms and Evolutionary Computation)*. Kluwer Academic Publishers, 2001.

[Bräysy et al. 2000] O. Bräysy , J. Berger and M. Barkaoui. « A New Hybrid Evolutionary Algorithm for the Vehicle Routing Problem with Time Windows ». Presented at the Route 2000 workshop, Danemark, 2000.

[Bräysy et al. 2001] O. Bräysy and M. Gendreau. « Metaheuristics for the Vehicle Routing Problem with Time Windows ». SINTEF Applied Mathematics, Research Council of Norway,

2001.

[Bräysy 2005] O. Bräysy and M. Gendreau. « Vehicle Routing Problem with Time Windows, Part I: Route Construction and Local Search Algorithms ». *Transportation Science* 39(1), pages 104-118, 2005

[Breedam 1995] A. Van Breedam. « Improvement Heuristics for the Vehicle Routing Problem Based on Simulated Annealing », *European Journal of Operational Research* 86, pages 480–490, 1995.

[Breedam 1996] A. Van Breedam. « An Analysis of the Effect of Local Improvement Operators in Genetic Algorithms and Simulated Annealing for the Vehicle Routing Problem ». RUCA, Faculty of Applied Economics, University of Antwerp, Antwerp, Belgium, Working Paper, pages 96-14, 1996.

[Brotcorne et al. 2003] L. Brotcorne, G. Laporte and F. Semet. « Ambulance location and relocation models ». *European Journal of Operational Research* 147(3), pages 451–463, 2003.

[Caricato et al. 2003] P. Caricato, G. Ghiani, A. Grieco and E. Guerriero. « Parallel Tabu Search For A Pickup And Delivery Problem Under Track Contention ». *Parallel Computing* 29 (5), pages 631–639, 2003.

[Cerny 1985] V. CERNY, « A thermodynamical approach to the travelling salesman problem : an efficient simulation algorithm ». *journal of optimization theory and applications* 45, pages 41-51, 1985.

[Charnes et al. 1963] A. Charnes and W. Cooper. « Deterministic equivalents for optimizing and satisfying under chance constraints ». *Operations Research* 11, pages 18–39. 1963.

[Chiang et al. 1997] W.C. Chiang and R.A. Russell. « A Reactive Tabu Search Metaheuristic for the Vehicle Routing Problem with Time Windows », *INFORMS Journal on Computing* 9, pages 417-430, 1997.

[Christiansen et al. 2007] C.H. Christiansen and J. Lysgaard, « A branch-and-price algorithm for the capacitated vehicle routing problem with stochastic demands ». *Operations Research Letters* 35, pages 773-781, 2007.

[Christofides et al. 1979] N. Christofides, A. Mingozzi and P. Toth. « The vehicle Routing Problem ». *Combinatorial Optimization*. Wiley, Chichster, pages 315–338, 1979.

[Christofides et al. 1981] N. Christofides, S. A. Mingozzi and P. Toth. « State Space Relaxation Procedures for the Computation of Bounds to Routing Problems ». *Networks* 11, pages 145–164, 1981.

[Christofides 1985] N. Christofides. « Vehicle Routing ». chapter 12 in *Traveling Salesman Problem* . pages 431- 448. John Wiley & Sons Ltd., 1985.

[Clarke et al. 1964] G. Clarke and J.W. Wright. « Sheduling of vehicules from a central depot to a number of delivery ponts ». *Operations Research* 12, pages 568–581, 1964.

[Coello 1998] C.A. Coello. « An update survey of ga-based multiobjective optimization techniques ». Technical Report, RD-98-08, Laboratorio Nacional de Informatica Avanzada (LANIA), México, Dec. 1998.

[Colorni et al. 1992] A. Colorni, M. Dorigo and V. Maniezzo. « Distributed optimization by ant colonies ». proceedings of the first European Conference on Artificial Life (ECAL 91), pages 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 105–119, 1997.

[Cordeau et al. 2001] J.F. Cordeau, G. Laporte and A. Mercier. « A unified tabu search heuristic for vehicle routing problems with time windows ». *Journal of the Operational Research Society* 52, pages 928–936, 2001.

[Cordeau et al. 2002] J.F. Cordeau, G. Desaulniers, J. Desrosiers, M. Solomon and F. Soumis. « The VRP with Time Windows ». *SIAM Monographs on Discrete Mathematics and Applications*, 9, P. Toth and D. Vigo(eds.), SIAM, Philadelphia, PA, pages 157–193, 2002.

[Cordeau et al. 2004] J.F. Cordeau, M. Gendreau, A. Hertz, G. Laporte and J.S. Sormany. « New Heuristics for the Vehicle 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 ». *Operations Research, Management Sciences* 6(1), pages 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(3), pages 322-332,1990.

