

UNIVERSITÉ DES SCIENCES ET TECHNOLOGIES DE LILLE

U.F.R. D'I.E.E.A.

Numéro d'Ordre : 3514

Année : 2004

THÈSE

pour obtenir le grade de

DOCTEUR DE L'U.S.T.L.

DISCIPLINE : INFORMATIQUE

présentée et soutenue publiquement,

le 06/12/2004, par

NICOLAS JOZEFOWIEZ

Titre :

MODÉLISATION ET RÉOLUTION APPROCHÉE DE PROBLÈMES DE
TOURNÉES MULTI-OBJECTIF

Jury :

Président :	M. Dauchet	Professeur, Université de Lille 1
Rapporteurs :	J. Figueira	Associate Professor, Université de Coimbra
	C. Prins	Professeur, Université de Troyes
Examineur :	X. Gandibleux	Professeur, Université de Nantes
Directeur :	E-G. Talbi	Professeur, Université de Lille 1
Co-directeur :	F. Semet	Professeur, Université de Valenciennes

Remerciements

Mes remerciements vont tout d'abord à mes directeurs de thèse. Merci à El-Ghazali Talbi pour m'avoir accueilli dans son équipe de recherche et m'avoir fourni un environnement de travail et des moyens techniques favorables. Que Frédéric Semet trouve ici toute ma reconnaissance pour ses conseils, sa présence et son soutien. Je suis conscient que sans lui cette thèse n'aurait pas été ce qu'elle est.

Je remercie très chaleureusement M. Max Dauchet de m'avoir fait l'honneur d'accepter d'être président du jury.

De même, j'adresse des remerciements très chaleureux à M. José Figueira et M. Christian Prins qui ont accepté d'être les rapporteurs de cette thèse. En particulier, je les remercie pour l'attention qu'ils ont portée à mon travail et pour les conseils et les améliorations lors de la rédaction du document final.

Que M. Xavier Gandibleux soit remercié d'avoir accepté d'être examinateur de cette thèse et de lui avoir porté un intérêt réel.

Je remercie les personnes du Laboratoire d'Informatique Fondamentale de Lille qui ont créé un environnement de travail agréable et chaleureux. Je ne peux pas toutes les citer, mais je n'ai aucun doute qu'elles se reconnaîtront. Je tiens à remercier chaleureusement Julien grâce à qui ces années n'ont pas été un enfer Linuxien. Je n'oublie pas les membres de mon équipe de recherche que j'ai eu la chance de connaître et avec lesquels j'ai pris beaucoup de plaisir à travailler. Ce sont aussi bien les permanents : Clarisse et Nounou, que les autres "thésards" présents et passés : Hervé, Laetitia, Ben, Matth et Julien (un autre).

Ces années de thèse n'auraient pas non plus été les mêmes sans toutes les personnes qui ont permis que ces années ne se limitent pas à résoudre des problèmes qui tournent en rond. Toutes les citer serait trop long mais je suis certain qu'elles se reconnaîtront.

Enfin, cette thèse n'aurait pu se faire sans le soutien constant de ma famille dont les nombreuses relectures ont permis qu'elle soit présentable et écrite dans un français correct.

Table des matières

Introduction	1
1 Optimisation multi-objectif	7
1.1 Généralités	8
1.2 Définitions	9
1.2.1 Problème d'optimisation	9
1.2.2 Problème d'optimisation multi-objectif	9
1.2.3 Solution d'un problème d'optimisation multi-objectif	10
1.3 Approches de résolution	13
1.3.1 Classification	13
1.3.1.1 Du point de vue du décideur	13
1.3.1.2 Du point de vue du concepteur	14
1.3.2 Approches scalaires	14
1.3.2.1 Méthode d'agrégation	14
1.3.2.2 Méthodes avec vecteur cible	16
1.3.2.3 Méthode ϵ -contrainte	18
1.3.3 Approches Pareto	18
1.3.4 Approches non scalaires et non Pareto	19
1.3.4.1 Sélection parallèle	19
1.3.4.2 Méthode lexicographique	20
1.3.4.3 Méthode avec genre	22
1.4 Métriques pour l'évaluation des performances	22
1.4.1 Ensemble \mathcal{P}^* connu	23
1.4.1.1 Proportion d'erreur	23
1.4.1.2 Distance générationnelle	23

1.4.1.3	Erreur maximale à la surface de compromis	25
1.4.2	Ensemble \mathcal{P}^* inconnu	25
1.4.2.1	Mesures évaluant la convergence	25
1.4.2.2	Mesures évaluant la diversité	27
1.4.2.3	Mesures évaluant la convergence et la diversité	28
1.5	Conclusion	29
2	Problèmes de tournées	31
2.1	Introduction	32
2.2	Présentation générale des problèmes de tournées	32
2.2.1	Le réseau	32
2.2.2	La demande	33
2.2.3	La flotte	33
2.2.4	Les coûts	34
2.2.5	L'objectif	34
2.3	Le problème de la tournée couvrante (PTC)	35
2.4	Le problème d'élaboration de tournées de véhicules (PTV)	37
2.5	Méta-heuristiques pour le PTV	38
2.5.1	Les méta-heuristiques à solution unique	39
2.5.1.1	Les méthodes de descente (Hill climbing)	39
2.5.1.2	Le recuit simulé (Simulated annealing)	39
2.5.1.3	La recherche Tabou (Tabu search)	40
2.5.2	Les méta-heuristiques à population de solutions	41
2.5.2.1	Les algorithmes génétiques (Genetic algorithms)	41
2.5.2.2	Les colonies de fourmis (Ant system)	43
2.6	Problèmes de tournées multi-objectif	44
2.6.1	Applications	44
2.6.1.1	Cas réels	45
2.6.1.2	Extension de problème	47
2.6.1.3	Généralisation de problème	48
2.6.2	Objectifs	49
2.6.2.1	Objectifs sur les tournées	49
2.6.2.2	Objectifs sur les nœuds et les arcs	51

2.6.2.3	Objectifs sur les ressources	51
2.6.3	Méthodes de résolution	52
2.6.3.1	Prise en compte du caractère multi-objectif	52
2.6.3.2	Méthodes d'optimisation	52
2.7	Conclusion	52
3	Algorithmes génétiques et méthodes de voisinage pour l'optimisation multi-objectif	59
3.1	Introduction	60
3.2	Méthodes coopératives	60
3.2.1	Coopération	60
3.2.2	Stratégie en deux phases : diversification-intensification	62
3.2.3	Modèles en îles	62
3.3	Algorithmes génétiques multi-objectif	63
3.3.1	Mécanismes pour l'optimisation multi-objectif	64
3.3.1.1	Assignation de l'efficacité	65
3.3.1.2	Préservation de la diversité	65
3.3.1.3	Elitisme	67
3.3.2	La diversification élitiste	68
3.3.3	Extension parallèle de la diversification élitiste	71
3.3.4	Présentation de l'algorithme NSGA II	71
3.3.5	Extension de NSGA II en modèle en îles	73
3.3.6	Inclusion de la diversification élitiste dans NSGA II	75
3.4	Méthodes de voisinage pour l'optimisation multi-objectif	75
3.4.1	Recherche guidée par l'archive	76
3.4.2	Recherche guidée par restriction de l'espace	78
3.4.3	Target Aiming Pareto Search	79
3.5	Conclusion	82
4	Modélisation et résolution du problème de la tournée couvrante bi-objectif	85
4.1	Introduction	86
4.2	Présentation, modélisation et remarques	86
4.2.1	Objectifs	87

4.2.2	Programme linéaire en nombre entier bi-objectif	87
4.2.3	Contraintes valides supplémentaires	88
4.2.4	Une heuristique pour le problème de la tournée couvrante	88
4.2.5	Remarques	89
4.3	Résolution exacte par algorithme de séparations et coupes et méthode ϵ - contrainte	91
4.3.1	Algorithme	92
4.3.2	Avantages	93
4.4	Résolution par algorithme génétique	94
4.4.1	Structure générale	94
4.4.2	Opérateurs de croisement	96
4.4.2.1	Opérateur select-first route-second (SRX)	96
4.4.2.2	Opérateurs route-first select-second (RSX)	97
4.4.3	Opérateurs de mutation	98
4.5	Coopération entre l'algorithme génétique et l'algorithme de séparations et coupes	98
4.5.1	Coopération de type I	100
4.5.2	Coopération de type II	100
4.6	Résultats expérimentaux	102
4.6.1	Problèmes tests utilisés	102
4.6.2	Protocole	102
4.6.3	Comparaison des opérateurs de croisement et de mutation	104
4.6.4	Comparaison entre la méthode exacte et l'algorithme génétique	105
4.6.5	Apport de la coopération	107
4.6.5.1	Intérêt de la coopération	107
4.6.5.2	Apport de la coopération de type I	108
4.6.5.3	Apport de la coopération de type II	109
4.6.5.4	Apport combiné des coopérations de type I et II	109
4.6.6	Tests sur le cas réel du district de Suhum au Ghana	109
4.7	Conclusion	112
5	Modélisation et résolution du problème d'élaboration de tournées de véhicules avec équilibrage des tournées	115
5.1	Introduction	116

5.2	Présentation et modélisation	116
5.2.1	Objectifs	117
5.2.2	Programme linéaire en nombre entier bi-objectif	118
5.3	NEMO : un algorithme génétique multi-objectif parallèle pour le PTVET	119
5.3.1	Première version de NEMO	119
5.3.1.1	Phase de sélection	119
5.3.1.2	Phase de recombinaison	120
5.3.1.3	Phase de mise à jour de l'archive	120
5.3.1.4	Phase de communication	120
5.3.1.5	Post-optimisation	121
5.3.2	Seconde version de NEMO	121
5.3.2.1	Modification de l'algorithme génétique par rapport à la version précédente	121
5.3.2.2	Post-optimisation	122
5.3.3	Troisième version de NEMO	123
5.3.3.1	Phase de sélection	123
5.3.3.2	Phase de recombinaison	124
5.3.3.3	Phase de mise à jour des archives	124
5.3.3.4	Phase de communication	124
5.4	Implémentation de NSGA II pour le PTVET	125
5.5	Opérateurs génétiques et heuristique	126
5.5.1	Opérateurs génétiques	127
5.5.1.1	Opérateur de croisement RBX	127
5.5.1.2	Opérateur de croisement SBX	127
5.5.1.3	Opérateur de croisement SPLIT	127
5.5.1.4	Opérateur de mutation Or-opt	128
5.5.2	Recherche locale	129
5.6	Implémentation de l'algorithme Target Aiming Pareto Search	129
5.6.1	Unified Tabu Search	129
5.6.2	Adaptation d'UTS	130
5.6.3	Phase de clustering	131
5.7	Expérimentation	131
5.7.1	Evaluation de NEMO	131

5.7.1.1	Protocole	131
5.7.1.2	Apport de la diversification élitiste à NEMO	132
5.7.1.3	Apport du modèle parallèle	132
5.7.1.4	Remarques complémentaires	133
5.7.2	Evaluation de NSGA II	134
5.7.2.1	Protocole	134
5.7.2.2	Apport de la parallélisation pour NSGA II	136
5.7.2.3	Apport de la diversification élitiste pour NSGA II	137
5.7.3	Evaluation de Target Aiming Pareto Search	139
5.7.3.1	Protocole	139
5.7.3.2	Efficacité de TAPaS	139
5.7.3.3	Efficacité de TAPaS comme méthode de post-optimisation .	140
5.7.3.4	Intérêt de l'hybridation	142
5.7.4	Etudes des bornes	142
5.8	Conclusion	144
Conclusions et perspectives		147
Bibliographie		149
A Résultats détaillés de la comparaison entre l'algorithme génétique et la méthode exacte sur le PTC bi-objectif		161

Liste des tableaux

2.1	Problèmes de tournées multi-objectif de la littérature.	54
2.2	Problèmes de tournées multi-objectif de la littérature.	55
2.3	Problèmes de tournées multi-objectif de la littérature.	56
2.4	Problèmes de tournées multi-objectif de la littérature.	57
4.1	Comparaison des opérateurs de croisement et de mutation.	104
4.2	Comparaison entre la méthode exacte et l'algorithme génétique.	105
4.3	Rapports et erreurs du nombre de solutions Pareto optimales trouvées et du nombre de couvertures trouvées.	107
4.4	Apport de la coopération de type I.	108
4.5	Apport de la coopération de type II.	109
4.6	Apport combiné des coopérations de type I et II.	110
4.7	Résultats sur le cas réel du district de Suhum, Ghana.	110
5.1	Valeurs des objectifs pour les meilleures solutions en termes de distance trouvées par Taburoute et l'algorithme génétique de Prins.	117
5.2	Apport de la diversification élitiste : $S(ED) - S(NED)$	132
5.3	Résultats de la mesure de couverture $C(ED, NED)$ (première entrée par cellule) et $C(NED, ED)$ (seconde entrée par cellule).	133
5.4	Apport du modèle en îles de la diversification élitiste : $S(PAR) - S(ED)$. . .	133
5.5	Résultats de la mesure de couverture $C(PAR, ED)$ (première entrée par cellule) et $C(ED, PAR)$ (seconde entrée par cellule).	134
5.6	Temps de calcul moyen en minutes.	134
5.7	Meilleures solutions trouvées par l'algorithme génétique multi-objectif pour les deux objectifs.	135
5.8	Valeurs moyennes et écarts types de la mesure \mathcal{S} pour NSGA II en fonction du nombre de processeurs utilisés.	136

5.9	Temps de calcul moyens en secondes de NSGA II en fonction du nombre de processeurs.	136
5.10	Valeurs moyennes et écarts types de la mesure \mathcal{S} pour NSGA II avec et sans diversification élitiste.	137
5.11	Meilleures solutions trouvées pour chaque objectif et valeurs associées du second objectif pour chaque implémentation de NSGA II.	138
5.12	Efficacité de TAPaS-std : D(TAPaS-std, NSGA II).	139
5.13	Efficacité de TAPaS-dp : D(TAPaS-dp, NSGA II).	140
5.14	Résultats des mesures C(TAPaS-std, NSGA II) (première entrée par cellule) et C(NSGA II, TAPaS-std) (seconde entrée par cellule).	140
5.15	Résultats des mesures C(TAPaS-dp, NSGA II) (première entrée par cellule) et C(NSGA II, TAPaS-dp) (seconde entrée par cellule).	142
5.16	Apport de l'hybridation : D(TAPaS-std, NSGA II) (première entrée par cellule) et D(TAPaS-dp, NSGA II) (seconde entrée par cellule).	143
5.17	Apport de l'hybridation : C(TAPaS-std, NSGA II) (première entrée par cellule) et C(TAPaS-dp, NSGA II) (seconde entrée par cellule).	144
5.18	Valeurs de la métrique D pour TAPaS-dp avec et sans hybridation.	144
5.19	Meilleures bornes trouvées pour les deux objectifs pour différentes méthodes.	146
A.1	Résultats pour $ V = 50$	162
A.2	Résultats pour $ V = 75$	163
A.3	Résultats pour $ V = 100$	164
A.4	Résultats pour $ V = 120$	165

Table des figures

1.1	Exemple de dominance.	11
1.2	Frontière Pareto : minimiser f_1 , minimiser f_2	12
1.3	Frontière Pareto : minimiser f_1 , maximiser f_2	12
1.4	Frontière Pareto : maximiser f_1 , minimiser f_2	12
1.5	Frontière Pareto : maximiser f_1 , maximiser f_2	12
1.6	La méthode d'agrégation.	15
1.7	La méthode du but à atteindre.	17
1.8	Une difficulté pour la méthode du but à atteindre.	17
1.9	La méthode ϵ -contrainte.	19
1.10	Sélection parallèle dans l'algorithme VEGA.	20
1.11	La méthode lexicographique dans le cas bi-objectif.	21
1.12	Les deux buts d'une méthode multi-objectif.	24
1.13	Une solution idéale.	24
1.14	Une solution qui répond au but de convergence mais qui est mauvaise pour le but de diversification.	24
1.15	Une solution qui répond au but de diversification mais qui est mauvaise pour le but de convergence.	24
1.16	(a) $S = 0$	28
1.17	(b) $S = 0$	28
1.18	(c) $S > 0$	28
1.19	La valeur de \mathcal{S} dépend du choix arbitraire d'un point de référence. Dans le cas du haut, deux ensembles de solutions non-dominées sont montrés, A et B , avec $\mathcal{S}(A) > \mathcal{S}(B)$. Dans le cas du bas, pour les mêmes ensembles, on a $\mathcal{S}(A) < \mathcal{S}(B)$	30
2.1	Un exemple de solution du problème de la tournée couvrante.	36

2.2	Un exemple de solution pour le problème d'élaboration de tournées de véhicules. (Les demandes des clients sont indiquées dans les parenthèses et la capacité des véhicules est de 12)	38
2.3	Fourmis à la recherche de sources de nourriture.	43
2.4	Objectifs pour les problèmes de tournées rencontrés dans la littérature.	45
3.1	Coopération chaînée.	62
3.2	Coopération imbriquée.	62
3.3	Coopération parallèle.	62
3.4	Coopération supervisée.	62
3.5	Etapes d'un algorithme génétique.	64
3.6	Etapes d'un algorithme génétique multi-objectif.	65
3.7	Assignation de l'efficacité d'un point de vue Pareto.	66
3.8	Illustration des techniques de préservation de la diversité.	67
3.9	Ensemble potentiellement Pareto optimal généré par un AG multi-objectif standard et la solution trouvée par TabuRoute pour deux benchmarks.	69
3.10	Frontières Pareto correspondantes aux différentes archives de la diversification élitiste pour deux objectifs.	70
3.11	Modèle coopératif de base pour la diversification élitiste - la structure torique n'est pas montrée pour ne pas obscurcir la figure.	70
3.12	Modèle coopératif de base pour la diversification élitiste - la structure torique n'est pas montrée pour ne pas obscurcir la figure.	71
3.13	L'algorithme évolutionnaire NSGA II.	72
3.14	Calcul de la distance de crowding.	73
3.15	Extension de NSGA II en modèle en îles.	74
3.16	Partition de l'espace objectif \mathcal{Y}	80
4.1	Bornes de l'espace des objectifs réalisable pour le problème de la tournée couvrante bi-objectif.	91
4.2	Résolution exacte par algorithme de séparations et coupes et méthode ϵ -contrainte.	92
4.3	Exemple d'application de la méthode de ranking rapide.	94
4.4	Exemple d'application de l'opérateur de croisement RSX.	99
4.5	Routes et villages, District de Suhum, Région Est, Ghana.	103
4.6	Réseau de transport modélisé, District de Suhum.	103
4.7	Ensemble Pareto optimal pour la saison sèche.	111

4.8	Ensemble Pareto optimal pour la saison des pluies.	111
4.9	Solution offrant la meilleure couverture pour la saison sèche.	112
4.10	Solution offrant la meilleure couverture pour la saison des pluies.	113
5.1	(a) est mieux équilibré que (b), mais (b) n'améliore pas l'équilibre de manière artificielle.	117
5.2	Topologie de communication du modèle en îles.	121
5.3	Ajout de la seconde archive à l'algorithme génétique multi-objectif.	122
5.4	Restriction de l'espace des objectifs.	123
5.5	Un exemple d'application de SBX.	127
5.6	D'une solution pour le PVC à une solution pour le PTV.	128
5.7	Un exemple d'application de l'opérateur Or-Opt.	129
5.8	Deux exemples où TAPaS est meilleur que NSGA II.	141

Introduction

Cette thèse s'inscrit dans le domaine de l'optimisation combinatoire multi-objectif et plus particulièrement dans l'étude de problèmes de logistique. Elle a été réalisée au sein de l'équipe Optimisation PARallèle Coopérative (OPAC) du Laboratoire d'Informatique Fondamentale de Lille (LIFL) en collaboration avec l'équipe Recherche Opérationnelle et Informatique (ROI) du Laboratoire d'Automatique, de Mécanique et d'Informatique industrielles et Humaines de Valenciennes (LAMIH) dans le cadre du Contrat Plan Etat Région "Méthodes pour l'Optimisation des Systèmes de Transport et de Télécommunication" (CPER MOST).

Le but de cette thèse est l'étude de problèmes de tournées de véhicules comportant plusieurs objectifs et leur résolution par le biais de méthodes et d'outils issus de l'optimisation multi-objectif. Les problèmes de tournées représentent une famille de problèmes fortement étudiée dans le cadre de la recherche opérationnelle. Le premier problème de cette famille à avoir été défini est sûrement le problème du voyageur de commerce dont les origines remontent au 19^{ème} siècle, les premières études mathématiques aux années 1930 et dont la première formulation a été présentée en 1954 [38]. Les problèmes de tournées, qui servent à modéliser de nombreux problèmes réels, sont le plus souvent étudiés et optimisés en ne prenant en compte qu'une unique fonction objectif qui vise le plus souvent à optimiser le coût des solutions identifiées. Or, la plupart des problèmes rencontrés dans l'industrie, et notamment en logistique, sont de nature multi-objectif. Par exemple, il est possible que plusieurs coûts soient associés à une tournée. Il est alors nécessaire de les optimiser tous. Cependant, l'ajout d'objectifs ne se limite pas à l'introduction de coûts supplémentaires car, s'il est important d'optimiser le coût d'une solution, de nombreux autres aspects peuvent être pris en compte au travers d'objectifs. Les objectifs supplémentaires peuvent par exemple avoir comme but d'introduire des notions d'équilibre ou de retard.

Dans les problèmes prenant en compte plusieurs objectifs simultanément, les objectifs sont la plupart du temps conflictuels les uns avec les autres. L'étude et la résolution de problèmes comportant plusieurs objectifs est l'objet de l'optimisation multi-objectif. Ce domaine possède ses sources dans les travaux de Edgeworth [52] et de Pareto [119] dans le cadre d'études d'économie au 19^{ème} siècle. Cependant, l'optimisation multi-objectif connaît un intérêt croissant depuis le milieu des années 1980 [148] et le domaine connaît une expansion importante depuis le milieu des années 1990 avec l'apparition de méthodes évolutionnaires pour l'optimisation multi-objectif [28]. Actuellement, l'optimisation multi-objectif est appliquée dans de nombreux domaines académiques et industriels. De manière formelle, un problème d'optimisation multi-objectif est un problème de la forme

$\min F(x) = (f_1(x), f_2(x), \dots, f_n(x))$ tel que x soit une solution réalisable. La solution d'un problème multi-objectif n'est pas une unique solution mais un ensemble de solutions appelé ensemble non-dominé. Les composantes du vecteur F sont les différentes fonctions à optimiser. Les solutions de cet ensemble sont nommées solutions non-dominées. Ces solutions sont celles pour lesquelles l'amélioration d'un des objectifs entraîne systématiquement la détérioration de la qualité d'au moins un autre objectif.

L'étude et la résolution, même approchée, de problèmes de tournées multi-objectif présentent de nombreux défis. En effet, les problèmes de tournées sont le plus souvent des problèmes NP-difficiles. De ce fait, dans la plupart des cas, une résolution exacte de ces problèmes devient inapplicable à cause de l'augmentation importante des temps de calcul en fonction de la taille des problèmes. Par ailleurs, dans le cadre des problèmes de tournées multi-objectif, la présence d'objectifs conflictuels complexifie encore plus la résolution, d'autant plus que la modélisation d'un problème de tournées multi-objectif n'est pas une tâche aisée car plusieurs choix sont envisageables. Il est en effet possible de transformer le problème multi-objectif en un problème mono-objectif. Par exemple, il est envisageable de ne garder qu'un seul objectif prioritaire et de transformer les autres objectifs en contraintes ou de regrouper tous les objectifs en un unique objectif à l'aide d'une méthode agrégative. Il est également possible de considérer tous les objectifs. Pour ce faire, on peut imposer une hiérarchie entre les objectifs ou bien encore utiliser des méthodes qui ne font pas de différence entre les objectifs. Le choix d'une modélisation influencera beaucoup la méthodologie de résolution employée. Cependant, même pour une modélisation fixée, de nombreux choix existent encore. D'autre part, la définition de méthodes d'optimisation multi-objectif n'est pas forcément chose aisée. En effet, la particularité d'une solution d'un problème multi-objectif, c'est-à-dire être constituée de plusieurs solutions, pose de nombreuses difficultés et demande la définition de stratégies et de mécanismes qui ne sont pas utilisés dans le cadre plus standard de l'optimisation mono-objectif. Enfin, des difficultés peuvent apparaître pour l'implémentation de certains mécanismes complexes.

Dans ce mémoire, nous nous intéressons à la résolution approchée de deux problèmes de tournées bi-objectif principalement à l'aide de méta-heuristiques. Aussi, les apports des travaux présentés dans ce mémoire peuvent se diviser en deux catégories. La première catégorie se situe au niveau de la classification des problèmes de tournées multi-objectif et l'étude de deux problèmes de tournées bi-objectif. La seconde catégorie concerne les apports pour les méthodes et mécanismes pour l'optimisation multi-objectif.

Dans la première catégorie, l'une des principales contributions de ce mémoire est la réalisation d'un état de l'art sur les problèmes de tournées multi-objectif. En effet, au cours de ces dernières années, de plus en plus d'études de ce type sont effectuées et la lecture des différents travaux tend à montrer que les auteurs ont peu de connaissances sur l'existant. Il est donc intéressant de classer et de dresser un bilan des travaux portant sur les problèmes de tournées multi-objectif. Par la suite, nous avons défini et résolu deux problèmes de tournées multi-objectif : le problème de la tournée couvrante bi-objectif et le problème d'élaboration de tournées de véhicules avec équilibrage des tournées. Ces deux problèmes sont intéressants car ils permettent d'illustrer les deux raisons qui peuvent motiver l'utilisation de plusieurs objectifs. En effet, le premier problème est une généralisation du problème de la tournée couvrante tandis que le second est une extension du problème

d'élaboration de tournées de véhicules. Dans le cadre de l'étude du problème de la tournée couvrante bi-objectif, nous avons aussi réalisé des tests sur des données réelles.

Parallèlement, dans cette thèse, nous nous sommes également intéressé aux méthodes et mécanismes employés pour résoudre les problèmes multi-objectif. De nombreuses techniques ont été proposées dans la littérature pour prendre en compte la présence de multiples objectifs. Depuis les années 1990, les algorithmes évolutionnaires utilisant des approches Pareto capables de prendre en compte tous les objectifs simultanément sont de plus en plus étudiés et utilisés [28]. De nombreux mécanismes et stratégies ont été proposés pour améliorer la qualité des approximations générées tant au niveau de la convergence vers l'ensemble non-dominé exact que de la diversification de l'approximation générée. Lors des expérimentations sur les problèmes que nous avons définis, il est apparu que l'utilisation des techniques standard pouvait ne pas être suffisante. Nous avons alors défini un nouveau mécanisme, la diversification élitiste, pour améliorer la qualité des résultats en termes de diversification. L'utilisation du parallélisme a aussi été étudiée à la fois pour améliorer l'impact de la diversification élitiste mais aussi pour améliorer de manière générale l'efficacité des algorithmes génétiques utilisés.

Il est généralement admis que les différentes méthodes d'optimisation possèdent chacune des avantages et des inconvénients, des forces et des faiblesses. Ainsi, de plus en plus d'études portent sur la coopération entre différents paradigmes d'optimisation. L'utilisation d'approche coopérative a aussi été prise en compte dans ce mémoire. En effet, pour résoudre les problèmes, une stratégie en deux phases a été définie. Dans un premier temps, l'algorithme génétique identifie une approximation la plus diversifiée possible. Puis, une seconde méthode est appliquée pour améliorer la qualité des résultats en termes de convergence. Pour cela, nous avons proposé plusieurs manières d'appliquer des méthodes à base de voisinage sur des ensembles non-dominés dans le but de les améliorer en termes de convergence. Dans le cas du problème de la tournée couvrante bi-objectif, nous avons utilisé un algorithme de séparations et coupes comme seconde méthode.

Il est à noter que le but dans cette thèse était de proposer des solutions de qualité pour les deux problèmes évoqués et de ce fait les méthodes et mécanismes proposés n'ont été validés que dans le cadre de ces problèmes. Cependant, la plupart des méthodes et mécanismes sont décrits de manière générale et il est donc possible de les adapter à d'autres problèmes.

Trois parties peuvent être identifiées dans ce mémoire. Les chapitres 1 et 2 présentent le contexte général du mémoire, à savoir l'optimisation multi-objectif et les problèmes de tournées. Le chapitre 3 est consacré à la présentation générale des méthodes d'optimisation utilisées. Enfin, les chapitres 4 et 5 contiennent respectivement la définition et les implémentations des méthodes d'optimisation pour le problème de la tournée couvrante bi-objectif et pour le problème d'élaboration de tournées de véhicules avec équilibrage des tournées. De manière plus précise, les différents chapitres abordent les points suivants.

Le chapitre 1 concerne l'optimisation multi-objectif en général. Dans un premier temps, les définitions et notations utilisées dans ce manuscrit sont fournies pour éviter les ambiguïtés dues au fait que les termes n'ont pas toujours le même sens dans tous les travaux de la littérature. Dans un second temps, les principales techniques pour traiter la présence

de plusieurs objectifs contradictoires sont présentées. Les avantages et inconvénients des différentes approches sont discutés. Le chapitre se clôt par une présentation des principales mesures et métriques pour évaluer ou comparer la qualité d'approximations.

Le chapitre 2 traite des problèmes de tournées et plus particulièrement les problèmes de tournées multi-objectif. Après avoir énuméré les principales caractéristiques des problèmes de tournées, les deux problèmes étudiés dans cette thèse sont présentés dans leur version mono-objectif. Ces problèmes sont le problème de la tournée couvrante et le problème d'élaboration de tournées de véhicules. Pour chacun des problèmes, une description et le modèle mathématique sont fournis. De plus, un état de l'art des méthodes d'optimisation, et plus particulièrement des méta-heuristiques, est fait pour chaque problème. L'apport majeur de ce chapitre est cependant l'état de l'art général des problèmes de tournées multi-objectif. Les études de ce type sont classées selon trois points. Le premier point concerne le contexte (cas réel, extension de problème ou généralisation de problème). Le second point est constitué des différents objectifs étudiés qui sont identifiés selon la composante du problème sur laquelle ils portent : tournées, nœuds/arcs ou ressources. Enfin, les approches de résolution et les méthodes d'optimisation utilisées sont répertoriées.

Le chapitre 3 porte sur les méthodes d'optimisation pour l'optimisation multi-objectif que nous avons proposées. Deux aspects sont étudiés : les algorithmes évolutionnaires utilisant des approches dites Pareto et l'utilisation de méthodes à base de voisinage pour l'optimisation multi-objectif. Au niveau des algorithmes évolutionnaires et plus particulièrement des algorithmes génétiques, une étude de l'existant en matière de techniques est tout d'abord réalisée. Puis nous présentons un nouveau mécanisme, la diversification élitiste, destiné à améliorer la qualité des approximations en termes de diversification. Un modèle parallèle en îles basé sur ce mécanisme est aussi présenté. Ensuite, deux modifications de l'algorithme multi-objectif standard NSGA II sont exposées. La première modification consiste dans la parallélisation de NSGA II au travers d'un modèle en îles. La seconde modification correspond à l'inclusion de la diversification élitiste dans NSGA II. Pour les méthodes à base de voisinage, après avoir identifié les caractéristiques qu'elles doivent remplir, nous proposons trois stratégies. Ces dernières prennent en compte le fait que nous désirons les faire coopérer avec des algorithmes génétiques.

Le chapitre 4 est dédié au problème de la tournée couvrante bi-objectif, une généralisation du problème de la tournée couvrante. Dans un premier temps, nous définissons le problème et proposons une modélisation sous la forme d'un programme linéaire en nombre entier. Puis trois approches pour le résoudre sont exposées. La première approche est une approche exacte qui consiste en une méthode ϵ -contrainte utilisant un algorithme de séparations et coupes pour le problème de la tournée couvrante. La seconde approche est un algorithme génétique multi-objectif pour lequel on a défini une procédure avancée pour la création de la population de départ ainsi que plusieurs opérateurs génétiques dédiés au problème de la tournée couvrante bi-objectif. Enfin, la dernière approche est constituée d'une méthode coopérative dans laquelle l'algorithme évolutionnaire est d'abord appliqué, puis des sous-problèmes sont extraits de l'approximation obtenue pour être résolus par l'algorithme de séparations et coupes pour le problème de la tournée couvrante. Les trois approches ont été évaluées sur des jeux de données générés ainsi que sur des données réelles.

Le chapitre 5 présente le problème d'élaboration de tournées de véhicules avec équi-

librage des tournées, une extension bi-objectif du problème d'élaboration de tournées de véhicules. Le second objectif permet d'inclure un équilibrage entre les tournées au travers d'un objectif linéaire. Nous avons modélisé le problème sous la forme d'un programme linéaire en nombre entier bi-objectif. Plusieurs méthodes de résolution sont proposées. La première est un algorithme génétique multi-objectif parallèle incluant la diversification élitiste. Ensuite, plusieurs implémentations de NSGA II ont été faites. Enfin, une implémentation de TAPaS, une méthode à base de voisinage que nous avons définie, utilisant une recherche Tabou est décrite. Toutes ces implémentations suivent les schémas généraux donnés dans le chapitre 3. Les méthodes et les différents mécanismes ont été testés sur des jeux de données classiques pour le problème d'élaboration de tournées de véhicules avec contraintes de capacité. Nous avons ainsi pu observer la pertinence des mécanismes et méthodes proposés.

Chapitre 1

Optimisation multi-objectif

Ce chapitre a pour objectif principal de définir les notions portant sur l'optimisation multi-objectif. Plus précisément nous mettons en évidence les choix et propositions faits par rapport à l'existant dans le domaine. Le chapitre s'articule autour de trois points.

Dans un premier point, on fournit les définitions et les notations utilisées dans l'ensemble du manuscrit. En effet, les termes utilisés dans la littérature ne sont pas toujours les mêmes surtout au niveau des notations.

Le second point traite des approches de résolution appliquées à l'optimisation multi-objectif. En effet, l'utilisation d'objectifs le plus souvent antagonistes pour évaluer une solution rend l'application de méthodes d'optimisation classiques difficile. Il est nécessaire de traiter le problème en le transformant ou en utilisant des mécanismes particuliers. De nombreuses approches sont proposées. Nous les avons classifiées par rapport à la manière dont elles prennent en compte la présence de plusieurs objectifs. Pour chaque méthode, des exemples d'utilisation de la littérature sont fournis et nous en discutons les avantages et inconvénients. Le troisième point porte sur les outils d'évaluation de performance et de comparaison entre les méthodes. Dans le cadre de l'optimisation multi-objectif, l'évaluation ou la comparaison de méthodes est une tâche difficile. De nombreuses méthodes existent et il n'est pas évident de savoir laquelle choisir car elles ne mesurent pas toutes la même chose. Pour les différencier, nous utilisons une classification à deux niveaux selon que l'ensemble optimal est connu ou non.

1.1 Généralités

De nombreux domaines de l'industrie, comme la mécanique, la chimie, les télécommunications, la logistique ou les transports, sont confrontés à des problèmes complexes. De nombreuses recherches sont menées afin de développer des méthodes efficaces pour résoudre ces problèmes, c'est-à-dire trouver une solution de bonne qualité. La qualité est le plus souvent exprimée par une fonction mathématique qui modélise typiquement le coût de la solution. Or, les problèmes sont rarement mono-objectif mais multi-objectif, c'est-à-dire qu'il est possible pour un même problème de définir plusieurs objectifs pertinents mais souvent contradictoires entre eux. Un objectif est dit contradictoire avec un autre objectif si une amélioration du premier objectif entraîne une dégradation du second.

Oszczka [114] décrit un problème multi-objectif comme la recherche :

“d'un vecteur de variables de décision qui satisfait des contraintes et optimise un vecteur de fonctions dont les éléments représentent les fonctions objectif. Ces fonctions forment une description mathématique des critères de performance qui sont usuellement en conflit les uns avec les autres. Ici, le terme "optimiser" signifie trouver une solution qui devrait donner des valeurs pour toutes les fonctions objectif acceptables pour le décideur.”

L'optimisation multi-objectif consiste à définir des méthodes efficaces pour la résolution de problèmes où tous les objectifs sont pris en compte. Cela signifie le plus souvent trouver de bons compromis entre les différents objectifs plutôt qu'une solution unique. La notion d'optimalité la plus généralement admise est celle introduite par Francis Ysidro Edgeworth en 1881 [52], généralisée plus tard par Vilfredo Pareto en 1896 [119]. Le terme le plus employé pour s'y référer est celui d'*optimum Pareto* ; cependant certains auteurs utilisent l'appellation d'*optimum d'Edgeworth-Pareto*.

L'optimisation multi-objectif a été initialement utilisée dans l'économie et le management, mais son utilisation dans les sciences pour l'ingénieur prend une place de plus en plus importante depuis la fin des années 1980. Il est à noter que dans la littérature on parle d'optimisation multi-objectif ou multi-critère ; le terme objectif étant employé quand ce qu'il modélise a une nature quantitative (coût, distance, ...), tandis que le terme critère correspond plus à une notion qualitative (appartenance à une classe, relation entre différents objets, ...). Dans ce manuscrit, le terme objectif sera utilisé indifféremment pour les deux cas.

Ce chapitre est organisé de la manière suivante : des définitions pour l'optimisation multi-objectif sont données dans la section 1.2. Une présentation de différentes approches de résolution est faite dans la section 1.3. Enfin, la section 1.4 présente différentes métriques pour évaluer la qualité des méthodes ou pour comparer plusieurs méthodes.

1.2 Définitions

1.2.1 Problème d'optimisation

Un problème d'optimisation consiste à rechercher la valeur maximale ou minimale, appelée optimum global, d'une fonction $f : \mathcal{S} \rightarrow \mathbb{R}$. f est appelée la fonction objectif. Comme maximiser une fonction f est équivalent à minimiser la fonction $-f$, nous considérons dans ce manuscrit que les fonctions doivent être minimisées.

\mathcal{S} est couramment appelé espace décisionnel et le plus souvent on utilise $\mathcal{S} = \mathbb{R}^m$. Un élément $x = (x_1, \dots, x_m)$ de \mathcal{S} est appelé une solution du problème d'optimisation. Les composantes x_i de x sont les variables de décision. C'est en faisant varier les valeurs de ces variables que l'on modifie la valeur de la fonction objectif. Une solution $x \in \mathcal{S}$ est dite réalisable si elle respecte un ensemble de contraintes C qui est de la forme $g(x) \geq 0$ avec $g : \mathcal{S} \rightarrow \mathbb{R}^k$. L'ensemble $\Omega \subseteq \mathcal{S}$ est l'ensemble des solutions réalisables que l'on appelle ensemble réalisable. Formellement, un problème d'optimisation s'écrit donc :

$$(PO) = \begin{cases} \min f(x) \\ \text{sous - contrainte} \\ g(x) \geq 0 \end{cases}$$

avec $x \in \mathcal{S}$.

L'optimum global d'un problème d'optimisation peut alors se définir ainsi [5] :

Définition 1 Pour $\Omega \neq \emptyset$, on note $f^* = f(x^*)$ avec $f^* > -\infty$ et $x^* \in \Omega$ si :

$$\forall x \in \Omega, f^* \leq f(x)$$

f^* est appelé l'optimum global de (Ω, f) .

Il est à noter que x^* peut ne pas être unique ; cependant il n'existe qu'un unique optimal global f^* .

1.2.2 Problème d'optimisation multi-objectif

Dans un problème d'optimisation multi-objectif, on ne cherche pas à optimiser une fonction objectif unique mais un vecteur de fonctions :

$$(PMO) = \begin{cases} \min F(x) = (f_1(x), f_2(x), \dots, f_n(x)) \\ \text{t.q. } x \in \Omega \end{cases}$$

où $n \geq 2$ est le nombre de fonctions objectif et $F(x) = (f_1(x), \dots, f_n(x))$ est le vecteur des fonctions à optimiser. Le vecteur x et les ensembles \mathcal{S} et Ω sont définis de la même manière que pour un problème d'optimisation mono-objectif.

Dans le cadre multi-objectif, le décideur raisonne plutôt en termes d'évaluation d'une solution pour chaque objectif et se place naturellement dans l'ensemble $F(\mathcal{S})$, appelé espace des objectifs. L'ensemble $\mathcal{Y} = F(\Omega)$ représente les points réalisables de l'espace des objectifs et $y = (y_1, \dots, y_n)$ avec $y_i = f_i(x)$, avec $x \in \Omega$, est un point de l'espace des objectifs.

1.2.3 Solution d'un problème d'optimisation multi-objectif

Alors que dans le cas d'un problème d'optimisation mono-objectif, il n'existe qu'un seul optimum global avec éventuellement plusieurs solutions correspondantes, ce n'est plus le cas pour un problème d'optimisation multi-objectif. Il faut donc utiliser une autre notion d'*optimum*. La plus communément admise est l'optimalité Pareto [52, 119]. On parle alors d'*optimum Pareto*. L'optimalité Pareto se définit comme suit :

Définition 2 Une solution $x^* \in \Omega$ est dite Pareto optimale si pour tout $x \in \Omega$, soit la proposition suivante est vraie :

$$\forall i \in \{1, \dots, n\}, f_i(x) = f_i(x^*)$$

soit la proposition suivante est vérifiée :

$$\exists i \in \{1, \dots, n\}, f_i(x) > f_i(x^*)$$

Cette définition peut s'interpréter de la manière suivante : une solution x^* est Pareto optimale s'il n'existe pas une solution réalisable x qui améliore la valeur d'un objectif sans détériorer au moins celle d'un autre objectif. Les solutions Pareto optimales sont aussi connues sous les noms de solutions admissibles, efficaces, non-dominées et non-inférieures.

D'autres notions sont associées à l'optimalité Pareto. Il est possible de définir une relation d'ordre partiel, appelée relation de dominance ou relation de dominance Pareto :

Définition 3 Une solution y domine (notée \preceq) une solution z si et seulement si $\forall i \in \{1, \dots, n\}, f_i(y) \leq f_i(z)$ et $\exists i \in \{1, \dots, n\}$ tel que $f_i(y) < f_i(z)$.

Dans l'exemple de la figure 1.1, les points A, C, E ne sont pas dominés par d'autres points. Ils sont donc Pareto optimaux. Tandis que le point B est dominé par le point C et le point D est dominé par les points C et E. Les points A, C, E sont au même niveau, c'est-à-dire que d'un point de vue multi-objectif il n'est pas possible d'établir une préférence pour un point plutôt que pour un autre.

Cette relation de dominance est parfois nommée relation de dominance *faible*. Cette définition peut être modifiée pour définir une relation de dominance *forte* :

Définition 4 Une solution y domine fortement une solution z (ou $y \prec z$) si la solution y est strictement meilleure que z sur tous les objectifs :

$$\forall i \in \{1, \dots, n\}, f_i(y) < f_i(z)$$

Le vecteur idéal est le point de $F(\mathcal{S})$ composé de l'optimum pour chaque objectif pris séparément. La condition nécessaire et suffisante pour que ce vecteur idéal soit atteint (c'est-à-dire qu'il corresponde à une solution réalisable) est que les fonctions objectif soient indépendantes. Si c'est le cas, il suffit de résoudre n problèmes mono-objectif. La solution d'un problème d'optimisation multi-objectif n'est donc pas une unique solution mais un ensemble de solutions appelé ensemble Pareto optimal :

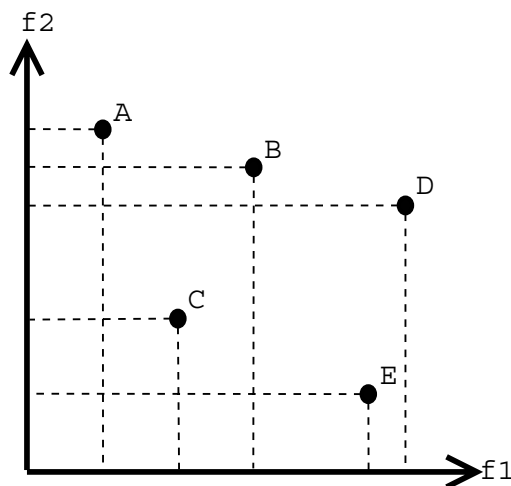


FIG. 1.1 – Exemple de dominance.

Définition 5 Pour un problème d'optimisation multi-objectif donné, l'ensemble Pareto optimal (\mathcal{P}^*) est défini comme suit :

$$\mathcal{P}^* = \{x \in \Omega \mid \nexists x' \in \Omega \ F(x') \preceq F(x)\}$$

L'ensemble Pareto optimal peut être divisé en deux sous-ensembles : l'ensemble des solutions supportées et l'ensemble des solutions non-supportées. Les solutions supportées sont les solutions Pareto optimales dont le point correspondant dans \mathcal{Y} se trouve sur l'enveloppe convexe de \mathcal{Y} . Les solutions Pareto n'appartenant pas à l'enveloppe convexe sont dites non-supportées.

L'image de l'ensemble Pareto optimal dans \mathcal{Y} est appelée frontière Pareto ou surface de compromis. Selon que les fonctions objectif doivent être minimisées ou maximisées, la frontière Pareto a une forme particulière. Les figures 1.2 à 1.5 illustrent les différentes possibilités dans le cas bi-objectif. Formellement, la frontière Pareto se définit comme suit :

Définition 6 Pour un problème d'optimisation multi-objectif donné et un ensemble Pareto optimal \mathcal{P}^* , la frontière Pareto (\mathcal{PF}^*) est définie comme suit :

$$\mathcal{PF}^* = \{u = (f_1(x), \dots, f_n(x)) \mid x \in \mathcal{P}^*\}$$

En général il n'est pas évident, voire impossible, de trouver une expression analytique de la ligne ou de la surface qui contient ces points. La procédure usuelle pour approximer la frontière Pareto est de calculer des solutions x et leur $F(x)$ correspondante. Quand un nombre suffisant de points a été généré, il est alors possible de déterminer les points non-dominés et de produire la frontière Pareto.

Les problèmes étudiés en optimisation sont souvent difficiles et appartiennent à la classe de complexité NP-difficile, ce qui rend impossible la recherche de l'optimum global

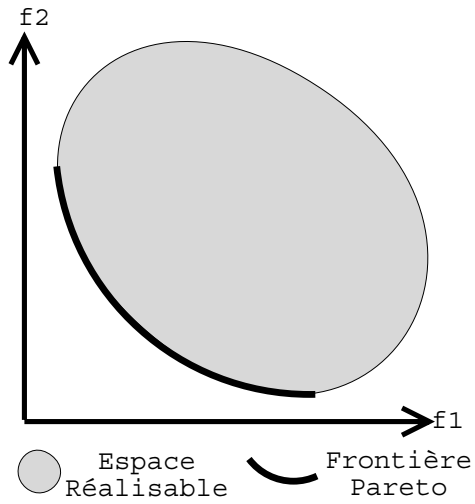


FIG. 1.2 – Frontière Pareto : minimiser f_1 , minimiser f_2 .

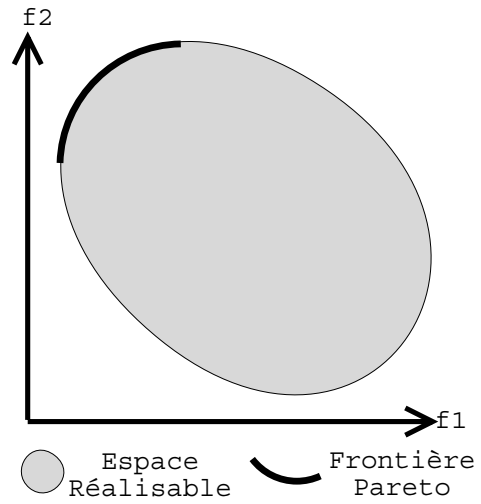


FIG. 1.3 – Frontière Pareto : minimiser f_1 , maximiser f_2 .

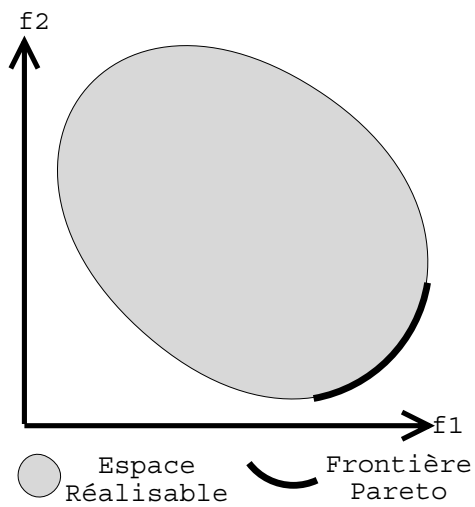


FIG. 1.4 – Frontière Pareto : maximiser f_1 , minimiser f_2 .

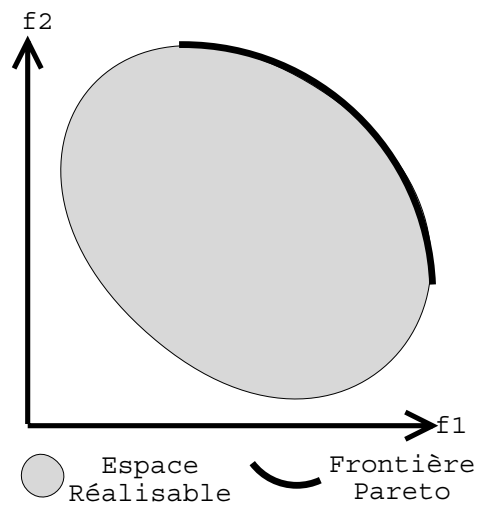


FIG. 1.5 – Frontière Pareto : maximiser f_1 , maximiser f_2 .

lorsque la taille du problème devient trop importante. Si un objectif d'un problème multi-objectif est NP-difficile alors le problème multi-objectif sera lui-même NP-difficile. Il est à noter que même si tous les objectifs sont solubles en un temps polynomial, alors, sauf dans certains cas particuliers, le problème multi-objectif n'est pas soluble en un temps polynomial [53]. Cela limite l'utilisation des méthodes exactes et encourage l'utilisation d'heuristiques dédiées et surtout de méta-heuristiques, notamment les méthodes à base de voisinage et les algorithmes génétiques. On ne cherche plus l'ensemble Pareto optimal mais une approximation de ce dernier. Pour un algorithme A donné, on parle alors de solutions potentiellement Pareto optimales qui forment l'ensemble potentiellement Pareto optimal :

Définition 7 Soit $A(\Omega)$ l'ensemble des solutions de Ω visitées par l'algorithme A . Une solution s_A^* est appelée solution potentiellement Pareto optimale par rapport à l'algorithme A si et seulement s'il n'existe pas de solution s appartenant à $A(\Omega)$ tel que $s \preceq s_A^*$.

Définition 8 Soit $A(\Omega)$ l'ensemble des solutions de Ω visitées par l'algorithme A . L'ensemble potentiellement Pareto optimal par rapport à A est :

$$\mathcal{P}_A = \{s_A^* \in A(\Omega) \mid \nexists s \in A(\Omega), s \preceq s_A^*\}$$

1.3 Approches de résolution

La présence de plusieurs objectifs contradictoires et l'absence d'ordre total entre les solutions demandent la mise au point de stratégies capables de prendre en compte ces particularités. Cette section présente les approches qui permettent de rendre les problèmes d'optimisation multi-objectif solubles. Nous présentons d'abord deux classifications de ces stratégies qui reposent sur deux points de vue différents du problème : celui du décideur et celui du concepteur. Puis nous classerons les principales stratégies par rapport à la classification du concepteur.

1.3.1 Classification

1.3.1.1 Du point de vue du décideur

La solution d'un problème multi-objectif est un ensemble de solutions. Cependant, pour un problème réel, une seule solution pourra être déployée. Un choix par un décideur doit donc être effectué ; le décideur pouvant intervenir en amont de la résolution, après celle-ci, ou de manière interactive [165] :

- Les méthodes à préférence *a priori* : le décideur définit ses préférences entre les différents objectifs avant d'utiliser la méthode d'optimisation.
- Les méthodes à préférence *progressive* : le décideur affine son choix de compromis au fur et à mesure du déroulement de la méthode d'optimisation.
- Les méthodes à préférence *a posteriori* : le décideur choisit la solution de son choix parmi l'ensemble des solutions fournies par la méthode d'optimisation.