[Desrochers et al. 1992] M. Desrochers, J. Desrosiers and M. Solomon. « A new Optimization Algorithm for the Vehicle Routing Problem with Time Windows ». *Operations Research* 40, pages 342–354, 1992.

[Doerner et al. 2004] K.F. Doerner, R.F. Hartl, G. Kiechle, M. Lucka and M. Reimann, « Parallel ant systems for the capacitated vehicle routing problem ». In: Gottlieb, J., Raidl, G.R. (Eds.), *Lecture Notes in Computer Science* 2004. Springer-Verlag, Heidelberg, Germany. Pages 72-83, 2004

- [Dorigo et al. 1996] M. Dorigo, V. Maniezzo and A. Colorni. « The Ant System: Optimization by a Colony of Cooperating Agents ». IEEE Transactions on Systems, Man and Cybernetics, 26(1), pages 29-41, 1996
- [Dorigo et al. 1997] M. Dorigo and L. Gambardella. « Ant Colony System: A Cooperative Learning Approach to the Travelling Salesman Problem ». IEEE Transactions on Evolutionary Computation 1(1), pages 53–66, 1997.
- [Dror 1993] M. Dror. « Modeling vehicle routing with uncertain demands as a stochastic program: Properties of the corresponding solution ». European Journal of Operational Research 64(3), pages 432–441. 1993.
- [Dror et al. 1985] M. Dror, M.O. Ball and B.L. Golden. « A computational comparison of algorithms for inventory routing ». Annals of Operations Research 4(1), pages 1–23, 1985.
- [Dror et al. 1986] M. Dror and P. Trudeau. « Stochastic vehicle routing with modified savings algorithm ». European Journal of Operational Research 23(2), pages 228–235, 1986
- [Dubois et al. 1988a] D. Dubois and H. Prade. Possibility Theory: An Approach to Computerized Processing of Uncertainty, Plenum, New York, 1988.
- [Dubois et al. 1988b] D. Dubois and H. Prade. « Fuzzy numbers: an overview ». in: J.C. Bezdek (Ed.), Analysis of Fuzzy Information 2, CRC Press, BocaRaton, pages 3–39, 1988.
- [Duhamel 2001] C. Duhamel. « Un Cadre Formel pour les Méthodes par Amélioration Itérative - Application à deux problèmes d'Optimisation dans les Réseaux ». PhD thesis, Université Blaise Pascal - Clermont-Ferrand II, Mars 22 2001.
- [Eglese et al. 2005] R.W. Eglese, Z. Fu and L. Li. « A tabu search heuristic for the open vehicle routing problem ». Journal of the Operational Research Society 56(3), pages 267–274, 2005.
- [Ergun et al. 2003] Ö. Ergun, J.B. Orlin and A. Steele-Feldman. « Creating very large scale neighborhoods out of smaller ones by compounding moves : A study on the vehicle routing problem ». Working paper, Massachusetts Institute of Technology, 2003.
- [Feo 1995] T.A. Feo and M.G.C. Resende. « Greedy randomized adaptive search procedures ». Journal of Global Optimization 6, pages 109-133, 1995.
- [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.

[Fisher et al. 1978] M.L. Fisher and R. Jaikumar. «A Decomposition Algorithm for Large-Scale Vehicle Routing ». Rapport technique n_ 78-11-05, Departement of Decision Sciences, University of Pennsylvania, 1978.

[Fisher et al. 1997] M.L. Fisher, K.J. Rnsten and O.B.G. Madsen. « Vehicle Routing with Time Windows : two optimization algorithms ». *Operations Research* 45(3), pages 488–492, 1997.

[Fogel et al. 1966] L.J. Fogel, A.J. Qwens and M.J. Walsh. *Artificial Intelligence through simulated Evolution*. New York: John whiley, 1966.

[Funke et al. 2005] B. Funke, T. Grünert and S. Irnich. « Local Search for Vehicle Routing and Scheduling Problems: Review and Conceptual Integration ». *Journal Heuristics* 11(4): pages 267-306, 2005.

[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 ». *MAS2003 – The International Workshop on Modeling & Applied Simulation*, pages 3–9, 2003.

[Garvin et al. 1957] W.M. Garvin, H.W. Crandall, J.B. Johnson and R.A. Spellman. «Application of linear programming to oil industry ». *Management Science* 3, pages 407-430, 1957.

[Gendreau et al. 1992] M. Gendreau, A. Hertz and G. Laporte. « New Insertion and Postoptimization Procedures for the Traveling Salesman Problem », *Operations Research* 40(6), pages 1086–1094, 1992.

[Gendreau et al. 1995] M. Gendreau, G. Laporte and R. Séguin. « An exact algorithm for the vehicle routing problem with stochastic demands and customers ». *Transportation science* 29(2), pages 143–155, 1995.

[Gendreau et al. 1996a] M. Gendreau, G. Laporte and R. Séguin. « Stochastic vehicle routing ». *European Journal of Operational Research* 88(1), pages 3–12, 1996.

[Gendreau et al. 1996b] M. Gendreau, G. Laporte and R. Séguin. « A tabu search heuristic for the vehicle routing problem with stochastic demands and customers ». *Operations Research* 44(3), pages 469-477, 1996.

[Gendreau et al. 1998a] M. Gendreau, A. Hertz, G. Laporte and M. Stan. « A Generalized Insertion Heuristic for the Traveling Salesman Problem with TimeWindows », *Operations Research* 46(3), pages 330–335, 1998.

[Gendreau et al. 1998b] M. Gendreau, F. Guertin, J.Y. Potvin and R. Séguin. « Neighborhood 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* 33(4), pages 381–390, 1999.

[Ghiani et al. 2003] G. Ghiani, F. Guerriero, G. Laporte and R. Musmanno. « Real-Time Vehicle Routing : Solution Concepts, Algorithms and Parallel Computing Strategies ». *European Journal of Operational Research* 151(1), pages 1–11, 2003.

[Gillett et al. 1974] B.E. Gillett and L.R. Miller. « A heuristic algorithm for the Vehicle Dispatch Problem ». *Operations Research* 22(2), pages 340–349, 1974.

[Glover 1977] F. Glover. « Heuristics for integer programming using surrogate constraints ». *Decision Science* 13(5), pages 156–166, 1977.

[Glover 1986] F. Glover. « Future paths for integer programming and links to artificial intelligence ». *Computers and Operations Research* 13, pages 533–549, 1986.

[Goldberg et al. 1985] D.E. Goldberg and R. Lingle. « Alleles, loci and the the Traveling Salesman Problem ». In *Proceedings of the First International Conference on Genetic Algorithms*, pages 154–159, 1985.

[Golden et al. 1979] B.L. Golden and J.R. Yee. « A framework for probabilistic vehicle routing ». *AIIE Transactions*, 11: pages 109–112, 1979.

[Goncalves et al. 2007] G. Goncalves, T. hsu, R. Dupas and H. Housroum. « Plateforme de simulation pour la gestion dynamique de tournées des véhicules ». *European Journal of Operational Research (JESA)* 2007, à paraître.

[Grefenstette 1986] J. J. Grefenstette. « Optimization of control parameters for genetic algorithms ». *IEEE Transactions on Systems, Man and Cybernetics*, Volume SMC-16, pages 122–128, 1986.

[Groth 2002] M.J. Groth. « Stochastic considerations in vehicle routing problems ». *Transport optimization*, Juin 2002.

[Guo et al. 2004] Z.G. Guo and K.L. Mak. « A heuristic algorithm for the stochastic vehicle routing problems with soft time windows ». *Congress on Evolutionary Computation (CEC'04)*, pages 1449–1456, 2004.

[Gutjahr 2004] W. Gutjahr. « S-ACO: An ant-based approach to combinatorial optimization under uncertainty ». in *Proceedings of ANTS 2004 – Ant Colony Optimization*

and Swarm Intelligence, volume 3172 of Lecture Notes in Computer Science, Springer, Berlin Heidelberg New York, pages 238–249, 2004.

[Haimovitch et al. 1985] M. Haimovitch and A.G.H. Rinnooy Kan. « Bounds and heuristics for capacitated routing problems ». *Mathematics of Operations Research* 10(4), pages 527–542, 1985.

[Hertz et al. 1994] A. Hertz, B. Jaumard, C. Ribeiro and W. Formosinho Filho. « A multicriteria tabu search approach to cell formation problems in group technology with multiple objectives ». *RAIRO Recherche Opérationnelle* 28(3), pages 303–328, 1994.

[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 de doctorat en informatique, Béthune, Mai 2005.

[Hvattum et al. 2003] L.M. Hvattum, A. Løkketangen and G. Laporte. « A heuristic solution method to a stochastic vehicle routing problem ». In *Norsk Informatikkonferanse 2003*, pages 229–240, 2003.

[Ioannou et al. 2003] G. Ioannou, M. Kritikos and G. Prastacos. « A problem generator-solver heuristic for vehicle routing with soft time windows ». *Omega*, Volume 31(1), Pages 41-53, 2003

[Ishibuchi et al. 1998] H. Ishibuchi and T. Murata. « A multi-objective genetic local search algorithm and its application to flowshop scheduling ». *IEEE Transaction on Systems, Man and Cybernetics, Part C*, pages 392–403, August 1998.