Il est à noter que certaines méthodes n'entrent pas forcément dans une seule catégorie. En effet, la méthode qui consiste à agréger les différents objectifs à l'aide de poids est une méthode à préférence a priori ; le décideur affectant les poids de manière à favoriser tel ou tel objectif. Cependant, si les poids sont affectés aléatoirement et si la méthode est itérée en changeant les poids à chaque exécution, il s'agit alors d'une méthode à préférence a posteriori.

1.3.1.2 Du point de vue du concepteur

Une autre classification utilisée dans la littérature se place du point de vue du concepteur. Les méthodes sont classées dans trois catégories en fonction de la manière dont elles traitent la présence de plusieurs objectifs. Les trois catégories sont :

- approches scalaires : les méthodes sont basées sur la transformation du problème en un problème mono-objectif.
- approches Pareto : les méthodes utilisent directement la notion d'optimalité Pareto dans leur processus de recherche.
- approches non scalaires et non Pareto : ces approches ne transforment pas le problème multi-objectif en un problème mono-objectif mais utilisent des mécanismes de recherche qui traitent séparément les différents objectifs.

1.3.2 Approches scalaires

Dans la résolution de problèmes multi-objectif, plusieurs méthodes traditionnelles transforment le problème multi-objectif en un problème mono-objectif. Parmi ces méthodes on trouve les méthodes d'agrégation, les méthodes avec vecteur cible et les méthodes ϵ -contraintes.

1.3.2.1 Méthode d'agrégation

C'est l'une des premières méthodes employées pour la génération de solutions Pareto optimales. Elle consiste à transformer un problème multi-objectif en un problème qui combine les différentes fonctions objectif du problème en une seule fonction U de façon linéaire :

$$U(x) = \sum_{i=1}^n \lambda_i f_i(x)$$

où les poids λ_i sont compris dans l'intervalle $[0, 1]$ et vérifient $\sum_{i=1}^n \lambda_i = 1$. Différents poids fournissent différentes solutions supportées ; une même solution pouvant être générée en utilisant des poids différents. Il existe d'autres approches agrégatives mais l'agrégation linéaire est la plus couramment utilisée.

La figure 1.6 illustre le fonctionnement de la méthode d'agrégation. Fixer un vecteur de poids revient à trouver un hyperplan dans l'espace objectif (une droite pour un problème bi-objectif) avec une orientation fixée. La solution optimale est le point où l'hyperplan possède une tangente commune avec l'espace réalisable (point x dans la figure).

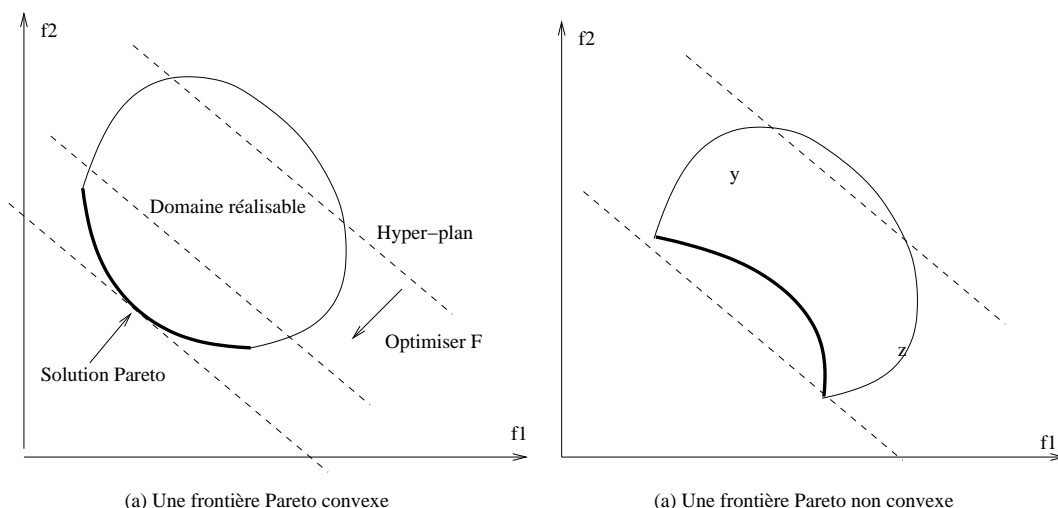


FIG. 1.6 – La méthode d'agrégation.

Les résultats obtenus dans la résolution du problème dépendent fortement des paramètres choisis pour le vecteur de poids λ . Les poids λ_i indiquent la préférence attribuée aux différents objectifs ; leur choix est donc une tâche délicate. Ainsi, une approche généralement utilisée consiste à résoudre le problème avec différentes valeurs de λ .

Si les différents objectifs sont non-commensurables, on peut transformer l'équation précédente sous la forme :

$$U(x) = \sum_{i=1}^n c_i \lambda_i f_i(x)$$

où c_i sont les constantes qui mettent les différents objectifs à la même échelle. Les constantes c_i sont généralement initialisées à $\frac{1}{f_i(x^*)}$ où $f_i(x^*)$ est la valeur optimale associée à la fonction objectif f_i . Dans ce cas, le vecteur est normalisé par rapport au vecteur idéal.

L'avantage de cette méthode est sa facilité d'implémentation et le fait qu'elle puisse être utilisée avec les méthodes et mécanismes définis pour l'optimisation mono-objectif. Cependant, Das et Dennis[41] ont montré que cette méthode est incapable de trouver les zones concaves du front Pareto quels que soient les poids choisis. Dans la figure 1.6b, où le problème possède une frontière Pareto non convexe, seules les solutions extrêmes y et z peuvent être générées. De plus, la solution trouvée peut ne pas être acceptable si l'espace de recherche est réduit de façon prématurée avant que des informations suffisantes soient disponibles.

Enfin, la détermination des poids peut être délicate si on ne dispose pas de connaissances suffisantes sur le problème. Plusieurs stratégies "aveugles" peuvent être utilisées pour générer les poids. Dans [84], les poids sont générés de manière aléatoire :

$$\lambda_i = \frac{random_i}{random_2 + \dots + random_n}, \quad i = 1, \dots, n$$

où les variables $random_i$ sont des entiers positifs.

cette méthode, largement employée dans la littérature, a été utilisée notamment pour le design de circuit de traitement de signal digital [4], de contrôle de la qualité de l'eau [66] ou le design de filtres optiques pour lampes [54].

1.3.2.2 Méthodes avec vecteur cible

Dans les méthodes de ce type, un ensemble de buts (ou cibles) que l'on désire atteindre est défini. La méthode d'optimisation essaie alors de minimiser la différence entre la solution courante et les buts. Bien que les méthodes de ce type puissent être considérées comme des approches agrégatives, elles sont capables de générer des solutions non-supportées. Ces méthodes ont l'avantage d'être efficaces et faciles à implémenter. Cependant, la définition des cibles peut être difficile et demander des calculs supplémentaires. Ces méthodes ont été utilisées pour des calculs en télécommunication [168], l'optimisation d'un modèle bioéconomique de la pêche [107] ou l'optimisation du contrepois d'un bras robot [25]. Des exemples d'approches avec vecteur cible sont la méthode du but à atteindre, la méthode du but programmé ou la méthode du min-max.

Méthode du but à atteindre : dans cette méthode, le décideur définit un ensemble de buts T_i à atteindre pour les différents objectifs, ainsi qu'un poids w_i . Le problème suivant est alors résolu :

$$(PMO) = \begin{cases} \min \alpha \\ t.q. \\ x \in \Omega \\ T_i + \alpha w_i \geq f_i(x), \quad i = 1, \dots, n \end{cases}$$

avec $\sum_{i=1}^n w_i = 1$. Les objectifs T_i représentent le point de départ de la recherche dans l'espace et les poids w_i indiquent la direction de recherche dans l'espace.

La figure 1.7 illustre cette méthode dans le cas bi-objectif. Le point A représente l'objectif de référence, le vecteur w représente une direction de recherche et le point B représente le but à atteindre.

Au contraire de la méthode de la somme pondérée, cette méthode est capable de générer des solutions non-supportées en faisant varier les valeurs des poids w_i [21]. Cependant, il peut exister des formes de domaine pour lesquelles la solution dépend du choix du but initial (cf. figure 1.8).

Méthode du but programmé : dans cette méthode, le décideur doit définir des buts ou références qu'il désire atteindre pour chaque objectif. Ces valeurs sont introduites dans la formulation du problème, le transformant en un problème uni-objectif. Par exemple, la fonction de coût peut intégrer une norme pondérée qui minimise les déviations par rapport aux buts. Le problème peut être formulé de la manière suivante :

$$(PMO) = \begin{cases} \min(\sum_{i=1}^n \lambda_i |f_i(x) - z_i|^p)^{\frac{1}{p}} \\ t.q. \\ x \in \Omega \end{cases}$$

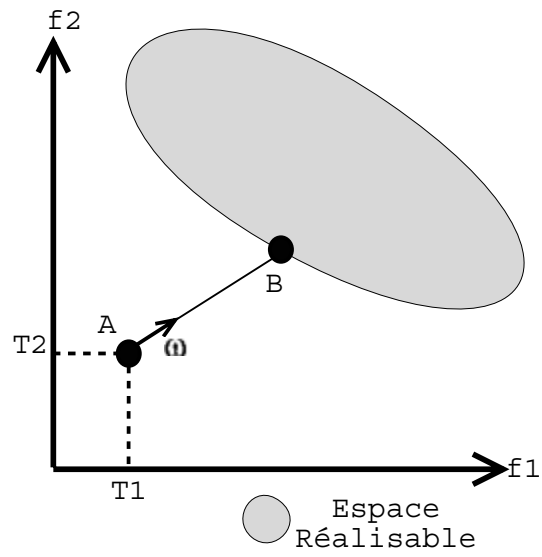


FIG. 1.7 – La méthode du but à atteindre.

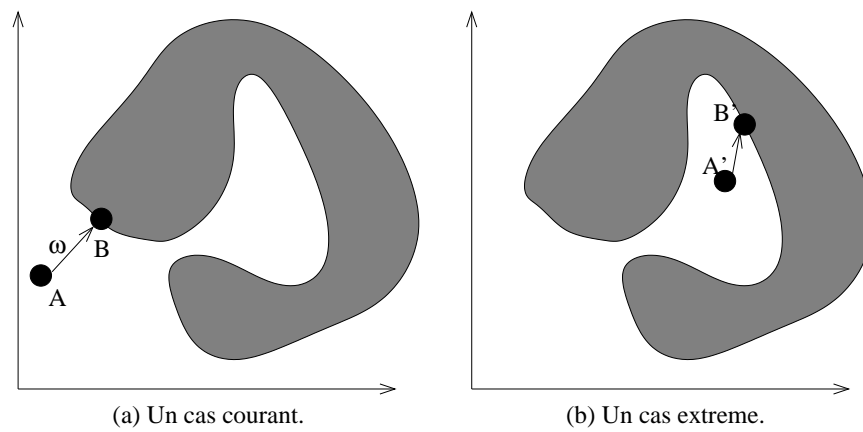


FIG. 1.8 – Une difficulté pour la méthode du but à atteindre.

où $1 \leq p \leq \infty$ et $z = (z_1, \dots, z_n)$ est le vecteur de référence. La norme utilisée est la métrique de Tchebycheff (L_p -métrique). Généralement on utilise $p = 2$, c'est-à-dire une métrique Euclidienne. Si $p = \infty$, l'équation revient à une fonction min-max.

La méthode est facile à mettre en place mais la définition des poids et des objectifs à atteindre est une question délicate qui détermine l'efficacité de la méthode. Cependant, cette méthode a l'avantage de fournir un résultat même si un mauvais choix initial a conduit le décideur à donner un ou plusieurs but(s) non réalisable(s).

Méthode Min-Max : Cette méthode est issue de la théorie des jeux qui traite de la résolution de situations conflictuelles. Elle minimise le maximum de l'écart relatif entre le $i^{\text{ème}}$ objectif et son but T_i associé par le décideur :

$$\min \max_i \left(\frac{f_i(x) - T_i}{T_i} \right) \text{ avec } i = 1, \dots, n$$

Plusieurs variantes de la méthode peuvent être trouvées dans la littérature et notamment dans [25].

1.3.2.3 Méthode ϵ -contrainte

dans cette méthode, une seule fonction objectif f_k est optimisée tandis que les autres fonctions objectif sont sujettes à des contraintes. Le problème d'optimisation s'écrit alors :

$$(PMO_\epsilon) = \begin{cases} \min f_k(x) \\ t.q. \\ x \in \Omega \\ f_j(x) \leq \epsilon_j, j = 1, \dots, n, j \neq k \end{cases}$$

La figure 1.9 illustre la méthode ϵ -contrainte où $F(C)$ est l'espace des objectifs original, restreint à $F'(C)$ par la transformation du problème, en minimisant la fonction f_1 et en ajoutant la contrainte $f_2(c) \leq \epsilon_2$.

L'objectif f_k représente l'objectif primaire ou l'objectif préféré. L'ensemble Pareto optimal est généré en faisant varier les valeurs des ϵ_j . Il est ainsi possible de générer l'ensemble Pareto optimal en n'ayant à résoudre qu'un ensemble de problèmes d'optimisation mono-objectif. Il est à noter que cette méthode est capable de générer les solutions non supportées. Elle peut cependant être coûteuse en temps de calcul et la méthode d'optimisation peut être difficile à programmer à cause du nombre de contraintes additionnelles.

Par exemple, cette méthode a été utilisée pour le design de véhicules marins [99] ou le design de systèmes tolérants aux pannes [141].

1.3.3 Approches Pareto

Les approches Pareto utilisent directement la notion de dominance au sens de Pareto. Cette idée a été introduite initialement dans les algorithmes génétiques par Goldberg [73]

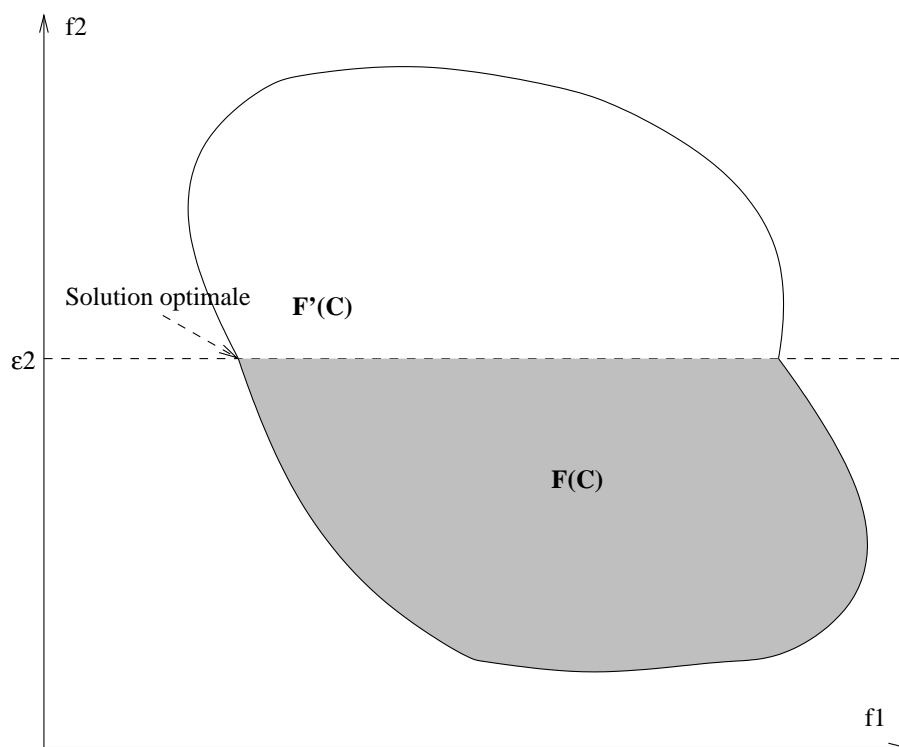


FIG. 1.9 – La méthode ϵ -contrainte.

pour résoudre les problèmes proposés par Schaffer [138]. Ce concept permet de ne pas favoriser un compromis plutôt qu'un autre mais il permet de fournir une aide précieuse au décideur. L'utilisation de la dominance est principalement à la base du développement des algorithmes évolutionnaires multi-objectif présentés dans le chapitre 3.

1.3.4 Approches non scalaires et non Pareto

Ces approches sont basées sur des populations de solutions. La recherche est réalisée en traitant séparément les différents objectifs.

1.3.4.1 Sélection parallèle

Le premier travail consistant à utiliser des algorithmes génétiques pour résoudre des problèmes d'optimisation multi-objectif est celui de Schaffer [138]. L'algorithme développé VEGA (Vector Evaluated Genetic Algorithm) sélectionne les individus de la population courante suivant chaque objectif, indépendamment des autres (sélection parallèle). A chaque génération, la population est donc divisée en un nombre de sous-populations qui est égal au nombre de fonctions objectif composant le vecteur objectif. Ainsi, dans le cas d'un problème à n objectifs résolu par un algorithme génétique travaillant sur k individus, n sous-populations de $\frac{k}{n}$ individus sont créées. Les n populations sont ensuite mélangées

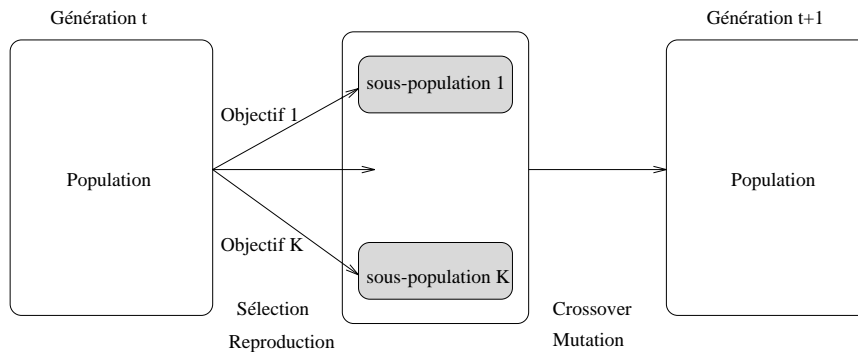


FIG. 1.10 – Sélection parallèle dans l’algorithme VEGA.

pour obtenir une nouvelle population à k individus sur laquelle les opérateurs génétiques sont appliqués. Une génération de VEGA est résumée dans la figure 1.10.

Comme l’algorithme affecte aléatoirement les individus aux sous-populations à chaque génération, un même individu peut être évalué différemment suivant l’objectif qui lui est affecté d’une génération à l’autre. L’analyse de l’algorithme VEGA a montré que son comportement est le même qu’un algorithme réalisant une agrégation linéaire [153]. Shaffer a cependant reporté de bons résultats dans la recherche des solutions Pareto optimales dans le cas de fonctions réelles. Les principaux problèmes de VEGA sont qu’il ne dispose pas de mécanisme pour maintenir la diversité et qu’il ne produit pas forcément que des solutions non-dominées. D’autre part, il est difficile d’obtenir des compromis entre les objectifs ; il encourage les solutions extrêmes.

Des exemples d’applications de VEGA sont l’optimisation aérodynamique [136], le design de circuits combinatoires [26] ou la retenue d’eau polluée [133]. Par ailleurs, de nombreuses variantes de cette technique ont été proposées : combinaison de VEGA avec la dominance Pareto [154], paramètre pour contrôler le taux de sélection [133], application à un problème contraint [149], utilisation d’un vecteur contenant les probabilités d’utiliser un certain objectif lors de la sélection [96].

1.3.4.2 Méthode lexicographique

Cette méthode, proposée par Fourman [64], classe les objectifs en fonction d’un ordre d’importance proposé par le décideur. Ensuite les fonctions objectif sont traitées dans cet ordre pour obtenir l’optimum.

La méthode lexicographique peut s’exprimer de la manière suivante : supposons que l’on ait les fonctions objectif f_i avec $i = 1, \dots, n$ classées de telle sorte que si $i < j$ alors f_i est prioritaire sur f_j . On résout d’abord le problème :

$$(PMO_{l_1}) = \begin{cases} \min f_1(x) \\ t.q. \\ x \in \Omega \end{cases}$$

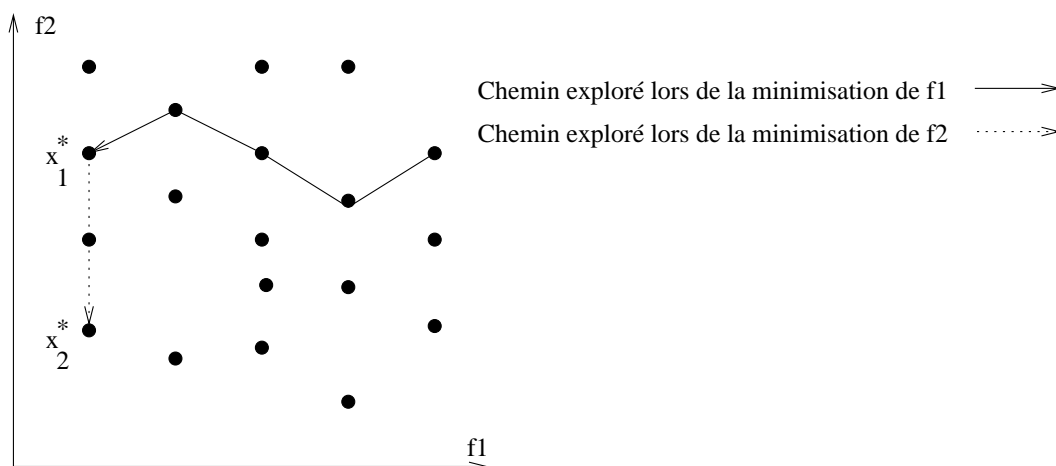


FIG. 1.11 – La méthode lexicographique dans le cas bi-objectif.

On obtient alors x_1^* la meilleure solution trouvée pour f_1 , associée à la valeur $f_1^* = f_1(x_1^*)$. f_1^* devient alors une contrainte du problème associé à f_2 :

$$(PMO_{l2}) = \begin{cases} \min f_2(x) \\ t.q. \\ x \in \Omega \\ f_1(x) = f_1^* \end{cases}$$

La meilleure solution trouvée est alors notée x_2^* et on pose $f_2^* = f_2(x_2^*)$ comme contrainte supplémentaire pour la résolution de f_3 . La procédure est itérée jusqu'à ce que tous les objectifs aient été traités. Ainsi le problème associé à la fonction objectif f_i est le suivant :

$$(PMO_{li}) = \begin{cases} \min f_i(x) \\ t.q. \\ x \in \Omega \\ f_1(x) = f_1^* \\ f_2(x) = f_2^* \\ \dots \\ f_{i-1}(x) = f_{i-1}^* \end{cases}$$

où $f_j^* = f_j(x_j^*)$ avec x_j^* la meilleure solution trouvée en optimisant la fonction objectif f_j avec $f_1(x) = f_1^*, f_2(x) = f_2^*, \dots, f_{j-1}(x) = f_{j-1}^*$ comme contraintes additionnelles. La figure 1.11 illustre le fonctionnement de la méthode lexicographique dans le cas d'un problème bi-objectif.

Fourman a aussi proposé une version de cet algorithme qui choisit aléatoirement la fonction objectif devant être prise en compte. Cette stratégie revient en fait à utiliser une somme pondérée dans laquelle un poids correspond à la probabilité que la fonction objectif associée soit sélectionnée.

Bien que cette méthode soit facile à implémenter et efficace sur certains problèmes, elle présente de nombreux inconvénients. Premièrement, elle demande de définir au préalable un ordre entre les objectifs, ordre qui a un impact sur les performances de la méthode. En effet, la première solution f_1^* trouvée va faire converger la méthode vers une zone restreinte de l'espace objectif. Deuxièmement, elle est inappropriée quand un grand nombre d'objectifs doit être traité.

Des exemples d'applications de la méthode lexicographique sont le tassement de tracés symboliques [64], la recherche d'un chemin pour un robot [65] et l'ordonnancement de personnel [55].

1.3.4.3 Méthode avec genre

Proposée par Allenson [3] pour traiter un problème bi-objectif, cette méthode utilise la notion de genre (masculin ou féminin) et d'attracteur. En affectant un objectif à chaque genre, l'auteur cherche à minimiser les deux objectifs simultanément car un individu sera toujours jugé d'après l'objectif qui lui a été associé.

Dans cette méthode, si on est par exemple dans un cas bi-objectif et que les deux genres sont mâle et femelle, la reproduction est permise uniquement entre un individu mâle et un individu femelle, le genre étant affecté aléatoirement à la création de l'individu. A l'initialisation, le nombre d'individus par genre est le même. Allenson utilise un algorithme génétique classique où les enfants remplacent les plus mauvais individus du même genre. Lors de la phase de sélection, Allenson introduit le principe d'attracteur pour éviter que des individus trop éloignés du point de vue de leur fitness puissent être croisés et pour permettre à des individus de mauvais fitness de se reproduire.

Lis et Eiben [105] ont généralisé cette approche dans les algorithmes génétiques. L'opérateur de croisement a été modifié pour porter sur plusieurs individus au lieu de deux seulement. La sélection d'un parent de chaque genre permet ainsi à tous les genres de contribuer à la génération d'un nouvel individu. Ce dernier aura le genre du parent dont il hérite le plus grand nombre de gènes et il réalise donc un compromis entre les différents objectifs.

L'utilisation des genres est une manière de définir des sous-populations séparées par rapport à chaque objectif. La différence avec la sélection parallèle et lexicographique est que la reproduction impose une restriction de voisinage, évitant la reproduction aléatoire entre individus. L'inconvénient de ces méthodes est qu'elles tendent à générer des solutions qui sont de très bonne qualité pour certains objectifs et l'inverse pour les autres objectifs. Les solutions compromises sont négligées.

1.4 Métriques pour l'évaluation des performances

De nombreuses mesures pour calculer l'efficacité des méthodes d'optimisation pour les problèmes multi-objectif ont été proposées. Dans cette section, les plus couramment utilisées sont décrites. Une des difficultés provient du fait qu'il existe plusieurs mesures

qui évaluent un même aspect. Cependant, comme chaque métrique présente à la fois des avantages et des inconvénients, il n'est pas évident de choisir laquelle appliquer. Toutefois, il est possible de trouver des études comme celle de Knowles et Corne [92] qui comparent plusieurs métriques.

Nous avons séparé les métriques en deux ensembles : celles qui offrent une mesure absolue par rapport à \mathcal{P}^* et celles pour lesquelles la connaissance de \mathcal{P}^* n'est pas nécessaire et qui offrent une mesure relative ou comparaison.

Les mesures peuvent servir à quantifier deux aspects des résultats. En effet, si le but de la méthode n'est pas de rechercher une unique solution en prenant en compte les préférences du décideur mais de générer un ensemble de solutions potentiellement Pareto (cas a posteriori), alors le but de la méthode est double. Premièrement, la méthode doit converger vers le front Pareto optimal ; deuxièmement, elle doit trouver des solutions tout le long de cette frontière. On parle alors de diversification. La figure 1.12 illustre les deux buts d'une méthode d'optimisation a posteriori. La figure 1.13 représente un ensemble de solutions répondant idéalement aux deux buts exposés ci-dessus. La figure 1.14 montre une approximation qui répond au premier but mais qui est mauvaise par rapport au second. La figure 1.15 illustre le cas opposé.

Dans le reste de cette section, \mathcal{P}_A représente l'ensemble des solutions potentiellement Pareto optimales trouvées par un algorithme A .

1.4.1 Ensemble \mathcal{P}^* connu

1.4.1.1 Proportion d'erreur

Cette mesure, proposée par Van Veldhuizen [165], compte le nombre de solutions de \mathcal{P}_A qui n'appartiennent pas à \mathcal{P}^* , soit :

$$PE(A) = \frac{\sum_{i=1}^{|\mathcal{P}_A|} e_i}{|\mathcal{P}_A|}$$

où $e_i = 1$ si la $i^{\text{ème}}$ solution de \mathcal{P}_A appartient à \mathcal{P}^* , sinon $e_i = 0$.

La méthode possède un désavantage. En effet, si aucune solution de \mathcal{P}_A n'appartient à \mathcal{P}^* , elle n'apporte aucune information quant à la proximité relative de \mathcal{P}_A par rapport à \mathcal{P}^* .

1.4.1.2 Distance générationnelle

Cette mesure, introduite par Van Veldhuizen [165], calcule la distance moyenne entre les solutions de \mathcal{P}_A et celles de \mathcal{P}^* . Elle est donnée par la formule suivante :

$$DG(A) = \frac{(\sum_{i=1}^{|\mathcal{P}_A|} d_i^p)^{\frac{1}{p}}}{|\mathcal{P}_A|}$$

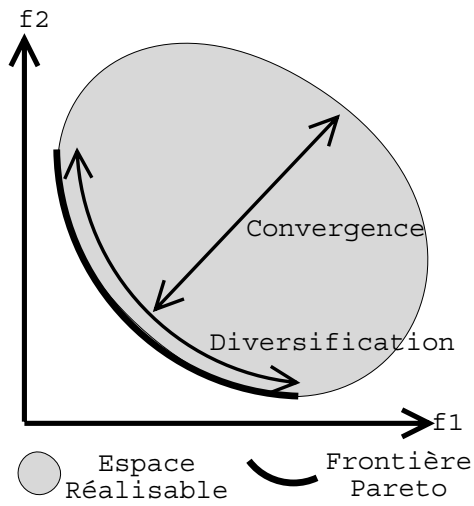


FIG. 1.12 – Les deux buts d'une méthode multi-objectif.

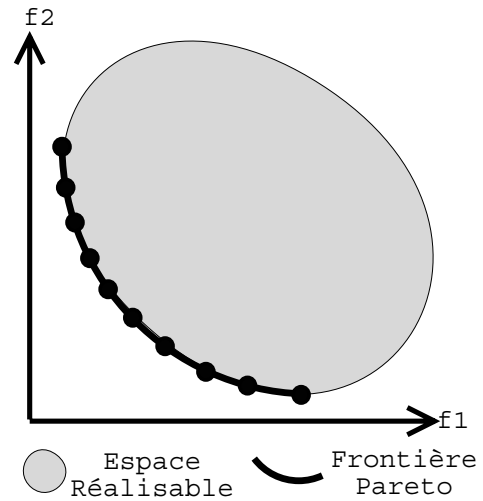


FIG. 1.13 – Une solution idéale.

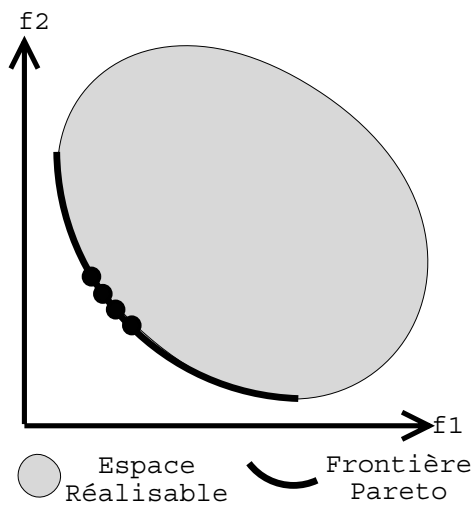


FIG. 1.14 – Une solution qui répond au but de convergence mais qui est mauvaise pour le but de diversification.

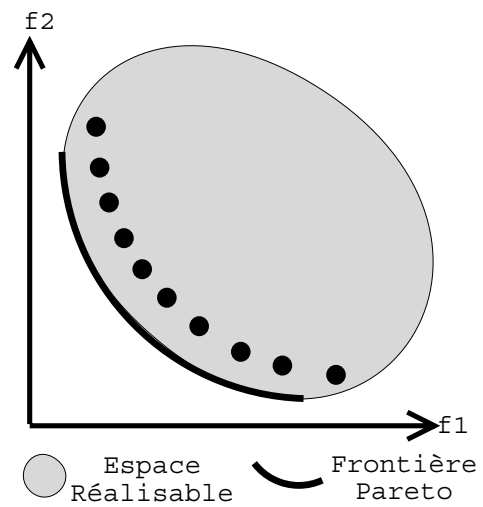


FIG. 1.15 – Une solution qui répond au but de diversification mais qui est mauvaise pour le but de convergence.

Pour $p = 2$, le paramètre d_i est la distance Euclidienne (dans l'espace des objectifs) entre la solution $i \in \mathcal{P}_A$ et le membre le plus proche de \mathcal{P}^* :

$$d_i = \min_{k=1}^{|\mathcal{P}^*|} \sqrt{\sum_{j=1}^n (f_j^i - f_j^k)^2}$$

où f_j^i est la valeur de la $j^{\text{ème}}$ fonction objectif de i et f_j^k est la valeur de la $j^{\text{ème}}$ fonction objectif de la $k^{\text{ème}}$ solution de \mathcal{P}^* .

Cette méthode présente deux difficultés. Premièrement, la métrique peut retourner une valeur faussée et peu représentative s'il y a une forte fluctuation dans les distances entre les points de l'approximation et ceux de l'ensemble optimal. Par exemple, dans le cas où l'approximation est très proche de l'ensemble optimal sauf pour un point très éloigné, c'est alors ce dernier qui augmentera de manière significative la distance générationnelle alors qu'il s'agit d'un cas isolé. Dans un tel cas, le calcul de l'écart type de la mesure GD est nécessaire. La seconde difficulté apparaît quand les fonctions objectif ne sont pas de même amplitude ; elles doivent alors être normalisées avant le calcul.

1.4.1.3 Erreur maximale à la surface de compromis

Cette métrique permet de mesurer la distance entre la surface de compromis et l'ensemble de solutions. Sa définition pour un problème à deux objectifs est la suivante :

$$EM(A) = \max_j (\min_i (|f_1^i(x) - f_1^j(x)|^p + |f_2^i(x) - f_1^j(x)|^{\frac{1}{p}}))$$

Elle retourne en fait la plus grande distance minimale entre les solutions de \mathcal{P}_A et les solutions les plus proches de \mathcal{P}^* .

1.4.2 Ensemble \mathcal{P}^* inconnu

Lorsque l'ensemble \mathcal{P}^* est connu, les mesures calculent le plus souvent la distance entre l'approximation et \mathcal{P}^* . Lorsque l'ensemble \mathcal{P}^* est inconnu, elles permettent alors de comparer de manière relative deux approximations. Il existe cependant des mesures qui renvoient une évaluation de la qualité d'une seule approximation. Les mesures évaluent la qualité des approximations soit par rapport à la convergence, soit par rapport à la diversification ou par rapport aux deux buts en même temps.

1.4.2.1 Mesures évaluant la convergence

La métrique \mathcal{C} : cette mesure, proposée par Zitzler [172], calcule la proportion de solutions de \mathcal{P}_B dominées par des solutions de \mathcal{P}_A :

$$\mathcal{C}(A, B) = \frac{|\{b \in \mathcal{P}_B | \exists a \in \mathcal{P}_A : a \prec b\}|}{|\mathcal{P}_B|}$$

$\mathcal{C}(A, B) = 1$ signifie que toutes les solutions trouvées par l'algorithme B sont dominées par celles trouvées par l'algorithme A . Tandis que $\mathcal{C}(A, B) = 0$ indique qu'aucune solution générée par B n'est dominée par une solution trouvée par A . Comme la relation de dominance n'est pas symétrique, $\mathcal{C}(A, B)$ n'est pas forcément égal à $1 - \mathcal{C}(B, A)$. Il est donc nécessaire de calculer $\mathcal{C}(A, B)$ et $\mathcal{C}(B, A)$. Il est intéressant de noter que les cardinalités de \mathcal{P}_A et \mathcal{P}_B ne doivent pas forcément être identiques.

Contribution : cette mesure [14, 108] calcule la proportion des solutions d'un ensemble potentiellement Pareto optimal \mathcal{P}_A généré par l'algorithme A par rapport à l'ensemble \mathcal{P}_B obtenu par l'algorithme B .

Soit C l'ensemble des solutions potentiellement Pareto optimales communes à A et B :

$$C = \mathcal{P}_A \cap \mathcal{P}_B$$

Soit W_A (respectivement W_B) l'ensemble des solutions de \mathcal{P}_A (respectivement \mathcal{P}_B) qui dominent des solutions dans \mathcal{P}_B (respectivement \mathcal{P}_A). Soit L_A (respectivement L_B) l'ensemble des solutions de \mathcal{P}_A (respectivement \mathcal{P}_B) dominées par des solutions dans \mathcal{P}_B (respectivement \mathcal{P}_A). L'ensemble N_A (respectivement N_B) est formé des solutions de \mathcal{P}_A (respectivement \mathcal{P}_B) étant non dominantes non dominées par rapport à toutes les solutions de \mathcal{P}_B (respectivement \mathcal{P}_A). Formellement, on a la relation suivante :

$$N_A = \mathcal{P}_A \setminus (C \cup W_A \cup L_B)$$

Soit PO l'ensemble des solutions potentiellement Pareto optimales trouvées par les deux algorithmes A et B :

$$PO = C \cup W_A \cup N_A \cup W_B \cup N_B$$

La contribution de l'algorithme A relativement à l'algorithme B , dénotée par $Cont(A, B)$, est le ratio des solutions non dominées produites par A par rapport à PO :

$$Cont(A, B) = \frac{\frac{|C|}{2} + |W_A| + |N_A|}{|C| + |W_A| + |N_A| + |W_B| + |N_B|} = \frac{\frac{|C|}{2} + |W_A| + |N_A|}{|PO|}$$

La contribution de l'algorithme B relativement à l'algorithme A est définie d'une manière similaire. La relation suivante est toujours vraie :

$$Cont(A, B) + Cont(B, A) = 1$$

D'autre part, notons que si les deux algorithmes produisent les mêmes solutions alors :

$$Cont(A, B) = Cont(B, A) = \frac{1}{2}$$

tandis que si toutes les solutions produites par B sont dominées par les solutions produites par A , on a :

$$Cont(B, A) = 0$$

Cette mesure permet d'avoir rapidement une idée de l'apport d'un algorithme par rapport à un autre algorithme. Cependant, l'information fournie est moins claire que celle de la métrique C de Zitzler présentée ci-dessus.

1.4.2.2 Mesures évaluant la diversité

La métrique Spacing : cette métrique, proposée par Schott [141], calcule la distance relative entre deux solutions consécutives de \mathcal{P}_A de la manière suivante :

$$S = \sqrt{\frac{1}{|\mathcal{P}_A|} \sum_{i=1}^{|\mathcal{P}_A|} (d_i - \bar{d})^2}$$

où $d_i = \min_{k \in \mathcal{P}_A \wedge k \neq i} \sum_{m=1}^n |f_m^i - f_m^k|$ et \bar{d} est la valeur moyenne de la distance précédente $\bar{d} = \frac{\sum_{i=1}^{|\mathcal{P}_A|} d_i}{|\mathcal{P}_A|}$. La distance d_i est la valeur minimale de la somme des différences absolues des valeurs des fonctions objectif entre la $i^{\text{ème}}$ solution et toutes les autres solutions de l'ensemble généré. Il est à noter que cette distance est différente de la distance Euclidienne minimale entre deux solutions.

Cette métrique calcule les écarts types des différentes valeurs de d_i . Ainsi, si les solutions sont uniformément espacées, la distance correspondante sera faible. Donc, un algorithme qui trouve un ensemble de solutions potentiellement Pareto optimales pour lequel cette mesure est faible est meilleur. La pertinence de l'information retournée par cette mesure est discutable. En effet, si on considère les trois cas exposés dans les figures 1.16, 1.17 et 1.18, il apparaît que même si les ensembles des figures 1.16 et 1.17 ont le même spacing, la seconde figure est tout de même plus intéressante. D'autre part, l'ensemble de la figure 1.18, qui offre de loin le meilleur résultat, obtient le plus mauvais spacing.

Métrique *Maximum spread* : Zitzler [172] définit une métrique mesurant la longueur de la diagonale d'une hyperboîte formée par les valeurs des fonctions objectif extrêmes de l'ensemble potentiellement Pareto optimal généré :

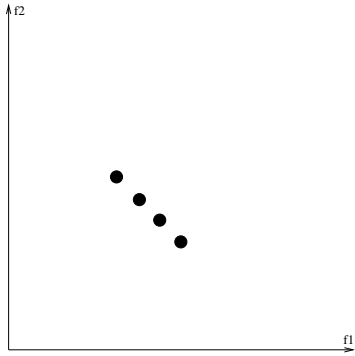
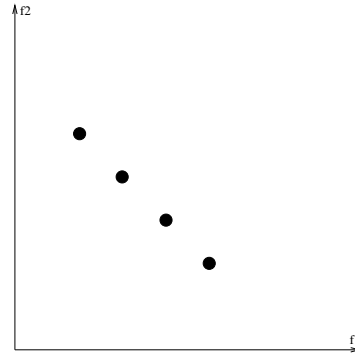
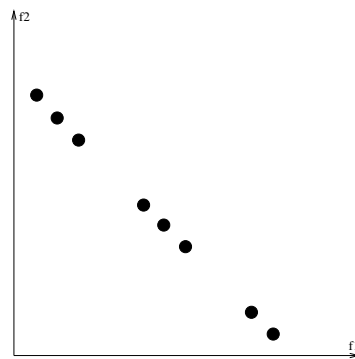
$$D = \sqrt{\frac{1}{|\mathcal{P}_A|} \sum_{m=1}^n \left(\frac{\max_{i=1}^{|\mathcal{P}_A|} f_m^i - \min_{i=1}^{|\mathcal{P}_A|} f_m^i}{F_m^{\max} - F_m^{\min}} \right)^2}$$

où F_m^{\max} et F_m^{\min} sont le maximum et le minimum pour le $m^{\text{ème}}$ objectif. Le problème de cette mesure est qu'elle ne fournit aucune information sur la distribution exacte des solutions de compromis.

Entropie : l'entropie [14, 108] permet d'évaluer la diversité d'une approximation \mathcal{P}_A produite par un algorithme A par rapport à la diversité d'une approximation \mathcal{P}_B produite par un algorithme B .

Dans cette mesure, une niche est associée à chaque solution. Les solutions présentes dans chaque niche sont considérées comme voisines de la solution associée à la niche. L'entropie est alors donnée par la mesure :

$$E(A, B) = \frac{-1}{\log \gamma} \sum_{i=1}^C \frac{1}{N_i} \frac{n_i}{C} \log \frac{n_i}{C}$$

FIG. 1.16 – (a) $S = 0$.FIG. 1.17 – (b) $S = 0$.FIG. 1.18 – (c) $S > 0$.

où N_i est le nombre de solutions de $\mathcal{P}_A \cup PO$ (PO étant l'ensemble des solutions non dominées de l'union de \mathcal{P}_A et \mathcal{P}_B) se trouvant dans la niche de la $i^{\text{ème}}$ solution de $\mathcal{P}_A \cup PO$; C est la cardinalité de $\mathcal{P}_A \cup PO$; n_i est le nombre de solutions de l'ensemble \mathcal{P}_A dans la niche de la $i^{\text{ème}}$ solution de $\mathcal{P}_A \cup PO$ et $\gamma = \sum_{i=1}^C \frac{1}{N_i}$ représente la somme des coefficients affectés à chaque solution.

Cette mesure permet donc une estimation de la diversité relative entre deux approximations. Toutefois, l'interprétation des résultats n'est pas toujours évidente car elle n'a pas d'ordre de grandeur et une faible fluctuation entre $E(A, B)$ et $E(B, A)$ peut cacher une différence importante en termes de diversité.

1.4.2.3 Mesures évaluant la convergence et la diversité

Métrique S : cette mesure, proposée par Zitzler [172], calcule l'hypervolume de la région multidimensionnelle fermée par \mathcal{P}_A et un point de référence; c'est-à-dire la taille de l'espace des objectifs que \mathcal{P}_A domine.

L'avantage de cette mesure, dont la signification est intuitive, est qu'elle offre un ordre total entre différentes approximations. Cependant, elle nécessite la définition d'une borne supérieure de la région dans laquelle se trouvent tous les points réalisables. Ce choix peut

avoir un impact sur l'ordre entre les approximations (cf figure 1.19). De plus, le temps de calcul est important : $O(|\mathcal{P}_A|^{n+1})$. De ce fait, la mesure est impraticable dans le cas où de nombreux objectifs sont considérés ou si la cardinalité des ensembles est trop importante.

Dans leur étude [92], Knowles et Corne recommandent l'utilisation de cette mesure.

1.5 Conclusion

Dans ce chapitre, nous avons introduit les notations et définitions portant sur l'optimisation multi-objectif utilisées dans ce manuscrit. Nous avons aussi présenté les principales méthodes pour traiter la présence d'objectifs conflictuels. Les méthodes proposées dans cette thèse sont centrées autour de l'utilisation d'un algorithme génétique utilisant une approche Pareto. Cependant, nous avons défini pour cet algorithme un mécanisme qui utilise des sous-populations et qui est donc proche de ce qui peut se faire dans des approches comme la sélection parallèle. D'autre part, nous avons aussi utilisé des méthodes à base de voisinage ainsi qu'un algorithme de séparations et coupes. Or, pour ces méthodes, l'application d'approches de type Pareto peut ne pas être aisée et nous avons donc exploré l'utilisation de méthodes scalaires comme la méthode du but programmé et la méthode ϵ -contrainte. Enfin, ce chapitre se clôt sur une présentation de différents outils pour la mesure et la comparaison dans le cadre de l'optimisation multi-objectif. Il s'agit d'un thème très important d'autant plus qu'aucune mesure proposée n'est capable de faire l'unanimité. Dans cette thèse, lorsque l'ensemble Pareto optimal est connu, nous avons utilisé le ratio du nombre de solutions Pareto optimales trouvées ainsi que la distance générationnelle. Lorsque l'ensemble optimal n'est pas connu nous avons utilisé la mesure \mathcal{C} et la mesure \mathcal{S} de Zitzler. Ces deux mesures sont complémentaires car la première permet de comparer deux approximations en termes de convergence tandis que la seconde permet de classer les différentes approximations.

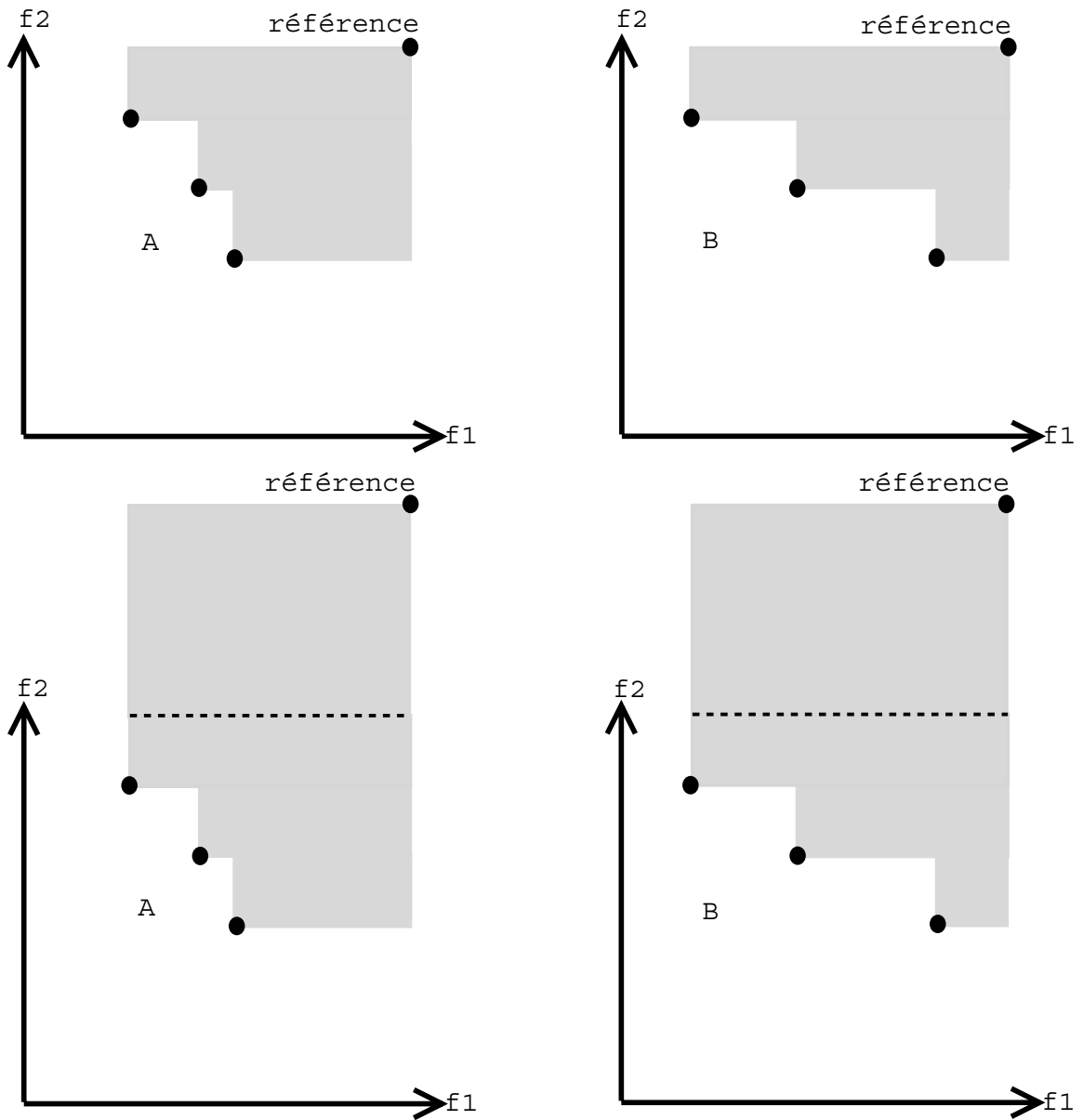


FIG. 1.19 – La valeur de \mathcal{S} dépend du choix arbitraire d'un point de référence. Dans le cas du haut, deux ensembles de solutions non-dominées sont montrés, A et B , avec $\mathcal{S}(A) > \mathcal{S}(B)$. Dans le cas du bas, pour les mêmes ensembles, on a $\mathcal{S}(A) < \mathcal{S}(B)$.

Chapitre 2

Problèmes de tournées

Ce chapitre est consacré aux problèmes de tournées en général et aux problèmes de tournées multi-objectif plus particulièrement. Tout d'abord, nous faisons une présentation générale des problèmes de tournées. Cette présentation est faite au travers d'une énumération des caractéristiques pouvant apparaître dans ce type de problèmes.

Dans un second temps, une introduction plus détaillée de deux problèmes de tournées particuliers est faite. Ces deux problèmes sont le problème de la tournée couvrante et le problème d'élaboration de tournées de véhicules qui sont les versions mono-objectif des problèmes multi-objectif étudiés au cours de cette thèse. Pour chaque problème, nous présentons un état de l'art des méthodes de résolution.

Après la présentation des problèmes mono-objectif, nous nous focalisons sur les problèmes de tournées multi-objectif. L'apport majeur de ce chapitre est l'état de l'art que nous avons effectué sur ces problèmes. Malgré un nombre de publications peu important dû à la jeunesse du domaine, nous constatons un réel essor ces dernières années. Il était donc temps de dresser un bilan des problèmes de tournées multi-objectif. Pour cela, nous proposons une classification selon trois points :

- Le contexte : cas réel, extension de problème, généralisation de problème.*
- Les différents objectifs étudiés. Ils ont été classés selon la composante du problème sur laquelle ils portent : tournées, nœuds/arcs ou ressources.*
- Les méthodes d'optimisation utilisées.*

2.1 Introduction

La logistique et les transports sont au centre de nombreux problèmes rencontrés dans l'industrie et les services. Ainsi, il n'est pas étonnant que les problèmes de tournées, qui sont une facette de la logistique, fassent partie des principaux problèmes étudiés en recherche opérationnelle. En effet, de nombreux enjeux théoriques, mais aussi pratiques et économiques, sont liés à cette famille de problèmes.