[Jacobs-Blecha et al. 1993] C. Jacobs-Blecha and M. Goetschalckx. « The Vehicle Routing Problem with Backhauls : An Optimization Based Approach ». *Proceedings of the 2nd Industrial Engineering Research Conference*, pages 504–509, 1993.

[Jagannathan 1985] R. Jagannathan. « Use of Sample Information in Stochastic Recourse and Chance-Constrained Programming Models », *Management Science* 31, pages 96-108. 1985.

[Jaillet 1985] P. Jaillet. « Probabilistic Traveling Salesman Problems ». PhD thesis, MIT, Cambridge, Massachusetts, 1985.

[Jaillet 1987] P. Jaillet. « Stochastic routing problems ». In: Andreatta, G., Mason, F., Serafini, P. (Eds.), *Stochastics in Combinatorial Optimization*. World Scientific, New Jersey, 1987.

[Jaillet 1988] P. Jaillet. « A priori solution of a travelling salesman problem in which a

random subset of the customers are visited ». *Operational Research* 36(6), pages 929–936, 1988.

[Jaillet et al. 1988] P. Jaillet and A. Odoni. « The probabilistic vehicle routing problems ». in chapter, B. L. Golden and A. A. Assad (eds.), *Vehicle Routing: Methods and Studies*. Elsevier, Amsterdam, The Netherlands, 1988.

[Jörnsten et al. 1986] K. Jörnsten, O. Madsen and B. Sorensen. « Exact solution of the Vehicle Routing and Scheduling Problem with Time Windows by Variable Splitting ». Research Report 5/1986, IMSOR, The Technical University of Denmark, Lyngby, Denmark, 1986.

[Jozefowicz et al. 2002] N. Jozefowicz, F. Semet and E-G. Talbi. « Parallel and Hybrid Models for Multi-objective Optimization: Application to the Vehicle Routing Problem». J.J. Merelo Guervós et al. (Eds.): *PPSN VII, LNCS 2439*, Springer-Verlag, pages 271–280, 2002.

[Kaufmann et Gupta 1985] A. Kaufmann and M.M. Gupta. « Introduction to Fuzzy Arithmetic». In *Theory and Applications*, Van Nostrand Reinhold, New York, 1985.

[Kawamura 1998] H. Kawamura, M. Yamamoto, T. Mitamura, K. Suzuki and A. Ohuchi. « Cooperative Search on Pheromone Communication for Vehicle Routing problems ». *IEEE Transactions on Fundamentals* E8 1- A, pages 1089–1096, 1998

[Kilby et al. 1998a] P. Kilby, P. Prosser and P. Shaw. « Dynamic VRPs : A Study of Scenarios». CSIRO, Canberra ACT 2601, Australia, September 1998.

[Kirkpatrick et al. 1983] S. Kirkpatrick, C.D. Gelatt and M.P. Vecchi. « Optimization by simulated annealing ». *Science* 220, 4598, pages 671-680, 1983.

[Kohl et al. 1997] N. Kohl and O. Madsen. « An optimization algorithm for the Vehicle Routing Problem with Time Windows based on Lagrangean Relaxation ». *Operations Research* 45(3), pages 395–406, 1997.

[Kolen et al. 1987] A. Kolen, H. Rinoy Kan and H. Trienekens. « Vehicle Routing with Time Windows ». *Operations Research* 35, pages 266–273, 1987.

[Koskosidis et al. 1992] Y.A. Koskosidis, W.B. Powell and M.M. Solomon, « An optimization-based heuristic for vehicle routing and scheduling with soft time window constraints ». *Transportation Science* 26 pages 69-85, 1992.

[Krumke et al. 2003] S.O. Krumke, W.E. de Paepe, D. Poensgen and L. Stougie. « News from the Online Traveling Repairman ». *Theoretical Computer Science* 295, pages 279–294, 2003.

[Lambert et al.1993] V. Lambert, G. Laporte and F.V. Louveaux. « Designing collection

routes through bank branches ». *Computers and Operations Research* 20, pages 783–791, 1993.

[Laporte et al. 1987] G. Laporte and Y. Nobert. « Exact algorithms for the vehicle routing problem ». *Annals of Discrete Mathematics* 31, pages 147-184, 1987.

[Laporte et al. 1986] G. Laporte , H. Mercure and Y. Nobert . « A exact algorithm for the asymmetrical capacitated vehicle routing problem ». *Networks*, 16, pages 33–46, 1986.

[Laporte et al. 1993] G. Laporte and F.V. Louveaux. « The integer l-shaped method for stochastic integer programs with complete recourse ». *Operations Research Letters* 33, pages 133–142, 1993.

[Laporte et al. 1998] G. Laporte and F.V. Louveaux. « Solving stochastic routing problems with the integer L-shaped method ». In: Crainic, G, Laporte, G (Eds.), *Fleet Management and Logistics*. Kluwer Academic Publishers, Boston, pages 159–167. 1998.

[Laporte et al. 2000] G. Laporte, M. Gendreau, J.Y. Potvin and F. Semet. « Classical and modern heuristics for the vehicle routing problem ». *International Transactions in Operational Research* 7, pages 285–300, 2000.

[Larson 1998] R.C. Larson. « Transportation of sludge to the 106-mile site: An inventory routing algorithm for fleet sizing and logistic system design ». *Transportation Science* 22, pages 186–198, 1988.

[Larsen 2001] A. Larsen. « The Dynamic Vehicle Routing Problem ». PhD thesis, IMM - Denmark, 2001.

[Larsen et al. 2004] A. Larsen, O. Madsen and M. Solomon. « The A-priori Dynamic Traveling Salesman Problem with Time Windows ». *Transportation Science* 38(4), pages 459–472, 2004.

[Lau et al. 2003] H.C. Lau, M. Sim and K.M. Teo. « Vehicle routing problem with time windows and a limited number of vehicles ». *European Journal of Operational Research* 148, pages 559–569, 2003.

[LeBouthillier 2000] A. LeBouthillier. « Modélisation UML pour une architecture cooperative 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.

[Lenstra et al. 1981] J.K. Lenstra and A.H.G. Rinnooy Kan. « Complexity of vehicle routing and scheduling problem ». *Networks* 11, pages 221-227, 1981.

[Li et al. 2005] F. Li, B.L. Golden and E.A. Wasil. « Very large-scale vehicle routing : New test problems, algorithms and results ». *Computers & Operations Research* 32, pages 1165–1179, 2005.

[Lin 1965] S. Lin. « Computer solutions of the traveling salseman problem ». *Bell System Technical Journal* 44, pages 2245–2269, 1965.

[Lucic et al. 2003] P. Lucic and D. Teodorovic. « Vehicle Routing Problem with Uncertain Demand at Nodes: The Bee System and Fuzzy Logic Approach ». In José-Luis Verdegay. *Fuzzy Sets Based Heuristics for Optimization*. Berlin, Springer, 2003.

[Lund et al. 1996] K. Lund, O. Madsen and J. M. Rygaard. « Vehicle Routing Problems with Varying Degrees of Dynamism ». Technical Report, IMM, The Department of Mathematical Modelling, Technical University of Denmark, 1996.

[Lysgaard 2004] J. Lysgaard, A.N. Letchford and R.W. Eglese. « A New Branch-and-cut Algorithm for the Capacitated Vehicle Routing Problem », *Mathematical Programming* 100, pages 423–445, 2004.

[Madsen 1990] O.B.G Madsen. « Lagrangean Relaxation and Vehicle Routing ». Working paper. IMSOR, The Technical University of Denmark, Lyngby, Denmark, 1990.

[Mamdani et al. 1975] E.H. Mamdani and S. Assilian. « An experiment in linguistic synthesis with a fuzzy logic controller ». *International Journal of Man-Machine Studies* 7(1), pages 1-13, 1975.

[Mester et al. 2005] D. Mester and O. Bräysy. « Active guided evolution strategies for the large scale vehicle routing problems with time windows ». *Computers & Operations Research* 32, pages 1593–1614, 2005.

[Mester et al. 2007] D. Mester, O. Bräysy and W. Dullaert. « A multi-parametric evolution strategies algorithm for vehicle routing problems ». *Expert Systems with Applications* 32(2), Pages 508-517, 2007

[Metropolis 1953] N. Metropolis, A.W. Rosenbluth, A.H. Teller and E. Teller. « Equation of state calculation by fast computing machines ». *Journal of Chemical Physics* 21, pages 1087-1091, 1953.

[Michalewicz 1994] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Berlin : Springer-Verlag, seconde édition, 1994.

[Miller 1995] D.L Miller. « A matching based exact algorithm for capacitated vehicle routing problems ». *ORSA Journal on Computing*, 7, pages 1–9, 1995.

[Mladenovic et al. 1997] N. Mladenovic and P. Hansen. « Variable neighborhood search». *Computers and Operations Research* 24, pages 1097-1100, 1997.

[Mole et al. 1976] R.H. Mole and S.R. Jameson. « A sequential route-building algorithm employing a generalized saving criterion ». *Operational Research Quarterly* 27(2), pages 503–511, 1976.

[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 and J.W. Barnes. « Solving the Pick-up and Delivery Problem with Time Windows using reactive tabu search ». *Transportation Research* 34, pages 107–121, 2000.

[Or 1976] I. Or. « Traveling Salesman-Type Combinatorial Problems and their Relation to the Logistics of Blood Banking ». PhD thesis, Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL, 1976.