Le terme *problème de tournées* englobe un grand nombre de problèmes très disparates. Le but ici n'est pas de présenter tous les problèmes que l'on peut rencontrer mais plutôt de donner une idée des différents éléments qui peuvent être considérés dans un problème de tournées. Cependant, deux problèmes, le problème de la tournée couvrante et le problème d'élaboration de tournées de véhicules, seront présentés en détail. Ils forment les versions mono-objectif des deux problèmes de tournées multi-objectif étudiés dans cette thèse. Les problèmes de tournées multi-objectif en général sont aussi abordés dans ce chapitre au travers d'un état de l'art.

Ce chapitre s'organise en trois parties. La première fournit une description générale des caractéristiques que l'on peut rencontrer dans un problème de tournées. La seconde partie introduit le problème de la tournée couvrante et le problème d'élaboration de tournées de véhicules. Cette partie comprend aussi un état de l'art de l'application des méta-heuristiques au problème d'élaboration de tournées de véhicules. Enfin, ce chapitre se clôt par un état de l'art des problèmes de tournées multi-objectif.

2.2 Présentation générale des problèmes de tournées

Un problème de tournées consiste à rechercher une tournée ou un ensemble de tournées sur un réseau ou un sous-réseau sous certaines contraintes en optimisant un (ou plusieurs) objectif(s) fixé(s). L'exemple de problèmes de tournées le plus connu est sans doute le problème du voyageur de commerce (PVC) [38]. Dans ce problème, étant donné un graphe orienté valué $G = (S, A)$, le but est de trouver un circuit de longueur minimale passant par tous les nœuds de S .

Du point de vue de la théorie des graphes, une tournée sur un graphe $G = (S, A)$ est donc un circuit sur un ensemble $H \subseteq S$. Il est possible de définir un problème de tournées en se basant sur les catégories suivantes : le réseau, la demande, la flotte, le coût et l'objectif.

2.2.1 Le réseau

Le réseau est un graphe $G = (S, A)$. Selon les problèmes, les sommets peuvent représenter des villes à visiter, des clients à servir, des points de passage, des dépôts ... Les arcs représentent les liaisons existantes entre ces derniers. Les arcs peuvent donc être des routes, des canalisations ou des connexions symboliques ... Une action de collecte ou de livraison peut être définie le long des liaisons.

Le réseau peut être symétrique, les liaisons entre deux nœuds étant représentées par des arêtes, ou asymétrique, une liaison étant symbolisée par un arc. Les réseaux urbains sont typiquement modélisés par des graphes asymétriques, les arêtes représentant les rues à double sens et les arcs les sens uniques par exemple. Les réseaux interurbains sont le plus souvent modélisés par des graphes symétriques. Aux arcs et aux arêtes sont souvent associées des valeurs numériques qui indiquent les longueurs des liaisons ainsi que des temps de parcours qui peuvent dépendre du type de véhicule utilisé et de la période de traversée.

Dans certains cas, les nœuds et les arcs ne peuvent être desservis qu'à certains moments précis, comme par exemple des horaires d'ouverture. Des fenêtres de temps sont alors associées à chaque nœud ou arc. Elles peuvent être de deux sortes : *dures* ou *souples*. Dans le premier cas, si la tournée permet que l'on arrive en avance, il faut attendre jusqu'à la borne inférieure de la fenêtre et il est interdit d'arriver en retard. Dans le cas des fenêtres de temps souples, la violation des contraintes est permise mais entraîne une pénalité pour l'objectif.

2.2.2 La demande

La demande peut être associée soit aux nœuds, on parle alors de problèmes de tournées sur les nœuds, soit aux arcs, on parle de problèmes de tournées sur les arcs. Elle peut être fixe ou stochastique, auquel cas elle est donnée par des formules probabilistes. Enfin, la demande peut porter sur un seul produit ou sur plusieurs.

Le plus souvent, on parle de demande lorsque le problème traité est un problème de distribution, c'est-à-dire un problème où il s'agit de livrer à chaque nœud - on parle alors de clients - ou arc, une certaine quantité de marchandise ou de produit. Les problèmes de ramassage, où il s'agit de prendre chez les clients une certaine quantité de marchandise, peuvent être considérés comme équivalents. Cependant, il existe des problèmes dits de collecte et de livraison dans lesquels une quantité de produit doit d'abord être collectée chez un premier client pour ensuite être livrée chez un autre.

Il est possible d'imposer que la demande soit satisfaite en une seule fois. Si ce n'est pas le cas, la demande d'un client peut être divisée entre plusieurs tournées.

Il se peut qu'il ne soit pas possible ou nécessaire de satisfaire toutes les demandes. Par exemple, dans un problème de livraison, certains clients peuvent être ignorés.

2.2.3 La flotte

Les tournées correspondent souvent aux chemins que doit suivre un véhicule ou une flotte de véhicules pour visiter les nœuds et arcs du graphe. Dans les problèmes rencontrés, le vocabulaire utilisé ne fait pas toujours référence à des véhicules. Ainsi dans le problème du voyageur de commerce, il est question d'un voyageur de commerce. Cependant, pour clarifier le texte dans le reste de cette section, seul le terme de véhicule sera employé.

La première caractéristique de la flotte est le nombre de véhicules la composant. Ce nombre peut être fixe ou non. Toutefois, même s'il n'y a qu'un seul véhicule, cela ne signifie pas que la solution du problème est une tournée unique. Dans les problèmes de distribution

par exemple, il est possible pour un seul véhicule de faire plusieurs tournées si l'ensemble de la demande est trop important pour y pourvoir en une seule fois.

Les véhicules composant la flotte peuvent avoir plusieurs caractéristiques qui limitent ou conditionnent leur utilisation :

Dépôt associé : un véhicule doit commencer et terminer sa route en un nœud précis du graphe appelé dépôt. Il se peut que le véhicule ait le choix entre plusieurs dépôts possibles. Le choix du dépôt est alors fixé lors de la résolution du problème.

Capacité : les véhicules peuvent avoir une capacité maximale exprimée en termes de poids, de volume ...

Limitations : différentes limitations peuvent être définies, comme la distance ou le temps maximal(e) avant que le véhicule ne retourne à son dépôt. Cela représente par exemple la limite de carburant pour le véhicule ou l'impossibilité humaine ou légale pour un conducteur de conduire de manière continue plus d'un certain nombre d'heures.

La flotte peut être homogène, c'est-à-dire composée d'un seul type de véhicules, ou hétérogène. Les différences peuvent alors porter sur la capacité des véhicules, certaines caractéristiques (couverts ou non couverts) ou certaines particularités (véhicules frigorifiques).

Il est enfin possible de définir des dépendances entre les véhicules, les conducteurs, les produits et les nœuds/arcs :

véhicule/produit : certains produits ne peuvent être transportés que par certains véhicules. Par exemple des denrées qui doivent être livrées par des véhicules frigorifiques.

véhicule/chauffeur : certains véhicules ne peuvent être conduits que par des chauffeurs possédant certaines compétences.

véhicule/client : certains clients ne peuvent être desservis que par certains véhicules à cause de la nature de leur demande.

véhicule/réseau : certains nœuds ou arcs ne sont accessibles qu'à un certain type de véhicules.

chauffeur/client : cette relation peut être utile car un chauffeur qui connaît bien une région peut permettre un gain de temps. Instaurer une relation de confiance entre le client et le chauffeur est aussi un but possible.

2.2.4 Les coûts

Généralement, le coût est fixe pour le véhicule et variable pour son utilisation en fonction de la distance parcourue ou du temps écoulé. Le coût prend aussi en compte les pénalités de service si les clients sont servis en retard ou de manière incomplète, voire non servis. Un gain peut être associé aux nœuds ou aux liaisons ; le gain étant collecté lors de la visite du nœud ou de la liaison.

2.2.5 L'objectif

De nombreux objectifs différents peuvent être imaginés. Les plus couramment utilisés sont :

- Minimisation de la longueur totale parcourue.
- Minimisation de la durée totale du parcours.
- Minimisation du coût total du parcours.
- Minimisation de la somme des coûts fixes et des coûts variables de l'utilisation des véhicules.
- Minimisation de la taille de la flotte.
- Maximisation de la qualité de service offert.
- Maximisation du profit collecté par la visite chez des clients choisis.
- ...

Ces fonctions objectif peuvent être calculées sur une période unique mais aussi sur plusieurs périodes (heure, jour, semaine ...). Il est alors nécessaire d'affecter les véhicules et les visites aux différentes périodes.

Les objectifs cités sont souvent contradictoires. Il est possible de les combiner dans la définition de problèmes de tournées multi-objectif. De tels problèmes sont présentés plus en détail dans la section 2.6.

2.3 Le problème de la tournée couvrante (PTC)

Le premier problème multi-objectif traité dans cette thèse est basé sur le problème de la tournée couvrante (*Covering Tour Problem* - CTP). Ce dernier se définit de la manière suivante : soit $G = (V \cup W, E)$ un graphe non orienté, où $V \cup W$ est l'ensemble des sommets et $E = \{(v_i, v_j) | v_i, v_j \in V \cup W, i < j\}$ est l'ensemble des arêtes. Le sommet v_1 est un dépôt, V est l'ensemble des sommets qui *peuvent* être visités, $T \subseteq V$ est l'ensemble des sommets qui *doivent* être visités et W l'ensemble des sommets qui doivent être *couverts*. Une matrice de distance $C = (c_{ij})$ satisfaisant l'inégalité triangulaire est définie sur E , où c_{ij} est la distance entre les sommets v_i et v_j . Le problème de la tournée couvrante consiste à déterminer un cycle Hamiltonien de longueur minimale sur un sous-ensemble de V contenant tous les nœuds de T de façon que tous les nœuds de W soient couverts, c'est-à-dire que chaque élément de W se trouve à moins d'une distance donnée c d'un sommet de la tournée. Il se peut qu'il n'y ait pas de solution réalisable. Un exemple de solution du PTC est donnée dans la figure 2.1

Le problème de la tournée couvrante est NP-difficile puisqu'il se réduit au problème du voyageur de commerce lorsque $c = 0$ et que tous les sommets de W coïncident avec ceux de V . Il est aussi lié au problème du voyageur de commerce avec collecte de prix [60][8] et au problème du voyageur de commerce sélectif [98]. Le problème de la tournée couvrante peut également être formulé comme un problème du voyageur de commerce généralisé [59].

Le problème a été introduit et formulé dans un premier temps par Current [34, 35]. Dans cette dernière étude ainsi que dans [36], deux objectifs sont pris en compte : la longueur de la tournée et le nombre de nœuds couverts par la tournée. Une nouvelle formulation du problème, présentée ci-dessous, ainsi qu'un algorithme de séparations et coupes ont été ensuite introduits par Gendreau et al. [70]. Peu d'études ont été proposées sur ce problème dans la littérature en dehors de celles citées précédemment. L'algorithme de séparations et coupes de Gendreau et al. [70] a été appliqué à un cas réel sur la gestion d'une unité de soins

$$y_k = 1 \quad (v_k \in T) \quad (2.6)$$

$$y_k \in \{0, 1\} \quad (v_k \in V \setminus T) \quad (2.7)$$

Les contraintes 2.2 assurent que tous les sommets de W sont couverts par la tournée. Les contraintes 2.3 sont des contraintes de degré. Les contraintes 2.4 sont les contraintes de connexité. Elles forcent la présence d'au moins deux arêtes connectant n'importe quels ensembles S et $V \setminus S$, où S est un sous-ensemble de V tel que $T \setminus S \neq \emptyset$ et S contiennent un sommet v_t appartenant à la tournée. Enfin, les contraintes 2.5 et 2.7 sont des contraintes de binarité. Les contraintes 2.6 forcent la visite des sommets de T . Dans cette formulation, le cas $x_{1k} = 2$ pour un k donné n'est pas autorisé car il ne peut survenir que si la tournée (v_1, v_k, v_1) est faisable. Si cela se produit, une stratégie évidente est de garder cette solution, de résoudre le problème avec $V \leftarrow V \setminus \{v_k\}$ et de sélectionner la meilleure des deux solutions.

2.4 Le problème d'élaboration de tournées de véhicules (PTV)

Le second problème étudié dans cette thèse est une extension du problème d'élaboration de tournées de véhicules (*Vehicle Routing Problem* - VRP). Introduit par Dantzig et Ramser [39], il s'agit d'un problème de distribution qui se définit sur un graphe $G = (V, E)$ où l'ensemble des sommets $V = \{1, \dots, n\}$ représente le dépôt (le sommet 1) et les $n - 1$ clients. E est l'ensemble des arcs reliant les sommets. On doit servir à chaque client une quantité q_i d'un bien ($i = 1, \dots, n$) depuis le dépôt. Pour effectuer les livraisons, une flotte de véhicules est disponible. Les véhicules sont limités par la quantité Q de biens qu'ils peuvent transporter. Une solution du problème d'élaboration de tournées de véhicules est une collection de tournées. Chaque client appartient à une et une seule tournée et la demande totale sur chaque tournée ne peut pas dépasser Q . A chaque arc (i, j) , on associe la distance entre les sommets i et j . Une solution de longueur minimale est recherchée lors de la résolution du problème. Un exemple de solution du (PTV) est donné dans la figure 2.2. Le problème a été prouvé NP-difficile [103].

Le problème possède plusieurs formulations mathématiques. Fisher et Jaikumar [61] ont proposé une formulation qui étend celle de Dantzig et al. [38] pour le problème du voyageur de commerce. Soit x_{ij}^k une variable binaire définie pour i différent de j et égale à 1 si le véhicule k visite le client j juste après le client i , 0 sinon. Soit y_i^k une variable binaire égale à 1 si le véhicule k effectue la livraison chez le client i et égale à 0 sinon. Le nombre de véhicules est fixé à M . Le problème se modélise alors de la manière suivante :

$$\min \sum_{k=1}^M \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}^k \quad (2.8)$$

sous les contraintes :

$$\sum_{i=1}^n q_i y_i^k \leq Q \quad (k = 1, \dots, M) \quad (2.9)$$

$$\sum_{k=1}^M y_i^k = \begin{cases} M & (i = 1) \\ 1 & (i = 2, \dots, n) \end{cases} \quad (2.10)$$

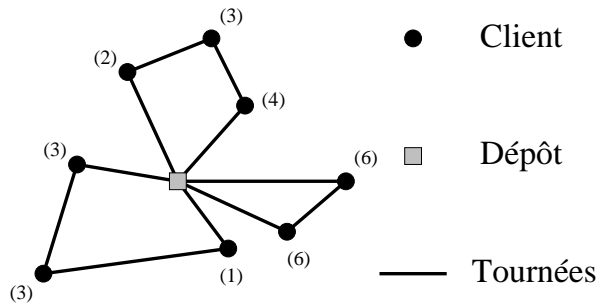


FIG. 2.2 – Un exemple de solution pour le problème d’élaboration de tournées de véhicules. (Les demandes des clients sont indiquées dans les parenthèses et la capacité des véhicules est de 12)

$$\sum_{i=1}^n x_{ij}^k = y_j^k \quad (j = 1, \dots, n; k = 1, \dots, M) \quad (2.11)$$

$$\sum_{i=1}^n x_{ij}^k = y_i^k \quad (i = 1, \dots, n; k = 1, \dots, M) \quad (2.12)$$

$$\sum_{i,j \in S} x_{ij}^k \leq |S| - 1 \quad (S \subset V; 2 \leq |S| \leq n - 2; k = 1, \dots, M) \quad (2.13)$$

$$x_{ij}^k \in \{0, 1\} \quad (i, j = 1, \dots, n; i \neq j; k = 1, \dots, M) \quad (2.14)$$

$$y_i^k \in \{0, 1\} \quad (i = 1, \dots, n; k = 1, \dots, M) \quad (2.15)$$

Les contraintes 5.2 assurent le respect de la capacité des différents véhicules. Les contraintes 5.3 forcent M véhicules à passer par le dépôt alors que pour chaque client le nombre de passages doit être égal à 1. Les contraintes 5.4 et 5.5 assurent la cohérence entre la visite d’un véhicule chez un client et le fait que le service est effectué par ce même véhicule. Enfin, les contraintes 5.6 sont l’extension naturelle des contraintes d’élimination de sous-tours (contraintes de connexité) proposées par Dantzig et al. [38] pour le problème du voyageur de commerce et qui ont déjà été exposées dans le cadre du problème de la tournée couvrante.

2.5 Méta-heuristiques pour le PTV

Les problèmes de tournées sont des problèmes d’optimisation combinatoire. Or, de nombreux problèmes d’optimisation combinatoire sont NP-difficiles et ne pourront donc pas être résolus de manière exacte dans un temps “raisonnable” puisque la capacité des machines évolue linéairement alors que le temps de calcul nécessaire à la résolution de ces problèmes augmente de manière exponentielle avec la taille des problèmes. Lorsque l’on s’attaque à des problèmes réels ou de taille importante, il faut se résoudre à un compromis entre la qualité des solutions obtenues et le temps de calcul utilisé.

Au milieu des années 1970 sont apparues des méthodes qui supervisent l'évolution de solutions fournies par des heuristiques pour assurer un compromis entre diversification et intensification. Ces méthodes, appelées *méta-heuristiques*, ont pour but de trouver des solutions dont la qualité est au-delà de ce qu'il aurait été possible de trouver à l'aide d'une simple heuristique.

Le but de cette section n'est pas d'exposer toutes les méta-heuristiques existantes ni le plus grand nombre de références pour les problèmes de tournées. L'objectif est de présenter de manière générale les méta-heuristiques les plus usitées pour le PTV. Certaines méta-heuristiques peu ou pas utilisées dans le cadre du PTV, par exemple la recherche par dispersion [71], la programmation génétique [95], les algorithmes à essaim de particules [90] ou les algorithmes à estimation de distribution [10, 11], ne sont pas évoquées. Les méta-heuristiques sont classées en deux groupes : les méthodes à solution unique et les méthodes à population de solutions. Pour chaque méthode, des applications sur le PTV seront présentées. Le PTC n'est pas pris en compte dans cette partie car, à notre connaissance, la seule méta-heuristique utilisée pour le résoudre est la recherche par dispersion comme indiqué précédemment.

2.5.1 Les méta-heuristiques à solution unique

Les méthodes itératives à solution unique sont toutes basées sur un algorithme de 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. Nous présentons ici les méthodes les plus usitées pour les problèmes de tournées : les méthodes de descente, le recuit simulé et la recherche Tabou.

2.5.1.1 Les méthodes de descente (Hill climbing)

Les méthodes de descente sont assez anciennes et doivent leur succès à leur rapidité et leur simplicité [116, 117]. A chaque pas de la recherche, cette méthode progresse vers une solution voisine de meilleure qualité. La descente s'arrête quand tous les voisins candidats sont moins bons que la solution courante ; c'est-à-dire lorsqu'un optimum local est atteint. On distingue plusieurs types de descente en fonction des stratégies de génération de la solution de départ et du parcours du voisinage : la descente déterministe, la descente stochastique et la descente vers le premier meilleur voisin.

2.5.1.2 Le recuit simulé (Simulated annealing)

Le recuit simulé est une technique d'optimisation de type Monte-Carlo généralisé à laquelle on introduit un paramètre de température qui sera ajusté pendant la recherche [91]. Elle s'inspire des méthodes de simulation de Metropolis développées dans les années 50 en mécanique statistique. L'analogie historique s'inspire du recuit des métaux en métallurgie : un métal refroidi trop vite présente de nombreux défauts microscopiques ; c'est l'équivalent d'un optimum local pour un problème d'optimisation combinatoire. Si on le

refroidit lentement, les atomes se réarrangent, les défauts disparaissent et le métal a alors une structure très ordonnée, équivalente à un optimum global.

La méthode du recuit simulé, appliquée aux problèmes d'optimisation, considère une solution initiale et recherche dans son voisinage une autre solution de façon aléatoire. L'originalité de cette méthode est qu'il est possible de se diriger vers une solution voisine de moins bonne qualité avec une probabilité non nulle. Ceci permet de sortir des optima locaux. Au début de l'algorithme, un paramètre T , apparenté à la température, est déterminé et décroît tout au long de l'algorithme pour tendre vers 0. De la valeur de ce paramètre va dépendre la probabilité p d'acceptation des solutions détériorantes (plus la température T est élevée, plus cette probabilité sera forte).

La performance du recuit simulé dépend notamment de la règle de refroidissement (c'est-à-dire la décroissance du paramètre T) utilisée. Un refroidissement trop rapide mènerait vers un optimum local pouvant être de très mauvaise qualité. Un refroidissement trop lent serait très coûteux en temps de calcul. Le réglage des différents paramètres (température initiale, nombre d'itérations par palier de température, décroissance de la température ...) peut être long et difficile.

Les premières implémentations d'un recuit simulé pour le PTV sont celles de Robusté et al. [134] et Alfa et al. [2]. Dans la première, le voisinage est défini par plusieurs mécanismes (renversements, mouvements de parties de la tournée ...). La seconde implémentation utilise une heuristique route-first, cluster-second [15] pour construire une première solution puis une recherche 3-opt [104] pour la phase de descente. Aucune des deux méthodes n'a été comparée avec d'autres approches. Une implémentation plus évoluée du recuit simulé est celle proposée par Osman [113]. Il utilise de meilleures solutions de départ, certains paramètres sont ajustés lors d'une phase d'essais et le refroidissement est plus sophistiqué. Bien que cette méthode produise généralement de bons résultats, elle identifie parfois des solutions de mauvaise qualité et trouve rarement la meilleure solution connue.

2.5.1.3 La recherche Tabou (Tabu search)

La recherche Tabou a été introduite par Glover [72] et a montré sa performance sur de nombreux problèmes d'optimisation. Les idées de base de la recherche Tabou se retrouvent aussi dans le travail d'Hansen [78]. Elles utilisent la notion de mémoire pour éviter d'être piégées par un optimum local.

Le principe de l'algorithme est le suivant : à chaque itération, le voisinage (complet ou sous-ensemble du voisinage) de la solution courante est examiné et la meilleure solution est sélectionnée. En appliquant ce principe, la méthode autorise à remonter vers des solutions qui semblent moins intéressantes mais qui peuvent permettre de s'échapper de l'optimum local. Le risque est de boucler entre deux solutions. Pour éviter ce phénomène, la méthode n'est pas autorisée à visiter une solution récemment visitée. Pour cela une liste taboue contenant les attributs des dernières solutions visitées est tenue à jour. Lorsqu'une nouvelle solution est considérée, l'attribut la définissant remplace le plus ancien attribut de la liste. Ainsi, la recherche de la solution courante suivante se fait dans le voisinage de la solution courante actuelle sans considérer les solutions appartenant à la liste taboue. Cette méthode

ne s'arrête pas d'elle-même et il faut déterminer un critère d'arrêt en fonction du temps de recherche que l'on s'octroie. Ce critère peut être, par exemple, l'exécution d'un certain nombre d'itérations ou la non-amélioration de la meilleure solution pendant un certain nombre d'itérations. Ainsi, tout au long de l'algorithme, la meilleure solution doit être conservée car l'arrêt se fait rarement sur la meilleure solution.

Durant la dernière décennie, la recherche Tabou a de plus en plus été appliquée au PTV. Même si les premières implémentations [169, 125] n'offraient pas des résultats impressionnants, les travaux suivants furent de plus en plus efficaces. Parmi celles-ci, on peut citer celles de Taillard [151], de Gendreau et al. [69], de Rochat et Taillard [135], de Xu et Kelly [170], de Rego et Roucairol [129], de Toth et Vigo [160], de Barbarosoglu et Ozgur [13]. Certaines de ces méthodes comme TabuRoute [69] sont capables de trouver des résultats de bonne qualité et notamment les meilleures solutions connues. Par ailleurs, Rochat et Taillard [135] ont introduit un concept novateur et efficace, la mémoire adaptative, qui est capable d'améliorer n'importe quelle recherche à base de recherche Tabou. De même, Toth et Vigo [160] proposent *granular tabu search* où le problème est prétraité de telle sorte qu'une dégradation minimale des résultats entraîne une baisse importante du coût en temps de calcul.

2.5.2 Les méta-heuristiques à population de solutions

Les méthodes 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.

2.5.2.1 Les algorithmes génétiques (Genetic algorithms)

Il existe de nombreux algorithmes évolutionnaires et il n'est pas possible de parler des algorithmes génétiques sans mentionner les méthodes de programmation évolutionnaire développées par Fogel et al. [62] et les stratégies évolutionnaires développées indépendamment par Rechenberg [128] et Schwefel [142]. Ils ont considérablement contribué à l'intérêt porté aux algorithmes évolutionnaires.

Les algorithmes génétiques sont des méthodes basées sur les mécanismes biologiques telles que les lois de Mendel et sur le principe fondamental (sélection naturelle) de Darwin [40]. Holland a exposé les principes de ces algorithmes pour permettre aux ordinateurs "d'imiter les êtres vivants en évoluant" pour rechercher la solution à un problème [80]. Il a d'abord expliqué comment ajouter de l'intelligence dans un programme informatique avec les croisements (échange du matériel génétique) et la mutation (source de la diversité génétique). Il a ensuite formalisé les principes fondamentaux des algorithmes génétiques :

- La capacité de représentations élémentaires, comme les chaînes de bits, à coder des structures complexes.
 - Le pouvoir de transformations élémentaires à améliorer de telles structures.
- Plus récemment, Goldberg a enrichi la théorie des algorithmes génétiques en s'appuyant sur le parallèle suivant [73] :
- Un individu est lié à un environnement par son code ADN.

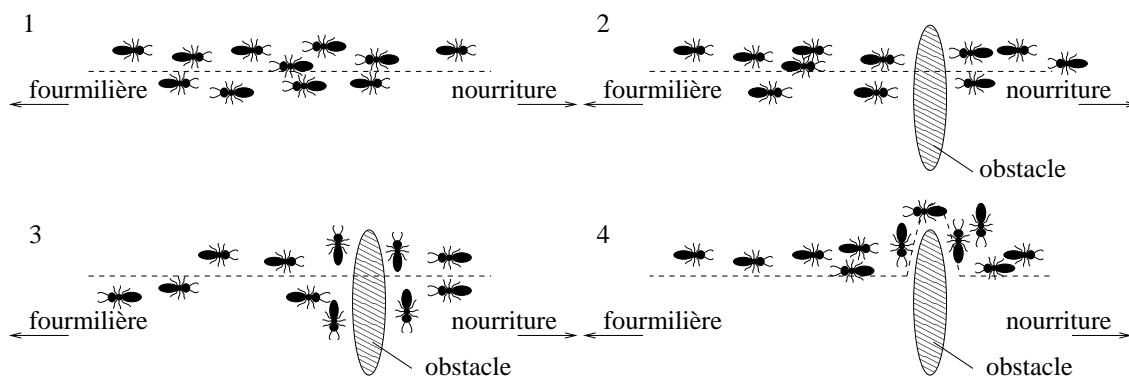
- Une solution est liée à un problème par son indice de qualité.
- Une “bonne” solution à un problème donné peut être vue comme un individu susceptible de survivre dans un environnement donné.

Les algorithmes génétiques simulent le processus d'évolution d'une population. A partir d'une population de N solutions du problème représentant des individus, on applique des opérateurs simulant les interventions sur le génome comme le croisement (*crossover*) ou la mutation pour arriver à une population de solutions de mieux en mieux adaptée au problème. Cette adaptation est évaluée grâce à une fonction coût.

Peu d'études portent sur l'application des algorithmes génétiques au PTV contrairement aux variantes plus complexes comme le PTV avec fenêtres de temps [17, 122, 157, 158]. Une première étude de comparaison d'un algorithme génétique avec des recuits simulés et des recherches Tabou a été faite par Van Breedam [164]. L'algorithme génétique travaillait sur une représentation étendue où le dépôt pouvait apparaître plusieurs fois à l'aide d'opérateurs classiques : l'opérateur de croisement PMX [74] et l'opérateur de mutation RAR. A l'issue des comparaisons, l'algorithme génétique offrait des performances en termes de qualité comparables aux autres méta-heuristiques mais pour un coût en temps de calcul plus important.

Par la suite, Schmitt [139, 140] proposa un algorithme génétique qui utilise une stratégie route-first cluster-second. En effet, les retours au dépôt, et donc les contraintes de distance et de capacité, sont ignorés et l'algorithme ne travaille que sur une tournée unique visitant tous les clients. Une telle approche permet de continuer à utiliser les opérateurs classiques pour le problème du voyageur de commerce. Dans un second temps, la solution est transformée en une solution pour le PTV à l'aide d'une procédure de balayage commençant par le premier nœud rencontré dans la chaîne. La solution est enfin améliorée par l'application d'une recherche locale 2-opt [104]. Les résultats de cet algorithme ne sont pas de très bonne qualité d'autant plus qu'il demande un temps de calcul important.

Récemment, deux algorithmes génétiques efficaces ont été proposés pour le PTV. Le premier algorithme est proposé par Baker et Ayechev [6]. Dans leur algorithme, les tournées des véhicules ne sont pas données explicitement. Une solution est encodée sous la forme d'un vecteur dont chaque composante correspond à un client et indique le numéro du véhicule qui le visite. Il est donc nécessaire de résoudre un PVC pour chaque véhicule pour connaître la longueur de la solution. Les opérateurs génétiques sont des opérateurs classiques : croisement 2-points et la mutation qui échange entre deux clients le numéro du véhicule qui effectue la visite. Des méthodes sont utilisées pour accélérer la convergence. La première sert à réduire la violation des contraintes de capacité qui sont permises. Ensuite, des recherches locales sont appliquées, à intervalles réguliers en termes de générations, sur toutes les solutions de la population. Les recherches locales sont une recherche 2-opt [104] et une recherche λ -interchange utilisée par Osman [113]. L'algorithme obtient de bons résultats et il est capable avec l'hybridation d'obtenir certaines des meilleures solutions connues. Le second algorithme génétique, proposé par Prins [124], utilise la même stratégie que l'algorithme génétique de Schmitt, mais le retour à une solution pour le PTV se fait par une procédure qui est capable de trouver le découpage optimal en tournées pour une séquence donnée en un temps polynomial. Par ailleurs, les solutions obtenues sont améliorées par une recherche locale 2-opt [104] qui travaille directement sur la méta-



Cette figure illustre le comportement de la colonie en quatre étapes :

1. Les fourmis suivent une piste de phéromone menant à la nourriture (pointillé).
2. Un obstacle coupe la piste de phéromone.
3. Les fourmis choisissent aléatoirement le chemin du haut ou le chemin du bas.
4. Une nouvelle piste de phéromone se construit : le meilleur chemin est gardé. La phéromone est effacée sur les éventuels mauvais choix et sous l'obstacle.

FIG. 2.3 – Fourmis à la recherche de sources de nourriture.

ournée qui visite tous les clients. Cet algorithme présente de très bons résultats et il est notamment capable de trouver un grand nombre de meilleures solutions connues.

2.5.2.2 Les colonies de fourmis (Ant system)

L'histoire de l'intelligence en essaim remonte à l'étude du comportement de fourmis à la recherche de nourriture au départ de leur nid, par Goss, Deneubourg et leur équipe [46, 45]. En se déplaçant du nid à la source de nourriture et vice-versa (ce qui, dans un premier temps, se fait essentiellement de manière aléatoire), les fourmis déposent au passage sur le sol une substance odorante appelée phéromone, ce qui a pour effet de créer une piste chimique. Les fourmis peuvent sentir cette phéromone qui a un rôle de marqueur de chemin : quand les fourmis choisissent leur chemin, elles ont tendance à choisir la piste qui porte la plus forte concentration de phéromone (voir figure 2.3). Ce procédé leur permet de retrouver le chemin vers leur nid lors du retour. D'autre part, les odeurs peuvent être utilisées par d'autres fourmis pour retrouver les sources de nourriture détectées par leurs consœurs.

Il a été montré expérimentalement que ce comportement permet l'émergence des chemins les plus courts entre le nid et la nourriture, à la condition que les pistes de phéromone soient utilisées par une colonie entière de fourmis. Le système de colonies de fourmis est une méthode d'optimisation basée sur ces observations proposée par Dorigo [50, 49, 51]. Cette

méthode a été utilisée avec succès sur de nombreux problèmes (voyageur de commerce, affectation quadratique ...). Mais les auteurs ont remarqué que les colonies de fourmis n'ont pas un comportement très exploratoire, ce qui les a conduits à utiliser des hybridations de colonies de fourmis avec des recherches locales.

A ce jour, peu d'études ont appliqué ce paradigme au PTV. La première étude a été proposée par Kawamura et al. [87]. La méthode est une approche hybride qui inclut des procédures d'amélioration basées sur la recherche 2-opt, ainsi que des règles d'acceptation probabilistes du recuit simulé. Peu de tests ont été effectués sur l'efficacité de cette méthode.

Une autre manière d'utiliser les colonies de fourmis pour le PTV a été proposée par Reiman et al. [131, 47, 130]. La principale caractéristique de leur méthode est l'inclusion de techniques provenant de l'algorithme des économies [24]. Par la suite, les auteurs ont proposé l'extension de leur méthode pour le PTV par le parallélisme [48].

2.6 Problèmes de tournées multi-objectif

Les problèmes de tournées académiques nécessitent souvent des adaptations pour des applications pratiques. Les adaptations sont le plus souvent sous la forme de nouvelles contraintes ajoutées au modèle du problème. Par exemple, la version de base du problème d'élaboration de tournées de véhicules consiste à construire une collection de longueur minimale de tournées pour une flotte de véhicules sur un ensemble de clients qui demandent à être servis d'une certaine quantité de marchandise depuis un dépôt, de telle sorte que la demande associée à une tournée ne dépasse pas la capacité du véhicule affecté. Pour prendre en compte d'autres aspects apparaissant dans des cas réels, plusieurs variantes du problème ci-dessus, incluant de nouvelles contraintes, ont été proposées [159]. Un exemple est le problème d'élaboration de tournées de véhicules avec fenêtres de temps où une fenêtre de temps est associée à chaque client qui doit être servi durant cet intervalle.

Une autre manière d'améliorer l'aspect réel d'un problème est de considérer plusieurs objectifs. Dans le cas des problèmes de tournées, les objectifs peuvent être classés selon la composante du problème sur laquelle ils portent : la tournée (travail, profit, makespan, équilibre, ...), les nœuds ou arcs (fenêtres de temps, service du client, ...) et les ressources (véhicules, marchandises, ...). La figure 2.4 recense les différents objectifs rencontrés dans la littérature. A la fin du chapitre, les tableaux 2.1 à 2.4 résument les différentes études de la littérature en indiquant : le problème, la méthode de résolution employée et les différents objectifs regroupés selon la classification proposée ci-dessus.

2.6.1 Applications

Les approches multi-objectif des problèmes de tournées ont principalement trois motivations : la résolution d'un problème réel dans lequel plusieurs objectifs ont clairement été définis par le décideur, l'adaptation ou l'extension de problèmes académiques dans le but de rendre les problèmes plus en phase avec des cas réels ou pour étudier des objectifs complémentaires sans abandonner toutefois l'objectif classique de minimisation de la distance parcourue ou encore pour offrir une généralisation d'un problème.

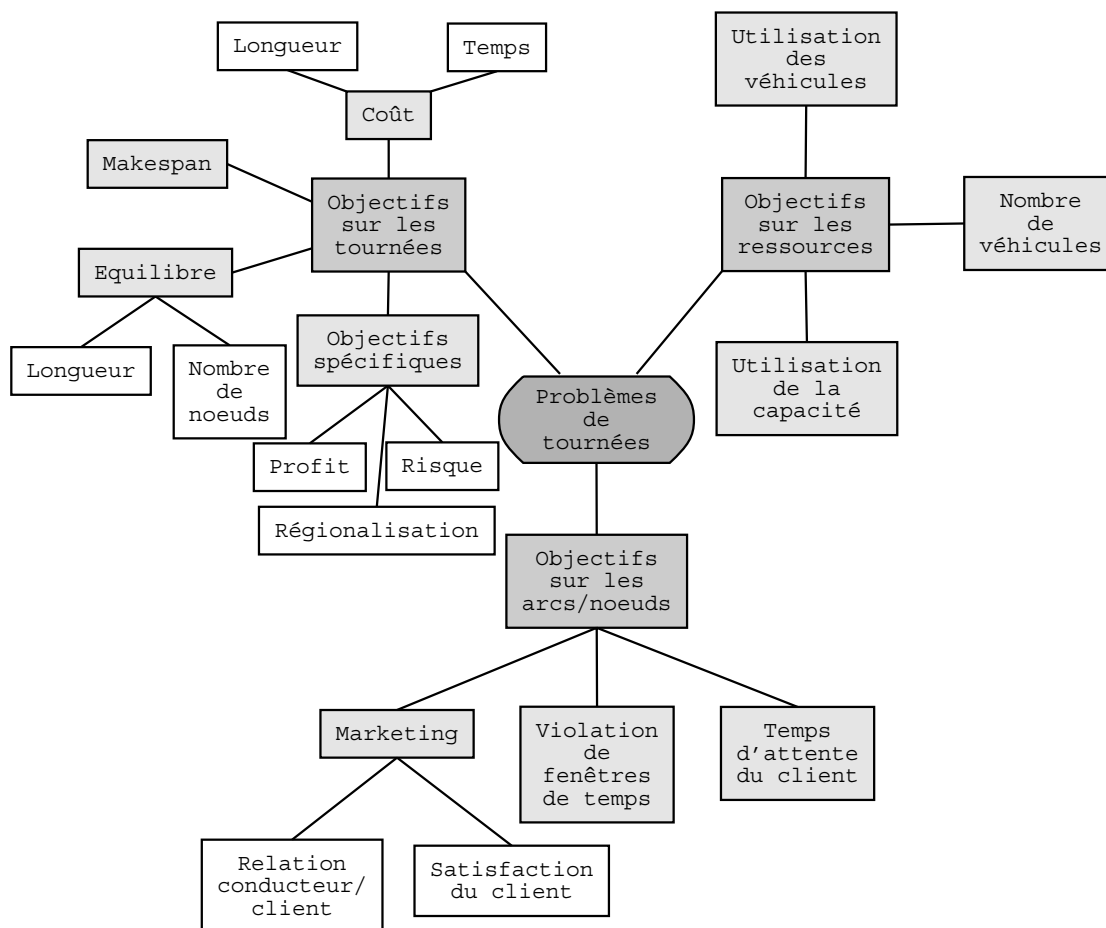


FIG. 2.4 – Objectifs pour les problèmes de tournées rencontrés dans la littérature.

2.6.1.1 Cas réels

Cas d'une compagnie de transport belge : dans cette étude [56], le problème envisagé comporte huit objectifs fixés par une compagnie de transport belge. Un ensemble de clients demande à se faire livrer une certaine quantité de marchandise depuis un emplacement autre que le dépôt de véhicules.

La livraison se fait donc en deux temps. D'abord, le camion se rend en un lieu pour charger la cargaison qui est ensuite livrée au destinataire final. Pour chaque lieu de chargement, une fenêtre de temps est définie. La livraison est effectuée par une flotte de camions hétérogènes. Il existe deux types de camions : les camions non-couverts qui n'ont pas de bâche et les camions couverts qui en ont une. Il n'y a pas de problème de capacité, en effet, si c'est nécessaire, une commande peut être décomposée entre plusieurs véhicules. La résolution doit être effectuée quotidiennement par rapport aux clients de la journée.

Tournées de cars scolaires : Bowerman et al. [19] s'intéressent à la planification de tournées de cars scolaires en milieu urbain et notamment dans le comté de Wellington dans l'Ontario. Selon les auteurs, le problème de tournées de cars est plus complexe que le problème d'élaboration de tournées de véhicules classiques. Ils spécifient le problème de la manière suivante : un groupe d'élèves répartis dans l'espace doit recevoir un service de transport public depuis le lieu de résidence jusqu'à l'école. Le problème est de trouver une série de tournées de cars scolaires qui assure que le service est équitable pour tous les élèves admissibles. Les auteurs proposent un modèle mathématique multi-objectif pour ce problème. Le modèle comporte quatre objectifs qui sont : la minimisation de la longueur totale, la minimisation de la distance de marche des élèves, l'équilibrage de la charge et l'équilibrage des longueurs. Cependant, les auteurs proposent une approche en deux phases qui ne prend pas en compte l'objectif d'équilibrage des longueurs. Dans la première phase, les élèves sont groupés en clusters qui peuvent être servis par un car unique. Les objectifs sont la longueur des tournées, l'aspect compact des clusters (distance de marche) et l'équilibrage de la charge. La seconde phase calcule la tournée des cars sur chacun des clusters définis précédemment. La résolution sur un cluster est indépendante de celles sur les autres clusters. Dans cette phase, il s'agit aussi de sélectionner les points d'arrêt du car de telle sorte que la distance que les élèves doivent parcourir à pied soit minimisée. Les deux objectifs envisagés dans cette phase sont : la minimisation de la longueur de la tournée sur chaque cluster et la minimisation de la distance que les élèves doivent parcourir à pied.

Les études de Corberan et al. [29] et de Pacheco et Marti [115] traitent le problème du transport d'élèves à leur école et leur retour ; le transport devant être effectué de la manière la plus sûre, la plus économique et la plus commode possible. D'autre part, l'environnement dans lequel le transport s'effectue est aussi particulier. Il s'agit en effet de zones rurales en Espagne où les distances entre deux points de ramassage et jusqu'aux écoles tendent à être longues. Ainsi, le problème de remplissage des cars n'intervient pas ; car un autobus est rarement plein avant qu'une certaine limite de temps ne soit atteinte. Cette limite vient du fait qu'il ne faut pas qu'un élève passe trop de temps dans un car et qu'il y ait trop d'iniquité entre le premier élève ramassé sur une tournée et le dernier.

Cas d'une société de ramassage des déchets : dans cette étude [97], des déchets doivent être collectés dans les rues de la ville de Troyes (France). Il s'agit donc d'un problème de tournées sur les arcs. Dans ce cas, tous les véhicules quittent le dépôt à six heures du matin et la collecte des déchets doit être terminée au plus vite car les travailleurs affectés aux véhicules doivent ensuite s'occuper du tri des déchets dans une usine de recyclage.

Transport de marchandises à Singapour : Tan et al. [155] proposent un problème bi-objectif défini à partir du système d'organisation de capacité de véhicules pour une entreprise de logistique de Singapour présenté dans [100]. Le modèle utilisé est un PTV avec camions et remorques (*Truck and Trailer Vehicle Routing Problem*) [20] où la longueur de la solution et la taille de la flotte de camions et de remorques sont prises en compte. Dans le PTVCR, les véhicules sont formés de deux éléments détachables. Certaines localisations sont accessibles par le véhicule complet. Pour d'autres, il est nécessaire de laisser la re-

morque le temps d'effectuer les livraisons. Dans l'étude de Tan et al. [155], des fenêtres de temps sont ajoutées au problème.

Cas de distribution de produits dangereux : Zografos et Androustopoulos [176] proposent la modélisation de la distribution de produits dangereux sous la forme d'un problème bi-objectif où la minimisation du coût ne se fait pas au détriment de la minimisation du risque et *vice versa*. Le problème est modélisé sous la forme d'un problème d'élaboration de tournées de véhicules avec fenêtres de temps. L'heuristique proposée a été intégrée à un système d'information géographique pour des opérations logistiques sur les matériaux dangereux.

Cas d'un problème de tournées de véhicules multipériodique : Dans [111], le problème étudié est un problème de tournées de véhicules multipériodique où l'équilibrage de charges (sur les véhicules ou les journées) ainsi qu'un objectif de régionalisation visant à générer des clusters de clients sont pris en compte. Lors de ce travail, un problème réel comprenant 6000 clients sur un délai de 20 jours a notamment été traité.

2.6.1.2 Extension de problème

Une autre utilisation de l'optimisation multi-objectif est la possibilité d'étudier des objectifs supplémentaires sans pour autant abandonner la minimisation du coût de la solution qui est souvent nécessaire. L'ajout d'objectifs a le plus souvent pour but de renforcer le côté réaliste du modèle en prenant en compte le fait que la plupart des problèmes en logistique sont soumis à des impératifs qui ne sont pas uniquement liés à la quantité de travail. Le problème d'élaboration de tournées de véhicules avec équilibrage des tournées présenté dans le chapitre 5 est une extension du problème d'élaboration de tournées de véhicules où un objectif supplémentaire sert à chercher des solutions où les tournées sont équilibrées entre elles en termes de longueur.

Prise en compte des conducteurs : Lee et Ueng [101] proposent une extension du problème d'élaboration de tournées de véhicules où un équilibrage entre les tournées est aussi recherché. L'ajout du second objectif est motivé par la volonté de rendre le travail plus équitable entre les conducteurs. D'après les auteurs, à Taiwan, l'affectation des tâches est effectuée manuellement par un décideur qui peut être influencé par des préférences personnelles, des informations insuffisantes ou d'autres facteurs humains. Les conducteurs comparent leurs emplois du temps et découvrent ainsi les anomalies, ce qui entraîne des plaintes et donc une baisse de la qualité du service. Or, les conducteurs sont considérés comme des moyens de compétition pour les compagnies de transport et c'est pourquoi leur bien-être doit être pris en compte.

Prise en compte de la satisfaction des clients : Sessomboon et al. [144] ont ajouté des objectifs à un problème d'élaboration de tournées de véhicules pour améliorer le niveau de satisfaction des clients par rapport à la date de livraison.

Problème de distribution commerciale : Ribeiro and Lourenço [132] proposent une extension du problème d'élaboration de tournées de véhicules avec période dans lequel le coût, l'équilibre entre les tournées et un objectif défini d'un point de vue marketing sont pris en compte. La motivation pour l'objectif d'équilibrage est la même que dans l'étude de Lee et Ueng [101]. Le problème a été défini par rapport aux problèmes survenant dans l'industrie de la nourriture et de la boisson où le planning de distribution est ordinairement défini pour une semaine et la relation entre le client et le conducteur a une grande importance pour améliorer les ventes et créer une bonne image de la compagnie.

Problème du voyageur de commerce multi-objectif : plusieurs études s'intéressent au problème du voyageur de commerce multi-objectif [118, 171]. Dans ce problème, plusieurs coûts sont associés aux arêtes. Par exemple, si le graphe représente un réseau routier, le temps nécessaire pour se rendre entre deux points peut ne pas être proportionnel à la distance parcourue. Une autre motivation de l'étude de ce problème vient du fait qu'il s'agit d'une extension d'un problème classique fortement étudié.

2.6.1.3 Généralisation de problème

Une autre manière d'utiliser l'optimisation multi-objectif est de généraliser un problème par l'inclusion d'objectifs supplémentaires. Typiquement, il s'agit de modéliser une ou plusieurs caractéristiques du problème, généralement des contraintes, et de les remplacer par des objectifs. Dans la littérature, cette stratégie a été plusieurs fois employée pour s'attaquer au problème d'élaboration de tournées de véhicules avec fenêtres de temps où les fenêtres de temps, qui sont très contraignantes, sont supprimées et remplacées par un ou plusieurs objectifs [81, 67, 127, 12, 126]. Pour le PTV avec fenêtres de temps, il peut aussi s'agir de traiter simultanément l'objectif de minimisation de la longueur de la tournée et de minimisation du nombre de véhicules utilisés [67, 127, 12, 126, 156].

Boffey [18] cite une liste de problèmes de tournées qu'il qualifie de problèmes implicitement multi-objectif. Dans ces problèmes, une contrainte est utilisée à la place de ce qui peut naturellement être modélisé par un objectif. Feillet et al. [58] exposent le cas de la classe de problèmes qu'ils appellent problèmes du voyageur de commerce avec profits. Dans ces problèmes, un profit, associé à chaque client, peut être collecté en visitant le client. Il n'est pas obligatoire de visiter tous les clients. Deux objectifs opposés peuvent être clairement envisagés :

1. Maximiser le profit et de ce fait visiter le maximum de clients et augmenter la distance parcourue.
2. Minimiser la distance parcourue en visitant peu de clients et de ce fait diminuer le profit.

Lorsque l'objectif est de maximiser le profit sans que la longueur de la tournée ne dépasse une constante, on parle de problème du voyageur de commerce sélectif. Lorsque le but est de minimiser la longueur de la tournée tout en assurant un profit minimum, le problème est appelé problème du voyageur de commerce avec quota. Cependant, d'un point de vue bi-objectif, il ne s'agit que d'un seul et même problème. Les auteurs ne rapportent qu'une seule étude qui prend en compte le problème sous forme bi-objectif [88, 89].

Le problème de la tournée couvrante bi-objectif, présenté dans le chapitre 4, est une généralisation du problème de la tournée couvrante [70] qui est l'un des problèmes cités par Boffey.

2.6.2 Objectifs

Nous allons maintenant décrire les différents objectifs en les classant dans les trois catégories définies précédemment.

2.6.2.1 Objectifs sur les tournées

Coût : la quasi-totalité des études cherchent à minimiser le coût des solutions générées. Le coût d'une solution peut s'exprimer en fonction de la distance parcourue, du temps nécessaire, du nombre de clients visités ... La plupart du temps, la minimisation du coût correspond à un critère économique. Dans le cadre du problème du voyageur de commerce multi-objectif, les différents objectifs correspondent à différents coûts que peut avoir un arc. Cependant, d'autres raisons peuvent être envisagées. Ainsi dans [120] et [121], la distance parcourue doit être minimisée pour éviter une détérioration des marchandises transportées.

Makespan : il existe des études où la minimisation du coût est ignorée. Ainsi, dans les études de Corboran et al. [29] et Pacheco et Marti [115], la minimisation du coût est écartée au profit de la minimisation du makespan, c'est-à-dire la durée de la plus longue tournée. Ce choix a été fait pour prendre en compte la particularité de l'environnement : une zone rurale en Espagne où du fait de l'éloignement entre les différents points de ramassage, les trajets ont tendance à être longs et les cars n'atteignent jamais leur capacité maximale. La minimisation du makespan permet d'assurer que les élèves ne passent pas trop de temps dans les transports et donc une certaine équité quant au temps passé dans le car entre l'élève qui est ramassé en début de tournée et celui qui est ramassé en fin de tournée.

Le makespan est aussi un objectif utilisé par Lacomme et al. [97] où le ramassage des déchets doit se terminer au plus vite car les conducteurs doivent ensuite trier les déchets.

Équilibrage : un autre type d'objectif naturel est celui qui permet d'induire un équilibrage entre les tournées. L'objectif peut être introduit pour apporter une notion d'équité entre les travailleurs, les clients ... Pour prendre en compte l'équilibrage entre les tournées, il est nécessaire de définir le travail ou la charge associé(e) à une tournée. Le travail d'une tournée peut s'exprimer en fonction du nombre de clients visités, de la quantité de marchandise livrée, de la longueur de la tournée, du temps nécessaire ...

Sutcliffe et Board [150] traitent un problème où des personnes handicapées mentales doivent être transportées et où l'égalisation des temps de voyage des véhicules doit être maximisée. Comme indiqué précédemment, Lee et Ueng incorporent l'équilibre des tournées dans le problème pour améliorer l'équité entre les travailleurs. Le travail d'une tournée est exprimé en fonction des temps de voyage, de chargement et de déchargement. L'objectif d'équilibre est modélisé par la minimisation de la somme des différences entre le travail de

chaque véhicule et celui du véhicule avec le moins de travail. L'équilibrage des tournées est aussi un objectif pour Ribeiro et Lourenço [132]. Le travail d'une tournée est donné par le volume transporté durant toutes les périodes. Selon les auteurs, ce travail est plus intéressant pour les industries de nourriture et de boisson où il est courant d'avoir un pourcentage de la rémunération en relation avec la quantité de produit vendue et distribuée.

Dans le cas de la firme de transport belge traité par El-Sherbeny [56], le travail d'une tournée correspond au temps nécessaire pour se rendre entre les différents points de ramassage et de livraison et aux temps de chargement et de déchargement. De plus, si un véhicule arrive en avance par rapport à la fenêtre de temps d'un client, il doit attendre, et le temps d'attente est ajouté au travail de la tournée. L'objectif est alors la minimisation de la différence entre le temps de la tournée la plus longue et le temps de la tournée la plus courte.

Dans le problème de tournées de véhicules multipériodique de [111], un objectif prend en compte l'équilibrage de la charge. En effet, il n'est pas acceptable que des solutions engendrent une forte charge certains jours, tandis que d'autres jours l'activité est moindre. Au contraire, d'après l'auteur, les industriels cherchent à lisser l'activité de livraison tout au long de la semaine. L'équilibrage de la charge de travail peut intervenir soit par jour, soit par véhicule. La première solution permet un planning homogène sur l'ensemble des journées alors que la seconde approche permet une utilisation homogène des véhicules sur tout l'horizon de temps.

Objectifs spécifiques : certains objectifs peuvent être définis en fonction de la nature du problème ou du contexte où s'inscrit le problème. Dans le problème traité par Keller et Goodchild [88, 89], un gain est associé aux différents nœuds du graphe. Ce gain peut être collecté en visitant le nœud, mais il n'est pas nécessaire de visiter tous les nœuds. Dans leur modèle, la maximisation du gain est prise en compte.

Dans le cas du transport de produits dangereux, un risque est associé au parcours qu'emprunte un véhicule. Du fait des conséquences pour les habitants et l'environnement des régions traversées qui seraient catastrophiques en cas d'accident au vu des produits transportés, il est nécessaire de minimiser le risque d'accident le long des tournées. Zografos et Androutsopoulos [176] proposent un modèle bi-objectif où l'un des objectifs minimise le risque d'accident. Le risque calculé est celui des populations le long des tournées. Le risque est exprimé par la somme des $p_s \times C_s$ sur les segments formant la tournée, où p_s représente la probabilité d'accident et C_s est la quantité d'individus qui se trouvent à moins d'une distance λ du segment s .

Dans [111], l'objectif de minimisation de la longueur des tournées est délaissé au profit d'un objectif de *régionalisation*. Cette notion est introduite pour répondre à un souhait de concentration géographique des tournées. *Régionaliser* une tournée revient à générer des clusters de points qui soient des regroupements de clients proches et qui forment une partition de l'espace en zone distincte. Or, les solutions obtenues lorsque l'on cherche à minimiser la distance parcourue sont le plus souvent constituées de tournées qui ne répondent pas à ce critère. En regroupant les clients, on crée des zones avec lesquelles un conducteur peut se familiariser ; cela entraîne un gain de temps. L'objectif traditionnel de

minimisation de la longueur des tournées est reporté au niveau des décisions opérationnelles. L'objectif de régionalisation est pris en compte par la minimisation de la somme des coûts intra-cluster.

2.6.2.2 Objectifs sur les nœuds et les arcs

La plupart des études qui incorporent des objectifs sur les nœuds ou les arcs traitent de problèmes qui disposent de fenêtres horaires [81, 56, 67, 127, 126, 12]. Dans ce cas, soit les fenêtres de temps sont ignorées et remplacées par un objectif qui minimise le nombre de fenêtres violées [127, 126], soit le temps d'attente des clients et/ou du conducteur exprimé par les avances et retards par rapport aux bornes de la fenêtre horaire est comptabilisé [81, 56, 12], ou ces deux objectifs sont pris en compte en même temps [67].

D'autres objectifs sur les nœuds/arcs peuvent prendre en compte des notions de priorité entre les nœuds ou arcs ainsi que des dépendances entre ces derniers [120, 121]. Il est possible aussi de définir des notions économiques ou marketing : satisfaction de clients [144] ou relation entre les clients et les conducteurs [132].

Dans [19], la somme des distances entre certains nœuds du graphe qui représentent les domiciles des élèves et d'autres nœuds représentant les arrêts du car doit être minimisée. Cela revient à minimiser la distance que doivent marcher les élèves pour aller prendre le car.

2.6.2.3 Objectifs sur les ressources

Les ressources rencontrées dans la littérature sont les véhicules et les marchandises. Un objectif couramment étudié est la minimisation du nombre de véhicules. Cet objectif peut avoir une signification économique : moins il y a de véhicules, moins il y a d'investissement (achats, essence, salaires des conducteurs ...) [56, 29, 115, 155].

L'autre cas où l'on rencontre cet objectif est celui du problème d'élaboration de tournées de véhicules avec fenêtres horaires. La modélisation classique de ce problème comprend deux objectifs qui sont traités de manière lexicographique. Le premier objectif est la minimisation du nombre de véhicules, puis pour un nombre de véhicules fixé, la minimisation de la longueur de la solution. La plupart des études multi-objectif portant sur le problème d'élaboration de tournées de véhicules avec fenêtres horaires, à l'exception de l'étude de Hong et Park [81], comportent les deux objectifs de minimisation du nombre de véhicules et de coût de la solution. Ces deux objectifs ne sont pas pris en compte de manière lexicographique comme auparavant mais sont considérés avec le même niveau de priorité. Cette approche peut être une modélisation justifiée ; le problème vient du fait que les études comparent leurs résultats, notamment en termes de longueur, avec le problème mono-objectif sans prendre en compte le fait que, dans le cas mono-objectif, la longueur dépend du nombre de véhicules.

D'autres objectifs sur les véhicules peuvent être définis pour maximiser leur rentabilité en temps d'utilisation [144, 56] ou en capacité [150].

Les objectifs liés à la marchandise peuvent être introduits du fait de la nature de la

marchandise. Par exemple, s'il s'agit d'une denrée périssable, il peut être utile d'inclure un objectif qui minimise la détérioration de la marchandise. [120, 121].

2.6.3 Méthodes de résolution

Les techniques employées pour résoudre les problèmes sont multiples. Elles sont présentées ici selon deux catégories. La première concerne l'approche de résolution qui correspond à la manière dont la présence de multiples objectifs est prise en compte. La seconde concerne la méthode d'optimisation employée.

2.6.3.1 Prise en compte du caractère multi-objectif

La plupart des études utilisent des méthodes scalaires pour traiter la présence de plusieurs objectifs. Une majorité utilise la méthode d'agrégation [120, 121, 150, 101, 29, 118, 176]. Dans [12], la méthode d'optimisation employée est une colonie de fourmis et le choix entre les objectifs est fait dans la définition de la phéromone. D'autres méthodes scalaires comme le but programmé [120, 121], l'approche ϵ -contrainte [115] sont aussi utilisées. Dans la dernière étude, les auteurs essaient les différents nombres de véhicules possibles et pour un nombre de véhicules fixé cherchent à minimiser le makespan. Une approche par méthode lexicographique a aussi été proposée [88, 89]. Les autres études emploient des approches Pareto.

2.6.3.2 Méthodes d'optimisation

De nombreuses études utilisent des heuristiques spécifiques. Ces méthodes sont le plus souvent couplées avec des approches de résolutions scalaires et notamment des agrégations. Corboran et al. [29] utilisent une méthode de recherche par dispersion. Dans [115], la méthode d'optimisation employée est une recherche Tabou. El-Sherbeny [56] utilise le recuit simulé multi-objectif proposé par Ulungu et al. [163]. Les autres études emploient des algorithmes évolutionnaires. Dans le cas de Sessomboon et al. [144], de Lacomme et al. [97] et Tan et al. [156, 155], l'algorithme génétique est hybridé avec une méthode de descente. Paquete et al. [118] proposent une méthode de descente en deux phases. Dans un premier temps, le problème est résolu de manière mono-objectif, puis lorsqu'un optimum local est atteint, les deux objectifs du problème sont combinés à l'aide d'une agrégation et le problème ainsi obtenu est résolu par une méthode de descente.

2.7 Conclusion

Dans ce chapitre, nous avons présenté les caractéristiques générales des problèmes de tournées. Les deux problèmes présentés, le problème de la tournée couvrante et le problème d'élaboration de tournées de véhicules, en font partie. Il s'agit des deux problèmes à la base des deux problèmes multi-objectif étudiés dans cette thèse. Les problèmes de tournées multi-objectif sont aussi traités dans ce chapitre au travers d'un état de l'art.

Pour ce dernier, nous avons proposé une classification des problèmes selon le contexte (cas réel, extension ou généralisation) et une classification des objectifs selon la composante du problème sur laquelle ils portent (tournées, nœuds/arcs ou ressources). Nous avons aussi répertorié les méthodes d'optimisation utilisées. Il apparaît au travers de cet état de l'art que, si le domaine est assez jeune comme le montre le nombre de publications qui n'est pas très important, il suscite néanmoins de plus en plus d'intérêt ces dernières années. En effet, à part les trois premières études qui datent d'avant 1990, les autres sont toutes postérieures à 1995. Notamment, plus de la moitié des travaux répertoriés ont été réalisés après 2000.

TAB. 2.1 – Problèmes de tournées multi-objectif de la littérature.

Auteurs	Problème	Méthode	Tournées	Noeuds/Arcs	Ressources
<ul style="list-style-type: none"> ●Keller (1985) [88] ●Keller et Goodchild (1988) [89] 	Problème du voyageur de commerce avec profits	<ul style="list-style-type: none"> ●Méthode lexico-graphique ●Heuristique 	<ul style="list-style-type: none"> ●Min. la longueur de la tournée ●Max. le profit 		
Park et Koelling (1986, 1989) [120, 121]	Problème de tournées de véhicules	<ul style="list-style-type: none"> ●But programmé ●Heuristique 	<ul style="list-style-type: none"> Min. la distance parcourue 	<ul style="list-style-type: none"> ●Max. la réa-lisation des demandes ur-dances condition-nelles des stations 	Min. la détériora-tion des marchan-disés
Sutcliffe et Board (1990) [150]	Problème de tournées de véhicules	Programmation linéaire	<ul style="list-style-type: none"> ●Min. la distance totale ●Max. l'égalisa-tion des temps de voyages des véhicules 		Max. l'utilisation de la capacité
Bowerman et al. (1995) [19]	Tournées de cars scolaires en milieu urbain	<ul style="list-style-type: none"> ●Agrégation ●Heuristiques 	<ul style="list-style-type: none"> ●Min. la longueur totale ●Equilibrer la charge 	Min. la distance de marche des élèves	
Lee et Ueng (1998) [101]	Problème de tournées de véhicules	<ul style="list-style-type: none"> ●Agrégation ●Heuristique 	<ul style="list-style-type: none"> ●Min. la distance totale ●Equilibrer les charges de travail (longueur) 		

TAB. 2.2 – Problèmes de tournées multi-objectif de la littérature.

Auteurs	Problème	Méthode	Tournées	Noeuds/arcs	Ressources
Sessomboon et al. (1998) [144]	Problème de tournées de véhicules avec fenêtres horaires	<ul style="list-style-type: none"> • Approche Pareto • Algorithme génétique hybride 	Min. la distance totale	Max. la satisfaction des clients	<ul style="list-style-type: none"> • Min. le nombre de véhicules • Min. le temps d'attente des véhicules
Hong et Park (1999) [81]	Problème de tournées de véhicules avec fenêtres horaires	<ul style="list-style-type: none"> • But programmé • Heuristique 	Min. les temps de parcours totaux	Min. les temps d'attente des clients	<ul style="list-style-type: none"> • Min. le nombre de camions • Min. le nombre de camions courts • Min. le nombre de camions découverts • Min. les temps de travail inutiles
El-Sherbeny (2001) [56]	Problème de tournées de véhicules adapté au cas d'une compagnie de transport belge	<ul style="list-style-type: none"> • Approche Pareto • Recuit simulé 	<ul style="list-style-type: none"> • Min. le temps total • Equilibrage (longueur) • Max. la flexibilité 	Min. les temps d'attente	<ul style="list-style-type: none"> • Min. le nombre de camions découverts • Min. les temps de travail inutiles
Geiger (2001) [67]	Problème de tournées de véhicules avec fenêtres horaires	<ul style="list-style-type: none"> • Approche Pareto • Algorithme génétique 	Min. la distance totale	<ul style="list-style-type: none"> • Min. la violation des fenêtres de temps • Min. le nombre de violations 	Min. le nombre de véhicules

TAB. 2.3 – Problèmes de tournées multi-objectif de la littérature.

Auteurs	Problème	Méthode	Tournées	Nœuds/arcs	Ressources
●Rahoual et al. (2001) [127] ●Rahoual et Djoukhdjoukh (2003) [126]	Problème de tournées de véhicules avec fenêtres horaires	●Approche Pareto ●Algorithme générique	Min. la distance totale	Min. le nombre de contraintes violées	Min. le nombre de véhicules
Ribeiro et Lourenço [132]	Problème de tournées de véhicules avec périodes	●Agrégation ●Recherche locale itérée	●Min. la distance totale ●Equilibrage (nb. de clients visités)	Marketing : relation conducteur/client	
Corboran et al. (2002) [29]	Tournées de cars scolaires en zone rurale	●Agrégation ●Recherche par dispersion	Min. le makespan		Min. le nombre de véhicules
Baràn et Schaefer (2003) [12]	Problème de tournées de véhicules avec fenêtres horaires	Colomie de fourmis	Min. le temps de voyage total	Min. les temps de livraison	Min. le nombre de véhicules
Lacomme et al. (2003) [97]	Problème de tournées de véhicules sur les arcs	●Approche Pareto ●Algorithme générique	●Min. la longueur totale ●Min. le makespan		
Padeco et Marti (2003) [115]	Tournées de cars scolaires en zone rurale	●Méthode ϵ -contrainte ●Recherche Tabou	Min. le makespan		Min. le nombre de véhicules
Paquete et Stitzle (2003) [118]	Problème du voyageur de commerce multi-objectif	●Agrégation ●Recherche locale	Min. les longueurs totales		

TAB. 2.4 – Problèmes de tournées multi-objectif de la littérature.

Auteurs	Problème	Méthode	Tournées	Noeuds/arcs	Ressources
Zhenyu et al. (2003) [171]	Problème du voyageur de commerce multi-objectif	<ul style="list-style-type: none"> ● Approche Pareto ● Algorithme génétique 	Min. les longueurs totales		
Zografos et Androustisopoulos (2004) [176]	Problème de tournées de véhicules avec fenêtres pour le transport de produits dangereux	<ul style="list-style-type: none"> ● Agrégation ● Heuristique 	<ul style="list-style-type: none"> ● Min. la longueur ● Min. le risque 		
Tan et al. (2003) [156]	Problème de tournées de véhicules avec fenêtres horaires	<ul style="list-style-type: none"> ● Approche Pareto ● Algorithme génétique hybride 	Min. les longueurs totales		Min. le nombre de véhicules utilisés
Tan et al. (2003) [155]	Problème de tournées de véhicules avec camions et remorques	<ul style="list-style-type: none"> ● Approche Pareto ● Algorithme génétique hybride 	Min. les longueurs totales		Min. le nombre de véhicules utilisés
Mourgaya (2004) [111]	Problème de tournées de véhicules multi-périodique	Heuristiques	<ul style="list-style-type: none"> ● Régionalisation des clusters de visites associés aux véhicules ● Équilibrage de la charge (par jour ou par véhicule) 		

Chapitre 3

Algorithmes génétiques et méthodes de voisinage pour l'optimisation multi-objectif

Ce chapitre porte sur des méthodes pour résoudre les problèmes d'optimisation multi-objectif. Les méthodes présentées ici sont indépendantes du problème étudié. Notre approche repose sur la coopération entre deux méthodes complémentaires. Dans un premier temps, une approche évolutionnaire génère une approximation la plus diversifiée possible ; puis une méthode à base de voisinage améliore l'approximation en termes de convergence.

Au niveau des algorithmes évolutionnaires et plus particulièrement des algorithmes génétiques, nous introduisons le mécanisme de diversification élitiste permettant une meilleure diversification grâce à l'utilisation d'archives supplémentaires. Un modèle parallèle en îles basé sur ce nouveau mécanisme est aussi défini. Ensuite, nous proposons deux modifications de l'algorithme génétique multi-objectif standard NSGA II. La première modification est la parallélisation de NSGA II au travers d'un modèle en îles. La seconde modification est l'inclusion de la diversification élitiste dans NSGA II.

Au niveau des stratégies à base de voisinage, nous caractérisons les propriétés qu'elles doivent remplir. Par ailleurs, nous proposons trois stratégies qui prennent en compte le cahier des charges ainsi défini :

- La première stratégie utilise directement la notion de dominance Pareto pour guider les recherches.*
- La seconde stratégie se base sur le découpage de l'espace des objectifs.*
- La troisième stratégie, Target Aiming Pareto Search (TAPaS), utilise la définition de cibles pertinentes.*

3.1 Introduction

D'un point de vue méthodologique, notre approche de résolution est centrée sur l'utilisation d'algorithmes génétiques multi-objectif [28]. Ces méthodes connaissent un intérêt croissant depuis le début des années 1990 car elles sont particulièrement bien adaptées au traitement de problèmes multi-objectif où l'on recherche un ensemble de solutions. Pour améliorer l'efficacité des algorithmes génétiques utilisés, un nouveau mécanisme, la diversification élitiste, a été défini. Nous avons aussi étudié l'utilisation du parallélisme sous la forme de modèles en îles qui prennent en compte le mécanisme proposé et les particularités des algorithmes génétiques multi-objectif. Ce chapitre comporte une présentation générale de la diversification élitiste et du modèle parallèle qu'elle induit ainsi qu'une parallélisation de l'algorithme génétique multi-objectif standard NSGA II [44] et l'inclusion de la diversification élitiste dans NSGA II.

L'utilisation de méthodes de voisinage dans le cadre de l'optimisation multi-objectif a aussi été étudiée dans cette thèse. Trois stratégies pour guider les recherches locales dans le cadre multi-objectif sont proposées. Le but est de faire coopérer l'algorithme génétique et la méthode de voisinage au sein d'un modèle de coopération en deux phases.

Ce chapitre est organisé de la manière suivante : la section 3.2 présente les différents enjeux et points à définir lors de la conception de méthodes coopératives. Cette section présente aussi les différentes stratégies de coopération qui sont prises en compte dans cette thèse. La section 3.3 est dédiée aux algorithmes génétiques pour l'optimisation multi-objectif et notamment à la présentation d'un nouveau mécanisme, la diversification élitiste, et aux modifications que nous avons apportées à NSGA II. Dans la section 3.4, plusieurs stratégies pour l'application de méthodes de voisinage sur des ensembles potentiellement Pareto optimaux sont présentées.

3.2 Méthodes coopératives

La plupart des méthodes utilisées dans cette thèse emploient une stratégie de coopération. Une telle stratégie consiste à faire travailler plusieurs entités ensemble. Dans ce paragraphe, les entités qui composent un modèle coopératif sont appelées "méthodes". D'autre part, lorsqu'on se réfère "aux méthodes" d'un modèle coopératif, il n'est pas exclu qu'il s'agisse du même algorithme utilisé plusieurs fois. Dans cette section, nous donnons les principaux points qui définissent une approche coopérative. Puis nous présentons les deux stratégies qui sont utilisées dans cette thèse.

3.2.1 Coopération

Quand ? La coopération intervient donc dès que deux méthodes au moins doivent être appliquées. Il s'agit alors d'organiser un échange d'informations entre les méthodes. Pour cela, il faut définir dans quel sens se fait la communication et les conditions selon lesquelles la communication se déclenche.

Pourquoi ? Il y a coopération dès que l'on souhaite utiliser au moins deux méthodes pour résoudre un problème. Il peut aussi s'agir d'utiliser plusieurs fois une seule méthode en parallèle. Dans le premier cas, l'intérêt est d'exploiter des avantages complémentaires des méthodes. Ainsi, dans le cas du problème d'élaboration de tournées de véhicules avec équilibrage des tournées, si les algorithmes génétiques sont bien adaptés à la nature bi-objectif du problème, il est connu qu'ils ne sont pas parmi les méthodes les plus efficaces pour le problème d'élaboration de tournées de véhicules. A l'inverse, les recherches Tabou sont connues pour être efficaces pour le PTV mais leur adaptation à l'optimisation multi-objectif est plus délicate. En effet, l'absence d'ordre entre de nombreuses solutions du point de vue de la dominance empêche la sélection de la prochaine solution courante sans l'intervention de mécanismes supplémentaires comme l'agrégation qui peuvent être difficiles à paramétrer. Il est donc intéressant de les faire coopérer. L'exécution multiple et parallèle d'une méthode peut permettre, par exemple, d'explorer un plus grand nombre de solutions en un temps donné ou de partitionner l'espace de recherche pour faciliter la recherche.

Comment ? Deux méthodes peuvent communiquer selon quatre schémas de coopération : la coopération chaînée, la coopération imbriquée, la coopération parallèle et la coopération supervisée. Ces quatre schémas sont illustrés dans les figures 3.1 à 3.4. Dans la coopération chaînée, la première méthode est exécutée puis son résultat ou une partie de son résultat est fourni en entrée à la seconde méthode. Dans la coopération imbriquée, la seconde méthode est utilisée comme sous-routine de la première méthode. Lors de la coopération parallèle, les deux méthodes sont exécutées simultanément et peuvent s'échanger de l'information en cours de résolution. En termes parallèles, il s'agit d'un modèle en îles. Enfin dans la coopération supervisée, les deux méthodes s'exécutent indépendamment et, si besoin, une *méta-méthode* s'occupe de l'échange d'informations entre les deux méthodes. Ce type de coopération se différencie des autres modes de coopération par le fait que les méthodes ne communiquent pas directement entre elles. D'autre part, elle ne peut pas être vue comme une coopération imbriquée car la *méta-méthode* ne sert que d'interface entre les méthodes et n'intervient pas dans la résolution du problème traité.

Ces types de coopération peuvent se généraliser et se combiner pour prendre en compte la présence de plus de deux méthodes. Pour définir le modèle complet, il suffit de définir les topologies de communication entre les méthodes deux à deux.

Pour terminer la topologie, il faut aussi définir les événements qui amènent à l'échange d'informations. Si pour certains types de coopération, cette tâche est évidente, elle peut cependant parfois être complexe, notamment pour la coopération parallèle. En effet, l'échange d'informations peut être synchrone ou asynchrone. Il peut aussi être déterministe, c'est-à-dire selon une politique définie par le concepteur, ou stochastique, et dans ce cas il faut définir l'ensemble des événements qui déclenchent les phases de communication.

Quoi ? Le dernier point à définir est le type d'information échangée par les méthodes. Celui-ci dépend fortement du problème étudié ainsi que de la nature des méthodes employées.

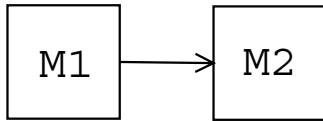


FIG. 3.1 – Coopération chaînée.

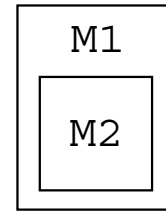


FIG. 3.2 – Coopération imbriquée.

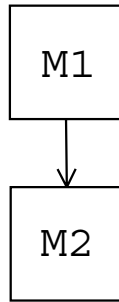


FIG. 3.3 – Coopération parallèle.

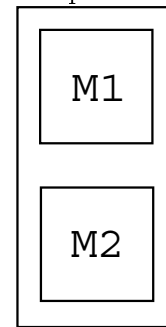


FIG. 3.4 – Coopération supervisée.

3.2.2 Stratégie en deux phases : diversification-intensification

La principale stratégie coopérative utilisée dans cette thèse est une coopération de type chaîné comportant deux étapes, chaque étape correspondant à une seule méthode. Durant la première étape, on applique une méthode pour obtenir un ensemble potentiellement Pareto optimal le plus diversifié possible. Au cours de la seconde étape, on tente d'améliorer cette première approximation. La méthode employée durant cette phase se focalise sur l'amélioration de l'approximation du point de vue de l'intensification, c'est-à-dire de la convergence vers la frontière Pareto. Il faut cependant veiller à ne pas dégrader l'approximation en termes de diversité.

Dans les faits, la première méthode est basée sur les algorithmes génétiques multi-objectif qui sont bien adaptés à cette tâche car ils travaillent sur des populations et on peut facilement leur incorporer des mécanismes pour améliorer la diversité des ensembles générés. La seconde phase utilise des méthodes de voisinage qui cherchent à améliorer individuellement les solutions de l'ensemble identifié lors de la première phase. Dans le cas du problème de la tournée couvrante bi-objectif, la seconde phase est construite autour d'un algorithme de séparations et coupes. Cependant, la politique de coopération étant spécifique au problème, elle est présentée en détail dans le chapitre 4 qui traite du problème de la tournée couvrante bi-objectif.

3.2.3 Modèles en îles

Dans cette thèse, nous avons aussi utilisé des modèles coopératifs parallèles à travers la définition de modèles en îles. Il s'agit d'exécuter plusieurs recherches avec différents degrés de synchronisation et de coopération. D'après le vocabulaire défini précédemment,

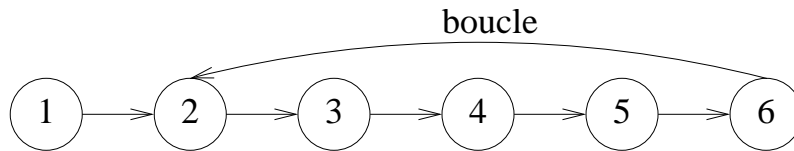
il s'agit donc de plusieurs méthodes liées par des schémas de coopération parallèle. Le but du parallélisme n'est pas ici d'accélérer la recherche pour réduire les temps de calcul nécessaires mais dans un temps donné d'améliorer la qualité des solutions générées. Dans ces travaux, nous avons principalement exploré l'utilisation de modèles en îles avec des topologies de communication en anneaux et utilisant des politiques de communication synchrones.

3.3 Algorithmes génétiques multi-objectif

Le but d'une méthode d'optimisation multi-objectif dans le cas d'une approche a posteriori est lui-même bi-objectif. Il s'agit en effet de trouver une approximation qui soit de bonne qualité aussi bien en termes de convergence vers le front Pareto optimal qu'en termes de diversification. Les algorithmes évolutionnaires, et notamment les algorithmes génétiques, semblent particulièrement bien adaptés à cette tâche du fait qu'ils travaillent sur une population de solutions. Cette caractéristique leur permet de trouver plusieurs solutions potentiellement Pareto optimales en une seule exécution plutôt que plusieurs exécutions indépendantes comme c'est le cas pour les méthodes classiques. D'autre part, les algorithmes évolutionnaires sont moins affectés par la forme et par la continuité de la frontière Pareto qui sont des problèmes pour les méthodes scalaires.

L'intérêt des algorithmes évolutionnaires pour l'optimisation multi-objectif a été suggéré par Rosenberg dans les années 1960 [137], mais c'est Schaffer qui proposa la première implémentation d'un algorithme génétique pour l'optimisation multi-objectif en 1985 [138]. Cependant, le domaine connaît un intérêt grandissant depuis le milieu des années 1990 avec l'apparition de plusieurs techniques et applications utilisant des approches Pareto. Actuellement, l'intérêt porté à ces méthodes est très important comme le montre la publication de livres dédiés à ce sujet [42, 28] ou le nombre important de publications de tout genre référencées sur la page de Coello Coello, <http://delta.cs.cinestav.mx/~coello/EM00>, qui comporte plus de 1600 entrées.

Coello Coello et al. [28] proposent une classification simple en deux générations de méthodes. Dans la première génération, ils rangent tous les algorithmes évolutionnaires qui prennent en compte la présence de plusieurs objectifs par des approches scalaires ou des approches non scalaires et non Pareto. Les méthodes de première génération comprennent aussi les premières études utilisant des approches Pareto : pure ranking Pareto [73], Multi-Objective Genetic Algorithm (MOGA) [63], Nondominated Sorting Genetic Algorithm (NSGA) [146], Niched-Pareto Genetic Algorithm (NPGA) [82, 83] et NPGA 2 [57]. Les méthodes de seconde génération utilisent toutes des approches Pareto. Elles se différencient des méthodes de première génération par l'inclusion de populations secondaires et de méthodes avancées pour la génération de solutions qui sont à la fois non-dominées et uniformément distribuées. Parmi ces méthodes, on peut citer : Pareto Archived Evolution Strategy (PAES) [94], Pareto Archived Envelope-based Selection Algorithm (PESA) [33], Pareto Envelope-based Selection Algorithm-II (PESA-II) [32], Strength Pareto Evolutionary Algorithm (SPEA) [175], Strength Pareto Evolutionary Algorithm 2 (SPEA2) [174], Nondominated Sorting Genetic Algorithm II (NSGA II) [44], Multi-objective Messy Genetic Algorithm (MOMGA) [166], Multi-objective Messy Genetic Algorithm II (MOMGA



Les étapes sont les suivantes :

1. *Initialisation de la population.*
2. *Évaluation de l'efficacité des individus de la population.*
3. *Sélection.*
4. *Croisement.*
5. *Mutation.*
6. *Remplacement.*

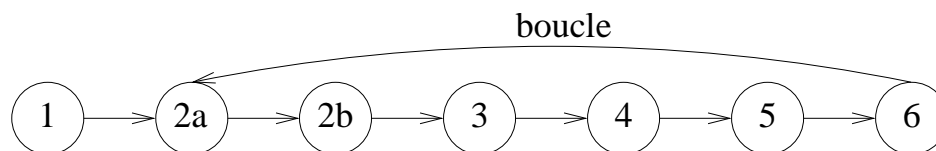
FIG. 3.5 – Etapes d'un algorithme génétique.

II) [177], Micro Genetic Algorithm for Multiobjective Optimization [27].

Cette section se concentre sur l'adaptation des algorithmes génétiques à l'optimisation multi-objectif à l'aide d'approches Pareto. Elle est organisée de la manière suivante : les principaux mécanismes pour les algorithmes génétiques sont tout d'abord décrits. Nous proposons ensuite un nouveau mécanisme pour favoriser la diversification dans les algorithmes génétiques : la diversification élitiste. Un modèle en îles se basant sur ce mécanisme est aussi défini. Puis, l'algorithme génétique classique NSGA II est présenté. Nous avons modifié cet algorithme au travers de deux approches. La première approche consiste en la parallélisation de NSGA II à l'aide d'un modèle en îles. Lors de la seconde approche, nous avons modifié NSGA II pour lui inclure le mécanisme de diversification élitiste.

3.3.1 Mécanismes pour l'optimisation multi-objectif

Un algorithme génétique peut se décomposer selon les différentes étapes données dans la figure 3.5 : l'adaptation des algorithmes génétiques à l'optimisation multi-objectif se fait principalement au niveau de l'étape d'évaluation de l'efficacité d'une solution qui est alors divisée en deux étapes : évaluation de l'efficacité de la solution en termes de convergence et évaluation de l'efficacité de la solution en termes de diversification. Le schéma de l'algorithme génétique avec ajout de ces phases est représenté dans la figure 3.6. L'étape 2.a correspond au calcul de l'efficacité par rapport à la convergence et l'étape 2.b correspond au calcul de la diversité de la solution. Les principaux mécanismes employés pour ces tâches sont présentés dans les sous-sections qui suivent. Il est à noter que selon les mécanismes employés, il est possible d'obtenir deux efficacités différentes pour une même solution : une pour la convergence, l'autre pour la diversification. Il est alors nécessaire de prendre en compte ce fait dans la phase de sélection lorsqu'il s'agit de comparer deux solutions. Un autre mécanisme, l'élitisme, est présenté dans la sous-section 3.3.1.3.



Les étapes de l'algorithme sont les mêmes que celles de la figure 3.5. L'étape 2 est scindée en deux sous-étapes. La première (2a) correspond à l'évaluation de la qualité de la solution en termes de convergence et la seconde (2b) à l'évaluation de la qualité de la solution en termes de diversité.

FIG. 3.6 – Etapes d'un algorithme génétique multi-objectif.

3.3.1.1 Assignment de l'efficacité

L'idée de calculer l'efficacité d'un individu en utilisant la dominance Pareto a été introduite par Goldberg [73]. En général, on parle alors du rang de l'individu. Plusieurs approches ont été proposées pour assigner un rang aux différentes solutions. La première technique consiste à attribuer à chaque solution un rang de dominance qui correspond au nombre de solutions dans la population qui domine l'individu. Le rang d'un individu est alors égal au nombre d'individus de la population qui le domine plus un. Cette technique est notamment employée par l'algorithme de Fonseca et Fleming [63].

Une autre possibilité est d'attribuer une profondeur de dominance. Il s'agit de diviser la population en plusieurs fronts et le rang d'une solution correspond à la profondeur du front auquel elle appartient. Le rang d'un individu est calculé par l'algorithme suivant. Les individus non-dominés de la population reçoivent le rang 1 et forment l'ensemble E_1 . Les individus qui ne sont dominés que par des solutions de E_1 reçoivent le rang 2 et forment l'ensemble E_2 . De manière générale, un individu ne reçoit le rang k que s'il est dominé par des individus de la population appartenant à l'ensemble $E_1 \cup E_2 \cup \dots \cup E_{k-1}$. Cette stratégie est notamment utilisée dans l'algorithme NSGA II [44].

Une dernière possibilité, le compte de dominance, consiste à compter le nombre de solutions qu'un individu domine. Cette mesure ne donne pas tout de suite une information sur l'efficacité de la solution et doit être utilisée, à l'instar de ce qui est fait dans SPEA et SPEA 2 [175, 174], en conjonction avec une autre technique.

Les différentes techniques sont illustrées dans la figure 3.7. Leur intérêt, par rapport à des méthodes scalaires ou autres, est qu'elles permettent d'évaluer l'efficacité d'une solution par rapport à tout l'ensemble de la population et qu'elles ne sont pas affectées par la forme ou la continuité de la frontière Pareto.

3.3.1.2 Préservation de la diversité

Les méthodes présentées précédemment tendent à favoriser la convergence vers le front Pareto optimal en avantageant les individus qui sont peu ou pas dominés. Cependant, ces méthodes ne sont pas capables de garantir que l'approximation obtenue sera de bonne qua-

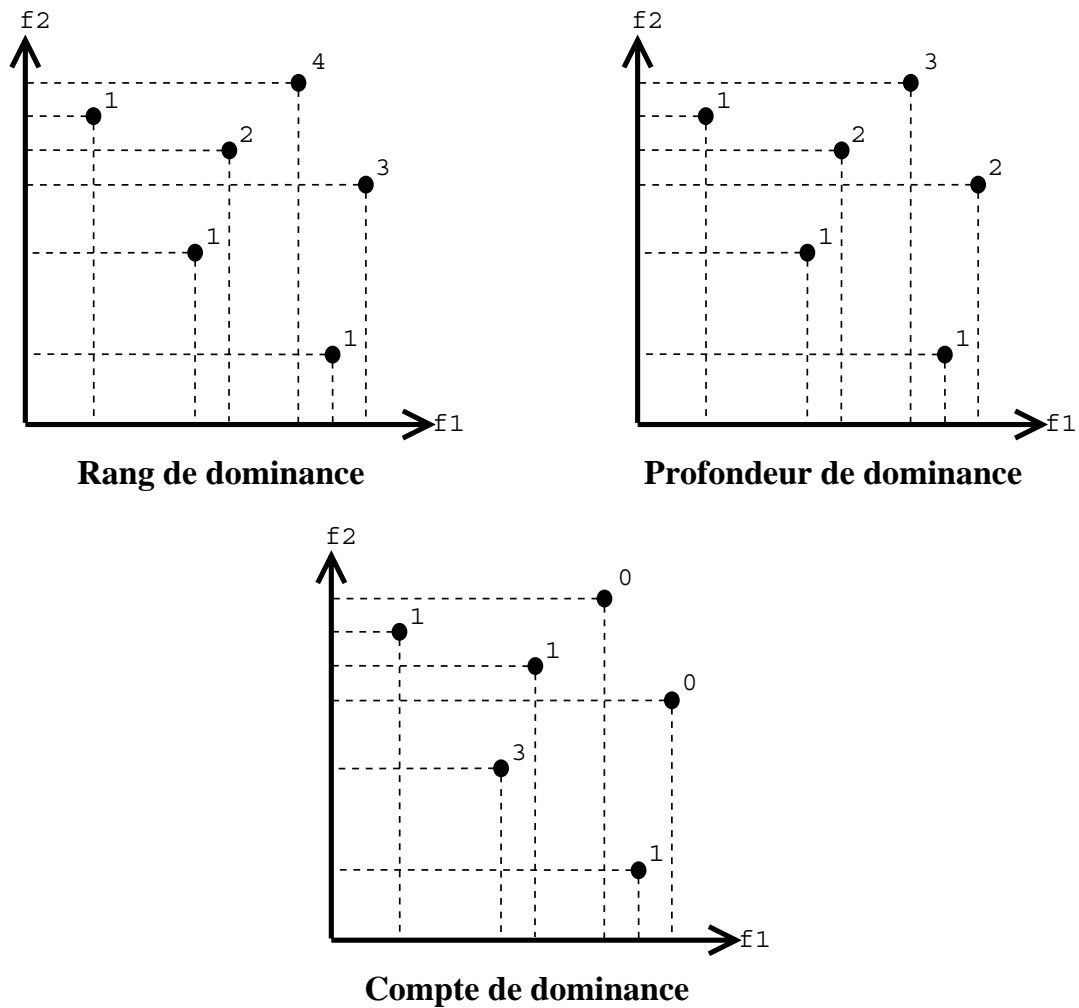


FIG. 3.7 – Assignment de l'efficacité d'un point de vue Pareto.

lité en termes de diversification. Ainsi, les algorithmes génétiques incorporent des méthodes qui évaluent la diversité d'une solution par rapport aux autres solutions de la population. Les différentes approches prennent en compte une notion de voisinage qu'il est possible de définir de trois manières différentes. Comme suggéré par Zitzler et al. [173], les trois catégories sont basées sur celles utilisées en estimation de densité statistique [145].

Les méthodes à noyau définissent le voisinage d'une solution i par rapport à une fonction K qui prend en paramètre la distance entre la solution et une autre solution. D'une manière générale, les distances d_{ij} entre i et les autres points j sont calculées et la somme des $K(d_{ij})$ effectuée. Cette somme représente la densité estimée autour de l'individu i . Le fitness sharing [75], utilisé notamment dans MOGA [63], NSGA [146] et NPGA [82], est certainement la technique la plus populaire dans les algorithmes évolutionnaires.

Les techniques de voisin le plus proche calculent la distance entre un point donné et son $k^{\text{ème}}$ voisin le plus proche pour estimer la densité dans son voisinage. Cette technique

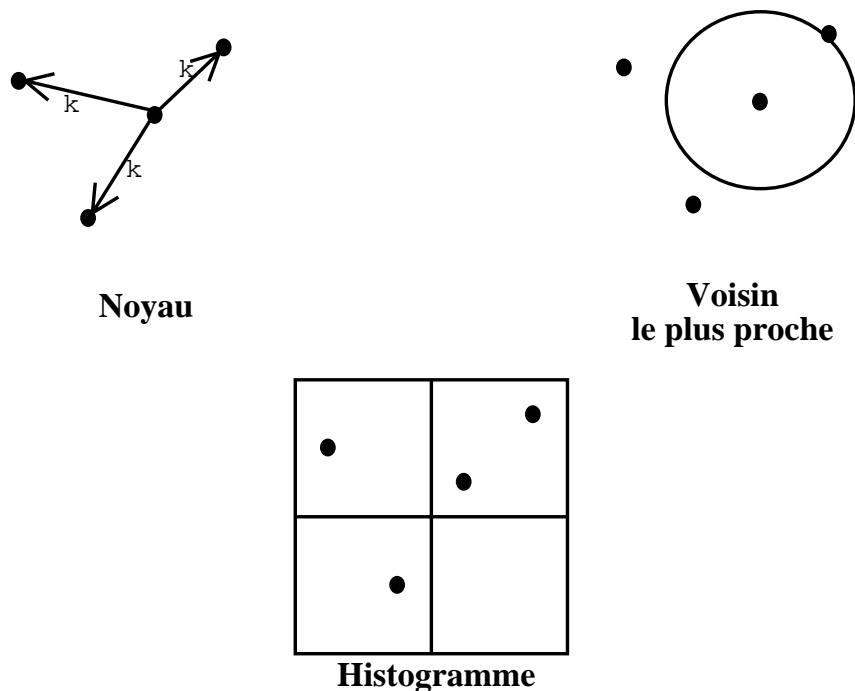


FIG. 3.8 – Illustration des techniques de préservation de la diversité.

a notamment été utilisée dans SPEA2 [174].

Une troisième catégorie utilise les histogrammes. Dans cette technique, l'espace est divisé en voisinages par une hypergrille. La densité autour d'une solution est estimée par le nombre de solutions se trouvant dans la même case de la grille. Cette technique a été employée dans PAES [94].

Les différentes approches sont illustrées dans la figure 3.8. Pour toutes ces méthodes, il est nécessaire de définir une métrique qui peut être calculée sur le génotype et/ou le phénotype dans l'espace décisionnel et/ou dans l'espace des objectifs.

3.3.1.3 Elitisme

L'élitisme consiste à garder les meilleures solutions rencontrées au cours de la recherche. Si cela peut être fait directement dans la population principale comme pour NSGA II [44], le plus souvent une population secondaire, appelée archive, est ajoutée à l'algorithme génétique. Son premier rôle consiste à éviter que les solutions non-dominées trouvées lors de la recherche ne soient perdues à cause de la nature stochastique des algorithmes génétiques. Une autre utilisation de cette archive est l'utilisation de solutions de l'archive pour la phase de sélection. Cette méthode permet une amélioration de la convergence vers la frontière Pareto.

Du fait de la limitation en mémoire et du coût de maintenance de l'archive, il peut ne pas être possible de garder toutes les solutions non-dominées trouvées. Plusieurs stratégies,

prenant en compte la dominance et la diversité, ont été proposées pour réduire la taille de l'archive sans perdre d'information intéressante.

3.3.2 La diversification élitiste

Durant notre étude sur le problème d'élaboration de tournées de véhicules avec équilibrage des tournées, il est apparu que le mécanisme de diversification standard utilisé, le fitness sharing [75], n'était pas capable de diversifier suffisamment la recherche pour que l'algorithme génétique puisse générer des solutions de bonne qualité par rapport à l'objectif de minimisation de la longueur parcourue. Il a été constaté que l'algorithme pour un problème de flowshop bi-objectif proposé par Talbi et al. [152] adapté au PTVET tend à converger prématurément vers une zone spécifique de l'espace de recherche. La figure 3.9 illustre les résultats obtenus sur deux benchmarks standard pour le problème d'élaboration de tournées de véhicules. Cela semble être dû au fait que l'algorithme génétique évite d'explorer l'espace des objectifs en direction de l'objectif le plus dur à optimiser.

Pour préserver et améliorer la diversification tout au long de la recherche, un nouveau mécanisme, qui peut s'utiliser en collaboration avec un mécanisme standard tel que le fitness sharing, est proposé : la diversification élitiste. Cette technique est inspirée de l'élitisme : des archives additionnelles sont maintenues et des solutions de ces archives sont utilisées à chaque génération durant la phase de recombinaison. Les archives stockent les solutions potentiellement Pareto optimales lorsqu'un objectif est maximisé au lieu d'être minimisé. Soit $S(A)$ l'ensemble des solutions trouvées par un algorithme A et k l'indice de la fonction objectif qui est maximisée. L'opérateur de dominance \preceq_k est alors défini de la manière suivante :

$$\begin{aligned} \forall y, z \in S(A), y \prec_k z \iff & (\forall i \in \{1 \dots n\} \setminus \{k\}, f_i(y) \leq f_i(z)) \\ & \wedge (f_k(y) \geq f_k(z)) \\ & \wedge ((\exists i \in \{1 \dots n\} \setminus \{k\}, f_i(y) < f_i(z)) \\ & \vee (f_k(y) > f_k(z))) \end{aligned}$$

On a alors $A_k = \{s \in S(A) \mid \forall s' \in S(A), s' \not\prec_k s\}$, avec $k = 1, \dots, n$. A_k est donc l'archive de solutions potentiellement Pareto optimales lorsque le $k^{\text{ème}}$ objectif est maximisé plutôt que minimisé. La mise à jour et l'utilisation des solutions se trouvant dans les archives supplémentaires de type A_k dépendent de la méthode définie par le concepteur. On définit \preceq_0 comme l'opérateur de dominance classique et A_0 correspond à l'archive standard ou à l'ensemble de solutions ayant le même rôle dans la méthode utilisée ; par exemple, la méthode NSGA II ne possède pas d'archive proprement dite et ce sont les individus de rang 1 de la population qui jouent ce rôle. Les frontières Pareto correspondantes aux différentes archives dans le cas bi-objectif sont illustrées dans la figure 3.10. A l'instar de ce qui peut être fait dans l'élitisme, des solutions des nouvelles archives sont utilisées durant la phase de recombinaison de l'algorithme génétique à chaque génération. Le rôle de ces solutions est d'attirer l'algorithme vers des zones inexplorées et d'éviter ainsi une convergence prématurée vers certaines zones de l'espace des objectifs.

Des tests ont montré que l'apport est moindre si toutes les nouvelles archives sont incluses dans le même algorithme génétique. D'autre part, les solutions présentes dans les

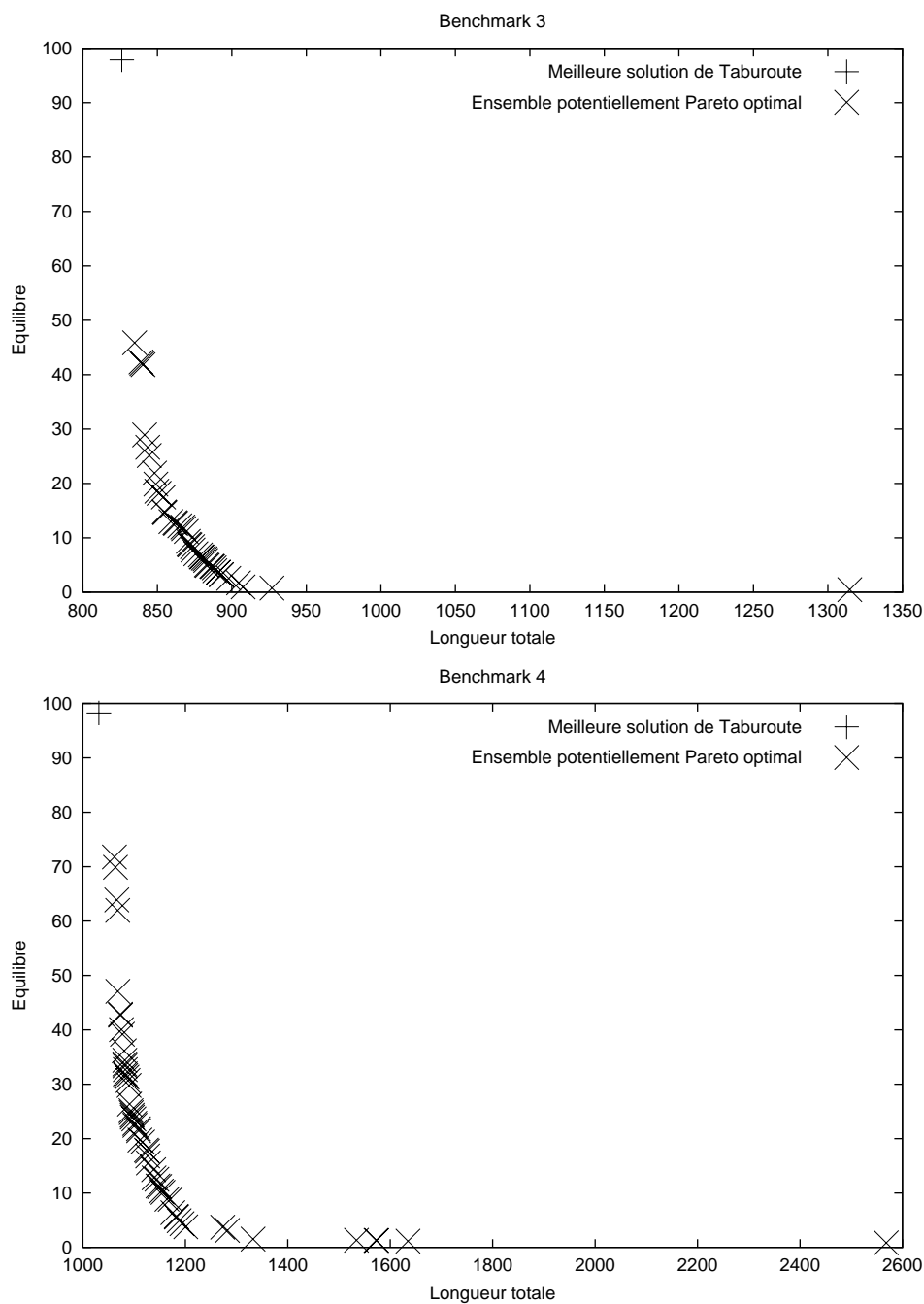


FIG. 3.9 – Ensemble potentiellement Pareto optimal généré par un AG multi-objectif standard et la solution trouvée par TabuRoute pour deux benchmarks.

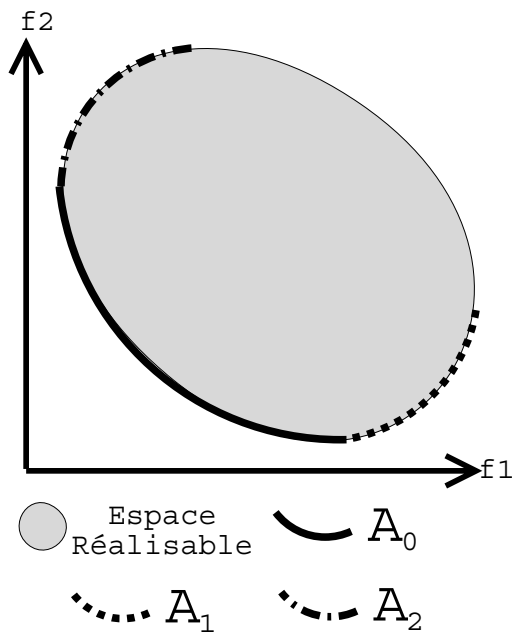


FIG. 3.10 – Frontières Pareto correspondantes aux différentes archives de la diversification élitiste pour deux objectifs.

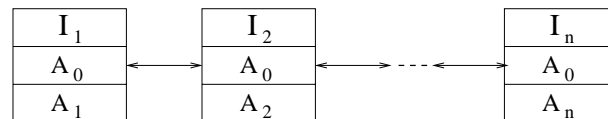


FIG. 3.11 – Modèle coopératif de base pour la diversification élitiste - la structure torique n'est pas montrée pour ne pas obscurcir la figure.

archives supplémentaires doivent intervenir durant la phase de recombinaison et il n'est pas envisageable d'utiliser un trop grand nombre d'archives, ce qui est le cas pour les problèmes comportant plus de deux objectifs. Ainsi, les différentes archives ont été réparties entre plusieurs recherches qui coopèrent à travers un modèle en îles. Dans le cas général avec n objectifs, le modèle en îles est composé de n îles correspondant chacune à un algorithme génétique. Une île est notée I_k . Chaque île I_k possède deux types d'archive : A_0 et A_k . A intervalles réguliers, I_k envoie son archive A_0 à ses deux voisines I_{k-1} et I_{k+1} . La topologie de communication étant torique, les calculs sur k se font modulo n . Ce modèle coopératif et sa topologie de communication sont illustrés dans la figure 3.11. Pour que l'information échangée ait un impact sur le déroulement de l'algorithme génétique, il faut que la stratégie élitiste employée prévoit l'utilisation de solutions de A_0 durant la phase de recombinaison. Ainsi, si on considère que les archives A_0 représentent l'avancée de l'algorithme génétique en termes de convergence, cet échange permet d'éviter qu'une des îles ne soit distancée par les autres du point de vue de la convergence.

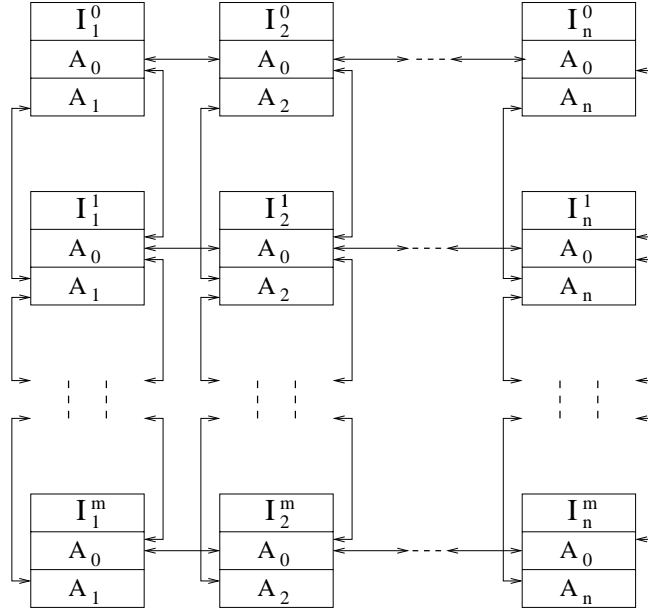


FIG. 3.12 – Modèle coopératif de base pour la diversification élitiste - la structure torique n'est pas montrée pour ne pas obscurcir la figure.