[Osman 1993] I.H. Osman. « Metastrategy Simulated Annealing and Tabu Search Algorithms for the Vehicle Routing Problem ». *Annals of Operations Research* 41, pages 421–451, 1993.

[Osman 1996] I.H. Osman and G. Laporte. « Metaheuristics: a bibliography ». *Annals of Operations Research* 63, pages 513-623, 1996

[Pankratz 2005] G. Pankratz. « Dynamic vehicle routing by means of a genetic algorithm ». *International Journal of Physical Distribution & Logistics Management* 35(5), pages 362-383, 2005.

[Pereira et al. 2002] F. Pereira, J. Tavares, P. Machado and E. Costa. « GVR : a New Genetic Representation for the Vehicle Routing Problem ». *Artificial Intelligence and Cognitive Science : 13th, Irish Conference Proceedings*, pages 95–102, 2002.

[Potvin et al. 1993] J.Y. Potvin and J.M. Rousseau. « A parallel route building algorithm for the Vehicle Routing and Scheduling Problem with Time Windows ». *European Journal of Operational Research* 66, pages 331–340, 1993.

[Prins 2001] C. Prins. « A simple and effective evolutionary algorithm for the vehicle routing problem ». 4th Metaheuristics International Conference(MIC'2001), pages 143-147, 16-20/07/01, Porto, Portugal, 2001.

[Prins 2002] C. Prins. « Efficient heuristics for the heterogeneous fleet multitrip vehicle routing problem ». *Journal of Mathematical Modelling and Algorithms* 1(2), pages 135–150,

2002.

[Prins 2004] C. Prins. « A simple and effective evolutionary algorithm for the vehicle routing problem ». *Computers and Operations Research* 31(12), pages 1985–2002, 2004.

[Psaraftis 1995] H.N. Psaraftis. « Dynamic vehicle routing: Status and prospects ». *Annals of Operations Research* 61(1), pages 143-164, 1995.

[Pureza et al. 1991] V.M. Pureza and P.M. Fransa. « Vehicle routing problems via tabu search metaheuristic ». Technical Report CRT-347. Montreal, Canada: Centre for Research on Transportation, 1991.

[Ralphs 2003] T. K. Ralphs. « Parallel branch and cut for capacitated vehicle routing ». *Parallel Computing* 29 (5), pages 607–629, 2003.

[Rechenberg 1965] I. Rechenberg. « Cybernetic solution path of an experimental problem ». Roy Aircraft Establishment, Library Translation 1222, Farnborough, Hants, U K, 1965

[Rego 2001] C. Rego. « Node ejection chains for the Vehicle Routing Problem : Sequential and Parallel Algorithms ». *Parallel Computing* 27(3), pages 201–222, 2001.

[Rego et al. 1994] C. Rego and C. Roucairol. « Le Problème de Tournées de Véhicules : Etude et Résolution Approchée ». Technical Report, inria, Février 1994.

[Reimann et al. 2004] M. Reimann, K. Doerner and R.F. Hartl. « D-Ants: Savings Based Ants divide and conquer the vehicle routing problem ». *Computers & Operations Research*(31), pages 563-591, 2004.

[Renaud et al.1996] J. Renaud, F.F. BOCTOR and L. GILBERT. « An Improved Petal Heuristic for the Vehicle Routing Problem », *Journal of the Operational Research Society* 47, pages 1156– 1167, 1996.

[Robuste et al. 1990] F. Robuste, C.F. Daganzo and R.R. Souleyrette. « II Implementing vehicle routing models ». *Transportation Research Part B* 24B(4), pages 263-286. 1990

[Rochat et al. 1995] Y. Rochat and E. Taillard. « Probabilistic diversification and intensification in local search for Vehicle Routing ». *Journal of Heuristics* 1, pages 147–167, 1995.

[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.

[Russell 1995] R.A. Russell. « Hybrid heuristics for Vehicle Routing Problem with Time

Windows ». *Transportation Science* 29, pages 156–166, 1995.

[Sandgren 1994] E. Sandgren. « Advances in design optimization, Multicriteria design optimization by goal programming ». Chapman and Hall, pages 225–265, 1994.

[Schaffer 1985] J.D. Schaffer. « Multiple objective optimization with vector evaluated genetic algorithms ». In *Proceedings of the First International Conference on Genetic algorithms*, pages 93–100, 1985.

[Secomandi 2000] N. Secomandi. « Comparing neuro-dynamic programming algorithms for the vehicle routing problem with stochastic demands ». *Computers and Operations Research* 27(5), pages 1201–1225, 2000.

[Secomandi 2001] N. Secomandi. « A rollout policy for the vehicle routing problem with stochastic demands ». *Operations Research* 49(5), pages 796–802, 2001.

[Solomon 1987] M.M. Solomon. « Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints ». *Operations Research* 35, pages 254–265, 1987.