3.3.3 Extension parallèle de la diversification élitiste

Le modèle coopératif décrit précédemment forme la brique élémentaire d'un modèle en îles plus général utilisé pour renforcer l'impact de la diversification élitiste et améliorer les résultats en termes de diversification et de convergence. La parallélisation n'est pas utilisée pour accélérer les temps de calcul mais pour identifier un plus grand nombre de solutions en un temps donné. Le nouveau modèle en îles est illustré dans la figure 3.12. Une île est alors notée I_j^i . Cela signifie qu'elle appartient à la $i^{\text{ème}}$ brique et que son archive supplémentaire est de type A_j . L'île I_j^i envoie son archive A_0 à tous ses voisins : $I_{j-1}^i, I_{j+1}^i, I_j^{i-1}$ et I_j^{i+1} . Par contre, elle ne communique son archive A_j qu'à ses voisines n'appartenant pas à la même brique, c'est-à-dire I_j^{i-1} et I_j^{i+1} . Du fait de la structure torique de la topologie de communication, les calculs sur i et j sont effectués modulo n .

3.3.4 Présentation de l'algorithme NSGA II

NSGA II a été proposé par Deb et al. [44] pour répondre aux critiques qui avaient été faites à NSGA [146]. NSGA II fonctionne de la manière suivante : sa population R_t , où t est le numéro de la génération courante, est divisée en deux sous-populations P_t et Q_t . Les tailles de P_t et Q_t sont égales à N et la taille de R_t est donc $2N$. La sous-population P_t correspond aux parents et Q_t aux enfants après la phase de recombinaison. Nous allons maintenant présenter les principales étapes d'une génération de NSGA II. Les mécanismes employés durant cette génération, comme la distance de crowding, sont expliqués par la suite. Les principales étapes d'une génération t de l'algorithme sont :

Etape 1 Combiner les populations de parents et d'enfants et créer $R_t = P_t \cup Q_t$. Appliquer

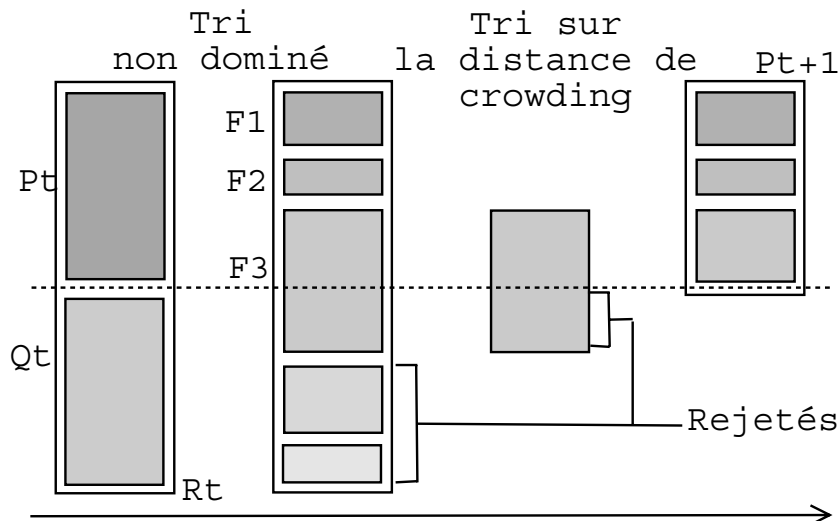


FIG. 3.13 – L'algorithme évolutionnaire NSGA II.

un tri non-dominé sur R_t et identifier les différents fronts \mathcal{F}_i , $i = 1, \dots$, etc.

Etape 2 Créer une nouvelle population $P_{t+1} = \emptyset$. Initialiser $i = 1$. Tant que $|P_{t+1}| + |\mathcal{F}_i| < N$, on fait $P_{t+1} = P_{t+1} \cup \mathcal{F}_i$ et $i = i + 1$.

Etape 3 Lorsque toutes les solutions de \mathcal{F}_i ne peuvent être intégrées à P_{t+1} , on fait un tri sur les distances de crowding et on inclut à P_{t+1} les $N - |P_{t+1}|$ solutions les plus éparpillées en utilisant la distance de crowding dans l'ensemble \mathcal{F}_i .

Etape 4 Créer la population d'enfants Q_{t+1} à partir de P_{t+1} en utilisant la sélection par tournoi à base de crowding et les opérateurs génétiques.

Le déroulement d'une génération est résumé dans la figure 3.13.

Dans l'étape 1, le tri non-dominé utilise la notion de profondeur de dominance. Les distances de crowding de l'étape 2 permettent d'améliorer la diversité de l'ensemble généré durant la recherche. Cette mesure donne l'estimation de la densité de solutions entourant une solution s dans la population. Cette densité est donnée par une approximation du périmètre du cuboïd formé par les voisins appartenant au même ensemble \mathcal{F}_i les plus proches de s (cf. figure 3.14). La distance de crowding des solutions d'un ensemble \mathcal{F}_i donné se calcule par l'algorithme 3.1.

La sélection par tournoi de l'étape 3 fonctionne de la manière suivante. Un individu i possède deux attributs : son rang r_i donné par le tri non-dominé et une distance de crowding d_i . Une solution i gagne le tournoi contre un individu j si l'une des conditions suivantes est vraie :

1. $r_i < r_j$
2. $r_i = r_j$ et $d_i > d_j$.

La première condition assure que la solution choisie appartient à un meilleur ensemble non-dominé. En cas d'égalité, la seconde condition permet de choisir la solution qui apporte le plus de diversification.

Algorithme 3.1 Calcul de la distance de crowding.

$l = |\mathcal{F}_i|$

pour tout objectif m **faire**

trier \mathcal{F}_i par rapport aux valeurs de l'objectif f_m par ordre décroissant. On obtient alors le vecteur d'indices triés I^m .

fin pour

$d_{I_1^m} = d_{I_l^m} = \infty$

pour $j = 2, \dots, l - 1$ **faire**

$d_{I_j^m} = \frac{f_m(s_{I_{j+1}^m}) - f_m(s_{I_{j-1}^m})}{f_m^{\max} - f_m^{\min}}$

fin pour

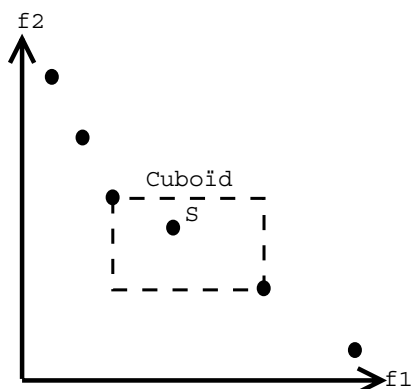


FIG. 3.14 – Calcul de la distance de crowding.

La solution de NSGA II est l'ensemble des solutions non-dominées de la population finale. Cependant, si le nombre de solutions Pareto optimales est plus grand que N , certaines de ces solutions seront perdues lors de la recherche. Pour palier à ce problème, une archive a été ajoutée et permet de conserver l'ensemble des solutions potentiellement Pareto optimales trouvées au cours de l'exécution de l'algorithme. Cette archive n'est utilisée que dans un but de stockage et les solutions qui s'y trouvent ne sont jamais utilisées durant le déroulement de l'algorithme. C'est notamment le cas lorsque l'on ajoute la diversification élitiste à NSGA II comme cela est présenté ci-dessous.

3.3.5 Extension de NSGA II en modèle en îles

Pour améliorer les résultats obtenus avec NSGA II, nous l'avons utilisé au sein d'un modèle en îles. Le modèle est construit de la manière suivante : chaque île correspond à une instanciation de NSGA II. Le réseau de communication est en forme d'anneau, chaque île ayant deux voisins. Une île envoie de l'information à ses deux voisines à intervalles réguliers en termes de générations. La phase d'échange a lieu entre la phase de sélection et la phase de recombinaison. Du fait de la topologie de communication, une île reçoit de l'information en même temps qu'elle en envoie. Les calculs propres à une île ne reprennent

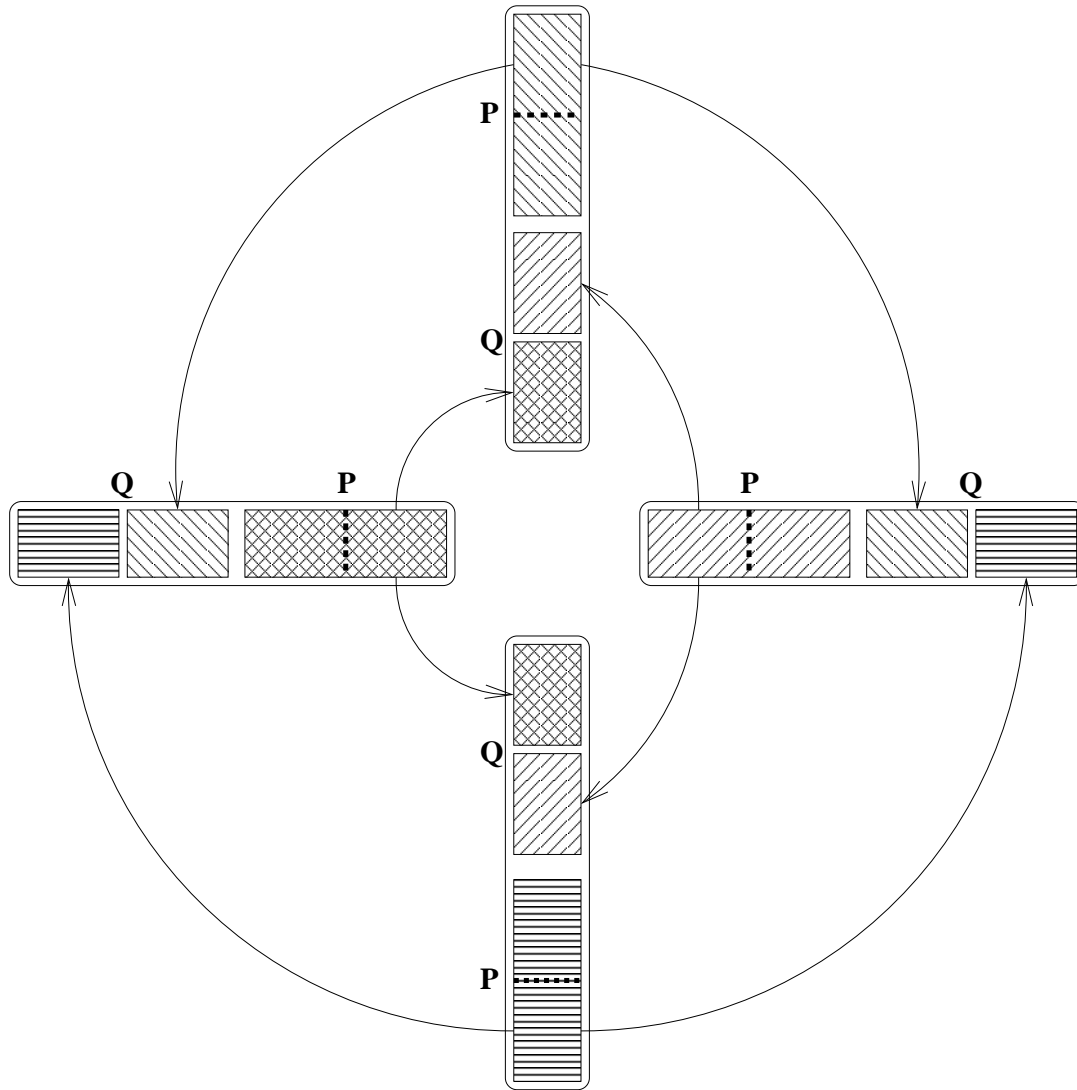


FIG. 3.15 – Extension de NSGA II en modèle en îles.

que lorsqu'elle a reçu les informations en provenance de ses deux voisines.

Il reste maintenant à décrire l'information envoyée par une île et la façon dont elle intègre l'information reçue dans son exécution. Une île envoie à ses deux voisines les $\frac{N}{2}$ meilleures solutions de sa population. C'est-à-dire les $\frac{N}{2}$ premières solutions de P_t après application du tri non-dominé et du tri sur le crowding. De la même manière, une île reçoit donc deux fois $\frac{N}{2}$ solutions. Les solutions reçues remplacent celles de Q_t qui seraient de toute façon perdues lors de la prochaine phase de recombinaison. Avant la phase de recombinaison, de nouveaux tris de dominance et de crowding sont effectués sur la nouvelle population ainsi obtenue. La figure 3.15 illustre les communications dans le cas où quatre îles sont utilisées.

3.3.6 Inclusion de la diversification élitiste dans NSGA II

Le but recherché ici est d'ajouter la gestion des archives supplémentaires en modifiant NSGA II le moins possible. Comme indiqué précédemment, il peut être remarqué que NSGA II n'utilise pas d'archive standard de type A_0 et que c'est la population principale qui assure la sauvegarde des solutions non dominées. L'application de la diversification élitiste à NSGA II consiste donc à ajouter une archive par île. Cette archive intervient dans la phase de recombinaison : k individus sont choisis parmi les individus se trouvant dans l'archive et forment l'ensemble C_t . Ensuite, la phase de recombinaison se passe normalement sauf en ce qui concerne le tournoi de sélection des parents. On choisit au hasard deux parents pa_1 et pa_2 parmi les solutions de $P_t \cup C_t$. Si soit pa_1 soit pa_2 appartient à C_t , l'individu qui gagne le tournoi est celui provenant de C_t . Si les deux solutions appartiennent à C_t , on en choisit une de manière aléatoire. Si aucune des deux solutions n'appartient à C_t , on utilise le tournoi classique de NSGA II. Les archives sont mises à jour après chaque phase de recombinaison ; on essaye d'inclure dans l'archive les individus générés durant cette phase.

La politique d'échange d'information est la même que celle définie en 3.3.5. Cependant, comme dans ce modèle une île possède quatre voisines, elle communique à ses voisines les $\frac{N}{4}$ meilleures solutions de P après application du tri non-dominé et du tri sur la distance de crowding. Ainsi une île reçoit quatre fois $\frac{N}{4}$ solutions qui viennent remplacer celles de Q . Il existe des cas particuliers. Dans le cas bi-objectif, une brique élémentaire n'est formée que de deux algorithmes génétiques. Donc, une des îles recevrait deux fois les $\frac{N}{4}$ meilleures solutions de la population P de la seconde île. Ce cas n'est pas très intéressant et on modifie alors la politique pour qu'une île envoie les $\frac{N}{2}$ meilleures solutions de sa population P à sa voisine. Le même cas de figure peut se produire entre deux îles ne se trouvant pas dans la même brique élémentaire si le modèle implémenté ne comporte que deux briques.

3.4 Méthodes de voisinage pour l'optimisation multi-objectif

Cette section traite de l'application de méthodes de voisinage sur des ensembles potentiellement Pareto optimaux. Les méthodes de voisinage peuvent être utilisées pour générer un ensemble potentiellement Pareto optimal [143, 161, 77, 93] ou pour être hybridées avec d'autres méthodes [1, 42, 86, 85]. L'idée dans les approches proposées ci-dessous n'est pas de créer un ensemble à partir de rien mais d'appliquer les recherches sur les solutions d'un ensemble diversifié généré par une première méthode et de l'améliorer par des méthodes de voisinage.

Les différentes stratégies proposées travaillent sur le principe suivant : une méthode de voisinage est appliquée sur un sous-ensemble représentatif de l'approximation fournie ou sur toutes les solutions de l'approximation de départ. Dans la définition de stratégie pour ce type d'approche, trois points doivent être pris en compte :

- Il faut guider les méthodes pour que ces dernières convergent vers la frontière Pareto. La plupart des études portant sur l'application de méthodes de voisinage utilisent des agrégations. Les poids sont choisis soit en prenant en compte la forme globale de l'ensemble potentiellement Pareto optimal courant [42], soit en appliquant plusieurs poids [1], soit en les choisissant aléatoirement ; dans ce cas, la procédure est le plus

souvent itérée [85]. Deux applications de recuit simulé dans le cadre de l'optimisation multi-objectif ont été proposées par Serafini [143] et par Czyzak et Jaszkiwicz [37]. Dans ces études, le critère d'acceptation de la solution courante suivante ne dépend plus seulement de la température mais aussi de la relation de dominance entre la solution courante et la solution candidate.

- Il faut éviter que deux algorithmes cherchent dans la même zone de l'espace des objectifs pour qu'il n'y ait pas des calculs inutiles tout en veillant à ce que la totalité de l'espace des objectifs dominant l'approximation de départ soit prise en compte.
- Il est souhaitable que les méthodes soient définies pour être les plus indépendantes possibles les unes des autres. En effet, le nombre de solutions dans une approximation peut être important et le nombre de recherches à lancer pour satisfaire le point précédent sera alors lui aussi important. Les temps de calcul dans cette stratégie peuvent donc devenir rapidement problématiques. De ce fait, si les différentes recherches sont suffisamment indépendantes les unes des autres, elles pourront être lancées parallèlement de manière facile et efficace. Cependant, il est clair qu'un minimum de coopération entre les méthodes de voisinage est intéressant.

Dans les trois parties qui suivent, nous proposons trois stratégies à base de méthode de voisinage qui tentent de prendre en compte les remarques faites précédemment.

3.4.1 Recherche guidée par l'archive

L'absence d'ordre total, si on souhaite éviter la définition de poids ou l'utilisation de méthodes équivalentes, entre les solutions d'un point de vue multi-objectif pose des problèmes pour les méthodes de voisinage qui travaillent sur une seule solution à la fois. En effet, dans le cas d'une recherche locale, le choix doit se faire entre toutes les solutions qui dominent la solution courante. Même en se restreignant aux seules solutions de cet ensemble qui ne sont pas dominées par d'autres solutions dominant la solution courante, le nombre de candidats peut ne pas être négligeable. D'autre part, les solutions de cet ensemble peuvent être extrêmement différentes les unes des autres et le choix est alors difficile.

Plusieurs études proposent l'utilisation d'une approche Pareto dans le cadre de méthodes de voisinage. L'une des méthodes possibles est la suivante [152] : la méthode travaille sur un ensemble potentiellement Pareto optimal P . Un nouvel ensemble N_P est généré en faisant l'union de tous les voisinages de toutes les solutions de P . Le processus est alors itéré sur P' qui est formé des solutions non-dominées de N_P . La méthode complète est décrite dans l'algorithme 3.2.

L'un des principaux désavantages de cette méthode est qu'elle devient vite coûteuse en temps de calcul si la cardinalité des approximations ou la taille des voisinages est importante. Pour limiter le temps de calcul, on divise l'exploration des voisinages selon les trois phases suivantes :

Phase d'intensification : dans un premier temps, pour chaque solution de l'archive, on ne considère dans son voisinage que les solutions qui la dominent ; ce qui revient à faire une vraie recherche locale. De plus, on ne garde qu'une seule solution à la fois. Comme il n'est pas possible de différencier toutes les solutions non-dominées, le

Algorithme 3.2 Recherche guidée par l'archive standard

```

 $P_0 \leftarrow \text{g\u00e9n\u00e9rer\_ensemble}()$ 
 $\text{continuer} \leftarrow \text{VRAI}$ 
 $t \leftarrow 0$ 
tant que  $\text{continuer} = \text{VRAI}$  faire
   $P_{t+1} \leftarrow \emptyset$ 
  pour tout  $s \in P_t$  faire
     $P_{t+1} \leftarrow P_{t+1} \setminus \{s' \in P_{t+1} \mid \exists s'' \in N(s), s'' \preceq s'\} \cup \{s' \in N(s) \mid \forall s'' \in P_{t+1}, s'' \not\preceq s'\}$ 
  fin pour
  si  $P_t \neq P_{t+1}$  alors
     $t \leftarrow t + 1$ 
  sinon
     $\text{continuer} \leftarrow \text{FAUX}$ 
  fin si
fin tant que
retourner  $P_t$ 

```

choix du voisin s\u00e9lectionn\u00e9 se fait al\u00e9atoirement et l'exploration du voisinage s'arr\u00eate d\u00e8s qu'un voisin qui domine la solution courante est trouv\u00e9. L'avantage de proc\u00e9der de cette mani\u00e8re est que la recherche est plus rapide et que l'on n'augmente pas la taille de l'archive qui peut m\u00eame diminuer. Cette phase se termine lorsqu'il n'est plus possible d'am\u00e9liorer l'archive, c'est-\u00e0-dire lorsqu'un optimum local est atteint pour chaque solution. On passe alors \u00e0 la phase de diversification.

Phase de diversification : pendant cette phase, on cherche \u00e0 \u00e9chapper au minimum local atteint lors de la phase pr\u00e9c\u00e9dente en faisant une recherche exhaustive du voisinage. Durant cette recherche, on prend en compte les solutions non domin\u00e9es non dominantes par rapport \u00e0 celles de l'archive. Le but est d'identifier des solutions qui ne sont pas locales d'un point de vue Pareto et qui pourront \u00eatre exploitées lors de la prochaine phase d'intensification. Comme le co\u00fbt en temps de calcul engendr\u00e9 par cette phase peut \u00eatre important, le nombre d'it\u00e9rations durant lesquelles elle est effectu\u00e9e ne doit pas \u00eatre trop important.

Phase de clusterisation : la taille de l'archive \u00e0 la fin de la phase de diversification peut avoir consid\u00e9rablement augment\u00e9. Ainsi, comme la proc\u00e9dure est it\u00e9r\u00e9e, il faut r\u00e9duire la taille de l'archive avant de relancer la phase d'intensification sous peine d'avoir des temps de calcul trop importants. D'autre part, cette m\u00e9thode est une approche a posteriori et il n'est pas envisageable d'offrir au d\u00e9cideur un trop grand nombre de solutions non-domin\u00e9es [175, 43]. Pour r\u00e9duire la taille de l'archive en gardant une information int\u00e9ressante, un algorithme de clustering est appliqu\u00e9 sur l'archive. Cependant, pour \u00e9viter de s\u00e9lectionner les solutions obtenues \u00e0 la fin de la phase d'intensification et que l'on sait \u00eatre des optima locaux, ces solutions ne sont pas prises en compte lors de la clusterisation. L'algorithme de clustering utilis\u00e9 est *average linkage method* [109] qui est connu pour bien fonctionner sur des donn\u00e9es comme les ensembles de solutions non-domin\u00e9es [175].

Cette strat\u00e9gie est r\u00e9sum\u00e9e dans l'algorithme 3.3.

Algorithme 3.3 Hybridation guidée par l'archive

```

{ENTREE : Une archive P}
P0 ← P
P0r ← P0
t ← 0
continuer ← VRAI
tant que continuer = VRAI faire
  clusteriser(Pt)
  Pt+1 ← Pt
  intensification(Pt+1) {cf. Algorithme 3.4}
  diversification(Pt+1) {cf. Algorithme 3.5}
  Pt+1r ← Ptr ∪ Pt+1
  si Ptr = Pt+1r alors
    continuer ← FAUX
  sinon
    Pt+1 ← Pt+1 \ Pt
    t ← t + 1
  fin si
fin tant que

```

3.4.2 Recherche guidée par restriction de l'espace

Dans cette méthode, pour chaque solution de l'ensemble potentiellement Pareto optimal, une recherche Tabou est appliquée. Plutôt que d'utiliser des techniques comme des fonctions agrégatives, la recherche utilise directement la notion de dominance. Le voisinage associé à une solution s est divisé dans les trois ensembles suivants :

- $N_1 = \{s' \in N(s) | s' \preceq s\}$
- $N_2 = \{s' \in N(s) | (s' \not\preceq s) \wedge (s \not\preceq s')\}$
- $N_3 = \{s' \in N(s) | s \preceq s'\}$

La solution courante suivante lors d'une itération est $s \in N_1$ tel qu'il n'existe pas de solution s' de N_1 qui domine s . Si l'ensemble N_1 est vide, l'ensemble N_2 et l'ensemble N_3 sont considérés dans cet ordre.

Le résultat de la recherche Tabou n'est pas une solution unique mais l'ensemble des solutions courantes non-dominées trouvées durant la recherche. Ces dernières sont stockées dans une archive.

Tel quel, le mécanisme de la recherche Tabou est peu efficace car il est peu probable que N_2 soit vide lors d'une itération et ainsi la recherche pourrait toujours s'échapper. D'autre part, il est nécessaire de guider la recherche pour que la solution générée soit bien une solution qui améliore s et non pas une autre solution de l'ensemble potentiellement Pareto optimal pour laquelle une autre recherche Tabou aurait été effectuée. Ainsi, chaque recherche Tabou se focalise sur une zone limitée de l'espace de recherche. La méthode pour restreindre la recherche dépend de l'implémentation. Un exemple de restriction est fourni dans le chapitre 5.

Algorithme 3.4 intensification($P : \text{ARCHIVE}$)

```

continuer  $\leftarrow$  VRAI
tant que continuer = VRAI faire
   $P_i \leftarrow \emptyset$ 
  pour tout  $s \in P$  faire
    si  $\exists s' \in N(s) | s' \preceq s$  alors
       $P_i \leftarrow P_i \cup \{s'\}$ 
    fin si
    si  $P_i = \emptyset$  alors
      continuer  $\leftarrow$  FAUX
    sinon
       $P \leftarrow P \cup P_i$ 
    fin si
  fin pour
fin tant que

```

Algorithme 3.5 diversification($P : \text{ARCHIVE}$)

```

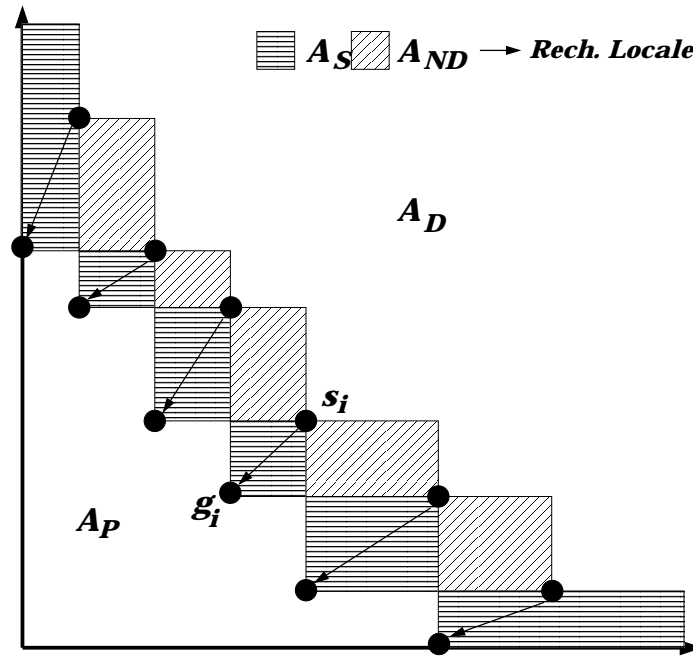
pour  $i = 1, \dots, K$  faire
   $P_i \leftarrow P$ 
  pour tout  $s \in P$  faire
    si  $\exists s' \in N(s) | s' \preceq s$  alors
       $P_i \leftarrow P_i \setminus \{s \in P_i | \exists s' \in N(s), s' \preceq s\} \cup \{s \in N(s) | \neg \exists s' \in P_i, s' \preceq s\}$ 
    fin si
   $P \leftarrow P_i$ 
fin pour
fin pour

```

D'autre part, comme les recherches sont toutes indépendantes les unes des autres, elles peuvent être exécutées en parallèle pour accélérer la recherche ou pour augmenter le nombre de points de départ. Cependant, il est clair que, puisque chaque recherche Tabou est restreinte à une zone de l'espace des objectifs, cette stratégie ne peut pas améliorer la diversité de l'espace potentiellement Pareto optimal mais uniquement la convergence vers la frontière Pareto.

3.4.3 Target Aiming Pareto Search

Dans la troisième politique que nous avons proposée, Target Aiming Pareto Search (TAPaS), le guidage des recherches est fait via un processus itératif basé sur une coopération supervisée et la définition de buts pertinents. Une recherche l_i est appliquée à chaque solution s_i d'un ensemble potentiellement Pareto optimal P . Une fonction objectif spécifique o_i est définie pour chaque recherche l_i . Cette fonction o_i prend en compte le fait que plusieurs recherches sont effectuées simultanément et que deux recherches ne doivent pas s'intéresser à la même zone de l'espace des objectifs et que la zone de l'espace des objectifs qui domine P doit être explorée dans son ensemble pour assurer la convergence globale de

FIG. 3.16 – Partition de l'espace objectif \mathcal{Y} .

Algorithme 3.6 but(s : solution, P : Pareto set)

 $\{M \text{ est le nombre d'objectifs}\}$
pour tout objectif m **faire**
 $f_m(g_p) \leftarrow \arg \min_{\{f_m(s') | (s' \in P) \wedge (f_m(s') < f_m(s))\}} (f_m(s') - f_m(s))$
fin pour

 retourner g_p

P vers l'ensemble Pareto optimal. La définition des fonctions objectif o_i est basée sur une partition de l'ensemble des objectifs \mathcal{Y} par rapport à P (voir figure 3.16) :

- $A_d = \{s \in O | \exists s' \in P, s' \preceq s\}$
- $A_{nd} = \{s \in O | \forall s' \in P, (s' \not\preceq s) \wedge (s \not\preceq s')\}$
- $A_s = \{s \in O | \exists ! s' \in P, s \preceq s'\}$
- $A_p = \{s \in O | \exists s_1, s_2 \in P, (s \preceq s_1) \wedge (s \preceq s_2)\}$

Chaque solution $s_i \in P$ est donc associée à une partie A_s^i de A_s . Si l_i est capable de générer une solution réalisable dans A_s^i , alors l'approximation est améliorée du point de vue de la convergence sans un appauvrissement de l'ensemble du point de vue de la diversité.

Pour guider la recherche, un but g_i est donné à chaque recherche l_i . g_i est défini comme le point de A_s^i qui domine tous les autres points de A_s^i . g_i ne correspond pas forcément à une solution réalisable. Dans le cas où certaines coordonnées de g_i ne peuvent pas être définies (comme par exemple pour les extrémités de P), il faut alors utiliser une borne inférieure pour la coordonnée manquante. L'algorithme 3.6 calcule les coordonnées du point g_i . La fonction objectif o_i visant à minimiser la distance avec le point g_i est alors définie de la

Algorithme 3.7 recherche_locale(s : solution, g_p : point, L : recherche locale)

```

{M est le nombre d'objectifs}
{N(s) est le voisinage de s}
L.s ← s
L.a ← {s}
L.o ← min(∑j=1M |fj(s) - fj(gp)|r)1/r
tant que (L.s ≠ gp) ∧ ¬ L.critère_d'arrêt faire
  sinter ← {s* ∈ N(L.s) | ∀s' ∈ N(L.s), L.o(s*) ≤ L.o(s')}
  L.s ← sinter
  L.a ← L.a \ {sa ∈ L.a | L.s < sa}
  si ∀sa ∈ L.a, sa ≠ L.s alors
    L.a ← L.a ∪ {L.s}
  fin si
fin tant que
retourner L.a

```

manière suivante :

$$\min\left(\sum_{j=1}^{j=M} \lambda_j |f_j(s) - f_j(g_i)|^r\right)^{\frac{1}{r}} \quad (3.1)$$

Il peut être nécessaire de normaliser les valeurs des objectifs s'ils ne sont pas du même ordre de grandeur.

La boucle générale d'une recherche l_i est décrite dans l'algorithme 3.7. Lorsque l_i atteint g_i ou trouve une solution qui domine g_i , la recherche s'arrête et renvoie une archive a_i qui contient les solutions courantes non-dominées trouvées lors de la recherche.

Lorsque toutes les recherches l_i sont terminées, un nouvel ensemble potentiellement Pareto optimal P' est formé par l'union Pareto des archives a_i . Comme il est clair que P' peut lui même être amélioré par une nouvelle application de la recherche, le processus complet est itéré jusqu'à ce que P' ne diffère plus de P . A chaque itération, un algorithme de clustering peut être appliqué à P de telle sorte que les solutions de départ ne soient pas trop proches les unes des autres et que les points g_i aient un sens. De plus, l'application de l'algorithme de clustering diminue le nombre d'exécutions nécessaires à chaque génération et donc le temps de calcul nécessaire. L'algorithme 3.8 résume la boucle principale de TAPaS.

Pour accélérer la recherche, il est possible de lancer les différentes recherches en parallèle. En effet, TAPaS est une stratégie coopérative car les points g_i et les fonctions objectif des recherches sont définis de telle sorte que deux recherches n'effectuent pas le même travail et que tout l'espace des objectifs soit exploré au maximum. Cependant, une fois les fonctions objectif définies, les recherches sont indépendantes les unes des autres et peuvent donc être effectuées en parallèle. Une autre manière d'économiser du temps de calcul est d'éviter de faire plusieurs fois une même recherche inutile. En effet, si lors d'une itération les solutions s_{i-1} , s_i et s_{i+1} ne sont pas améliorées, il n'est pas nécessaire de lancer lors de l'itération suivante une nouvelle recherche sur s_i . Dans ce cas, on part de la même solution de départ pour essayer d'atteindre le même but ; or, cette recherche s'étant déjà révélée

Algorithme 3.8 TAPaS(P : Ensemble potentiellement Pareto optimal, L : Recherche locale)

```

 $t \leftarrow 0$ 
 $P_t \leftarrow P$ 
continuer  $\leftarrow$  VRAI
tant que continuer faire
   $P_{t+1} \leftarrow P_t$ 
  cluster( $P_t$ )
  pour tout  $s \in P_t$  faire
     $g \leftarrow \text{but}(s, P_t)$ 
     $a \leftarrow \text{recherche\_locale}(s, g_p, L)$ 
     $P_{t+1} \leftarrow P_{t+1} \cup a$ 
  fin pour
  si  $P_t \neq P_{t+1}$  alors
     $t \leftarrow t + 1$ 
  sinon
    continuer  $\leftarrow$  FAUX
  fin si
fin tant que
retourner  $P_t$ 

```

infructueuse à l'itération précédente, il n'est pas utile de la recommencer. Eviter ces calculs inutiles est facile à mettre en place au travers d'une liste taboue simple. En effet, une recherche l_i est définie par sa solution de départ s_i et son but g_i . Il suffit donc de former, lors de la première itération, l'ensemble T_0 de tous les couples (s_i, g_i) . Puis lors de l'itération j , la recherche définie par le couple (s_i, g_i) est effectuée si et seulement si $(s_i, g_i) \notin T_{j-1}$; d'autre part, on a $T_j \leftarrow T_j \cup \{(s_i, g_i)\}$ que la recherche soit effectuée ou non. En effet, si la recherche n'est pas réalisée, c'est qu'elle l'a été lors d'une phase précédente et il faut donc faire passer l'information à l'itération suivante.

TAPaS fonctionne avec n'importe quelle méthode qui accepte une solution de départ et qui cherche ensuite à l'améliorer en passant de voisin en voisin. TAPaS a particulièrement été conçu dans l'optique d'appliquer des recherches Tabou. En effet, lorsque ces dernières rencontrent une solution près du but qui est un optimum local, la recherche ne s'arrête pas comme le ferait une simple recherche locale mais oscille autour de cette solution. De cette manière, une certaine amélioration de la diversité est possible, notamment au niveau des extrémités des ensembles potentiellement Pareto optimaux.

3.5 Conclusion

Dans ce chapitre, nous avons exposé la stratégie de coopération générale utilisée. Il s'agit d'une stratégie en deux phases où un algorithme génétique multi-objectif génère une approximation la plus diversifiée possible qui est ensuite améliorée du point de vue de la convergence par des stratégies à base de voisinage. Pour les algorithmes génétiques, nous

avons proposé un nouveau mécanisme, appelé la diversification élitiste, pour améliorer la diversification des ensembles identifiés. Ce mécanisme a été inclus dans l'algorithme génétique multi-objectif classique NSGA II. Nous avons aussi étudié l'utilisation du parallélisme au travers de l'utilisation de modèles en îles. Au niveau des stratégies à base de voisinage, nous avons proposé trois stratégies. Ces différentes stratégies ont été implémentées dans le cadre de l'étude du problème de tournées de véhicules avec équilibrage des tournées. La première stratégie utilise une archive et la dominance Pareto pour guider les recherches. Dans la seconde stratégie, on limite l'espace de recherche alloué à chaque recherche. Enfin, la troisième stratégie, Target Aiming Pareto Search, utilise la définition de buts pertinents à partir d'un découpage de l'espace des objectifs en fonction de l'approximation courante. La stratégie de coopération en deux phases est appliquée sur les deux problèmes étudiés. Pour les deux problèmes, la première correspond à l'application d'un algorithme génétique. Pour le problème de la tournée couvrante bi-objectif, on utilise un algorithme de séparations et coupes lors de la seconde phase, tandis que pour le problème d'élaboration de tournées de véhicules, il s'agit de méthodes à base de voisinage employées dans le cadre des stratégies définies dans ce chapitre.

Chapitre 4

Modélisation et résolution du problème de la tournée couvrante bi-objectif

Ce chapitre présente la modélisation et la résolution d'un problème de tournées multi-objectif : le problème de la tournée couvrante bi-objectif. Il s'agit d'une généralisation du problème de la tournée couvrante présenté dans le chapitre 2. Dans la version bi-objectif, la contrainte de couverture est remplacée par un objectif. Nous proposons aussi une modélisation du problème sous la forme d'un programme linéaire en nombre entier bi-objectif.

Pour ce problème, nous proposons trois approches :

- *Une méthode exacte : elle consiste en une méthode ϵ -contrainte utilisant un algorithme de séparations et coupes pour le problème de la tournée couvrante. Cette approche est capable de générer l'ensemble Pareto optimal sans faire de calcul inutile.*
- *Une méta-heuristique : c'est une version steady-state de NSGA II. Elle utilise une procédure avancée pour la création de la population de départ et plusieurs opérateurs génétiques dédiés au problème de la tournée couvrante ont été définis.*
- *Une méthode coopérative : nous faisons coopérer l'algorithme génétique et l'algorithme de séparations et coupes. A partir des approximations du premier, on construit des problèmes de petite taille qui sont résolus par le second.*

Une étude expérimentale nous a permis de mettre en évidence la qualité des résultats obtenus par l'algorithme génétique. Les résultats montrent aussi que la coopération est utile car elle permet une amélioration des approximations identifiées sans entraîner une augmentation pénalisante des temps de calcul. Nous avons aussi testé nos méthodes sur des données réelles.

4.1 Introduction

L'objectif de ce chapitre est de modéliser et résoudre le problème de la tournée couvrante bi-objectif. Il s'agit d'une généralisation du problème de la tournée couvrante [70]. L'idée d'étudier ce dernier problème d'un point de vue bi-objectif est venue d'une remarque de Boffey [18] selon laquelle plusieurs problèmes de tournées définis et étudiés d'un point de vue mono-objectif sont en fait des problèmes implicitement multi-objectif où les objectifs supplémentaires sont le plus souvent traités sous la forme de contraintes. Parmi les problèmes cités par Boffey, on trouve le problème de la tournée couvrante [70]. Dans la version bi-objectif, les contraintes liées à la couverture maximale sont retirées et remplacées par un objectif qui vise à minimiser la plus longue distance entre un sommet à couvrir et le sommet visité le plus proche.

Pour résoudre ce problème, nous avons exploré trois approches. La première est une résolution exacte exploitant un algorithme de séparations et coupes pour le problème de la tournée couvrante au sein d'une stratégie ϵ -contrainte. Cependant, comme il s'agit d'un problème NP-difficile, on ne peut espérer le résoudre avec cette méthode en un temps raisonnable pour des instances de grandes tailles. Ainsi, une approche heuristique basée sur un algorithme évolutionnaire multi-objectif a aussi été explorée. Pour cela, des opérateurs dédiés au problème de la tournée couvrante bi-objectif ont été définis. Enfin, une dernière approche combine l'algorithme génétique avec l'algorithme de séparations et coupes au sein d'une stratégie en deux phases.

Des expérimentations ont été menées sur des jeux de données générés. Nous avons évalué la qualité de l'algorithme génétique par rapport aux ensembles exacts trouvés par la méthode ϵ -contrainte. La comparaison a été faite du point de vue de la qualité des ensembles et des temps de calcul. L'intérêt de la coopération a aussi été validé. Enfin, nous avons résolu des instances correspondant à des données réelles.

Ce chapitre s'organise ainsi : le problème de la tournée couvrante bi-objectif est présenté dans la section 4.2. L'algorithme exact est introduit et discuté dans la section 4.3. La section 4.4 porte sur la construction de l'algorithme génétique multi-objectif. La stratégie de coopération est exposée dans la section 4.5. Les résultats expérimentaux sont reportés et discutés dans la section 4.6. Enfin, des conclusions sont données en 4.7.

4.2 Présentation, modélisation et remarques

Cette section est organisée de la manière suivante : tout d'abord, une modélisation bi-objectif du problème est donnée. Ensuite, nous présentons une heuristique pour le problème de la tournée couvrante utilisée dans nos méthodes. Puis quelques remarques générales sur le problème sont faites. Ces remarques portent notamment sur les couvertures possibles et leur identification.

4.2.1 Objectifs

Le problème de la tournée couvrante bi-objectif correspond au problème de la tournée couvrante où les contraintes qui imposent que tous les sommets de W se trouvent à moins d'une distance donnée c d'un sommet visité sont retirées et remplacées par un objectif. Toutes les autres contraintes du problème restent inchangées. Les objectifs du problème de la tournée couvrante bi-objectif traitent donc de la minimisation de :

1. la longueur de la tournée,
2. la couverture.

La couverture d'une solution est définie comme la plus grande distance entre un sommet $v_l \in W$ et le sommet visité $v_k \in V$ le plus proche de v_l .

4.2.2 Programme linéaire en nombre entier bi-objectif

Nous proposons la formulation du problème sous la forme d'un programme linéaire en nombre entier bi-objectif basé sur le programme linéaire en nombre entier de Gendreau et al. [70] énoncé dans le chapitre 2. Le formalisme associé au programme est le suivant. Le graphe $G = (V \cup W, E)$ est défini comme pour le PTC. Il en est de même pour l'ensemble $T \subseteq V$ et la matrice des distances C . Le but est toujours de trouver un cycle Hamiltonien sur un sous-ensemble $V' \subseteq V$, tel que $T \subseteq V'$, de longueur minimale, mais il n'y a plus de constante c qui impose une couverture maximale.

Un vecteur $d \in \mathbb{R}^{|W|}$ est introduit. Chaque composant d_l de d donne la plus petite distance nécessaire afin de couvrir le sommet $v_l \in W$. On définit également la matrice $S = (s_{lk})$ à valeur dans $[1, \dots, n]$ de dimension $|W| \times |V|$. s_{lk} correspond à l'indice du $k^{\text{ème}}$ sommet de V le plus proche de $v_l \in W$. La fonction σ est ajoutée et vaut :

$$\begin{aligned} \sigma : W \times V &\rightarrow [1, \dots, n] \\ (v_l, v_k) &\mapsto k' \text{ tel que } s_{lk'} = k \end{aligned}$$

A chaque $v_k \in V$, on associe une variable binaire y_k égale à 1 si et seulement si le sommet v_k appartient à la tournée. Dans certaines contraintes, une *grande valeur* (HV) a dû être mise en place pour être certain d'obtenir la valeur souhaitée. Le programme linéaire en nombre entier bi-objectif est alors le suivant :

$$\begin{cases} \min \sum_{i < j} c_{ij} x_{ij} \\ \min c_{max} \end{cases} \quad (4.1)$$

sujet à :

$$c_{max} \geq d_l \quad (v_l \in W), \quad (4.2)$$

$$d_l \geq c_{lk} \times y_k - HV \times \left(\sum_{i=0}^{\sigma(v_l, v_k)-1} y_{s_{li}} \right) \quad (v_l \in W, v_k \in V), \quad (4.3)$$

$$\sum_{i < k} x_{ik} + \sum_{j > k} x_{kj} = 2y_k \quad (v_k \in V), \quad (4.4)$$

$$\sum_{v_i \in S, v_j \in V \setminus S \text{ ou } v_j \in S, v_i \in V \setminus S} x_{ij} \geq 2y_t \quad (S \subset V, 2 \leq |S| \leq n-2, T \setminus S \neq \emptyset, v_t \in S), \quad (4.5)$$

$$x_{ij} \in \{0, 1\} \quad (1 \leq i < j \leq n), \quad (4.6)$$

$$y_k = 1 \quad (v_k \in T), \quad (4.7)$$

$$y_k \in \{0, 1\} \quad (v_k \in V \setminus T). \quad (4.8)$$

Les deux objectifs dans 4.1 correspondent respectivement à la minimisation de la longueur de la tournée et à la minimisation de la couverture. Les contraintes 4.2 assurent que c_{max} est bien le maximum des distances minimales entre un sommet de W et un sommet visité. Les contraintes 4.3 donnent une borne inférieure pour d_l . En effet, le terme représentant la somme partielle est plus grand ou égal à 1 si et seulement si, pour un sommet plus proche v_k , on a déjà $y_k = 1$. Les autres contraintes ont la même signification que dans le programme linéaire en nombre entier pour le problème de la tournée couvrante.

4.2.3 Contraintes valides supplémentaires

Dans le cadre du problème de la tournée couvrante bi-objectif, des contraintes supplémentaires peuvent être ajoutées. La première est très fortement liée à la modélisation choisie. En effet, les variables d_l correspondent au minimum de la couverture associée au sommet v_l de W . Or, dans l'ensemble des contraintes rencontrées, aucune borne inférieure indiquant clairement la nature de ce minimum n'a été introduite. Pour forcer la variable d_l à avoir tout de suite la bonne valeur, il est possible d'ajouter la contrainte suivante :

$$d_l \leq c_{lk} + HV \times (1 - y_k) \quad (v_l \in W, v_k \in V) \quad (4.9)$$

Cette formule assure bien que d_l est une distance minimale. Le "HV" de cette formule permet de ne pas prendre en compte les sommets v_k pour lesquels $y_k = 0$.

Les contraintes suivantes sont aussi valides :

$$c_{max} \geq m_l \quad (v_l \in W) \quad (4.10)$$

où les m_l représentent les minima de la distance avec un point de V quelle que soit la tournée.

4.2.4 Une heuristique pour le problème de la tournée couvrante

Nous allons maintenant décrire une heuristique pour le problème de la tournée couvrante mono-objectif utilisée dans l'algorithme génétique multi-objectif. L'heuristique est celle développée par Gendreau et al. [70]. Elle combine l'heuristique GENIUS pour le PVC [68] avec l'heuristique PRIMAL1 *set covering* de Balas et Ho [9] pour le problème de recouvrement maximal. L'heuristique a été légèrement modifiée pour limiter les appels à GENIUS.

L'heuristique PRIMAL1 ajoute des sommets v dans la solution selon un critère glouton qui cherche à minimiser une fonction $f(c_v, b_v)$ où, à chaque étape, c_v est le coût d'inclusion du sommet v dans la solution et b_v est le nombre de sommets de W couverts par v . Les trois fonctions suivantes suggérées par Balas et Ho sont utilisées :

1. $f(c_v, b_v) = \frac{c_v}{\log_2 b_v}$
2. $f(c_v, b_v) = \frac{c_v}{b_v}$
3. $f(c_v, b_c) = c_v$

Dans un premier temps, PRIMAL1 applique le critère 1 de manière gloutonne jusqu'à ce que tous les sommets de W soient couverts. Puis, les sommets qui couvrent un sommet de W couvert par plus d'un sommet sont retirés. Ensuite, la solution est complétée en utilisant le critère 2 et, après la phase de complétion, les sommets qui ne sont pas les seuls à couvrir un même sommet de W sont retirés. Le processus est alors itéré avec le troisième critère. Une seconde solution est construite en utilisant les critères dans l'ordre 1, 3, 2. La meilleure des deux solutions est retenue.

L'heuristique suivante est aussi exécutée deux fois avec les deux séquences de critères :

ETAPE 1 Initialisation. $H \leftarrow T$, $\bar{z} = +\infty$. Le critère courant est 1.

ETAPE 2 Terminaison. Si au moins un sommet de W n'est pas couvert par un sommet de H , alors aller à l'ETAPE 3, sinon un cycle Hamiltonien est construit sur les sommets de H en utilisant GENIUS. Soit z la longueur du cycle Hamiltonien obtenu. Si $z \leq \bar{z}$ alors $\bar{z} \leftarrow z$ et $\bar{H} \leftarrow H$. Si le critère courant est le dernier, l'heuristique s'arrête et la solution est donnée par la tournée sur \bar{H} de longueur \bar{z} . Sinon, il faut retirer les sommets de H associés aux sommets de W qui ne sont pas couverts par un unique sommet de H et prendre en compte le critère suivant.

ETAPE 3 Sélection du sommet. Pour chaque sommet $v \in V \setminus H$, on calcule les coefficients c_v et b_v par rapport à l'ensemble H courant et on détermine le meilleur sommet v^* selon le critère courant. c_v est donné par la valeur de l'arbre couvrant minimal du graphe partiel $(H \cup \{v\}, E')$ où $H \subseteq V$ est l'ensemble des sommets actuellement présents dans la solution et $E' = \{(v_i, v_j) \in E \mid v_i \in H \cup \{v\} \wedge v_j \in H \cup \{v\}\}$. L'arbre couvrant est construit par l'algorithme de Prim [123]. On met à jour H : $H \leftarrow H \cup \{v^*\}$. Aller à l'ETAPE 2.

4.2.5 Remarques

Nous allons maintenant faire quelques remarques sur le problème et les particularités qui ont été exploitées dans les méthodes pour le résoudre. Une solution est définie par deux éléments : i) l'ensemble des sommets visités, ii) la tournée sur cet ensemble de sommets. Il est à noter que la longueur de la tournée dépend de ces deux éléments mais que la couverture ne dépend que de l'ensemble des sommets visités et pas de l'ordre dans lequel ils sont visités.

D'autre part, il est possible de connaître l'ensemble des couvertures qui correspondent au moins à une solution. En effet, une couverture est définie par un couple $(v_k, v_l) \in V \times W$ et sa valeur est c_{kl} . Ainsi pour tout couple $(v_k, v_l) \in V \times W$, on associe une couverture candidate. Cependant, une couverture candidate n'est pas forcément une couverture réalisable, c'est-à-dire qu'il n'existe pas forcément une solution de couverture c_{kl} et dont la couverture est définie par le couple (v_k, v_l) . Pour tester la faisabilité d'une couverture donnée par

$v_k \in V$ et $v_l \in W$, on utilise le critère suivant :

$$c_{lk} \text{ une couverture admissible} \Leftrightarrow \begin{array}{l} 1) \forall v_i \in T, v_i \neq v_k, \quad c_{il} \geq c_{kl} \\ 2) \forall v_j \in W, v_k \neq v_l, \quad \exists v_i \in V, \\ \quad \text{t.q. } c_{ij} \leq c_{kl} \leq c_{il} \end{array} \quad (4.11)$$

La démonstration de ce critère est immédiate. En effet, dans le sens \Rightarrow , il s'agit de la définition d'une couverture, et dans l'autre sens, si on suit les règles, on construit une solution qui a bien c_{kl} comme couverture.

Il apparaît donc que le problème de minimisation de la couverture n'est pas NP-difficile et qu'il est possible de le résoudre en un temps polynomial. En effet, la plus petite couverture est rencontrée lorsque la tournée se fait sur V . Il se peut néanmoins que certains sommets de la tournée soient inutiles. La pire couverture est atteinte lorsque seuls les sommets de T sont visités. Comme indiqué précédemment, il est possible de trouver toutes les autres valeurs de couverture réalisable en un temps polynomial. Il n'est cependant pas possible de trouver la longueur minimale associée à chaque couverture. Pour la plus mauvaise couverture, il faut résoudre un problème du voyageur de commerce sur T . Pour une valeur quelconque de la couverture que l'on sait réalisable, il faut résoudre un problème de la tournée couvrante. Le sous-espace de l'espace des objectifs où l'on peut trouver des solutions est illustré avec ses bornes dans la figure 4.1.

Cependant, lorsque la couverture est fixée à une valeur c , il est possible de réduire le problème, c'est-à-dire de mettre des sommets de V dans T et de supprimer des sommets de $V \setminus T$ et de W . Le premier cas est utile car il apparaît que plus la cardinalité de T est élevée plus le problème est simple à résoudre et le deuxième cas, surtout la suppression de sommets de V , permet de réduire la combinatoire du problème [70]. Tout d'abord, on crée l'ensemble T' qui est défini de la manière suivante :

$$T' = T \cup \{v_k \in V \mid \exists v_l \in W, c_{kl} \leq c \wedge \nexists v_{k'} \in V \setminus \{v_k\}, c_{k'l} \leq c\}$$

Il faut ensuite construire l'ensemble $W' \subseteq W$ qui correspond à W auquel on a retiré les sommets qui sont couverts par un sommet de T' :

$$W' = W \setminus \{v_l \in W \mid \exists v_k \in T', c_{kl} \leq c\}$$

Enfin, il est possible de supprimer les sommets de V qui sont trop loin de tous les sommets de W' . On a alors

$$V' = V \setminus \{v_k \in V \mid \nexists v_l \in W', c_{kl} \leq c\}$$

Les réductions suivantes sont proposées par Gendreau et al. [70] pour une couverture maximale c . Une matrice $\delta = (\delta_{lk})$ de dimension $|W'| \times |V'|$ à valeur dans $\{0, 1\}$ est définie. Pour chaque couple $v_l \in W'$ et $v_k \in V'$, δ_{lk} est égal à 1 si $c_{lk} \leq c$, 0 sinon. Les simplifications suivantes sont alors possibles : i) chaque sommet $v_l \in W'$ avec $\delta_{lk} = 1$ pour tout $v_k \in V'$ est supprimé ; ii) s'il existe plusieurs sommets de W' qui ont les mêmes valeurs δ_{lk} pour tout sommet $v_k \in V'$, un seul sommet est conservé ; iii) les sommets dominés sont retirés : un sommet $v_l \in W'$ domine un sommet $v_{l'} \in W'$ si $\delta_{lk} \leq \delta_{l'k}$ pour tout k ; (iv) si un sommet $v_k \in V'$ ne couvre aucun sommet à cause des réductions précédentes, il est inutile et peut être éliminé.

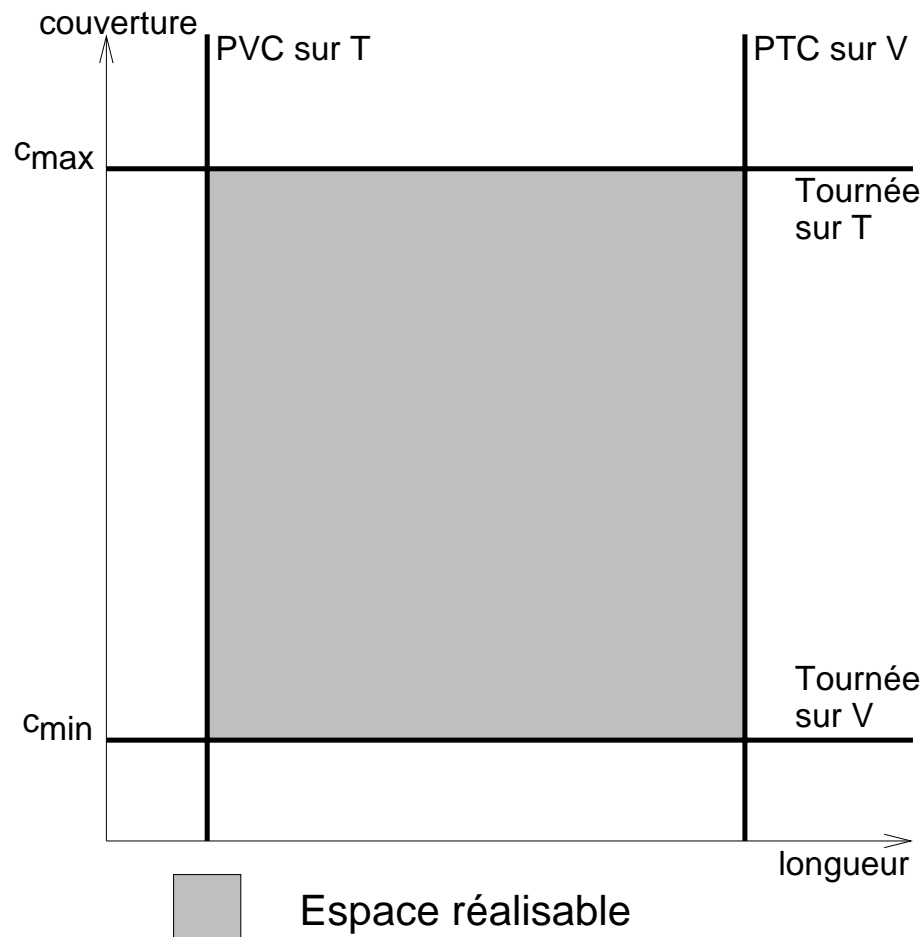


FIG. 4.1 – Bornes de l'espace des objectifs réalisable pour le problème de la tournée couvrante bi-objectif.

4.3 Résolution exacte par algorithme de séparations et coupes et méthode ϵ -contrainte

Peu de méthodes ont été proposées pour générer de manière exacte l'ensemble Pareto optimal. Par exemple, on peut citer la méthode deux phases de Ulungu et Teghem [162]. Cette méthode a notamment été utilisée pour résoudre un problème de flowshop bi-objectif [102]. Ici, une possibilité plus simple s'offre à nous. En effet, si on contraint l'objectif de couverture, on retrouve le problème de la tournée couvrante standard pour lequel il existe un algorithme de séparations et coupes [70]. De plus, comme il est possible de connaître toutes les couvertures qui correspondent à une solution admissible, on peut les utiliser pour définir une méthode ϵ -contrainte employant l'algorithme de séparations et coupes pour le problème mono-objectif pour générer l'ensemble Pareto optimal.

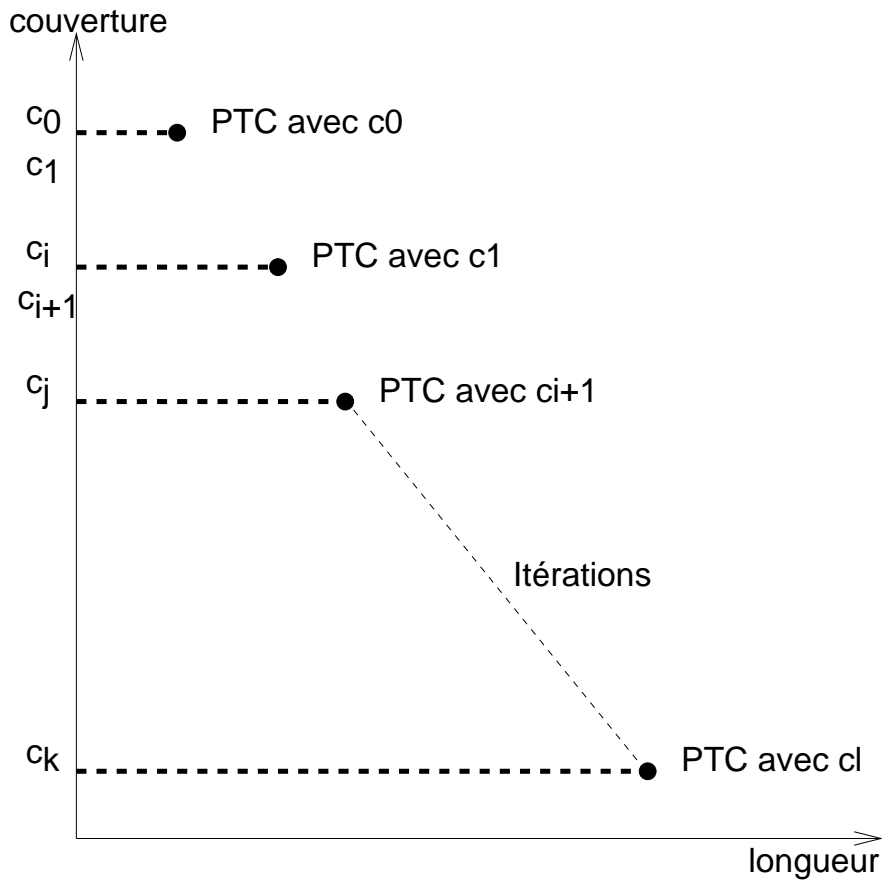


FIG. 4.2 – Résolution exacte par algorithme de séparations et coupes et méthode ϵ -contrainte.

4.3.1 Algorithme

L'algorithme de séparations et coupes n'a besoin que d'un seul paramètre qui est la couverture maximale autorisée. La méthode est définie de la manière suivante :

Étape 1 Calculer les couvertures admissibles à l'aide du critère 4.11 et les trier par valeur décroissante dans une liste $l = \{c_0, c_1, \dots, c_k\}$. On pose $c \leftarrow c_0$.

Étape 2 Résoudre le problème de la tournée couvrante avec c comme paramètre.

Étape 3 Calculer la couverture c de la solution s fournie par l'algorithme de séparations et coupes à l'étape précédente. On garde s comme une solution de l'ensemble Pareto optimal. Puis, on cherche $c_i = \max_{c_j \in l} (c_j < c)$. Si une telle couverture c_i existe alors $c \leftarrow c_i$ et on va à l'Étape 2, sinon l'algorithme s'arrête.

Cette méthode est illustrée dans la figure 4.2.

Proposition 1 *L'algorithme donné ci-dessus trouve l'ensemble Pareto optimal.*

Démonstration Soit s une solution de couverture c appartenant à l'ensemble Pareto optimal. Soit c' la plus grande des couvertures admissibles strictement inférieure à c . Soit s_i la solution de longueur l_i et de couverture c_i trouvée, si elle existe, par l'algorithme de séparations et coupes avec en paramètre c' . Nous allons montrer que : i) s_i est aussi une solution Pareto optimale ; ii) si $c_i \neq c'$ alors il n'existe pas une autre solution Pareto optimale s_j de couverture c_j tel que $c_i < c_j < c'$. En fait, il se peut que deux solutions aient une même longueur mais des couvertures différentes. L'algorithme de séparations et coupes trouve indifféremment l'une des deux solutions selon l'ordre dans lequel les variables sont fixées. Cependant, ce cas est rare et il peut être pris en compte en utilisant les contraintes du programme linéaire en nombre entier donné précédemment et en utilisant la fonction objectif suivante :

$$\min \sum_{i < j} c_{ij} x_{ij} + M c_{max}$$

avec M une constante suffisamment petite.

Nous allons démontrer i) par l'absurde. Supposons qu'il existe une solution s_j de longueur l_j et de couverture c_j tel que s_j domine s_i , c'est-à-dire tel que $c_j \leq c_i$ et $l_j \leq l_i$. Dans ce cas, les deux solutions respectent la contrainte sur la couverture qui doit être inférieure ou égale à c et l'algorithme de séparations et coupes aurait dû trouver s_j car elle possède une meilleure longueur. Donc, une telle solution s_j ne peut pas exister et s_i est donc une solution Pareto optimale.

Nous allons maintenant démontrer la proposition ii) qui dit qu'il n'existe pas de solution s_j qui soit Pareto optimale et dont la couverture c_j est telle que $c_i < c_j < c'$. Supposons qu'une solution Pareto optimale s_j de longueur l_j et de couverture c_j existe. Alors, étant donné que $c_j > c_i$, on a $l_j < l_i$ ou s_j serait dominée par s_i , ce qui n'est pas possible puisque s_j est une solution Pareto optimale. Or, comme les deux solutions respectent la contrainte sur la couverture et puisque $l_j < l_i$, l'algorithme de séparations et coupes aurait dû trouver s_j et non pas s_i . Il y a donc contradiction et une solution s_j Pareto optimale ayant une couverture c_j tel que $c_i < c_j < c'$ n'existe pas.

La première exécution de l'algorithme de séparations et coupes se fait avec le paramètre c_0 . Comme indiqué précédemment, la solution optimale de couverture c_0 est la tournée optimale sur T . Pour obtenir une meilleure couverture, il est obligatoire d'ajouter au moins un sommet à visiter et par conséquent, la longueur de la tournée sera forcément plus grande que la longueur de la tournée optimale sur T . Aussi, la solution qui visite uniquement les points de T de couverture c_0 est une solution Pareto optimale. D'après les points i) et ii), comme cette première solution est Pareto optimale, l'algorithme trouve bien toutes les solutions Pareto optimales.

4.3.2 Avantages

La méthode possède plusieurs avantages. D'abord, sauf pour le cas particulier cité précédemment, elle ne fait pas de calculs inutiles puisqu'à chaque itération une nouvelle solution Pareto optimale est trouvée et que l'on est capable de connaître la couverture minimale et de l'utiliser comme critère d'arrêt. Ensuite, en fixant la couverture, on revient au problème mono-objectif. Il est donc possible d'appliquer tout ce que l'on sait dans ce

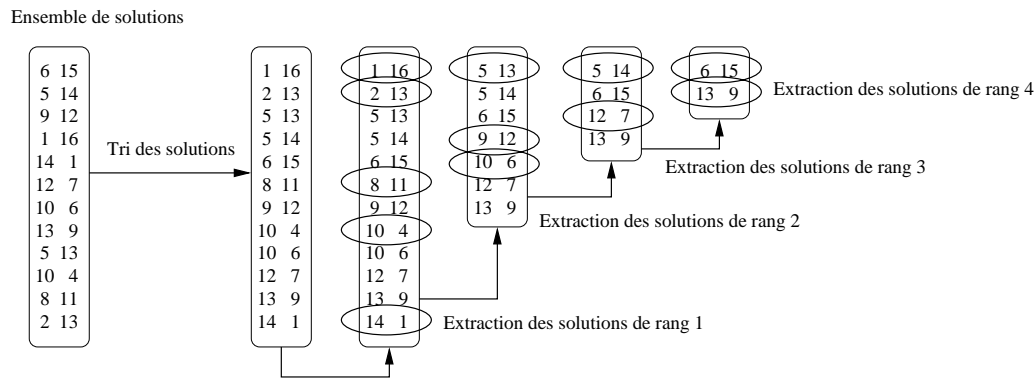


FIG. 4.3 – Exemple d’application de la méthode de ranking rapide.

cas et en particulier les règles de réduction données précédemment, ce qui permet d’espérer une accélération des temps d’exécution des algorithmes de séparations et coupes effectués.

4.4 Résolution par algorithme génétique

Comme le problème de la tournée couvrante est NP-difficile, on ne peut pas espérer le résoudre de manière exacte lorsque la cardinalité des ensembles augmente. Ceci est d’autant plus vrai pour la méthode bi-objectif décrite précédemment où plusieurs résolutions exactes sont effectuées. Il est donc nécessaire d’utiliser des méthodes heuristiques. Dans cette étude, un algorithme génétique multi-objectif est défini pour s’attaquer au problème de la tournée couvrante bi-objectif. Cette partie est organisée de la manière suivante : la structure générale de l’algorithme est présentée en 4.4.1, deux opérateurs de croisement sont proposés dans la section 4.4.2 et les opérateurs de mutation utilisés sont décrits dans la section 4.4.3.

4.4.1 Structure générale

Notre approche est basée sur une variante steady-state de NSGA II [44]. Les principaux mécanismes et caractéristiques de notre algorithme sont :

1. La fonction de ranking est celle de NSGA II. Dans le cas général, le tri des solutions dans les différentes profondeurs se fait en $O(n^2 \log n)$. Cependant, comme nous nous intéressons à un problème bi-objectif, il est possible d’attribuer les rangs des différentes solutions en $O(n \log n)$ en moyenne et en $O(n^2)$ dans le pire des cas en effectuant des tris sur les différents objectifs comme indiqué dans l’algorithme 4.1 qui est illustré dans la figure 4.3.
2. Une archive, contenant les solutions non-dominées, a été ajoutée à l’algorithme. Son but est double. Premièrement, elle permet de garder toutes les solutions potentiellement Pareto optimales trouvées durant la recherche. Mais elle intervient aussi dans le critère d’arrêt de l’algorithme génétique. En effet, si aucune nouvelle solution n’a été incorporée dans l’archive durant les M dernières générations, l’algorithme s’arrête.

Algorithme 4.1 Fonction de ranking rapide.

```

{Q : ensemble des solutions non-traitées}
{tri(A) : cette fonction trie les solutions de l'archive A de manière croissante selon le
premier objectif; si plusieurs solutions ont même valeur pour le premier objectif, elles
sont triées par rapport au second objectif de manière décroissante. Les solutions sont
notées  $s_i$  avec  $i = 1, \dots, |A|$ .}
tri(A)
Q ← A
r ← 1
tant que Q ≠ ∅ faire
  s ←  $s_i$  tel que  $s_i \in Q$  et  $\nexists s_j \in Q$  avec  $j < i$ 
  s.rang ← r
  meilleur ←  $f_2(s)$ 
  Q ← Q \ {s}
  pour i = 1, ..., |A| faire
    si ( $s_i \in Q$ ) ∧ ( $f_2(s_i) < meilleur$ ) alors
       $s_i$ .rang ← r
      meilleur ←  $f_2(s)$ 
    fin si
  fin pour
  r ← r + 1
fin tant que

```

3. La population initiale est construite de la manière suivante : tout d'abord, les différentes couvertures réalisables sont calculées à l'aide du critère 4.11. Ensuite, on sélectionne différentes couvertures que l'on donne en paramètre à l'heuristique décrite précédemment. L'intérêt d'utiliser l'heuristique est qu'elle permet de commencer la recherche avec des solutions de bonne qualité. D'autre part, si les couvertures sont bien choisies, il est possible d'obtenir des solutions bien réparties le long de l'espace réalisable et notamment une première approximation des extrémités de l'ensemble Pareto optimal. $N + 1$ solutions sont générées durant cette phase.
4. L'approche est steady-state. Une génération se déroule de la manière suivante : le rang et la distance de crowding sont calculés pour les $N + 1$ solutions de la population. Deux parents sont choisis parmi les N meilleures solutions à l'aide de la sélection par tournoi standard de NSGA II. Un enfant est alors généré par application des opérateurs génétiques. Les doublons sont interdits dans la population. Si l'enfant appartient déjà à la population, un nouvel enfant est généré à partir des mêmes parents. Le processus est itéré jusqu'à ce qu'un enfant non présent dans la population soit créé ou jusqu'à ce que 50 enfants soient générés sans succès. Dans le premier cas, l'enfant prend la place dans la population du plus mauvais individu, sinon la population ne change pas.

4.4.2 Opérateurs de croisement

Deux composants constituent une solution A de longueur l_A et de couverture $c_A : V_A$ l'ensemble des sommets visités par A et σ_A la séquence qui constitue la tournée sur V_A . La longueur de la solution dépend à la fois de V_A et σ_A , tandis que la couverture ne dépend que de V_A . D'autre part, les deux objectifs sont clairement opposés puisque, d'une manière générale, pour faire diminuer la couverture il faut visiter de plus en plus de sommets, ce qui a pour effet d'augmenter la distance. Il est à noter que cette affirmation peut ne pas être toujours vraie ; mais, si l'on parcourt l'ensemble Pareto optimal dans le sens des longueurs croissantes, et donc des couvertures décroissantes, on observe que le nombre de sommets visités a tendance à augmenter.

Par conséquent, il est délicat de définir un opérateur de croisement qui prend en compte les deux objectifs. Aussi, nous avons défini des opérateurs qui travaillent en deux phases. L'une des phases sélectionne les sommets à visiter, tandis que l'autre phase construit la tournée. Deux types d'opérateur sont alors possibles : un opérateur select-first route-second et un opérateur route-first select-second.

4.4.2.1 Opérateur select-first route-second (SRX)

Dans cet opérateur, on choisit d'abord les sommets qui seront visités, puis on construit une tournée sur les sommets choisis. Le but de la phase de sélection est de mettre le moins possible de sommets dans la solution. Ensuite, la tournée est construite par une heuristique pour le problème du voyageur de commerce.

Le but est de minimiser le nombre de sommets dans la solution. Pour cela, un sommet n'est ajouté que s'il a un effet sur la couverture. L'insertion d'un sommet ne peut avoir que deux conséquences sur la couverture : soit elle diminue, soit elle reste inchangée. Le premier cas intervient lorsque la couverture est donnée par un couple (v_k, v_l) et que l'on inclut un sommet v_i tel que $c_{il} < c_{kl}$. En prenant cette remarque en compte, la phase de sélection est définie de la manière suivante :

ETAPE 1 Initialisation. $H \leftarrow T$.

ETAPE 2 Identifier le couple (v_k, v_l) qui donne la couverture courante. Construire l'ensemble $H' \leftarrow \{v_i \in V \setminus H \mid c_{il} < c_{kl}\}$.

ETAPE 3 Si $H' = \emptyset$ alors aller à l'ETAPE 4, sinon choisir un sommet v_i de H' et le retirer de H' . On essaie d'inclure v_i dans H avec une probabilité p . Si v_i est inclus dans H , aller à l'ETAPE 2, sinon réitérer l'ETAPE 3.

ETAPE 4 Construire un ensemble U des sommets de H de façon que la couverture reste inchangée si on retire un sommet de U de H . Si U est vide, la phase s'arrête et l'ensemble des sommets visités par l'enfant est H . Sinon, on choisit $u \in U$ de sorte que la valeur de l'arbre couvrant minimal sur $H \setminus \{u\}$ soit minimale et on met à jour $H \leftarrow H \setminus \{u\}$ et $U \leftarrow U \setminus \{u\}$. L'ETAPE 4 est réitérée.

La probabilité p de l'ETAPE 3 est calculée de la même manière que dans l'opérateur de croisement *fusion* pour le problème de recouvrement minimal [16]. Soit A et B les deux parents et v le sommet que l'on souhaite inclure dans l'enfant, alors la probabilité p est calculée ainsi :

1. Si $v \in V_A$ et $v \in V_B$, alors $p = 1$.
2. Si $v \notin V_A$ et $v \notin V_B$, alors $p = 0$.
3. Si v n'apparaît que dans un seul parent, alors on pose $p' = \frac{r_B}{r_A+r_B}$ si $r_A \neq r_B$, sinon on pose $p' = \frac{d_A}{d_A+d_B}$. r_A (respectivement r_B) est la profondeur de dominance de A (respectivement B) et d_A (respectivement d_B) est la distance de crowding de A (respectivement B). Si $v \in V_A$, alors on a $p = p'$, sinon $p = 1 - p'$.

Il est possible de construire la tournée sur les sommets sélectionnés par le biais d'une méthode de résolution du PVC. Dans cette implémentation, nous utilisons l'heuristique GENIUS [68]. Comme GENIUS est une heuristique, elle ne sera pas toujours capable de trouver la tournée optimale. Cependant, notre hypothèse est la suivante : si V_1 et V_2 sont deux sous-ensembles de V et si le tour optimal sur V_1 est plus court que le tour optimal sur V_2 , alors, dans la plupart des cas, le tour généré par GENIUS sur V_1 sera de meilleure qualité que le tour généré par GENIUS sur V_2 . Ainsi, l'algorithme génétique est défini pour identifier les couvertures et leurs ensembles de sommets associés, qui sont de bons candidats pour correspondre à des solutions Pareto optimales. Ensuite, il est possible d'utiliser des méthodes dédiées au PVC sur les solutions générées, ou sur celles choisies par un décideur, pour améliorer leur longueur. Le travail de la méthode de post-optimisation sera alors simplifié car de bons ensembles candidats auront été identifiés et il apparaît, d'après les expérimentations de Gendreau et al. [70], que le nombre de sommets visités dans les solutions optimales est en général petit par rapport à $|V|$.

4.4.2.2 Opérateurs route-first select-second (RSX)

Le problème pour définir un opérateur de croisement qui travaille directement sur la tournée est que cette dernière peut être très différente entre deux solutions en termes de sommets visités. Il n'est donc pas intéressant d'appliquer directement sur les deux parents des opérateurs issus des recherches sur le PVC. Pour palier à ces difficultés, nous proposons un opérateur de croisement qui transforme les tournées des parents en deux tournées qui sont effectuées sur les mêmes nœuds et applique ensuite sur ces nouvelles tournées un opérateur classique du PVC, *Edge Recombination Crossover* (ERX) [167, 147].

Nous allons maintenant expliquer comment les tournées \tilde{t}_A et \tilde{t}_B , sur lesquelles est appliqué ERX, sont construites. Les deux parents sont appelés A et B et visitent respectivement les ensembles V_A et V_B par les tournées t_A et t_B . Dans un premier temps, on identifie les nœuds qui sont communs aux deux parents et on forme l'ensemble $\tilde{V} = V_A \cap V_B$. S'il n'y a pas au moins deux sommets en commun entre V_A et V_B , il n'est pas possible de continuer l'opérateur. On crée alors une tournée pour A et B sur \tilde{V} . La tournée \tilde{t}_A formée à partir de t_A sur \tilde{V} est construite de la manière suivante : les sommets de \tilde{V} sont visités dans le même ordre dans \tilde{t}_A que dans t_A . La valuation de l'arc entre deux sommets de \tilde{V} consécutifs dans \tilde{t}_A est égale à la somme des valuations des arêtes formant la chaîne entre les deux sommets dans V_A . La tournée \tilde{t}_B formée à partir de t_B est créée de la même manière.

Après l'application de ERX sur \tilde{t}_A et \tilde{t}_B , il est nécessaire d'appliquer une phase de réparation sur la tournée \tilde{t} de l'enfant obtenu. En effet, il est possible qu'une arête (v_i, v_j)

de \tilde{t} corresponde en fait à une chaîne dans la tournée initiale du parent dont elle est issue. Il faut alors inclure entre v_i et v_j les sommets constituant la chaîne dans le parent. L'ordre d'inclusion est celui du chemin ayant v_i pour origine et v_j pour extrémité. Lorsque tous les sommets devant être ajoutés l'ont été, on obtient alors la tournée t de l'enfant. Cet opérateur de croisement est illustré dans la figure 4.4.

La phase de sélection consiste en une recherche locale pour supprimer les sommets qui ne servent à rien dans le calcul de la couverture. La recherche est semblable à celle utilisée dans l'opérateur SRX, à part que le gain de retirer un sommet n'est pas évalué par la recherche d'un arbre couvrant minimal mais par le coût marginal puisqu'on dispose d'une tournée.

4.4.3 Opérateurs de mutation

Deux opérateurs de mutation sont utilisés. Le premier, dénoté M_1 , change l'état de chaque sommet $v \in V \setminus T$ avec une probabilité $\frac{1}{|V \setminus T|}$. Changer l'état d'un sommet signifie le retirer s'il fait partie des sommets sélectionnés, l'ajouter sinon. Lorsque le crossover SRX est utilisé, on applique l'opérateur M_1 avant la construction de la tournée par GENIUS.

La seconde mutation est une variante du Or-opt [112]. De un à trois sommets consécutifs de la tournée sont retirés puis réinclus dans la tournée de telle sorte que le coût d'inclusion soit minimal. L'ordre d'inclusion des sommets est le même que l'ordre dans lequel ils étaient visités dans la tournée originale. Cet opérateur n'est pas utilisé en combinaison avec l'opérateur SRX.

4.5 Coopération entre l'algorithme génétique et l'algorithme de séparations et coupes

L'algorithme évolutionnaire présenté ne sera pas capable de trouver l'ensemble Pareto optimal lorsque la taille des instances deviendra de plus en plus grande. Cependant, son temps de calcul ne deviendra pas trop important. A l'inverse, la méthode exacte deviendra trop longue. Nous avons donc cherché à faire coopérer les deux méthodes. Dans cette partie, l'algorithme de séparations et coupes est vu comme une boîte noire qui prend en entrée un ensemble de sommets et une couverture et renvoie une tournée valide du point de vue du PTC. La stratégie retenue est la suivante : on utilise l'information obtenue par l'algorithme génétique pour créer des problèmes de petite taille qui peuvent être traités rapidement par l'algorithme de séparations et coupes.

Deux procédures de création de problèmes ont été définies. La première dite de type I a pour but d'améliorer les solutions trouvées par l'algorithme génétique du point de vue de la longueur de la tournée sans modifier la couverture. La seconde méthode dite de type II construit des ensembles dans le but de trouver des solutions dont la couverture n'a pas été identifiée par l'algorithme génétique. Les expérimentations (voir section 4.6) montrent que les deux situations sont intéressantes et justifiées.

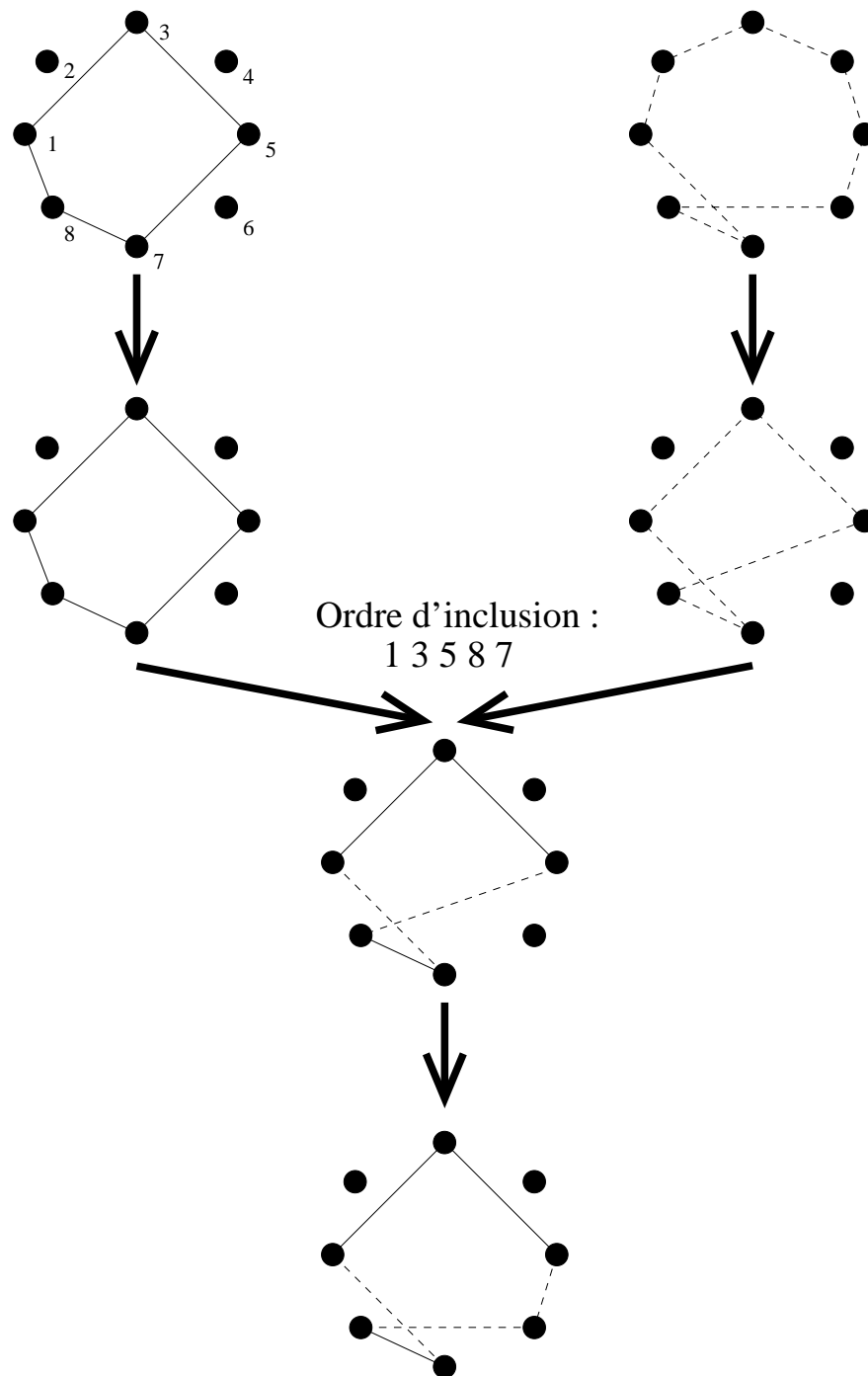


FIG. 4.4 – Exemple d'application de l'opérateur de croisement RSX.

4.5.1 Coopération de type I

On suppose que l'algorithme génétique fournit des ensembles qui ne comptent pas de sommets inutiles du point de vue de la couverture. L'idée est donc de voir s'il n'est pas possible de remplacer certains sommets de l'ensemble visité \tilde{V} par un ensemble de sommets $R \subseteq V \setminus \tilde{V}$ de telle sorte que cette modification laisse la couverture \tilde{c} donnée par le couple (v_t, v_c) inchangée. Un sommet $v_k \in \tilde{V}$ peut être remplacé par l'ensemble R si :

1. Aucun sous-ensemble de R ne peut remplacer v_k .
2. Aucun sommet de R n'est capable de fournir une meilleure couverture :

$$\forall v_i \in R, c_{tc} \leq c_{ic}$$

3. Pour tous les sommets de W qui peuvent être couverts par v_k , il doit exister un autre sommet de \tilde{V} ou un sommet de R qui peut remplacer v_k . Il faut pour cela que la condition suivante soit vraie pour tout $v_i \in W \setminus \{v_c\}$ tel que $\nexists v_j \in \tilde{V} \setminus \{v_k\}, c_{ji} \leq c_{kj}$:

$$\exists v_j \in (\tilde{V} \setminus \{v_k\}) \cup R, c_{jk} \leq \tilde{c}$$

Lorsque la taille de R augmente, le test devient de plus en plus coûteux. D'autre part, plus l'ensemble R contient de sommets, plus il a de chance de pouvoir remplacer un sommet de \tilde{V} . Or, si l'ensemble qu'on construit devient trop grand, alors l'algorithme de séparations et coupes sera trop coûteux en temps de calcul. La cardinalité des ensembles a donc été limitée à un ou deux éléments.

L'ensemble V_I fourni à l'algorithme de séparations et coupes est constitué de l'union de \tilde{V} et de tous les sous-ensembles de V de cardinalité 1 ou 2 qui peuvent remplacer un sommet de \tilde{V} . La méthode de construction de l'ensemble V_I est donnée dans l'algorithme 4.2.

4.5.2 Coopération de type II

Le premier type d'hybridation consiste à améliorer les solutions trouvées par l'algorithme génétique. Cependant, toutes les couvertures admissibles qui correspondent à une solution Pareto optimale peuvent ne pas avoir été identifiées. Les solutions qui manquent se trouvent forcément entre deux solutions de l'approximation et il est vraisemblable que toutes les solutions ne sont pas entre les deux mêmes solutions voisines de l'ensemble potentiellement Pareto optimal. Il est légitime de penser qu'en effectuant une recherche entre chaque couple de solutions voisines avec des ensembles adaptés, il sera possible de trouver de nouvelles solutions Pareto optimales.

Soit A et B deux solutions voisines dans l'ensemble de solutions générées par l'algorithme génétique. A (respectivement B) est une solution de couverture c_A (respectivement c_B) qui visite les sommets de l'ensemble V_A (respectivement V_B). On suppose qu'on a $c_A < c_B$. On va alors construire un ensemble de sommets V_{II} à partir de ces informations, puis exécuter l'algorithme de séparations et coupes avec en entrée V_{II} et la première couverture strictement inférieure à c_B .

Algorithme 4.2 Construction de l'ensemble V_I

```

 $V_1 \leftarrow \emptyset$ 
pour tout  $v_k \in \tilde{V} \setminus \{v_t\}$  faire
  pour tout  $v_i \in V \setminus \tilde{V}$  faire
    si  $\{v_i\}$  peut remplacer  $v_k$  alors
       $V_1 \leftarrow V_1 \cup \{v_j\}$ 
    fin si
  fin pour
fin pour
 $V_2 \leftarrow \emptyset$ 
pour tout  $v_k \in \tilde{V} \setminus \{v_t\}$  faire
  pour tout  $v_i \in V \setminus (\tilde{V} \cup V_1)$  faire
    pour tout  $v_j \in V \setminus (\tilde{V} \cup V_1)$  faire
      si  $\{v_i, v_j\}$  peut remplacer  $v_k$  alors
         $V_2 \leftarrow V_2 \cup \{v_j, v_j\}$ 
      fin si
    fin pour
  fin pour
fin pour
 $V_I \leftarrow \tilde{V} \cup V_1 \cup V_2$ 

```

Algorithme 4.3 Construction de l'ensemble V_{II}

```

 $V_{II} \leftarrow V_A \cup V_B$ 
pour tout  $c$  tel que  $c_A \leq c \leq c_B$  faire
  pour tout  $v_l \in W$  faire
     $V_{II} \leftarrow V_{II} \cup \{v_k \in V \setminus V_{II} | c_{kl} \leq c\}$ 
  fin pour
fin pour

```

L'ensemble V_{II} est construit de la manière suivante : tout d'abord, il contient V_B . En effet, il apparaît lors des tests que les solutions voisines dans l'ensemble Pareto optimal ont un nombre de sommets en commun non négligeable. De même, on inclut aussi V_A , ce qui assure que l'algorithme de séparations et coupes trouvera au pire A ou une solution de même couverture mais avec une meilleure longueur au cas où la tournée sur V_A n'est pas optimale. Pour compléter V_{II} , on procède de la manière suivante : pour chaque couverture faisable c , tel que $c_A \leq c \leq c_B$, on ajoute des sommets à V_{II} pour qu'il existe au moins un sous-ensemble de V_{II} de couverture c . L'algorithme 4.3 montre comment construire l'ensemble V_{II} .

4.6 Résultats expérimentaux

4.6.1 Problèmes tests utilisés

Données générées : pour générer les ensembles de sommets, $|V| + |W|$ points ont été choisis selon une distribution uniforme dans un carré de taille $[0, 100] \times [0, 100]$. Les ensembles T et V sont respectivement définis par les $|T|$ et $|V|$ premiers points et W est formé des points restants. La taille de V a été fixée à 50, 75, 100 et 120 ; la taille de T à 1, $\lfloor 0.10|V| \rfloor$, $\lfloor 0.20|V| \rfloor$; et la taille de W à $|V|, 2|V|, 3|V|$.

Données réelles : les données de ce problème correspondent au district de Suhum au Ghana décrit par Hodgson et al. [79]. Il s'agit de définir une route pour une unité de soins mobile. Il n'est cependant pas possible au vu des différents types de routes d'accéder à l'ensemble des villages. Le but est de définir une tournée depuis Suhum qui traverse une partie des villages accessibles par la route de telle sorte que les habitants des villages non desservis ne doivent pas marcher plus d'une certaine distance jusqu'à un village desservi. Ce cas peut donc être traité par la résolution d'un PTC.

Le réseau routier est modélisé sous forme d'un graphe composé de 148 sommets représenté dans la figure 4.5 et la modélisation utilisée dans la figure 4.6. Deux valuations sont définies sur les arcs. La première correspond à la distance que doit parcourir le véhicule pour se rendre d'un village à un autre, la seconde correspond à la distance de marche pour un villageois pour se rendre d'un village à un autre. La longueur de la tournée est donc calculée avec la première valuation et la couverture avec la seconde. Pour chacune des valuations, il existe deux matrices de distance : une pour la saison sèche, une pour la saison des pluies. Il n'existe pas forcément un lien direct entre deux villages. Ainsi, la matrice a été complétée en affectant comme valuations aux arcs manquants la valeur du plus court chemin entre les deux extrémités. Cela peut entraîner une dégénérescence de la solution. En effet, si dans une solution la tournée comprend l'arc (v_i, v_j) alors que les deux villages v_i et v_j ne sont pas directement connectés dans la réalité, alors l'unité visite également les villages qui se situent sur le plus court chemin entre v_i et v_j . Or, dans le codage de la solution, ces villages n'apparaissent pas. De ce fait, il existe plusieurs codages pour une même solution. De plus, il se peut que la valeur de la couverture soit faussée. D'autre part, tous les sommets ne sont pas accessibles par le véhicule. Le nombre de villages que l'on peut visiter est de 104 lors de la saison sèche et de 52 lors de la saison des pluies.

4.6.2 Protocole

L'algorithme génétique est codé en C et l'algorithme de séparations et coupes est implémenté en C avec CPLEX 8.1. Les tests ont été effectués sur un Pentium IV 2.67Ghz avec 512 Mo sous Debian Linux 3.0.

Pour les instances générées, pour chaque combinaison de tailles de V , T et W , cinq instances sont générées. L'algorithme génétique est exécuté dix fois sur chacune des instances ainsi que sur les deux cas correspondant au district de Suhum. L'algorithme génétique a

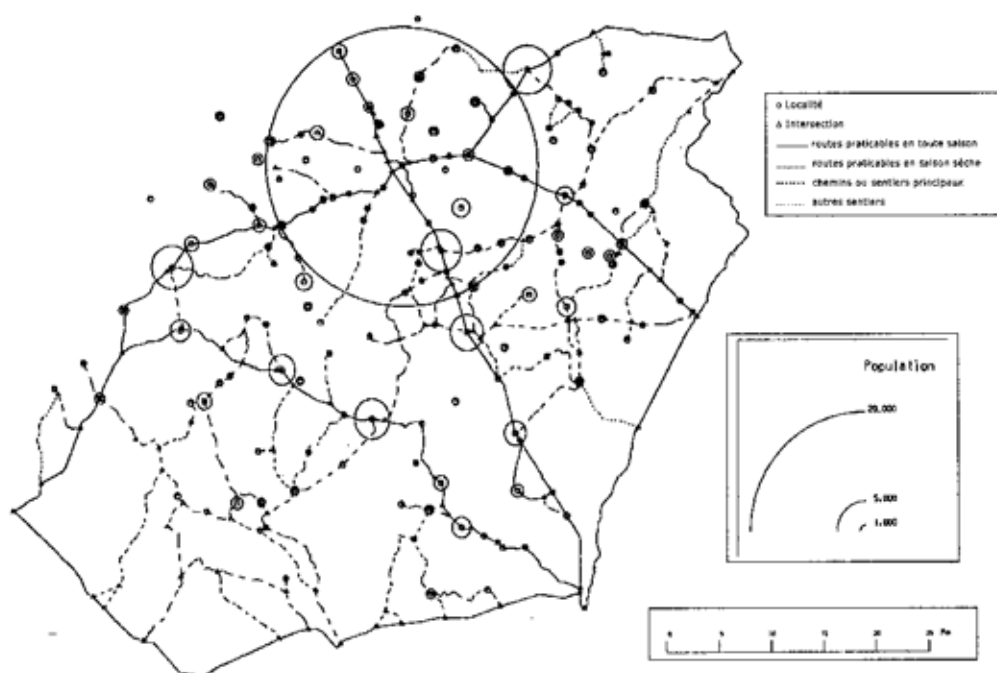


FIG. 4.5 – Routes et villages, District de Suhum, Région Est, Ghana.

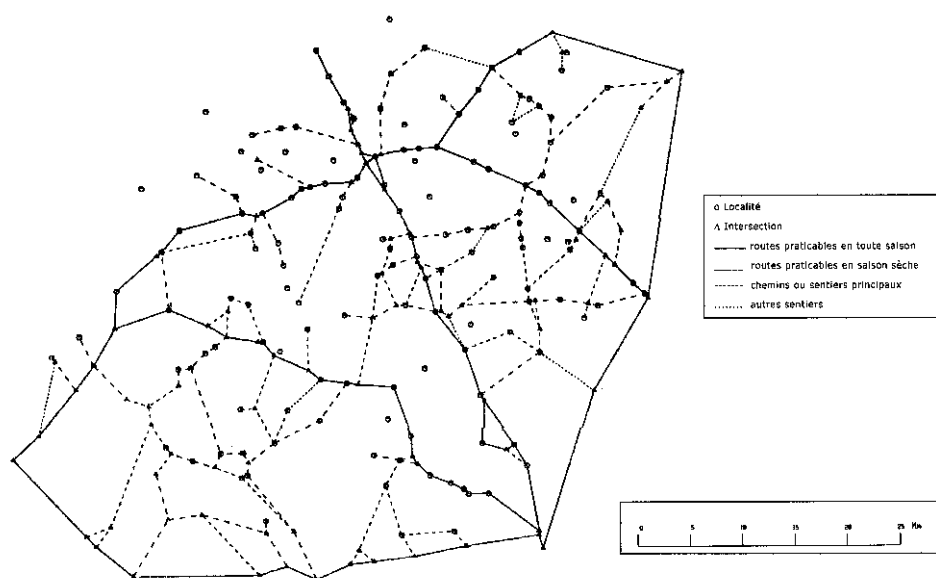


FIG. 4.6 – Réseau de transport modélisé, District de Suhum.

TAB. 4.1 – Comparaison des opérateurs de croisement et de mutation.

			I1			I2			I3		
$ V $	$ T $	$ W $	temps	ratio	DG	temps	ratio	DG	temps	ratio	DG
100	1	100	481.9	0.65	4.03	751.4	0.49	6.68	851.9	0.57	4.23
100	1	200	668.1	0.66	2.31	965.4	0.48	7.86	1065.5	0.58	3.48
100	1	300	770.7	0.63	2.14	955.0	0.47	7.48	1157.7	0.59	3.25
100	10	100	439.5	0.74	2.66	282.8	0.48	11.1	507.7	0.70	4.20
100	10	200	416.1	0.77	2.80	311.1	0.50	10.9	443.7	0.59	5.49
100	10	300	854.6	0.72	1.61	652.1	0.45	9.39	867.2	0.63	2.75
100	20	100	448.6	0.69	1.31	91.3	0.66	10.2	328.2	0.74	5.96
100	20	200	570.4	0.69	3.39	284.1	0.52	9.34	584.6	0.71	3.50
100	20	300	560.3	0.61	4.59	188.4	0.59	10.5	477.8	0.70	5.86
120	12	120	541.8	0.80	2.00	336.4	0.50	11.9	517.0	0.68	3.97
120	12	240	1119.0	0.69	2.78	582.0	0.34	16.4	1311.5	0.55	4.08
120	12	360	1286.5	0.51	3.19	892.7	0.30	13.1	1301.2	0.45	4.56
120	24	120	529.2	0.68	1.73	126.1	0.46	18.6	472.2	0.63	1.39
120	24	240	863.8	0.64	3.73	299.9	0.40	12.7	798.1	0.67	4.43
120	24	360	871.6	0.76	2.52	517.6	0.35	11.1	784.3	0.65	2.88

été paramétré expérimentalement. La population est fixée à 256 individus et il s'arrête après que 5000 générations sans mise à jour de l'archive se soient écoulées.

Deux mesures ont été utilisées pour les évaluations et les comparaisons. Tout d'abord, nous avons calculé le ratio de solutions Pareto optimales trouvées par l'algorithme génétique. Nous avons aussi employé la distance générationnelle [165] présentée dans le chapitre 1. Comme les coordonnées des points dans l'espace des objectifs ont dû être normalisées, toutes les valeurs x relatives à cette mesure doivent être lues $x \times 10^{-4}$.

4.6.3 Comparaison des opérateurs de croisement et de mutation

Trois implémentations ont été testées :

I1 : L'opérateur de croisement est SRX et l'opérateur de mutation est M1.

I2 : L'opérateur de croisement est RSX et les deux opérateurs de mutation sont utilisés.

I3 : Les deux opérateurs de croisement et les deux opérateurs de mutation sont utilisés.

Pour chaque implémentation, le tableau 4.1 reporte les informations suivantes :

Temps : Temps moyen d'exécution.

ratio : Ratio de solutions Pareto optimales trouvées.

DG : Distance générationnelle moyenne entre l'approximation générée et l'ensemble Pareto optimal.

Du point de vue du nombre de solutions Pareto optimales trouvées, c'est l'implémentation I1 qui en trouve toujours un plus grand nombre. D'autre part, lorsque c'est l'implémentation I3 qui parvient à identifier le plus grand nombre de solutions Pareto optimales, la différence avec I1 est faible alors que pour plusieurs jeux de tests, I1 trouve clairement

TAB. 4.2 – Comparaison entre la méthode exacte et l’algorithme génétique.

$ V $	$ T $	$ W $	NB	temps ϵ	ratio	DG	temps AG
75	1	75	84.8	661	0.82	3.18	205.4
75	1	150	86.8	1424.6	0.75	2.33	281.0
75	1	225	100.4	2654.4	0.73	1.33	310.8
75	7	75	65	1126.8	0.78	1.65	375.6
75	7	150	59.4	235	0.82	0.79	171.0
75	7	225	56	583.2	0.79	3.07	253.0
75	15	75	62.6	276	0.84	1.23	180.4
75	15	150	28.4	80	0.90	0.95	163.2
75	15	225	34.2	42.4	0.88	2.75	218.6
100	1	100	124.8	19622.2	0.65	4.04	481.9
100	1	200	131.4	21976.8	0.66	2.31	668.1
100	1	300	122.4	21823.8	0.63	2.14	770.7
100	10	100	61.4	951.8	0.74	2.66	439.5
100	10	200	56.6	936	0.77	2.80	416.1
100	10	300	82.4	5779.4	0.72	1.61	854.6
100	20	100	31.4	135.6	0.69	1.31	448.6
100	20	200	44.2	477.4	0.69	3.39	570.4
100	20	300	34.4	293	0.61	4.59	560.3
120	12	120	52.2	5864.8	0.80	2.00	541.8
120	12	240	83.2	11824.2	0.69	2.78	1119.0
120	12	360	93.6	15481.4	0.51	3.19	1286.5
120	24	120	35.2	334.4	0.68	1.73	529.2
120	24	240	52.6	3062	0.64	3.73	863.8
120	24	360	50	1063.8	0.76	2.52	871.6

plus de solutions Pareto optimales. L’implémentation I2 est toujours largement battue à ce niveau. D’autre part, il peut être remarqué, à une exception près, que l’implémentation I1 génère toujours des ensembles plus proches de l’ensemble optimal même lorsqu’elle trouve moins de solutions Pareto optimales que I3. Enfin, sauf pour les trois premiers jeux de données, I2 est toujours sensiblement plus rapide que les deux autres implémentations qui ont des temps d’exécution similaires.

De ces tests, il apparaît que I1 est l’implémentation qui offre le meilleur compromis entre les trois caractéristiques étudiées. Par conséquent, dans le reste des expériences de ce chapitre, lorsqu’on fait référence à l’algorithme génétique, il s’agit toujours de l’implémentation I1.

4.6.4 Comparaison entre la méthode exacte et l’algorithme génétique

Nous avons ensuite évalué la qualité et l’intérêt de l’algorithme génétique en le comparant aux ensembles Pareto optimaux générés par l’algorithme exact. Le tableau 4.2 rappelle les informations suivantes :

NB : Nombre de solutions Pareto optimales.

temps ϵ : Temps d'exécution moyen de la méthode exacte.

ratio : Ratio moyen de solutions Pareto optimales trouvées par l'algorithme génétique.

DG : Distance générationnelle moyenne entre l'approximation générée par l'algorithme génétique et l'ensemble Pareto optimal.

Temps AG : Temps d'exécution moyen de l'algorithme génétique.

Des résultats plus détaillés peuvent être trouvés dans l'annexe A.

Nous allons maintenant discuter ces résultats par rapport à deux points : la qualité des approximations et les temps de calcul. Premièrement, l'algorithme génétique est capable de générer des approximations de bonne qualité. En effet, il identifie toujours une portion significative de l'ensemble Pareto optimal, en moyenne au moins les deux tiers. De plus, la distance générationnelle indique que les solutions trouvées qui ne sont pas Pareto optimales sont proches des solutions Pareto optimales. L'augmentation de la taille de V entraîne une dégradation des résultats mais ces derniers restent de bonne qualité. Les tailles de T et de W ne semblent pas avoir un impact important sur la qualité des approximations.