[Stewart et al. 1983] W.R. Stewart and B.L. Golden. « Stochastic vehicle routing: A comprehensive approach ». *European Journal of Operational Research* 14(4), pages 371–385, 1983.

[Stützle 1999] T. Stützle. « Local Search Algorithms for Combinatorial Problems – Analysis, Algorithms and New Applications ». DISKI – Dissertationen zur Künstlichen Intelligenz. Infix, Sankt Augustin, Germany, 1999.

[Sushil et al. 1999] J. L. Sushil, X. Yin and Z. Y. Yuan. « Multiple Vehicle Routing with Time Windows Using Genetic Algorithms ». In *Proceedings of IEEE Congress on Evolutionary Computation*. Piscataway: IEEE Press, pages 1804–1808, 29 January 1999.

[Taillard 1993] E. Taillard. « Parallel Iterative Search Methods for Vehicle Routing Problems », *Networks* 23, pages 661–673, 1993.

[Taillard 1999] E. Taillard. « A heuristic column generation method for the heterogeneous fleet VRP ». *RAIRO-Operations Research* 33, pages 1–14, 1999.

[Taillard et al. 1997] E. 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, pages 170–186, 1997.

[Talbi 2001] E. Talbi. « Métaheuristiques pour l’optimisation combinatoire multi-objectif : Etat de l’art ». Equipe OPAC (Optimisation PARallèle Coopérative), Laboratoire d’Informatique Fondamentale de Lille Université de Lille 1, France, 2001. citeseer.ist.psu.edu/382692.html.

[Tan et al. 2007] K.C. Tan, C.Y. Cheong, et C.K. Goh. « Solving multiobjective vehicle routing problem with stochastic demand via evolutionary computation ». *European Journal of Operational Research* 177(2), pages 813-839, 2007.

[Tarantilis et al. 2002] C.D. Tarantilis and C.T. Kiranoudis. « Bone Route : An adaptive memory-based method for effective fleet management ». *Annals of Operations Research* 115, pages 227–241, 2002.

[Teghem et al. 2003] J. Teghem and M. PIRLOT. *Résolution de Problèmes de RO par les Métaheuristiques*, Hermès, Paris, 2003.

[Teodorovic et al. 1992] D. Teodorovic and G. Pavkovic. « A simulated annealing technique approach to the vehicle routing problem in the case of stochastic demand ». *Transportation Planning and Technology* 16(4), pages 261–273, 1992.

[Teodorovic et al. 1996] D. Teodorovic and G. Pavkovic. « The fuzzy set theory approach to the vehicle routing problem when demand at nodes is uncertain ». *Fuzzy Sets and Systems* 82(3), Pages 307-317, 1996.

[Teodorovic et al. 2000] D. Teodorovic and P. Lucic. « Intelligent vehicle routing system ». In: *Proceedings of the IEEE International Conference on Intelligent Transportation Systems*, Dearborn, MI, USA, 1–3 October 2000, pages 482–487, 2000.

[Thompson et al. 93] P.M. Thompson and H.N. Psaraftis. « Cyclic Transfer Algorithms for Multivehicle Routing and Scheduling Problems ». *Operations Research* 41(5), pages 935-946, 1993.

[Tillman 1969] F. Tillman. « The multiple terminal delivery problem with probabilistic demands ». *Transportation Science* 3: pages 192–204, 1969.

[Toth et al. 2002] P. Toth and D. Vigo. *The Vehicle Routing Problem*. SIAM Monographs on Discrete Mathematics and Applications, 2002.

[Toth et al. 2003] P. Toth and D. Vigo. « The granular tabu search and its application to the vehicle routing problem ». *INFORMS Journal on Computing* 15(4), pages 333–346, 2003.

[Voudouris 1997] C. Voudouris. « Guided local search ». PhD thesis, Department of Computer Science, University of Essex, UK, 1997.

[Voß et al. 1998] S. Voß, S. Martello, I.H. Osman and C. Roucairol. *MetaHeuristics: Advances and Trends in Local Search Paradigms for Optimization*. Springer, 1998.

[Werners et al. 2003] B. Werners and M. Drawe. « Capacitated Vehicle Routing Problem with Fuzzy Demand ». In José-Luis Verdegay *Fuzzy Sets Based Heuristics for Optimization*.

Berlin, Springer, 2003.

[Willard 1989] J.A.G. Willard. « Vehicle Routing Using r-Optimal Tabu Search ». M.Sc. Dissertation. The Management School, Imperial College, London, 1989

[Xu et al. 2006] J. Xu, T. Hsu, R. Dupas and G. Goncalves. « Une Approche de résolution Multiobjectif du problème DVRPTW avec demandes floues ». Metaheuristique 06, Hamamet, Tunisie, 2-4 Novembre 2006.