Du point de vue des temps de calcul, la méthode exacte est toujours plus rapide quand on a $|V| = 50$. Ce résultat peut s'expliquer en deux points. Premièrement, l'algorithme de séparations et coupes n'est exécuté qu'un nombre de fois égal au nombre de solutions Pareto optimales qui n'est pas très important en moyenne. Pour sa part, l'algorithme génétique doit au moins s'exécuter pendant 5000 générations avant de pouvoir s'arrêter. Deuxièmement, dans la méthode exacte, il est possible d'appliquer les règles de simplification. Ces règles ayant pour principal intérêt de diminuer la cardinalité de V , les problèmes traités par l'algorithme de séparations et coupes sont donc plus *simples* par rapport à ceux résolus par l'algorithme génétique qui doit travailler sur les ensembles V et W complets. Or, il a été remarqué que la taille de V avait une importance dans les temps d'exécution de l'algorithme génétique. Lorsque $|V| = 75$, l'algorithme génétique est significativement plus rapide lorsque $|T| = 1$. Ceci peut s'expliquer par le fait que les règles de simplification ne réduisent pas le problème suffisamment pour que l'algorithme exact soit plus rapide. Quand la taille de T augmente, les simplifications sont plus efficaces et la méthode exacte redevient plus rapide. Ces observations sont confirmées lorsque $|V| = 100$. En effet, pour $|T| = 1$, la différence entre les deux méthodes est importante et en faveur de l'algorithme génétique. En outre, lorsqu'on a $|T| = 10$ et $|W| = 200$ ou 300 , l'algorithme génétique est toujours plus rapide du fait des réductions moins importantes. Pour $|T| = 100$ et $|W| = 100$, les temps de calcul sont quasiment identiques. Pour $|T| = 20$, les réductions sont toujours suffisantes pour donner un avantage à la méthode exacte. A partir de ces observations, on peut déduire qu'avec l'augmentation de la taille de V , les réductions seront de moins en moins efficaces, ce qui entraînera une augmentation du temps de calcul de l'algorithme exact tandis que les temps de calcul de l'algorithme génétique ne connaîtront pas une augmentation aussi importante. Ces remarques sont confirmées lorsque $|V| = 120$. D'ailleurs, les résultats pour $|V| = 120$ et $|T| = 1$ ne sont pas reportés à cause du coût en temps prohibitif requis par la méthode ϵ -contrainte.

TAB. 4.3 – Rapports et erreurs du nombre de solutions Pareto optimales trouvées et du nombre de couvertures trouvées.

$ V $	$ T $	$ W $	NB	Ratio	Erreur	R. Couv	E. Couv
100	1	100	124.8	0.65	0.32	0.83	0.06
100	1	200	131.4	0.66	0.26	0.84	0.05
100	1	300	122.4	0.63	0.29	0.85	0.07
100	10	100	61.4	0.74	0.19	0.89	0.02
100	10	200	56.6	0.77	0.18	0.92	0.02
100	10	300	82.4	0.72	0.21	0.90	0.02
100	20	100	31.4	0.69	0.27	0.92	0.02
100	20	200	44.2	0.69	0.28	0.93	0.02
100	20	300	34.4	0.61	0.36	0.92	0.03
120	12	120	52.2	0.80	0.16	0.93	0.02
120	12	240	83.2	0.69	0.25	0.88	0.04
120	12	360	93.6	0.51	0.40	0.80	0.06
120	24	120	35.2	0.68	0.31	0.86	0.08
120	24	240	52.6	0.64	0.32	0.87	0.07
120	24	360	50.0	0.76	0.21	0.90	0.06

4.6.5 Apport de la coopération

L'intérêt de la coopération entre l'algorithme génétique et la méthode exacte est évalué selon deux points. Premièrement, on vérifie qu'il y a un apport en termes de solutions Pareto optimales trouvées ou une diminution de la distance générationnelle. Deuxièmement, il est vérifié que les temps de calcul ne sont pas trop importants et lorsque ce n'est pas le cas, que les temps de calcul cumulés de l'algorithme génétique et des algorithmes de séparations et coupes ne deviennent pas supérieurs aux temps de la méthode exacte seule.

4.6.5.1 Intérêt de la coopération

Les résultats obtenus par l'algorithme génétique montrent que l'application des deux approches de coopération est intéressante. En effet, si on observe les informations suivantes reportées dans le tableau 4.3 :

NB : Nombre moyen de solutions Pareto optimales.

Ratio : Ratio de solutions Pareto optimales trouvées par l'algorithme génétique.

Erreur : Ratio de solutions trouvées par l'algorithme génétique qui ne sont pas Pareto optimales.

R. Couv : Ratio de couvertures qui correspondent à une solution Pareto optimale trouvées par l'algorithme génétique.

E. Couv : Ratio de solutions trouvées par l'algorithme génétique dont la couverture ne correspond pas à la couverture d'une solution Pareto optimale.

par une coopération de type I, on peut espérer gagner en pourcentage la différence entre R. Couv et Ratio. C'est-à-dire les solutions pour lesquelles on a trouvé la bonne couverture

TAB. 4.4 – Apport de la coopération de type I.

			AG			AG+BC			
$ V $	$ T $	$ W $	ratio	DG	temps	ratio	DG	temps	TT
100	1	100	0.65	4.04	481.9	0.67	2.90	80.7	562.6
100	1	200	0.66	2.31	668.1	0.70	2.05	102.7	770.8
100	1	300	0.63	2.14	770.7	0.69	2.03	131.5	902.2
100	10	100	0.74	2.66	439.5	0.80	2.14	30.5	470.0
100	10	200	0.77	2.80	416.1	0.82	2.54	33.1	449.2
100	10	300	0.72	1.61	854.6	0.76	1.43	60.6	915.2
100	20	100	0.69	1.31	448.6	0.76	0.77	18.6	467.2
100	20	200	0.69	3.39	570.4	0.76	2.59	30.9	604.1
100	20	300	0.61	4.59	560.3	0.68	4.10	20.6	580.9
120	12	120	0.80	2.00	541.8	0.84	1.16	55.7	597.5
120	12	240	0.69	2.78	1119.0	0.74	2.52	97.7	1206.7
120	12	360	0.51	3.19	1286.5	0.55	2.70	117.6	1404.1
120	24	120	0.68	1.73	529.2	0.75	0.34	37.9	567.1
120	24	240	0.64	3.73	863.8	0.69	3.21	75.7	939.5
120	24	360	0.76	2.52	871.6	0.78	2.49	53.8	925.4

mais qui ne sont pas Pareto optimales. Il apparaît donc que le gain envisagé est non négligeable. Par ailleurs, comme les valeurs de E. Couv sont faibles, on ne lance pas trop de recherches inutiles. En outre, la deuxième coopération permet de rechercher un nombre de solutions important, plus précisément K solutions où K est le complément à 1 de R. Couv.

4.6.5.2 Apport de la coopération de type I

Le tableau 4.4 reporte les informations suivantes :

ratio : Ratio moyen de solutions Pareto optimales trouvées.

DG : Distance générationnelle moyenne entre l'ensemble généré et l'ensemble Pareto optimal.

temps : Temps d'exécution moyen.

TT : Temps d'exécution cumulé moyen.

Les trois premières colonnes autres que les tailles des ensembles correspondent à l'algorithme génétique seul. Les trois colonnes suivantes reportent les résultats après application des algorithmes de séparations et coupes. Le temps ne correspond qu'au temps nécessaire pour l'application de l'algorithme de séparations et coupes. Le temps cumulé des deux méthodes est donné dans la dernière colonne.

Il apparaît intéressant d'appliquer cette hybridation car pour des temps relativement faibles, elle est capable d'obtenir en moyenne 5 ratios de solutions Pareto optimales supplémentaires et dans tous les cas de faire baisser l'écart entre l'ensemble généré et l'ensemble Pareto optimal.

TAB. 4.5 – Apport de la coopération de type II.

V	T	W	AG			AG+BC			TT
			ratio	DG	temps	ratio	DG	temps	
100	1	100	0.65	4.04	481.9	0.65	3.14	34.9	516.8
100	1	200	0.66	2.31	668.1	0.69	1.97	120.5	788.6
100	1	300	0.63	2.14	770.7	0.65	1.62	113.2	883.9
100	10	100	0.74	2.66	439.5	0.77	2.25	7.5	447.9
100	10	200	0.77	2.80	416.1	0.78	2.71	8.8	424.9
100	10	300	0.72	1.61	854.6	0.76	1.53	23.5	878.1
100	20	100	0.69	1.31	448.6	0.73	1.21	3.5	452.1
100	20	200	0.69	3.39	570.4	0.74	2.67	10.0	580.4
100	20	300	0.61	4.59	560.3	0.65	3.56	7.6	567.9
120	12	120	0.80	2.00	541.8	0.82	1.93	8.2	550.0
120	12	240	0.69	2.78	1119.0	0.73	2.74	24.2	1243.2
120	12	360	0.51	3.19	1286.5	0.54	2.87	35.2	1321.7
120	24	120	0.68	1.73	529.2	0.69	1.53	6.8	536.0
120	24	240	0.64	3.73	863.8	0.68	3.32	13.3	877.1
120	24	360	0.76	2.52	871.6	0.80	2.41	18.4	890.0

4.6.5.3 Apport de la coopération de type II

Les mêmes types d'information que dans le tableau 4.4 sont reportés dans le tableau 4.5. Des conclusions similaires peuvent être effectuées, à savoir que pour un coût en temps de calcul faible par rapport au temps d'exécution de l'algorithme génétique, l'application de l'algorithme de séparations et coupes permet un gain tant au niveau du nombre de solutions Pareto optimales trouvées que de l'écart entre l'approximation et l'ensemble Pareto optimal.

4.6.5.4 Apport combiné des coopérations de type I et II

Enfin, il a été vérifié que les deux approches de coopération ne retrouvaient pas les mêmes solutions. Pour cela, nous avons appliqué les deux approches sur l'ensemble fourni par l'algorithme génétique ; le résultat de la coopération étant l'union des ensembles obtenus. Le tableau 4.6 reporte les résultats des différentes mesures pour l'algorithme génétique et pour la version où les deux hybridations sont appliquées sur les résultats de l'algorithme génétique. Il apparaît que combiner les deux approches apporte de meilleurs résultats sans entraîner une dégradation substantielle des temps de calcul.

4.6.6 Tests sur le cas réel du district de Suhum au Ghana

Le tableau 4.7 reporte les informations suivantes :

NB : Le nombre de solutions Pareto optimales.

Temps : Le temps d'exécution. Dans le cas de l'algorithme génétique, il s'agit d'une

TAB. 4.6 – Apport combiné des coopérations de type I et II.

			AG			AG+BC			T. Exact
$ V $	$ T $	$ W $	ratio	DG	temps	ratio	DG	temps	
100	1	100	0.65	4.04	481.9	0.67	2.79	597.5	19622.2
100	1	200	0.66	2.31	668.1	0.73	1.97	891.3	21976.8
100	1	300	0.63	2.14	770.7	0.70	1.62	1015.4	21823.8
100	10	100	0.74	2.66	439.5	0.82	1.95	477.7	956.3
100	10	200	0.77	2.80	416.1	0.83	2.48	458.0	936.0
100	10	300	0.72	1.61	854.6	0.78	1.37	938.7	5779.4
100	20	100	0.69	1.31	448.6	0.78	0.74	470.7	135.6
100	20	200	0.69	3.39	570.4	0.79	1.95	614.1	477.4
100	20	300	0.61	4.59	560.3	0.73	3.24	588.5	293.0
120	12	120	0.80	2.00	541.8	0.86	1.06	605.7	5694.8
120	12	240	0.69	2.78	1119.0	0.77	2.51	1230.9	11824.2
120	12	360	0.51	3.19	1286.5	0.57	2.55	1439.3	15481.4
120	24	120	0.68	1.73	529.2	0.76	0.32	573.9	334.4
120	24	240	0.64	3.73	863.8	0.72	2.89	952.8	3062.0
120	24	360	0.76	2.52	871.6	0.81	2.39	943.8	1063.8

TAB. 4.7 – Résultats sur le cas réel du district de Suhum, Ghana.

	Algorithme exact		Algorithme génétique		
	NB	Temps	Ratio	DG	Temps
saison sèche	48	5577	0.96	0.65	108.8
saison des pluies	19	36	1.00	0.00	5.2

moyenne sur l'ensemble des exécutions.

Ratio : Moyenne des pourcentages de solutions Pareto optimales trouvées par l'algorithme génétique.

DG : Moyenne des distances générationnelles entre les ensembles générés par l'algorithme génétique et l'ensemble Pareto optimal.

Les ensembles Pareto optimaux sont représentés dans les figures 4.7 et 4.8.

Une première remarque est que la résolution lors de la saison des pluies n'est pas intéressante. En effet, dans l'algorithme génétique, la totalité ou la quasi-totalité des solutions Pareto optimales est identifiée dès la phase de création de la population initiale par l'heuristique. Cela s'explique par le fait qu'un des villages se trouve très loin d'un village qui peut être visité par le véhicule durant la saison des pluies. Ainsi, si on regarde la solution offrant la meilleure couverture, le véhicule ne peut visiter que trois villages. Tout village supplémentaire dans la tournée ne peut pas améliorer la couverture.

Les figures 4.9 et 4.10 représentent les solutions offrant la meilleure couverture durant la saison sèche et la saison des pluies. Il peut être nécessaire de rappeler que les distances entre deux points ne correspondent pas à la distance à vol d'oiseau entre les deux points sur la carte. Il apparaît que le PTC bi-objectif n'est pas adapté au cas du district de Suhum. En effet, si on regarde la meilleure couverture trouvée, elle est d'environ neuf kilomètres

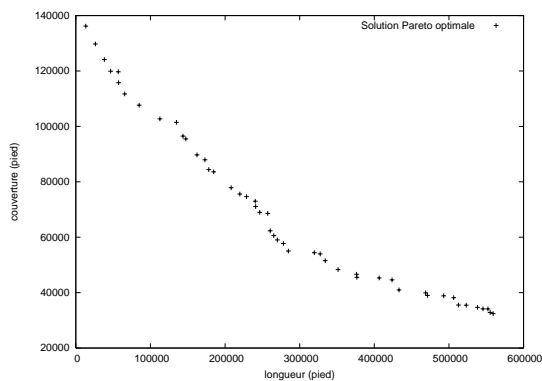


FIG. 4.7 – Ensemble Pareto optimal pour la saison sèche.

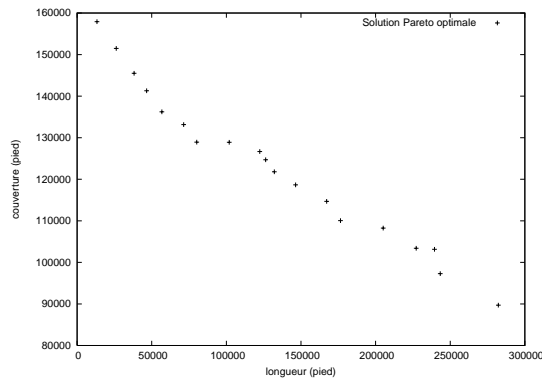


FIG. 4.8 – Ensemble Pareto optimal pour la saison des pluies.

pour la saison sèche et vingt-cinq kilomètres pour la saison des pluies. Or, dans l'étude de Hodgson et al. [79], le ministère Ghanéen de la santé avait fixé la couverture maximale à huit kilomètres. Pour pouvoir traiter le problème, les auteurs avaient retiré de la carte les villages qui ne pouvaient pas être couverts. Une telle politique n'est possible que si la couverture est fixée et on ne peut donc pas l'utiliser dans le cas du PTC bi-objectif.

Une manière de traiter les cas où toute la population n'est pas couverte serait d'ajouter un objectif supplémentaire qui aurait pour but de maximiser le nombre de personnes couvertes. Cependant, il ne s'agirait plus de résoudre un PTC. En effet, le choix des sommets à visiter doit se faire sur deux points :

1. Un sous-ensemble $W' \subseteq W$ de sommets que l'on souhaite pouvoir couvrir.
2. Un sous-ensemble de sommets $V' \subseteq V$ sur lequel s'effectue une tournée et qui couvre tous les sommets de W' .

Une autre manière d'aborder ce problème serait de rendre l'objectif associé à la couverture plus discriminant. En effet, pour une couverture donnée, il peut exister un grand nombre de tournées. Cependant, dans le PTC bi-objectif, on cherche la tournée la plus courte qui est probablement aussi l'une des tournées qui visite le moins de sommets. De ce fait, pour les sommets de W autres que celui qui définit la couverture, la distance avec le sommet visité le plus proche peut être très importante. Il existe en effet des solutions non Pareto pour ce problème qui sont intéressantes d'un point de vue pratique. Par exemple, il peut être intéressant de détériorer la longueur de la tournée sans modifier toutefois la couverture pour diminuer les distances entre les autres sommets de W et les sommets visités. Si on prend le cas de la solution dans la figure 4.10, cela laisserait toujours le village le plus éloigné à vingt-cinq kilomètres mais permettrait de raccourcir efficacement les distances pour les autres sommets. Un tel objectif pourrait être la minimisation de la somme des distances entre les sommets de W et les sommets les plus proches de V :

$$\min \sum_{v_l \in W} \min_{v_k \in \Theta} (c_{kl})$$

où $\Theta \subseteq V$ est l'ensemble des sommets visités par la tournée.

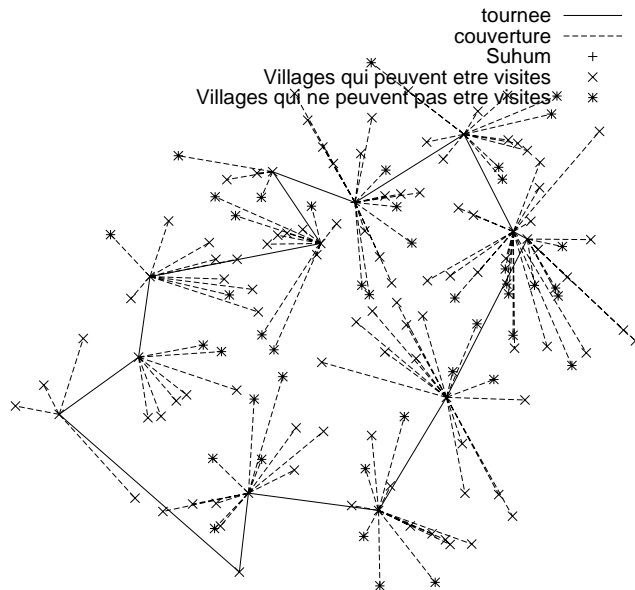


FIG. 4.9 – Solution offrant la meilleure couverture pour la saison sèche.

4.7 Conclusion

Dans ce chapitre, nous avons étudié le problème de la tournée couvrante bi-objectif qui est une généralisation du problème de la tournée couvrante. Dans cette généralisation, une contrainte du problème original a été exprimée naturellement sous la forme d'un second objectif. Pour le résoudre, nous avons conçu et implémenté une stratégie exacte multi-objectif ainsi qu'un algorithme génétique multi-objectif. Nous avons aussi exploré l'utilisation de la coopération au travers d'une stratégie en deux phases combinant l'algorithme génétique multi-objectif et un algorithme de séparations et coupes pour le problème de la tournée couvrante. Des expérimentations ont été faites sur des jeux de données générés. Dans un premier temps, la méta-heuristique a été comparée avec la méthode exacte multi-objectif. Les résultats ont montré que l'algorithme génétique était capable de générer des approximations de bonne qualité sans que les temps de calcul requis ne deviennent prohibitifs, ce qui est le cas pour la méthode exacte. Dans un second temps, nous avons évalué l'intérêt de la coopération qui se révèle être capable d'améliorer les ensembles générés sans être pénalisante au niveau des temps de calcul. Enfin, nous avons résolu des instances qui correspondent à un cas réel d'élaboration de tournées d'une unité de soins mobile au Ghana.

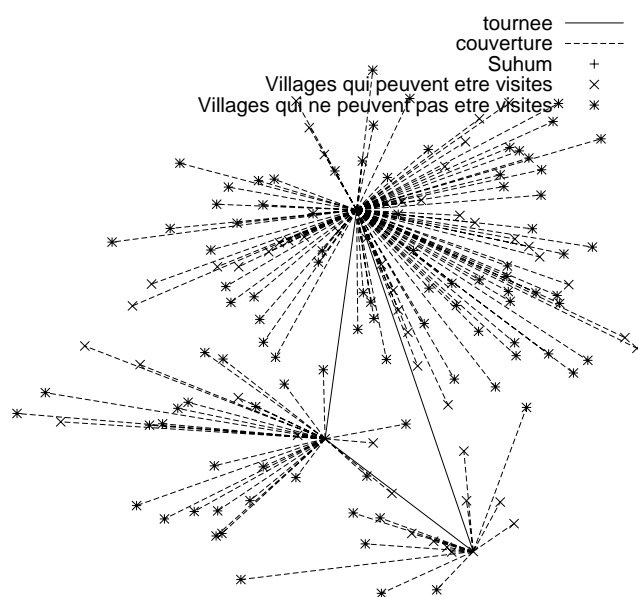


FIG. 4.10 – Solution offrant la meilleure couverture pour la saison des pluies.

Chapitre 5

Modélisation et résolution du problème d'élaboration de tournées de véhicules avec équilibrage des tournées

Ce chapitre présente la modélisation et la résolution du problème d'élaboration de tournées de véhicules avec équilibrage des tournées qui est une extension bi-objectif du problème d'élaboration de tournées de véhicules. Le second objectif permet d'inclure un équilibrage entre les tournées au travers d'un objectif linéaire. Nous avons modélisé le problème sous la forme d'un programme linéaire en nombre entier bi-objectif.

Nous avons proposé plusieurs méthodes de résolution. La première est un algorithme génétique multi-objectif parallèle incluant la diversification élitiste. Ensuite, plusieurs implémentations de NSGA II ont été faites. Enfin, une implémentation de notre stratégie TAPaS utilisant une recherche Tabou est décrite. Toutes ces implémentations suivent les schémas généraux donnés dans le chapitre 3.

Les méthodes et les différents mécanismes ont été évalués sur des jeux de données classiques pour le problème d'élaboration de tournées de véhicules avec contraintes de capacité. Nous avons ainsi pu observer la pertinence des mécanismes et méthodes proposés.

5.1 Introduction

Dans ce chapitre, nous étudions la résolution d'un problème d'élaboration de tournées de véhicules bi-objectif nommé le problème d'élaboration de tournées de véhicules avec équilibrage des tournées. Le second objectif vise à limiter les différences de longueur entre les tournées.

Deux algorithmes génétiques multi-objectif sont proposés pour le problème d'élaboration de tournées de véhicules avec équilibrage des tournées. Le premier algorithme est l'adaptation d'un algorithme génétique initialement conçu pour un problème de flowshop bi-objectif [152]. Plusieurs implémentations sont proposées. On utilise notamment les mécanismes présentés dans le chapitre 3. Dans le cadre du problème de flowshop bi-objectif, les expérimentations montrent que l'algorithme génétique initial offre de meilleures performances que NSGA II. Dans le cadre du problème de tournées bi-objectif étudié dans cette thèse, il apparaît que NSGA II offre sur certains jeux de données des performances légèrement meilleures. Nous proposons donc aussi une implémentation de NSGA II pour le problème d'élaboration de tournées de véhicules avec équilibrage des tournées. Des implémentations de NSGA II parallèle et avec la diversification élitiste sont aussi proposées. Nous présentons aussi une implémentation de TAPaS utilisant une recherche Tabou de littérature.

Des évaluations ont été menées sur des jeux de données standard de la littérature pour le problème d'élaboration de tournées de véhicules. Nous avons étudié l'apport des mécanismes et des politiques parallèles dans le cadre des algorithmes génétiques multi-objectif. Nous avons aussi vérifié que l'hybridation d'un algorithme génétique avec TAPaS offrait des résultats intéressants.

Ce chapitre est organisé de la manière suivante : le problème d'élaboration de tournées de véhicules avec équilibrage des tournées est défini dans la section 5.2. Le premier algorithme génétique est détaillé dans la section 5.3 et l'implémentation de NSGA II dans la section 5.4. Les opérateurs génétiques pour ces deux algorithmes sont des méthodes dédiées au problème d'élaboration de tournées de véhicules et elles sont présentées dans la section 5.5. L'implémentation de TAPaS pour le problème d'élaboration de tournées de véhicules avec équilibrage des tournées est décrite dans la section 5.6. Les résultats expérimentaux sont reportés dans la section 5.7. Enfin, nous présentons les conclusions de ce chapitre dans la section 5.8.

5.2 Présentation et modélisation

Le problème d'élaboration de tournées de véhicules avec équilibrage des tournées (PTVET) est une extension du problème d'élaboration de tournées de véhicules (PTV) où la longueur totale des tournées est minimisée et les tournées sont équilibrées. Cette partie est organisée de la manière suivante. Tout d'abord, les objectifs étudiés sont présentés. Ensuite, un programme linéaire en nombre entier bi-objectif est donné pour le PTVET.

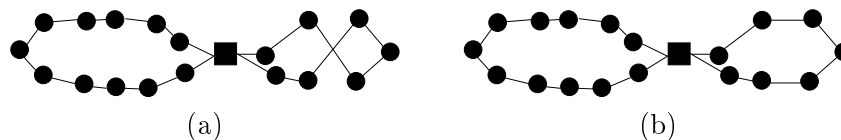


FIG. 5.1 – (a) est mieux équilibré que (b), mais (b) n’améliore pas l’équilibre de manière artificielle.

TAB. 5.1 – Valeurs des objectifs pour les meilleures solutions en termes de distance trouvées par Taburoute et l’algorithme génétique de Prins.

Instance	Taburoute		AG de Prins	
	Distance	Equilibre	Distance	Equilibre
E51-05e	524.61	20.07	524.61	20.07
E76-10e	835.32	78.10	835.26	91.08
E101-08e	826.14	97.88	826.14	97.88
E151-12c	1031.17	98.24	1031.63	100.34
E200-17c	1311.35	106.70	1300.23	82.31
E121-07c	1042.11	146.67	1042.11	146.67
E101-10c	819.56	93.43	819.56	93.43

5.2.1 Objectifs

Le PTVET est une extension bi-objectif du PTV où l’on considère les deux objectifs suivants :

1. Minimisation de la distance parcourue par les véhicules.
2. Minimisation de la différence entre la longueur de la plus longue tournée et la longueur de la plus courte tournée.

Le second objectif est ajouté pour identifier des solutions dans lesquelles la charge de travail entre les véhicules est répartie équitablement. Ici, le travail d’un véhicule est exprimé par la distance qu’il parcourt. D’autre part, les tournées doivent être optimales pour éviter que les solutions ne soient artificiellement équilibrées comme c’est le cas dans la figure 5.1.

Le tableau 5.1 reporte les valeurs de ces deux objectifs pour les meilleures solutions trouvées par Taburoute [69] et l’algorithme génétique de Prins [124] sur sept jeux de données pour le PTV avec contraintes de capacité proposés par Christofides et Eilon [22] et Christofides et al. [23]. En suivant la nomenclature utilisée dans [159], le nom de chaque instance est de la forme $Ei-jk$. E signifie que la distance Euclidienne est utilisée. i est le nombre de sommets, dépôt inclus, et j est le nombre de véhicules disponibles. k est une lettre qui identifie l’article où les données sont présentées. $k = e$ réfère à Christofides et Eilon [22] et $k = c$ à Christofides et al. [23]. Taburoute et l’algorithme génétique de Prins, qui peuvent être considérés comme faisant partie des meilleurs algorithmes pour le PTV, ne prennent pas en compte l’équilibrage des tournées. Cela apparaît clairement dans le tableau 5.1 où les meilleures solutions identifiées sont de mauvaise qualité pour l’objectif additionnel.

5.2.2 Programme linéaire en nombre entier bi-objectif

Le programme linéaire en nombre entier pour le PTVET est basé sur celui pour le PTV donné dans le chapitre 2. Deux autres nouvelles variables, l_{\min} et l_{\max} , correspondent respectivement à la longueur de la tournée la plus courte et à la longueur de la tournée la plus longue. Le programme linéaire en nombre entier pour le PTVET est alors le suivant :

$$\begin{cases} \min \sum_{k=1}^M \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}^k \\ \min l_{\max} - l_{\min} \end{cases} \quad (5.1)$$

sous les contraintes :

$$\sum_{i=1}^n q_i y_i^k \leq Q \quad (k = 1, \dots, M) \quad (5.2)$$

$$\sum_{k=1}^M y_i^k = \begin{cases} M & (i = 1) \\ 1 & (i = 2, \dots, n) \end{cases} \quad (5.3)$$

$$\sum_{i=1}^n x_{ij}^k = y_j^k \quad (j = 1, \dots, n; k = 1, \dots, M) \quad (5.4)$$

$$\sum_{i=1}^n x_{ij}^k = y_i^k \quad (i = 1, \dots, n; k = 1, \dots, M) \quad (5.5)$$

$$\sum_{i,j \in S} x_{ij}^k \leq |S| - 1 \quad (S \subset V; 2 \leq |S| \leq n - 2; k = 1, \dots, M) \quad (5.6)$$

$$l_{\max} \geq \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}^k \quad (k = 1, \dots, M) \quad (5.7)$$

$$l_{\min} \leq \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}^k \quad (5.8)$$

$$x_{ij}^k \in \{0, 1\} \quad (i, j = 1, \dots, n; i \neq j; k = 1, \dots, M) \quad (5.9)$$

$$y_i^k \in \{0, 1\} \quad (i = 1, \dots, n; k = 1, \dots, M) \quad (5.10)$$

$$v_k \in \{0, 1\} \quad (k = 1, \dots, M) \quad (5.11)$$

$$l_{\max} \geq 0 \quad (5.12)$$

$$l_{\min} \geq 0 \quad (5.13)$$

Les contraintes 5.7 assurent que l_{\max} sera plus grande ou égale à la plus longue des tournées. De même, les contraintes 5.8 permettent d'assurer que l_{\min} sera toujours plus petite que la longueur de la tournée la plus courte. Les autres contraintes sont les mêmes que pour le programme linéaire du problème d'élaboration de tournées de véhicules donné dans le chapitre 2.

Ces nouvelles contraintes, ou des contraintes équivalentes, peuvent être ajoutées à toute modélisation du PTV qui permet de connaître la longueur de la tournée associée à un véhicule.

5.3 NEMO : un algorithme génétique multi-objectif parallèle pour le PTVET

Cette partie présente NEMO (*Non-panmictic Evolutionary algorithm for Multi-objective Optimization*), un algorithme génétique développé pour résoudre le PTVET. Plus précisément, trois implémentations de l'algorithme sont décrites. Chaque version correspond à la version précédente qui a été modifiée ou à laquelle on a ajouté des mécanismes. Les trois versions sont présentées pour permettre de voir l'évolution de la méthode. Cependant, pour ne pas surcharger le document de tableaux et comme la meilleure implémentation est la dernière, on ne parlera que de cette méthode dans la section 5.7 consacrée aux résultats expérimentaux.

5.3.1 Première version de NEMO

Cet algorithme est une adaptation au PTV de l'algorithme génétique multi-objectif pour le flowshop de Talbi et al. [152]. Nous allons présenter les principales étapes : la phase de sélection, la phase de recombinaison, la phase de mise à jour de l'archive et la phase de communication.

5.3.1.1 Phase de sélection

Durant la phase de sélection, un rang k est d'abord attribué à chaque solution par une technique de profondeur de rang. Ensuite, on calcule pour chaque solution s son fitness $\pi(s)$ de la manière suivante :

$$\pi(s) = \frac{S(N + 1 - R_k) + R_k - 2}{N(N - 1)} \quad (5.14)$$

où S est la pression de sélection, N la taille de la population et

$$R_k = 1 + |E_k| + 2 \times \sum_{i \in \{1 \dots k-1\}} |E_i| \quad (5.15)$$

Pour maintenir la diversité durant la recherche, une technique de fitness sharing [75] est appliquée dans l'espace des objectifs. Le fitness $\pi(u)$ d'un individu u est divisé par une valeur appelée *compteur de niche* :

$$\pi'(u) = \frac{\pi(u)}{m(u)} \quad (5.16)$$

Le compteur de niche est calculé à l'aide de la formule $\sum_{v \in \text{population}} sh(d(u, v))$ où sh est la fonction de sharing définie de la manière suivante :

$$sh(d(u, v)) = \begin{cases} 1 - \left(\frac{d(u, v)}{\gamma}\right)^\alpha & \text{si } d(u, v) < \gamma \\ 0 & \text{sinon.} \end{cases} \quad (5.17)$$

où γ est la taille de la niche. La distance $d(u, v)$ est la distance de Manhattan entre les solutions $F(u)$ et $F(v)$.

On choisit alors par le mécanisme de sélection *stochastic universal sampling* [7] les $\frac{N}{2} - M$ individus qui appartiendront à la population lors de la génération suivante et qui serviront à générer de nouveaux individus. M individus de l'archive sont inclus dans la population intermédiaire pour favoriser la convergence à travers le mécanisme d'élitisme.

5.3.1.2 Phase de recombinaison

Durant cette phase, $\frac{N}{2}$ individus sont créés à partir des $\frac{N}{2}$ individus choisis lors de la phase de sélection. On procède de la manière suivante : les parents sont regroupés par couples de manière aléatoire. Puis deux enfants sont créés par couple par application d'un opérateur de croisement. Il y a le choix entre deux opérateurs de croisement, chacun ayant une même probabilité d'être appliqué. Cependant, selon la nature du jeu de données, on n'utilise pas les deux mêmes opérateurs de croisement. En effet, si les clients sont répartis de manière uniforme sur l'ensemble de la carte, on utilise les opérateurs de croisement RBX et SPLIT. Cette combinaison ne conduisant pas à de bonnes performances sur les cartes où les clients sont groupés, on utilise les opérateurs de croisement RBX et SBX. Chaque individu nouvellement généré a une probabilité de muter par application de l'opérateur de mutation Or-opt. Une recherche locale 2-opt est appliquée sur les tournées de la solution nouvellement créée. Tous les individus générés sont conservés et forment avec les solutions sélectionnées durant la phase de sélection la nouvelle population.

5.3.1.3 Phase de mise à jour de l'archive

Après la phase de recombinaison, on inclut les individus non-dominés de la nouvelle population dans l'archive. La taille de l'archive est petite, 50 solutions, pour éviter trop de pression lorsqu'on utilise l'élitisme [175]. Quand l'archive est pleine, on retire des solutions en appliquant l'algorithme de clustering *average linking method* [109]. Cet algorithme est connu pour bien fonctionner sur des ensembles structurés comme les ensembles potentiellement Pareto optimaux. Les solutions extrêmes de l'ensemble potentiellement Pareto optimal sont toujours gardées même lorsqu'elles ne sont pas sélectionnées par l'algorithme de clustering.

5.3.1.4 Phase de communication

L'utilisation du parallélisme a deux buts : explorer plus de solutions en un temps donné et améliorer la qualité des résultats. La parallélisation est effectuée sous la forme d'un modèle en îles où chaque île correspond à un algorithme génétique. À intervalles réguliers en termes de générations, les algorithmes génétiques échangent leur archive. La topologie de communication est un anneau (voir figure 5.2) pour minimiser les communications et aussi maintenir l'aspect connexe du graphe et assurer qu'un bon individu aura atteint tous les algorithmes génétiques après un certain nombre de migrations.

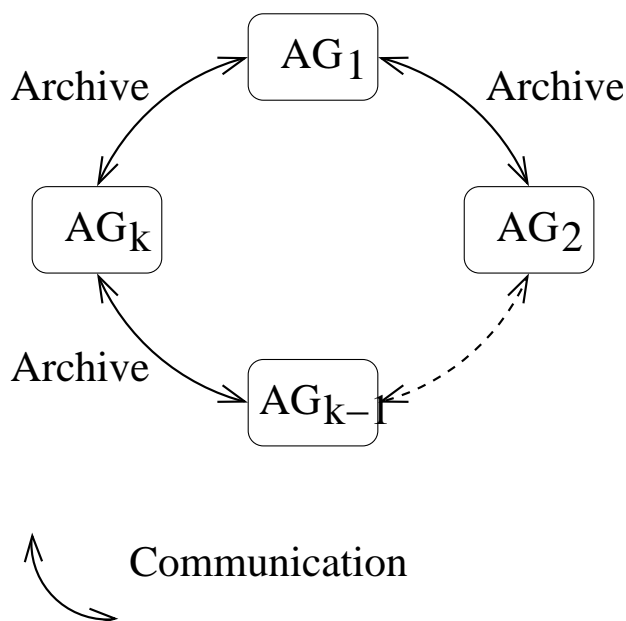


FIG. 5.2 – Topologie de communication du modèle en îles.

5.3.1.5 Post-optimisation

La phase de post-optimisation est une recherche guidée par l'archive telle qu'elle est décrite dans le chapitre 3. Le voisinage est défini par le mouvement Or-opt. Cependant, tout le voisinage n'est pas parcouru pour réduire les temps de calcul. Les voisins sont alors visités aléatoirement.

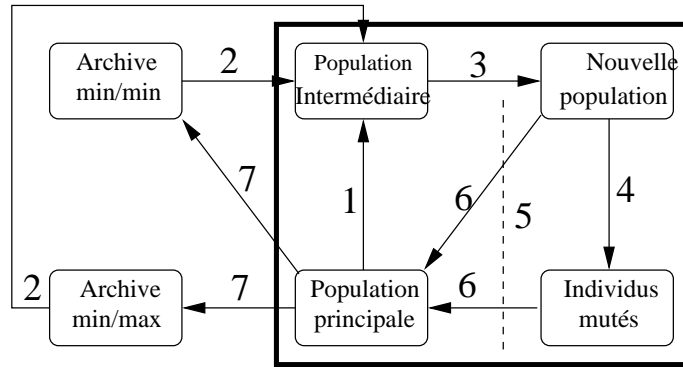
5.3.2 Seconde version de NEMO

Pour améliorer les résultats, l'algorithme précédent a été modifié sur deux points : l'ajout de la diversification élitiste et une nouvelle procédure de post-optimisation utilisant une recherche Tabou.

5.3.2.1 Modification de l'algorithme génétique par rapport à la version précédente

Par rapport à la version précédente, cette version se différencie par l'ajout partiel de la diversification élitiste. En effet, on a seulement ajouté l'archive où l'on minimise la longueur et maximise l'équilibre pour favoriser la recherche de solutions de bonne qualité vis-à-vis du premier objectif qui est le plus dur à optimiser. Les modifications apportées à la méthode présentée en 5.3.1 sont :

1. On ajoute une nouvelle archive A_{mM} . Dans la phase de sélection, on ne choisit plus alors que $\frac{N}{2} - 2M$ solutions de la population précédente. La nouvelle population



- | | | |
|----------------------|---------------------|-----------------------------|
| 1. Sélection par SUS | 4. Mutation | 7. Mise à jour des archives |
| 2. Elitisme et DE | 5. Recherche locale | |
| 3. Recombinaison | 6. Remplacement | |

FIG. 5.3 – Ajout de la seconde archive à l’algorithme génétique multi-objectif.

intermédiaire est complétée par M individus de l’archive standard de type A_0 , comme cela était le cas auparavant, et par M individus de la nouvelle archive A_{mM} . De la même manière, lors de la phase de mise à jour des archives, on tente d’inclure les nouvelles solutions générées dans la nouvelle archive. La boucle principale de l’algorithme génétique avec prise en compte de la nouvelle archive est illustrée dans la figure 5.3.

- Le modèle parallèle en îles ne change pas puisque nous utilisons une seule archive supplémentaire et tous les algorithmes génétiques sont donc du même type. Il suffit juste de modifier la phase de communication pour envoyer les deux archives aux deux îles voisines et recevoir les solutions en provenance des îles voisines.

5.3.2.2 Post-optimisation

La phase de post-optimisation n’est pas non plus la même que lors de l’implémentation précédente. Le guidage de la recherche est fait par restriction de l’espace de recherche (cf. chapitre 3). La zone associée à une recherche est définie comme expliqué ci-après. Tout d’abord, les objectifs sont normalisés. Cela permet de ne pas favoriser un objectif par rapport à l’autre. On a alors défini δ la droite de pente $\frac{1}{2}$ qui passe par la solution de départ s_0 . Soit $d_\delta(s)$ la distance entre l’image de la solution s dans l’espace des objectifs et sa projection sur δ . La recherche qui part de la solution s_0 est alors restreinte aux solutions de l’ensemble $\{s \mid d_\delta(s) \leq \Gamma\}$ avec $\Gamma \geq 0$. Le principe de cette restriction est illustré à la figure 5.4.

L’utilisation d’un guidage par restriction de l’espace permet l’utilisation d’une recherche Tabou dont les principaux mécanismes sont expliqués ci-après. L’opérateur de voisinage est le mouvement Or-opt. Quand un client est retiré d’une route, il ne peut pas y être réinséré avant que θ itérations se soient écoulées. Comme suggéré par Gendreau et al. [69], les solutions qui ne respectent pas la contrainte de capacité sont prises en compte mais elles sont dégradées selon les mêmes règles que dans [69]. Enfin, le critère d’arrêt est vérifié

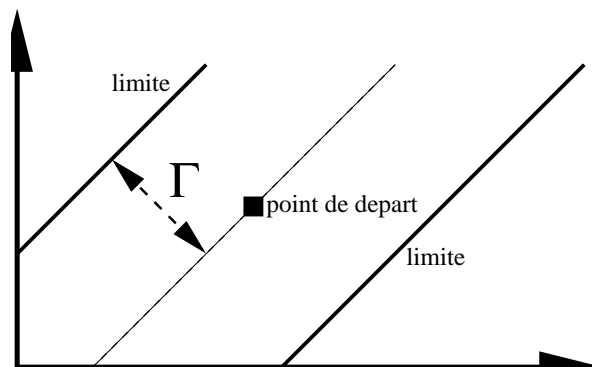


FIG. 5.4 – Restriction de l'espace des objectifs.

Algorithme 5.1 Boucle principale de l'île I_k^i

```

{N : taille de la population}
P0 ← population_initiale()
A0 ← {s ∈ P0 | ∀s' ∈ P0, s' ≠ s}
Ak ← {s ∈ P0 | ∀s' ∈ P0, s' ≠k s}
t ← 0
tant que t < max_generation faire
  communication(t, A0, Ak) {c.f. 5.3.3.4}
  sélection(Pt, Pt+1, A0, Ak) {c.f. 5.3.3.1}
  recombinaison(Pt+1) {c.f. 5.3.3.2}
  mise_à_jour_des_archives(Pt+1, A0, Ak) {c.f. 5.3.3.3}
  t ← t + 1
fin tant que

```

lorsqu'un nombre d'itérations fixé a été effectué.

5.3.3 Troisième version de NEMO

Cette dernière version inclut une implémentation complète de la diversification élitiste et du modèle parallèle correspondant ainsi que la modification de certaines phases et mécanismes. D'autre part, la phase de post-optimisation n'est pas appliquée. Tous les algorithmes génétiques ont donc deux archives A_0 , l'archive standard, et A_k , avec $k = 1$ ou 2, l'archive ajoutée par la diversification élitiste. La boucle principale pour une île I_k^i est donnée dans l'algorithme 5.1.

5.3.3.1 Phase de sélection

L'attribution du rang, le calcul du fitness et l'application du sharing se font de la même manière que précédemment si ce n'est que la taille des niches pour le sharing est fixée à $\frac{N}{2}$ comme préconisé dans [14] et que les valeurs des objectifs sont normalisées. Les solutions

Algorithme 5.2 sélection(P_t, P_{t+1} : Population, A_0, A_k : Archive)

{*élitisme*(A, M) : choisit M individus dans A }
 ranking(P_t)
 sharing(P_t)
 calcul_fitness(P_t)
 tri_sur_fitness(P_t)
 $P_{t+1} \leftarrow P_t[1 \dots N/2 - 2M] \cup \text{élitisme}(A_0, M) \cup \text{élitisme}(A_k, M)$

de la population sont alors triées par fitness croissant et les $\frac{N}{2} - 2M$ premières solutions sont choisies pour faire partie de la population intermédiaire. Comme précédemment, les $2M$ solutions restantes proviennent des deux archives. L'algorithme 5.2 correspond à cette phase.

5.3.3.2 Phase de recombinaison

La génération d'un enfant fonctionne de la manière suivante : d'abord deux parents de la population intermédiaire sont sélectionnés à l'aide d'un tournoi binaire. Une solution u est préférée à une solution v si elle a un meilleur rang ou si elles ont toutes deux le même rang et si $m(u)$ est plus petit que $m(v)$.

Deux enfants sont alors créés par application d'un opérateur de croisement choisi entre RBX et l'opérateur SPLIT. Finalement, l'opérateur de mutation Or-opt est appliqué avec une probabilité $prob_{mutation}$ sur chaque enfant. Une recherche locale 2-opt est appliquée sur les tournées de la nouvelle solution.

Pour chaque couple de parents sélectionnés, on crée deux nouvelles solutions qui deviennent des individus de la population pour la prochaine génération. Le processus précédent est répété jusqu'à ce que $\frac{N}{2}$ individus aient été générés. La phase de recombinaison est résumée dans l'algorithme 5.3.

5.3.3.3 Phase de mise à jour des archives

Les deux archives sont mises à jour après la phase de recombinaison. Si leur taille après inclusion des nouvelles solutions dépasse une limite fixée, des solutions sont retirées par application de l'algorithme *average linking method* [109] (voir l'algorithme 5.4).

5.3.3.4 Phase de communication

Le modèle parallèle complet associé à la diversification élitiste a été implémenté. L'île I_k^i a donc quatre voisines et elle leur communique ses archives A_0 et A_k selon la politique de communication donnée dans l'algorithme 5.5.

Algorithme 5.3 recombinaison(P : Population)

```

i ← 0
tant que i <  $N/2$  faire
  parent1 = tournoi_binaire( $P[1 \dots N/2]$ )
  parent2 = tournoi_binaire( $P[1 \dots N/2]$ )
  si rand() < Probrbx alors
    croisement ← RBX
  sinon
    croisement ← SPLIT
  fin si
   $P[N/2 + i]$  ← croisement( $P_{t+1}[parent_1]$ ,  $P_{t+1}[parent_2]$ )
   $P[N/2 + i + 1]$  ← croisement( $P_{t+1}[parent_2]$ ,  $P_{t+1}[parent_1]$ )
  si rand() < probmutation alors
    or_opt( $P[N/2 + i]$ )
  fin si
  si rand() < probmutation alors
    or_opt( $P[N/2 + i + 1]$ )
  fin si
  recherche_locale_2opt( $P[N/2 + i]$ )
  recherche_locale_2opt( $P[N/2 + i + 1]$ )
  i ← i + 2
fin tant que

```

Algorithme 5.4 mise_à_jour_des_archives(P : Population, A_0, A_k : Archive)

```

 $A_0$  ←  $A_0 \setminus \{s \in A_0 \mid \exists s' \in P, s' \prec s\} \cup \{s \in P \mid \forall s' \in A_0, s' \not\prec s\}$ 
 $A_k$  ←  $A_k \setminus \{s \in A_k \mid \exists s' \in P, s' \prec_k s\} \cup \{s \in P \mid \forall s' \in A_k, s' \not\prec_k s\}$ 
si  $A_0.size > archive\_max\_size$  alors
  clustering( $A_0$ )
fin si
si  $A_k.size > archive\_max\_size$  alors
  clustering( $A_k$ )
fin si

```

5.4 Implémentation de NSGA II pour le PTVET

L'implémentation de NSGA II pour le PTVET suit le schéma donné dans le chapitre 3. La phase de recombinaison, qui est la seule phase qu'il faut définir dans le cadre particulier du PTVET, est donnée par l'algorithme 5.6. Pour prendre en compte la diversification élitiste, il suffit de modifier dans cette phase la sélection des parents pa_1 et pa_2 en considérant les solutions se trouvant dans les archives additionnelles. Les opérateurs génétiques utilisés sont l'opérateur RBX, l'opérateur SPLIT et l'opérateur de mutation Or-opt. La taille des populations P et Q est fixée à 128 individus chacune.

Il est apparu lors des expérimentations que le nombre de solutions potentiellement Pareto optimales identifiées pouvait être largement supérieur à la taille de la population. Une

Algorithme 5.5 *communication*(t : Entier, A_0, A_k : Archive)

```

si  $t \bmod Migration_t = 0$  alors
  communiquer( $A_0, I_{k+1}^i, I_k^{i-1}, I_k^{i+1}$ )
  communiquer( $A_k, I_k^{i-1}, I_k^{i+1}$ )
fin si

```

Algorithme 5.6 *phase_de_recombinaison*(P, Q : POPULATION)

```

 $Q \leftarrow \emptyset$ 
pour  $i \leftarrow 1, \dots, N$  faire
   $pa_1 \leftarrow \text{tournoi}(P \cup C)$ 
   $pa_2 \leftarrow \text{tournoi}(P \cup C)$ 
  si  $\text{rand}() < 0.5$  alors
     $s \leftarrow RBX(pa_1, pa_2)$ 
  sinon
     $s \leftarrow SPLIT(pa_1, pa_2)$ 
  fin si
  si  $\text{rand}() < 0.4$  alors
     $s \leftarrow \text{or\_opt}(s)$ 
  fin si
   $\text{recherche\_locale\_2opt}(s)$ 
   $Q \leftarrow Q \cup \{s\}$ 
fin pour

```

archive de grande capacité a donc été ajoutée à NSGA II pour sauvegarder les solutions potentiellement Pareto optimales identifiées. On tente d'inclure dans l'archive toute nouvelle solution créée.

Deux critères d'arrêt différents ont été utilisés lors des expérimentations. Pour tester l'apport du parallélisme et de la diversification élitiste, le critère d'arrêt est réalisé lorsqu'un nombre fixé de générations est atteint. Dans certaines expériences, le critère d'arrêt est lié à l'archive. Ainsi, si aucune inclusion dans l'archive n'est réalisée durant 5000 générations consécutives, l'algorithme s'arrête.

5.5 Opérateurs génétiques et heuristique

Dans les différents algorithmes, trois opérateurs de croisement sont utilisés ainsi qu'un opérateur de mutation. Dans cette section, nous expliquons aussi l'intérêt de la recherche locale qui est appliquée sur les tournées des solutions.

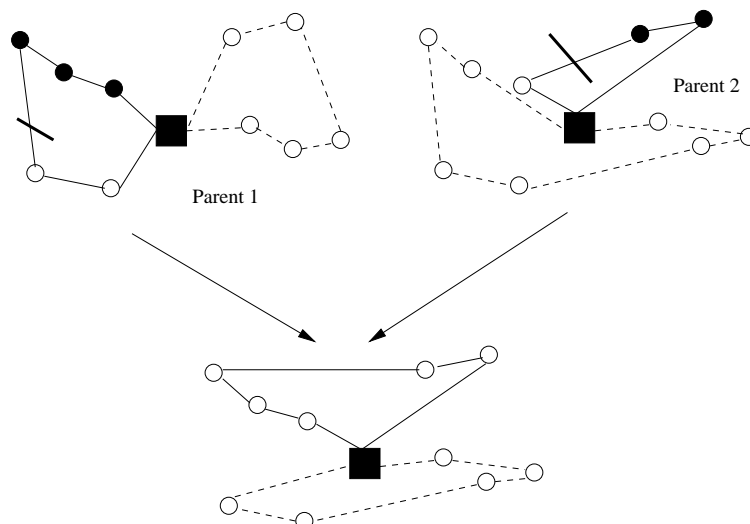


FIG. 5.5 – Un exemple d'application de SBX.

5.5.1 Opérateurs génétiques

5.5.1.1 Opérateur de croisement RBX

Le crossover RBX (pour *route based crossover*) [122] peut être décrit de la manière suivante : certaines tournées du premier parent choisies aléatoirement sont gardées. Ensuite, l'enfant est complété avec les tournées du second parent desquelles les clients déjà visités ont été retirés. Cet opérateur est utilisé pour favoriser l'objectif d'équilibrage des tournées. En effet, il travaille au niveau des tournées et non pas au niveau des sommets.