[Xu et al. 2007a] J. Xu, T. Hsu and G. Goncalves. « A genetic algorithm for the stochastic vehicle routing problem with time windows ». 2nd Conference on Optimization Methods & Software and 6th EUROPT Workshop on Advances in Continuous Optimization (EUROPT-OMS), Prague, Czech Republic, 4-7 July, 2007.

[Xu et al. 2007b] J. Xu, T. Hsu and G. Goncalves. « A genetic algorithm for solving the vehicle routing problem with time windows and fuzzy demands ». International Conference on Artificial Intelligence and Systems, AIS'07/CAD07, Divnomorskoe, Russie, 2-10 September, 2007.

[Xu et al. 2007c] J. Xu, G. Goncalves and T. Hsu. « Un algorithme génétique pour le problème de tournées de véhicules avec demandes stochastiques et fenêtres de temps ». Logistique & Transport 2007, pages 379-384, Sousse, Tunisie, 18-20 Novembre, 2007.

[Yang et al.2000] W. Yang, K. Mathur and R.H. Ballou. « Stochastic vehicle routing problem with restocking ». Transportation Science 34(1), pages 99-112, 2000

[Yang et al. 2004] J. Yang, P. Jaillet and H. S. Mahmassani. « Real-Time Multi-Vehicle Truckload Pick-Up and Delivery Problems ». Transportation Science 38(2), pages 135-148, 2004.

[Zadeh 1978] L.A. Zadeh. « Fuzzy sets as a basis for a theory of possibility ». Fuzzy Sets and Systems 1, pages 3-28, 1978.

Résumé

Le travail réalisé dans cette thèse traite le problème de la gestion de tournées de véhicules avec fenêtres de temps et demandes floues (VRPTWFD : Vehicle Routing Problem with Time Windows and Fuzzy Demands). Ce problème consiste à trouver des chemins avec un coût minimum pour que les véhicules puissent visiter exactement une fois chaque client en respectant des contraintes. Les clients spécifient leur demande à l'aide d'un nombre flou pour une plus grande souplesse. Le VRPTWFD est étudié dans un contexte aussi bien statique que dynamique.

La théorie des possibilités nous a permis d'exprimer la contrainte de capacité floue en fixant des valeurs de seuils de possibilité et de nécessité. En utilisant cette contrainte de capacité floue, un modèle de programmation sous contraintes probabilistes (CCP) et un modèle à deux-étapes de programmation stochastique avec recours (SPR) ont été proposés pour traiter le VRPTWFD. Des algorithmes génétiques qui intègrent ces modèles, ont été proposés pour la recherche de bonnes solutions.

Dans le VRPTWFD dynamique, des nouveaux clients arrivent au cours de la journée. Une plateforme de simulation ayant la capacité de simuler la journée de service, nous a permis de résoudre le VRPTWFD dynamique « en ligne ».

Afin de vérifier les performances de ces modèles, nous avons construit un benchmark pour le VRPTWFD statique et un benchmark pour le VRPTWFD dynamique en modifiant le jeu de problèmes fournis par Solomon pour le VRPTW, puis nous avons évalué la qualité des solutions fournies par les modèles dans un environnement réel en simulant les situations réelles à l'aide de scénarios « test ».

Mots clés : Tournées de véhicules, Fenêtres de temps, Demandes floues, Stochastique, Dynamique, Théorie des possibilités, Algorithme génétique, Simulation

Abstract

The work of this thesis considers the vehicle routing problem with time windows and fuzzy demands (VRPTWFD). The goal of the problem is to find the routes of the vehicles with a minimal cost so that the vehicles can service each client exactly only once respecting some constraints. The customers specify their demands by a fuzzy number. The VRPTWFD is studied as the static case as well as the dynamic case.

We use the possibility theory to handle the constraint of capacity by setting certain thresholds for the degrees of the possibility and necessity. Using this capacity constraint, a chance constrained programming model (CCP) and a two-stages stochastic programming with recourse model (SPR) in stochastic programming were proposed to treat the VRPTWFD. The genetic algorithms integrating these models have been proposed as the optimization approach in order to find the optimal solutions.

In the dynamic VRPTWFD, some customers can call in their orders during the daily operation. A simulation platform, which has the capability of simulating the daily operation, has been developed to solve online the dynamic VRPTWFD.

In order to assess the performance of the proposed models, we have constructed a benchmark for the static VRPTWFD and a benchmark for the dynamic VRPTWFD adapting from Solomon's benchmark for the VRPTW, then we have evaluated the quality of the solutions, which were obtained by using these models, by simulating the real world situations with the help of the "test" scenarios.

Keyword: Vehicle routing, Time windows, Fuzzy demands, Stochastic, Dynamic, Possibility theory, Genetic Algorithm, Simulation