5.5.1.2 Opérateur de croisement SBX

Dans l'opérateur SBX (*sequence based crossover*) [122], un arc choisi aléatoirement est retiré de chaque parent. Les clients de la tournée qui se trouvent avant l'arc retiré du premier parent sont recopiés dans l'enfant. Ensuite, les clients de la tournée après l'arc retiré dans le second parent sont ajoutés à la suite de ceux du premier parent. Finalement, les autres tournées du premier parent sont ajoutées dans l'enfant. Les clients déjà présents dans la nouvelle solution ne sont pas pris en compte. Un exemple d'application de cet opérateur est donné dans la figure 5.5.

5.5.1.3 Opérateur de croisement SPLIT

Dans un article récent, Prins a proposé un algorithme génétique pour la résolution du PTV avec contraintes de capacité [124]. Chaque solution est codée comme une tournée pour le PVC et transformée en une solution pour le PTV pour la phase d'évaluation par une procédure nommée SPLIT. L'opérateur de croisement SPLIT, présenté ci-dessous, suit les

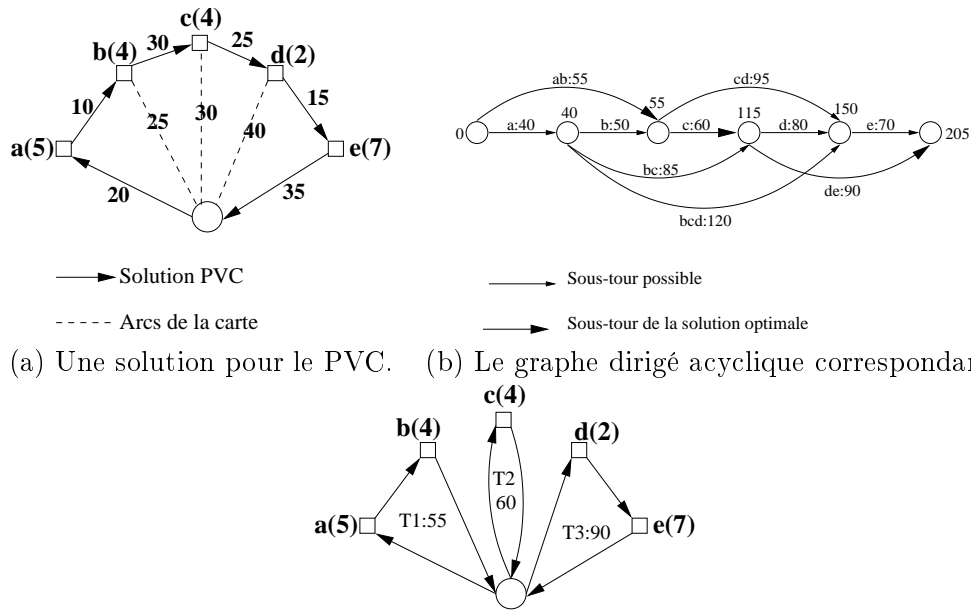


FIG. 5.6 – D’une solution pour le PVC à une solution pour le PTV.

mêmes lignes. Les parents sont considérés comme deux solutions pour le PVC (c’est-à-dire que les arcs sortant du dépôt ou y entrant sont retirés et les tournées sont concaténées). L’opérateur de croisement OX (*Ordered crossover*) pour le PVC est appliqué et une solution pour le PTV est déduite à partir de l’enfant obtenu. Pour ce faire, on construit un graphe dirigé acyclique $D = (V, A)$ avec $V = \{0, 1, \dots, m\}$ et on cherche un plus court chemin. La tournée pour le PVC est représentée par la permutation σ . On a toujours $\sigma(0) = 0$. Les arcs de D représentent les différentes routes faisables. L’ensemble A est défini ainsi : pour tout $i, j \in V$ ($i < j$), l’arc (i, j) appartient à A si et seulement si la charge totale sur la tournée $0\sigma(i+1)\dots\sigma(j)0$ est plus petite ou égale à la capacité maximale du véhicule. La valuation de l’arc (i, j) est alors donnée par la formule :

$$c_{0\sigma(i+1)} + \sum_{l=0}^{j-1} c_{\sigma(i+l)\sigma(i+l+1)} + c_{\sigma(j)0}$$

Trouver la partition optimale de la tournée pour le PVC est équivalent à trouver un plus court chemin dans le graphe dirigé acyclique. Un exemple donné par Prins [124], avec $Q = 10$, est présenté à la figure 5.6.

5.5.1.4 Opérateur de mutation Or-opt

L’opérateur de mutation utilisé est l’opérateur Or-opt [112]. Il consiste à déplacer de un à N clients consécutifs d’une tournée à une autre tournée. Dans notre implémentation, N était fixé à trois. Cet opérateur a deux avantages : i) il est particulièrement bien adapté

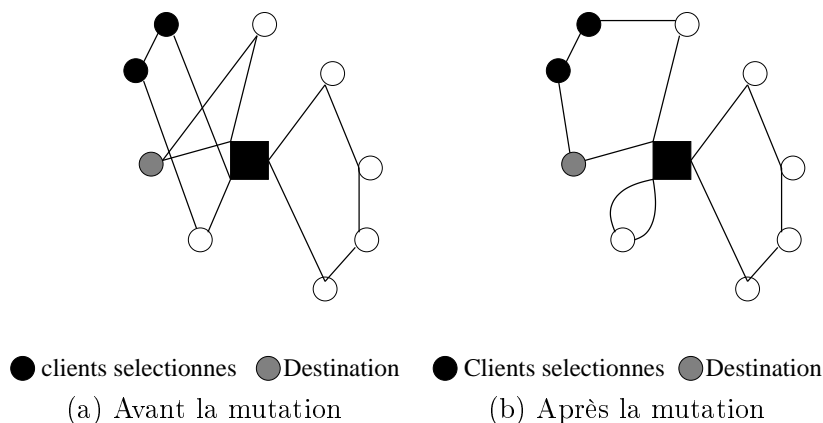


FIG. 5.7 – Un exemple d'application de l'opérateur Or-Opt.

au PTV; ii) il permet la création de nouvelles tournées. Un exemple d'application de l'opérateur est donné dans la figure 5.7.

5.5.2 Recherche locale

Une recherche locale 2-opt [104] est appliquée sur chaque tournée de chaque solution. Le but de cette recherche locale est d'améliorer la longueur de la solution et de répondre à la nécessité d'optimalité des tournées exprimée dans la définition du problème. La recherche locale 2-opt a été retenue car elle offre un bon compromis entre le temps de calcul et la qualité des tournées obtenues sachant que ces dernières comprennent rarement un nombre important de clients et ne sont donc pas difficiles à résoudre.

5.6 Implémentation de l'algorithme Target Aiming Pareto Search

Dans cette section, nous présentons l'implémentation de TAPaS faite pour le PTVET. Dans la section 5.6.1, les principaux mécanismes d'Unified Tabu Search, la recherche Tabou utilisée comme méthode d'optimisation, sont donnés. Puis une adaptation de la méthode pour prendre en compte la présence de deux objectifs est expliquée dans la section 5.6.2. Enfin, des détails sur la phase de clusterisation de TAPaS sont fournis dans la section 5.6.3.

5.6.1 Unified Tabu Search

L'implémentation de TAPaS pour le PTVET repose sur Unified Tabu Search (UTS) [30, 31]. UTS a initialement été conçue pour le PTV avec fenêtres de temps, mais cette méthode obtient aussi de bons résultats sur le PTV avec contraintes de capacité. Les principales caractéristiques de UTS sont :

1. UTS permet de prendre en compte les solutions infaisables durant la recherche. Une solution qui ne respecte pas la contrainte de capacité peut donc être visitée mais son coût $c(s)$ est dégradé selon une règle spécifique. Soit $q(s)$ la violation totale des contraintes de capacité, alors le coût de la solution s devient $f(s) = c(s) + \alpha q(s)$ où α est un paramètre strictement positif ajusté dynamiquement.
2. Le voisinage d'une solution s est défini en utilisant un ensemble d'attributs $B(s) = \{(i, k) \mid \text{le client } i \text{ est visité par le véhicule } k\}$. Le voisinage $N(s)$ est défini en retirant un attribut (i, k) de $B(s)$ et en le remplaçant par un nouvel attribut (i, k') . Quand un client est retiré de la tournée k , cette dernière est réparée en reliant les clients visités avant et après i . L'ajout de i dans la tournée k' se fait de telle sorte que $f(s)$ soit minimisée. Quand un client i est retiré de la tournée k , l'attribut (i, k) est marqué tabou pour les θ itérations suivantes.
3. UTS propose aussi un critère d'aspiration qui permet de révoquer le statut tabou d'un attribut. Cependant, puisque le nombre d'itérations d'une application de UTS est limité et que le critère d'aspiration tend à ralentir la convergence durant les premières itérations, il n'a pas été utilisé dans le cadre de TAPaS.
4. Pour diversifier la recherche, toute solution $s' \in N(s)$ tel que $f(s') \geq f(s)$ est pénalisée de la manière suivante. Soit ρ_{ik} le nombre de fois où l'attribut (i, k) a été ajouté à la solution durant la recherche. Une pénalité $p(s') = \lambda c(s') \sqrt{nm} \sum_{(i,k) \in B(s')} \rho_{ik}$, où λ est une constante positive, est alors ajoutée à $f(s')$.
5. UTS ne requiert qu'un unique paramètre : le nombre maximal d'itérations fixé à 5000.

5.6.2 Adaptation d'UTS

Deux versions d'UTS ont été utilisées. La première, notée UTS-std, implémente UTS telle qu'elle est décrite précédemment. La seconde implémentation, notée UTS-dp, inclut deux pénalités de diversité qui peuvent être formulées de manière générale : $\lambda r(s) \sqrt{nm} \sum_{(i,k) \in B(s)} \rho_{ik}$. La première pénalité, p_l , porte sur la longueur de la solution et dans ce cas on a $r(s)$ qui est égale à la longueur de la solution. La seconde pénalité, p_b , porte sur l'équilibre et dans ce cas $r(s)$ est égale à la différence entre la longueur de la plus longue tournée et la longueur de la plus courte tournée. Le choix de la pénalité à appliquer pour une recherche Tabou donnée dépend de la position de la solution de départ dans l'ensemble potentiellement Pareto optimal courant. Dans notre implémentation, la recherche Tabou qui a s comme solution de départ utilise la pénalité p_l avec la probabilité suivante :

$$\frac{1}{2} \times \left(1 - \frac{f_1(s) - f_1^{\min}}{f_1^{\max} - f_1^{\min}}\right) + \frac{1}{2} \times \frac{f_2(s) - f_2^{\min}}{f_2^{\max} - f_2^{\min}} \quad (5.18)$$

où f_1^{\min} (respectivement f_2^{\min}) est la meilleure valeur trouvée courante pour le premier objectif (respectivement le second) et f_1^{\max} (respectivement f_2^{\max}) est la plus mauvaise valeur dans l'approximation courante. Ces pénalités ont été introduites pour aider UTS à s'échapper d'optima locaux pour les solutions qui se trouvent près de la meilleure solution pour l'objectif d'équilibre et pour lesquelles une pénalité basée sur la longueur n'est pas toujours significative.

Ces pénalités ont aussi une influence sur le mécanisme tabou empêchant d'effectuer des calculs inutiles. En effet, si pour une solution de départ s et un but g , une recherche a déjà été effectuée, une nouvelle recherche est cependant toujours possible avec s comme solution de départ et g comme but mais avec la pénalité qui n'a pas encore été appliquée. Si les deux pénalités ont déjà été utilisées dans des recherches partant de s et ayant pour but g , toutes nouvelles recherches de ce type sont interdites.

5.6.3 Phase de clustering

Pour choisir les solutions de départ de l'itération suivante, l'algorithme *average linking method* [109] est appliqué sur l'approximation courante. Dix solutions sont choisies lors de cette phase.

5.7 Expérimentation

Trois points ont été évalués : l'apport de la diversification élitiste dans le cadre de NEMO (5.7.1), l'apport de la parallélisation et de la diversification élitiste dans le cadre de NSGA II (5.7.2) et les performances de TAPaS (5.7.3).

5.7.1 Evaluation de NEMO

5.7.1.1 Protocole

L'algorithme génétique a été codé en C/MPI. Les expérimentations ont été conduites sur un IBM RS6000/SP équipé de processeurs Power4 1.1Ghz. Huit processeurs ont pu être utilisés. Une brique du modèle a été implémentée en un seul processus.

Les évaluations ont été faites sur les jeux de données standard de Christofides et Eilon [22] et de Christofides et al. [23] qui correspondent au PTV avec contraintes de capacité. Chaque instance a été résolue cinq fois.

L'algorithme génétique a été paramétré expérimentalement. Les paramètres utilisés sont les suivants :

Taille de la population : $N = 256$.

Taille maximale de l'archive : $archive_max_size = 50$.

Taille maximale de l'archive après clustering : $M = 20$.

Pression du sharing : $S = 1.7$.

Probabilité d'appliquer l'opérateur RBX : $prob_{rbx} = 0.5$.

Probabilité de mutation : $prob_{mutation} = 0.4$.

Nombre de briques (implémentation parallèle) : 8.

Fréquence des migrations : $Migration_t = 250$.

TAB. 5.2 – Apport de la diversification élitiste : $S(ED) - S(NED)$.

Instance	Meilleur	Moyenne	Pire	écart type
E51-05e	0.01597	0.00729	-0.00013	0.00415
E76-10e	0.03301	0.02193	0.00505	0.00710
E101-08e	0.04291	0.03249	0.01506	0.00787
E151-12c	0.09167	0.06603	0.04362	0.01196
E200-17c	0.09281	0.06157	0.03746	0.01311
E121-07c	0.03051	0.02300	0.01588	0.00426
E101-10c	0.05898	0.04126	0.02252	0.00926

Le critère d'arrêt de l'algorithme est lié au nombre de générations maximum qu'il peut effectuer. Trois implémentations ont été testées : l'algorithme génétique sans diversification élitiste (NED), l'algorithme génétique multi-objectif avec diversification élitiste (ED) et l'algorithme génétique multi-objectif avec diversification élitiste parallèle (PAR). NED est composé d'une seule île et ED de deux îles. Pour assurer que les deux algorithmes génèrent le même nombre de solutions, le nombre maximum de générations était fixé à 100000 pour NED et 50000 pour ED. Pour assurer que ED et PAR utilisent le même temps de calcul hors communication, le nombre maximum de générations pour PAR est aussi 50000.

5.7.1.2 Apport de la diversification élitiste à NEMO

Nous avons d'abord évalué l'intérêt du mécanisme de diversification élitiste. Pour cela, nous avons utilisé la métrique \mathcal{S} pour évaluer l'apport du mécanisme. Les résultats sont reportés dans le tableau 5.2. La version avec le mécanisme de diversification élitiste est toujours capable d'améliorer la qualité des approximations sauf pour certaines exécutions sur l'instance E51-05e. Du fait qu'il s'agit de l'instance la plus facile, *NED* était déjà capable de trouver de bonnes solutions sur les deux objectifs et notamment par rapport à l'objectif de longueur.

La technique de diversification élitiste a été conçue pour améliorer la diversité de l'approximation générée. Pour évaluer son impact sur la convergence de la méta-heuristique, nous avons utilisé la mesure de couverture \mathcal{C} [172]. Il apparaît (cf. le tableau 5.3) que non seulement la diversification élitiste ne ralentit pas la convergence mais elle peut l'améliorer.

5.7.1.3 Apport du modèle parallèle

Les mêmes métriques ont été utilisées pour évaluer l'apport du modèle parallèle. Les résultats montrent qu'il permet une amélioration des approximations générées tant au niveau de la diversification que de la convergence (cf. tableaux 5.4 et 5.5).

TAB. 5.3 – Résultats de la mesure de couverture $C(ED, NED)$ (première entrée par cellule) et $C(NED, ED)$ (seconde entrée par cellule).

Instance	Meilleur	Moyenne	Pire	écart type
E51-05e	0.59	0.42	0.12	0.12
	0.45	0.15	0.00	0.14
E76-10e	0.89	0.52	0.26	0.17
	0.44	0.21	0.00	0.12
E101-08e	1.00	0.94	0.69	0.09
	0.31	0.06	0.00	0.09
E151-12c	1.00	0.99	0.87	0.02
	0.07	0.01	0.00	0.01
E200-17c	1.00	0.88	0.45	0.18
	0.25	0.04	0.00	0.07
E121-07c	1.00	0.97	0.89	0.03
	0.05	0.01	0.00	0.02
E101-10c	0.97	0.93	0.79	0.04
	0.04	0.01	0.00	0.01

TAB. 5.4 – Apport du modèle en îles de la diversification élitiste : $S(PAR) - S(ED)$.

Instance	Meilleur	Moyenne	Pire	Ecart type
E51-05e	0.01045	0.00531	-0.00267	0.00267
E76-10e	0.02071	0.00858	0.00014	0.00553
E101-08e	0.01437	0.00688	0.00114	0.00359
E151-12c	0.03270	0.02061	-0.00241	0.01004
E200-17c	0.05525	0.03123	0.00621	0.01234
E121-07c	0.01399	0.09198	0.00430	0.00256
E101-10c	0.02966	0.01633	0.00421	0.00654

5.7.1.4 Remarques complémentaires

Pour les différentes implémentations, les temps de calcul moyens en minutes sont reportés dans le tableau 5.6. Il apparaît que la maintenance des archives supplémentaires et les communications entre les îles n'entraînent pas une hausse significative du temps de calcul nécessaire.

Puisque les ensembles Pareto optimaux ne sont pas connus, nous avons observé les valeurs des objectifs pour les solutions qui forment les extrémités de l'approximation. Le tableau 5.7 reporte la meilleure valeur connue pour la longueur totale, la meilleure valeur trouvée pour la longueur totale et la valeur du second objectif qui lui est associée, la meilleure valeur trouvée pour l'équilibre et la longueur totale associée. Le tableau indique aussi le nombre moyen de solutions dans l'approximation générée. Même si l'algorithme génétique n'a pas été spécialement conçu pour traiter l'objectif de longueur totale, les valeurs des solutions identifiées sont proches des meilleures valeurs connues. Aucune meilleure solution n'est connue pour le second objectif, mais les meilleures valeurs trouvées varient entre 0.10 et 1.38. On peut donc en déduire que les solutions trouvées pour cet objectif

TAB. 5.5 – Résultats de la mesure de couverture C(PAR, ED) (première entrée par cellule) et C(ED, PAR) (seconde entrée par cellule).

Instance	Meilleur	Moyenne	Pire	Ecart type
E51-05e	0.66	0.50	0.25	0.10
	0.34	0.07	0.00	0.09
E76-10e	1.00	0.82	0.43	0.15
	0.61	0.19	0.00	0.21
E101-08e	0.97	0.80	0.48	0.13
	0.46	0.16	0.00	0.13
E151-12c	1.00	0.94	0.42	0.12
	0.55	0.04	0.00	0.11
E200-17c	1.00	0.75	0.43	0.27
	0.42	0.17	0.00	0.21
E121-07c	0.91	0.77	0.62	0.07
	0.24	0.11	0.02	0.06
E101-10c	0.86	0.79	0.68	0.05
	0.15	0.05	0.01	0.04

TAB. 5.6 – Temps de calcul moyen en minutes.

Instance	E51-05e	E76-10e	E101-08e	E151-12c	E200-17c	E121-07c	E101-10c
NED	10.10	25.10	34.37	63.13	81.39	38.57	34.41
ED	10.09	25.24	34.59	63.40	81.46	39.05	34.58
PAR	10.22	25.38	35.22	65.60	82.20	40.30	35.43

sont de bonne qualité puisqu'une borne inférieure évidente est zéro. Enfin, le nombre important de solutions pour chaque approximation ainsi que la différence entre les valeurs des extrémités de l'approximation tendent à indiquer que les deux objectifs ne sont pas fortement corrélés.

5.7.2 Evaluation de NSGA II

5.7.2.1 Protocole

NSGA II, le modèle parallèle pNSGA II, la variante avec diversification élitiste NSGAED et la version parallèle de NSGAED, notée pNSGAED, ont été codés en C et MPI a été utilisé pour la partie parallèle. Les tests ont été effectués sur un IBM RS6000/SP équipé de processeurs Power4 1.1 Ghz.

Les évaluations ont été faites sur les jeux de données de Christofides et Eilon [22] et Christofides et al. [23] pour le PTV avec contraintes de capacité. Chaque instance a été résolue dix fois par chaque méthode.

NSGA II et pNSGA II s'arrêtent après 100000 générations et NSGAED et pNSGAED après 50000. Ainsi, on assure que toutes les méthodes génèrent le même nombre de solutions à nombre de processeurs égal. Dans les implémentations utilisant la diversification élitiste

TAB. 5.7 – Meilleures solutions trouvées par l’algorithme génétique multi-objectif pour les deux objectifs.

Instance	NB	Longueur totale			Equilibre		
		Meilleur connu	Meilleur trouvé	%	Equilibre	Meilleur trouvé	Longueur
E51-05e	49.8	524.61	524.61	0.00	20.07	0.24	618.22
E76-10e	108	835.26	835.32	0.01	78.10	0.59	1203.98
E101-08e	154	826.14	827.39	0.15	67.55	0.29	1871.06
E151-12c	242.4	1028.42	1047.35	1.84	74.78	0.80	1484.48
E200-17c	243	1291.45	1352.46	4.72	76.60	1.38	1902.64
E121-07c	741.8	1042.11	1042.11	0.00	146.67	0.10	2388.30
E101-10c	348	819.56	819.56	0.00	93.43	1.15	1429.90

TAB. 5.8 – Valeurs moyennes et écarts types de la mesure \mathcal{S} pour NSGA II en fonction du nombre de processeurs utilisés.

Instance		1 proc.	4 proc.	8 proc.	16 proc.
E51-05e	Moyenne	0.511232	0.527863	0.527733	0.530235
	écart type	0.006132	0.001838	0.004329	0.001987
E76-10e	Moyenne	0.414035	0.420253	0.425498	0.426979
	écart type	0.002988	0.001714	0.002892	0.002052
E101-08e	Moyenne	0.570935	0.576901	0.577431	0.579026
	écart type	0.001779	0.001638	0.000724	0.000418
E151-12c	Moyenne	0.618357	0.631726	0.634581	0.637956
	écart type	0.006315	0.001637	0.003460	0.001426
E200-17c	Moyenne	0.607886	0.628112	0.632612	0.639964
	écart type	0.014343	0.005537	0.008276	0.002474
E121-07c	Moyenne	0.516538	0.526248	0.527154	0.527934
	écart type	0.007145	0.001480	0.001405	0.000637
E101-10c	Moyenne	0.584904	0.620338	0.627408	0.629321
	écart type	0.018182	0.004675	0.003061	0.002398

TAB. 5.9 – Temps de calcul moyens en secondes de NSGA II en fonction du nombre de processeurs.

Instance	E51-05e	E76-10e	E101-08e	E151-12c	E200-17c	E121-07c	E101-10c
1 proc.							
4 proc.	993.4	1453.4	2451.3	4082.1	4996.3	4615.1	2640.1
8 proc.	937.8	1300.8	2406.1	3621.9	4463.7	4791.1	2425.3
16 proc.	1080.7	1329.0	2289.6	3794.5	4677.7	5171.1	2451.6

quinze solutions étaient choisies dans les archives à chaque génération.

5.7.2.2 Apport de la parallélisation pour NSGA II

Nous avons évalué si l'utilisation du parallélisme avait un impact sur les performances en faisant varier le nombre de processeurs utilisés : 1, 4, 8 ou 16. Le tableau 5.8 reporte les valeurs moyennes et l'écart type de la métrique \mathcal{S} pour les différents cas. Il apparaît qu'utiliser 4 processeurs améliore sur la plupart des instances les résultats de manière significative. Utiliser plus de processeurs apporte aussi une amélioration des résultats mais dans une moindre mesure. Par contre, il apparaît en observant l'évolution des écarts types qu'en général plus il y a de processeurs plus les résultats sont robustes.

L'impact des communications sur les temps de calcul a aussi été évalué. Le tableau 5.9 reporte les temps moyens d'exécution en secondes en fonction du nombre de processeurs. Il apparaît que les communications n'influent pas sur les temps.

TAB. 5.10 – Valeurs moyennes et écarts types de la mesure \mathcal{S} pour NSGA II avec et sans diversification élitiste.

Instance		1 processeur		8 processeurs	
		NSGA II	NSGAED	pNSGA II	pNSGAED
E51-05e	Moyenne	0.511232	0.521232	0.527733	0.529467
	écart type	0.006132	0.004139	0.004329	0.001282
E76-10e	Moyenne	0.414035	0.415599	0.425498	0.425809
	écart type	0.002988	0.003651	0.002892	0.002992
E101-08e	Moyenne	0.570935	0.573612	0.577431	0.577501
	écart type	0.001779	0.001800	0.000724	0.001430
E151-12c	Moyenne	0.618357	0.619450	0.634581	0.635170
	écart type	0.006315	0.007012	0.003460	0.003016
E200-17c	Moyenne	0.607886	0.617594	0.632612	0.643165
	écart type	0.014343	0.006185	0.008276	0.004848
E121-07c	Moyenne	0.516538	0.518553	0.527154	0.527442
	écart type	0.007145	0.007998	0.001405	0.000478
E101-10c	Moyenne	0.584904	0.602430	0.627408	0.629226
	écart type	0.018182	0.020408	0.003061	0.003343

5.7.2.3 Apport de la diversification élitiste pour NSGA II

Nous avons évalué la version avec diversification élitiste par rapport à celle sans diversification élitiste. Nous avons exécuté NSGAED avec une seule brique élémentaire et avec huit briques élémentaires. Nous les avons comparés à NSGA II et à pNSGA II avec huit processeurs. Les valeurs moyennes et les écarts types de la métrique \mathcal{S} sont reportés dans le tableau 5.10.

Il apparaît que la diversification élitiste est toujours capable d'améliorer les résultats de NSGA II lorsqu'un seul processeur est utilisé. L'apport est le plus important sur les instances de grande taille. Ce fait est notamment visible sur l'instance E200-17c. Nous avons aussi comparé l'apport de la diversification élitiste lorsque des modèles parallèles avec huit îles étaient utilisés. Sur les instances de petite taille, l'apport est moindre car l'apport du parallélisme était déjà important comme déjà remarqué précédemment. Ce phénomène est particulièrement visible sur l'instance E121-07c. Par contre, l'apport est toujours important sur l'instance de plus grande taille.

Le tableau 5.11 reporte les meilleures solutions trouvées en termes de longueur avec l'équilibre associé (première ligne des cellules), les meilleures solutions trouvées pour l'équilibre avec la longueur associée (seconde ligne des cellules), ainsi que le nombre de solutions potentiellement Pareto optimales trouvées (troisième ligne des cellules). Il apparaît, qu'à nombre égal de processeurs, les implémentations incluant la diversification élitiste sont capables d'améliorer les extrémités de l'approximation générée. D'autre part, ces bornes sont très proches des meilleures solutions connues en termes de longueur et les bornes pour l'équilibre ne sont pas loin de la limite évidente zéro. Il apparaît donc que les solutions sont très équilibrées.

TAB. 5.11 – Meilleures solutions trouvées pour chaque objectif et valeurs associées du second objectif pour chaque implémentation de NSGA II.

Instance	NSGA II								NSGAED							
	1 proc.		4 proc.		8 proc.		16 proc.		1 proc.		1 proc.		8 proc.			
E51-05e	524.61	20.07	524.61	20.07	524.61	20.07	524.61	20.07	524.61	20.07	524.61	20.07	524.61	20.07		
	0.51	694.79	0.17	648.80	0.23	643.89	0.18	779.88	0.48	1093.31	0.32	645.29	44.8	44.1		
E76-10 e	835.32	78.10	842.06	67.55	835.32	78.10	835.32	78.10	835.89	85.60	835.32	78.10	1.01	1268.02		
	1.01	1268.02	1.10	1117.29	0.64	997.97	0.48	1153.25	0.85	1110.78	0.33	1380.01	109.7	124.7		
E101-08e	827.39	67.55	827.39	67.55	827.39	67.55	827.39	67.55	827.39	67.55	827.29	67.55	0.64	1596.59		
	0.64	1596.59	0.09	1695.55	0.35	984.76	0.27	1132.28	0.43	1129.10	0.10	1610.86	129.4	161.1		
E151-12c	1047.02	87.10	1044.69	91.14	1043.83	77.89	1032.95	97.57	1043.71	87.67	1034.91	109.06	1.22	2290.53		
	1.22	2290.53	0.56	1443.31	0.51	1382.28	0.44	1551.50	0.66	2523.00	0.21	1523.94	204.1	310.10		
E200-17c	1358.08	68.77	1349.48	79.14	1332.82	87.50	1328.44	96.61	1353.17	87.84	1327.89	97.20	3.04	1838.32		
	3.04	1838.32	2.17	3006.26	1.85	1976.10	0.86	3490.14	2.18	3436.43	0.58	3746.57	206.6	316.4		
E121-07c	1043.78	146.67	1042.11	146.67	1042.11	146.67	1042.11	146.67	1043.11	147.27	1042.11	146.67	0.15	2258.21		
	0.15	2258.21	0.06	2314.61	0.06	2348.74	0.07	2296.43	0.06	2285.22	0.07	1649.76	441.5	847.40		
E101-10c	819.56	93.43	819.56	93.43	819.56	93.43	819.56	93.43	819.56	93.43	819.56	94.43	1.16	2035.80		
	1.16	2035.80	0.77	2107.14	0.65	1425.43	0.21	1324.53	1.10	1303.67	0.35	1664.80	269.1	441.4		

TAB. 5.12 – Efficacité de TAPaS-std : $D(\text{TAPaS-std}, \text{NSGA II})$.

Instance	Meilleur	Moyenne	Pire	Ecart type
E51-05e	1.818×10^{-2}	$- 1.27 \times 10^{-3}$	$- 2.784 \times 10^{-2}$	1.003×10^{-2}
E76-10e	1.720×10^{-2}	$- 1.655 \times 10^{-2}$	$- 3.884 \times 10^{-2}$	1.449×10^{-2}
E101-08e	$- 7.32 \times 10^{-3}$	$- 2.662 \times 10^{-2}$	$- 7.532 \times 10^{-2}$	1.790×10^{-2}
E151-12c	3.612×10^{-2}	1.421×10^{-2}	$- 4.462 \times 10^{-2}$	1.865×10^{-2}
E200-17c	3.335×10^{-2}	1.154×10^{-2}	$- 5.867 \times 10^{-2}$	2.802×10^{-2}
E121-07c	6.23×10^{-3}	$- 2.12 \times 10^{-3}$	$- 2.638 \times 10^{-2}$	8.27×10^{-3}
E101-10c	7.235×10^{-2}	7.9×10^{-4}	$- 1.163 \times 10^{-2}$	4.927×10^{-2}

5.7.3 Evaluation de Target Aiming Pareto Search

5.7.3.1 Protocole

TAPaS pour le PTVET a été implémenté en C. Les évaluations sont réalisées sur les jeux de données standard de Christofides et Eilon [22] et Christofides et al. [23] qui correspondent à des instances avec contraintes de capacité. Les ensembles utilisés pour l’hybridation ont été générés par NSGA II. Dix exécutions ont été faites par jeu de données. Nous avons utilisé la mesure \mathcal{S} de Zitzler [172]. Dans les faits, nous avons calculé pour deux algorithmes A et B la valeur $D(A, B) = S(A) - S(B)$. Il est à noter que les valeurs des objectifs ont été normalisées par rapport au point de référence utilisé dans la métrique \mathcal{S} . Donc, si $D(A, B)$ est plus grand que zéro, cela signifie que l’algorithme A est meilleur que l’algorithme B , sinon l’algorithme B est meilleur que A . Nous avons aussi utilisé la métrique \mathcal{C} [172]. Dans les tableaux de résultats, nous reportons les valeurs moyennes, maximales et minimales ainsi que les écarts types. Dans la suite de cette partie, TAPaS-std signifie que la recherche Tabou utilisée est UTS-std et TAPaS-dp signifie qu’on utilise UTS-dp.

5.7.3.2 Efficacité de TAPaS

Dans un premier temps, nous avons évalué l’efficacité de TAPaS-std et TAPaS-dp par rapport à NSGA II. Une population est générée en utilisant une méthode gloutonne. Cette population forme la population de départ de NSGA II tandis que les individus non-dominés de la population sont les points de départ de TAPaS-std et TAPaS-dp. Les résultats de ces comparaisons sont reportés dans les tableaux 5.12 et 5.13. Il apparaît que TAPaS-dp est meilleur en moyenne que NSGA II quatre fois sur sept. De son côté, TAPaS-std est meilleur que NSGA II en moyenne que trois fois sur sept. TAPaS-std trouve toujours des approximations de moins bonne qualité que celles identifiées par NSGA II sur l’instance E101-08e tandis que ce n’est jamais le cas pour TAPaS-dp. Il apparaît donc que la pénalité de diversité adaptative est utile. De manière générale, TAPaS-dp semble être capable de produire des ensembles de bonne qualité. Il est à noter que le seul mécanisme dans notre implémentation de TAPaS qui favorise la diversification est la liste tabou de UTS. Cette liste permet d’"osciller" autour des optima locaux trouvés par la recherche Tabou.

D’autre part, si on évalue la qualité des ensembles potentiellement Pareto optimaux par rapport à la convergence vers la frontière Pareto à l’aide de la métrique \mathcal{C} (tableaux

TAB. 5.13 – Efficacité de TAPaS-dp : D(TAPaS-dp, NSGA II).

Instance	Meilleur	Moyenne	Pire	Ecart type
E51-05e	9.73×10^{-3}	1.38×10^{-3}	- 4.24×10^{-3}	3.39×10^{-3}
E76-10e	9.48×10^{-3}	- 1.156×10^{-2}	- 2.597×10^{-2}	9.28×10^{-3}
E101-08e	7.38×10^{-3}	2.64×10^{-3}	- 2.37×10^{-3}	2.66×10^{-3}
E151-12c	3.878×10^{-2}	9.50×10^{-3}	- 3.12×10^{-3}	1.098×10^{-2}
E200-17c	2.671×10^{-2}	3.84×10^{-3}	- 1.143×10^{-2}	1.106×10^{-2}
E121-07c	3.45×10^{-3}	- 2.54×10^{-3}	- 9.67×10^{-3}	3.56×10^{-3}
E101-10c	2.141×10^{-2}	- 3.419×10^{-2}	- 7.489×10^{-2}	2.713×10^{-2}

TAB. 5.14 – Résultats des mesures C(TAPaS-std, NSGA II) (première entrée par cellule) et C(NSGA II, TAPaS-std) (seconde entrée par cellule).

Instance	Meilleur	Moyenne	Pire	Ecart type
E51-05e	1.00	0.46	0.16	0.24
	0.73	0.28	0.00	0.17
E76-10e	0.83	0.35	0.10	0.19
	0.62	0.27	0.00	0.14
E101-08e	0.78	0.42	0.02	0.22
	0.93	0.30	0.03	0.19
E151-12c	1.00	0.79	0.20	0.27
	0.33	0.03	0.00	0.07
E200-17c	0.99	0.55	0.00	0.30
	0.27	0.10	0.00	0.07
E121-07c	0.67	0.45	0.18	0.10
	0.40	0.25	0.08	0.08
E101-10c	0.55	0.30	0.11	0.11
	0.69	0.50	0.37	0.08

5.14 et 5.15), TAPaS est toujours meilleur que NSGA II. On peut donc en conclure que TAPaS est capable de converger plus loin que NSGA II alors que ce dernier est capable de générer des ensembles de solutions plus diversifiés. Cette observation ne fait que confirmer le but de TAPaS qui est d'améliorer la convergence d'ensembles potentiellement Pareto optimaux bien diversifiés. La figure 5.8 montre deux exemples d'ensembles générés sur les instances E151-12c et E200-17c où TAPaS est meilleur que NSGA II.

5.7.3.3 Efficacité de TAPaS comme méthode de post-optimisation

Nous avons aussi testé l'efficacité de TAPaS quand il est appliqué dans le but d'améliorer des approximations générées par une autre méthode. Dans les tests effectués, les ensembles générés par NSGA II durant la première série d'expérimentations ont été utilisés comme ensembles de départ pour TAPaS. Le tableau 5.16 reporte les apports de TAPaS sur les différents jeux de données. Le tableau montre que les deux implémentations sont capables d'améliorer les résultats et cela même sur les instances où NSGA II était meilleur

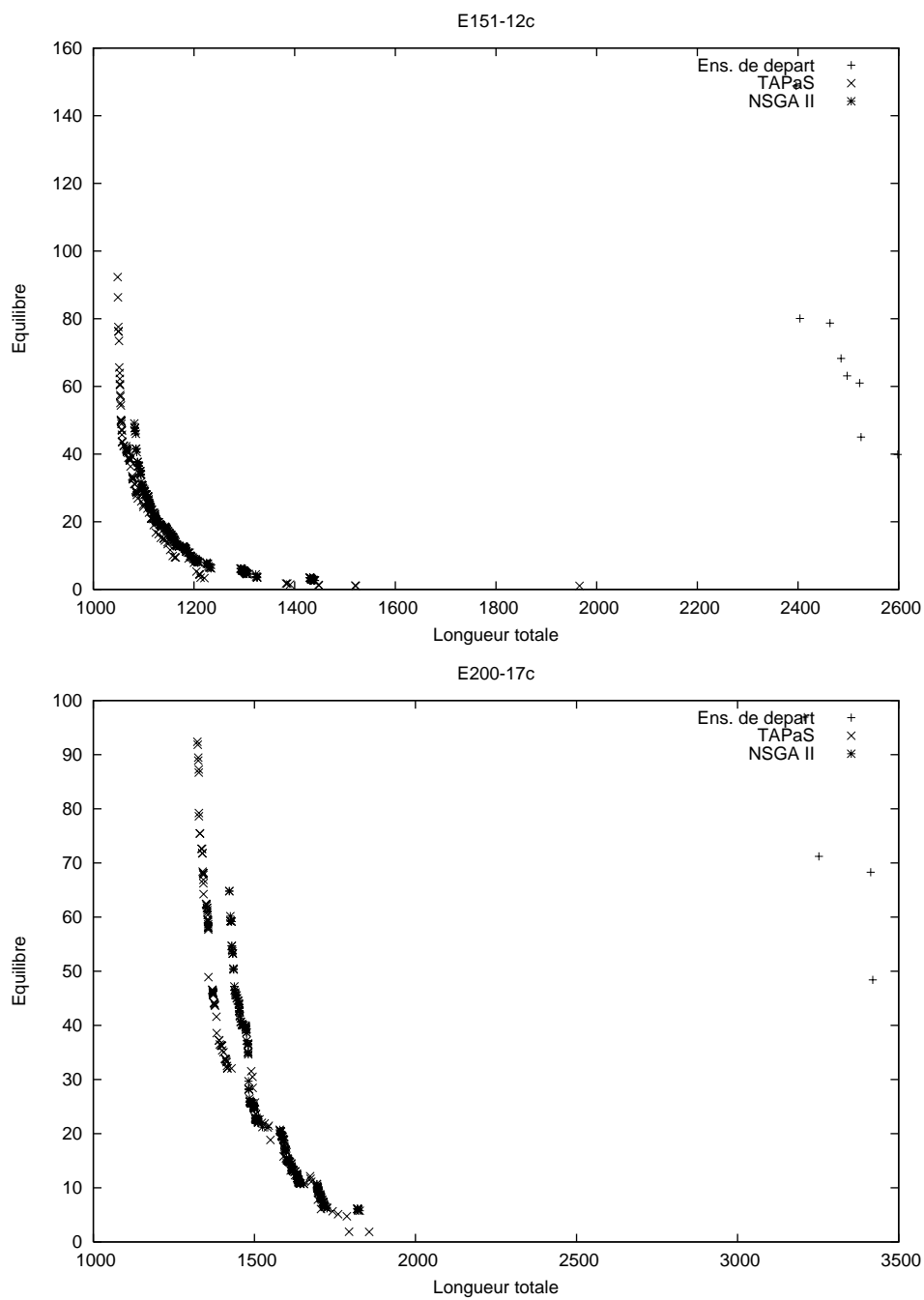


FIG. 5.8 – Deux exemples où TAPaS est meilleur que NSGA II.

TAB. 5.15 – Résultats des mesures \mathcal{C} (TAPaS-dp, NSGA II) (première entrée par cellule) et \mathcal{C} (NSGA II, TAPaS-dp) (seconde entrée par cellule).

Instance	Meilleur	Moyenne	Pire	Ecart type
E51-05e	0.90	0.39	0.09	0.22
	0.76	0.34	0.00	0.19
E76-10e	0.81	0.45	0.02	0.23
	0.74	0.37	0.02	0.21
E101-08e	1.00	0.88	0.64	0.08
	0.28	0.10	0.00	0.08
E151-12c	1.00	0.75	0.48	0.11
	0.10	0.02	0.00	0.03
E200-17c	0.76	0.45	0.21	0.15
	0.09	0.03	0.00	0.02
E121-07c	0.76	0.61	0.37	0.09
	0.23	0.13	0.06	0.04
E101-10c	0.54	0.26	0.05	0.13
	0.77	0.42	0.17	0.21

que TAPaS. Le tableau 5.17 donne le ratio de solutions de l'ensemble généré par NSGA II amélioré par TAPaS. A part pour les instances E51-05e et E101-10c qui sont les instances les plus faciles des jeux de données et pour lesquelles NSGA II était déjà capable de générer des solutions de bonne qualité, TAPaS est capable d'améliorer, en moyenne, plus de cinquante pour cent des solutions des approximations. Enfin, les deux mesures montrent que TAPaS-dp est toujours meilleur que TAPaS-std, ce qui confirme que la pénalité de diversification adaptative est un mécanisme intéressant.

5.7.3.4 Intérêt de l'hybridation

Si on compare la mesure \mathcal{C} pour TAPaS-dp dans les tableaux 5.15 et 5.17, il apparaît que, si elles sont sensiblement identiques pour l'instance E101-08e, les valeurs sont meilleures dans le cas où TAPaS-dp est utilisé pour hybrider NSGA II. Le tableau 5.18 présente les mesures D entre les approximations générées par TAPaS-dp en partant des ensembles fournis par NSGA II et les approximations de TAPaS-dp utilisé seul. Il apparaît qu'en moyenne les approximations issues de l'hybridation sont de meilleure qualité et cela même sur les instances où TAPaS-dp était en moyenne meilleur que NSGA II. Ces résultats montrent donc clairement l'intérêt d'une telle hybridation.

5.7.4 Etudes des bornes

Nous allons maintenant étudier les bornes obtenues par nos méthodes sur les deux objectifs. Le tableau 5.19 reporte pour différents algorithmes les meilleures valeurs trouvées par chacun des objectifs ainsi que la valeur de l'autre objectif qui lui est associée. Les méthodes sont les implémentations les plus efficaces des différents algorithmes que nous avons proposés, c'est-à-dire, la troisième version de NEMO parallélisée, le modèle parallèle

TAB. 5.16 – Apport de l’hybridation : D(TAPaS-std, NSGA II) (première entrée par cellule) et D(TAPaS-dp, NSGA II) (seconde entrée par cellule).

Instance	Meilleur	Moyenne	Pire	Ecart type
E51-05e	9.63×10^{-3}	4.47×10^{-3}	1.3×10^{-4}	3.26×10^{-3}
	1.198×10^{-2}	4.98×10^{-3}	1.21×10^{-3}	3.05×10^{-3}
E76-10e	1.757×10^{-2}	9.08×10^{-3}	6.38×10^{-3}	4.28×10^{-3}
	1.913×10^{-2}	9.27×10^{-3}	9.26×10^{-3}	1.36×10^{-3}
E101-08e	5.62×10^{-3}	4.55×10^{-3}	2.69×10^{-3}	1.20×10^{-3}
	7.79×10^{-3}	4.91×10^{-3}	2.55×10^{-3}	1.47×10^{-3}
E151-12c	2.567×10^{-2}	1.730×10^{-2}	1.087×10^{-2}	5.01×10^{-3}
	2.668×10^{-2}	2.068×10^{-2}	1.398×10^{-2}	3.08×10^{-3}
E200-17c	4.887×10^{-2}	3.886×10^{-2}	3.195×10^{-2}	5.69×10^{-3}
	5.669×10^{-2}	4.514×10^{-2}	3.654×10^{-2}	6.49×10^{-3}
E121-07c	8.36×10^{-3}	4.14×10^{-3}	1.03×10^{-3}	2.51×10^{-3}
	1.057×10^{-2}	4.86×10^{-3}	9.4×10^{-4}	2.10×10^{-3}
E101-10c	7.62×10^{-3}	1.72×10^{-3}	1×10^{-5}	2.95×10^{-3}
	1.519×10^{-2}	5.56×10^{-3}	1.13×10^{-3}	4.82×10^{-3}

de NSGA II exécuté sur 16 processeurs, NSGA II avec la diversification élitiste et exécuté sur huit processeurs et le modèle coopératif composé de NSGA II et de TAPaS.

Ces comparaisons sont intéressantes car comme l’ensemble Pareto optimal n’est pas connu, elles permettent de se comparer avec le seul élément étudié dans la littérature à savoir la longueur totale des tournées. Il est intéressant de remarquer sur les instances E76-10e et E101-08e que si les solutions sont très proches en termes de longueur de la meilleure solution de la littérature, il existe un écart important en termes d’équilibre. Par exemple, la meilleure solution trouvée par TabuRoute et l’AG de Prins pour l’instance E101-08e a une longueur de 826.14 pour un équilibre de 97.88 alors que la meilleure solution trouvée par nos méthodes a une longueur de 827.39 pour un équilibre de 67.55. Il apparaît que pour gagner un peu en longueur (0.15 %), il est nécessaire de perdre une valeur significative du point de vue de l’équilibre. Il est donc probable que les solutions présentent de grandes différences dans la composition des tournées. Si de tels aspects se retrouvent dans les instances E151-12c et E200-17c, cela peut expliquer que les algorithmes génétiques définis pour prendre en compte les deux objectifs rencontrent les difficultés présentées dans la section 3.3.2 et que l’on ait dû définir un nouveau mécanisme pour la diversification.

Pour l’équilibre, on peut observer que les meilleures solutions sont proches de 0 ; on est donc capable de générer des tournées bien équilibrées. On observe cependant pour ces solutions une forte fluctuation de la longueur de la solution.

Au niveau des méthodes, il apparaît que NEMO est celle qui est la moins efficace. En effet, sur les deux plus grands problèmes, c’est elle qui identifie la borne la plus mauvaise du point de vue de la longueur totale. D’autre part, lorsque l’on observe les bornes pour l’équilibre, elle trouve la plus mauvaise solution 5 fois sur 7 et, par ailleurs, il y a un écart important entre la valeur trouvée par NEMO et celles trouvées par les autres méthodes. Il faut cependant remarquer que NEMO, ainsi que les autres méthodes, trouvent des solutions

TAB. 5.17 – Apport de l’hybridation : C(TAPaS-std, NSGA II) (première entrée par cellule) et C(TAPaS-dp, NSGA II) (seconde entrée par cellule).

Instance	Meilleur	Moyenne	Pire	Ecart type
E51-05e	0.84	0.32	0.06	0.27
	0.96	0.38	0.15	0.27
E76-10e	0.77	0.46	0.11	0.28
	0.98	0.60	0.17	0.26
E101-08e	0.97	0.76	0.15	0.31
	0.98	0.86	0.53	0.10
E151-12c	1.00	0.82	0.51	0.17
	1.00	0.97	0.83	0.05
E200-17c	0.71	0.51	0.31	0.17
	1.00	0.89	0.63	0.09
E121-07c	0.65	0.53	0.40	0.09
	0.88	0.70	0.40	0.11
E101-10c	0.36	0.12	0.00	0.13
	0.70	0.48	0.26	0.12

TAB. 5.18 – Valeurs de la métrique D pour TAPaS-dp avec et sans hybridation.

Instance	Meilleur	Moyenne	Pire	Ecart type
E51-05e	9.37×10^{-3}	4.04×10^{-3}	-4.17×10^{-3}	4.23×10^{-3}
E76-10e	2.291×10^{-2}	1.467×10^{-2}	6.05×10^{-3}	4.81×10^{-3}
E101-08e	2.28×10^{-3}	2×10^{-5}	-2.24×10^{-3}	1.15×10^{-3}
E151-12c	2.826×10^{-2}	7.11×10^{-3}	-1.02×10^{-3}	9.80×10^{-3}
E200-17c	4.430×10^{-2}	2.794×10^{-2}	-6.75×10^{-3}	1.796×10^{-2}
E121-07c	1.333×10^{-2}	7.14×10^{-3}	2.11×10^{-3}	3.54×10^{-3}
E101-10c	7.805×10^{-2}	4.370×10^{-2}	-6.44×10^{-3}	3.023×10^{-2}

bien équilibrées par rapport à la longueur moyenne des tournées. Inversement, la méthode coopérative composée de NSGA II et de TAPaS trouve toutes les meilleurs valeurs pour la longueur totale sans obtenir de mauvais résultats pour l’équilibre. Une explication pour la bonne performance de cette méthode pour les valeurs de la longueur totale sur les instances E151-12c et E200-17c peut être le fait que l’implémentation de TAPaS utilise une recherche Tabou qui est initialement conçue pour l’optimisation de l’objectif visant à minimiser la longueur totale des tournées.

5.8 Conclusion

Dans ce chapitre, nous avons proposé plusieurs implémentations de méthodes d’optimisation pour le problème d’élaboration de tournées de véhicules avec équilibrage des tournées. Ce problème est une extension bi-objectif du problème d’élaboration de tournées de véhicules avec contraintes de capacité où l’on minimise la longueur totale des tournées mais où l’on cherche aussi à équilibrer celles-ci entre elles. L’équilibrage est pris en compte

au travers d'un objectif qui vise à minimiser la différence entre la longueur de la plus longue tournée et la longueur de la plus courte tournée.

Au niveau des méthodes d'optimisation, nous avons proposé l'implémentation de deux algorithmes génétiques multi-objectif. L'un des algorithmes est l'algorithme standard NSGA II. Le mécanisme de diversification élitiste a aussi été implémenté ainsi que les modèles parallèles définis dans le chapitre 3. Nous avons aussi implémenté TAPaS à partir d'une recherche Tabou car les méta-heuristiques de ce type sont connues pour être efficaces sur le problème d'élaboration de tournées de véhicules.

Des expérimentations ont été faites sur des jeux de données standard de la littérature. Il apparaît que les mécanismes proposés sont capables d'améliorer les performances des algorithmes génétiques multi-objectif. Pour TAPaS, les résultats expérimentaux montrent que : i) TAPaS est capable de générer des approximations de bonne qualité quand il est comparé à NSGA II ; ii) TAPaS est efficace quand il est utilisé comme méthode de post-optimisation ; iii) l'hybridation de TAPaS avec un autre algorithme semble générer de meilleures approximations que si un seul algorithme est utilisé.

TAB. 5.19 – Meilleures bornes trouvées pour les deux objectifs pour différentes méthodes.

Instance	Méthode	Longueur totale			Equilibre		
		Meilleur connu	Meilleur trouvé	%	Equilibre	Meilleur trouvé	Longueur
E51-05e	PAR (NEMO)	524.61	524.61	0.00	20.07	0.24	618.22
	NSGA II (16 proc.)		524.61	0.00	20.07	0.18	779.88
	NSGAED (8 proc.)		524.61	0.00	20.07	0.32	645.29
E76-10e	NSGA II + TAPaS		524.61	0.00	20.07	0.19	633.58
	PAR (NEMO)	835.26	835.32	0.01	78.10	0.59	1203.98
	NSGA II (16 proc.)		835.32	0.01	78.10	0.48	1153.25
E101-08e	NSGAED (8 proc.)		835.32	0.01	78.10	0.33	1380.01
	NSGA II + TAPaS		835.32	0.01	78.10	0.65	1665.13
	PAR (NEMO)	826.14	827.39	0.15	67.55	0.29	1871.06
E151-12c	NSGA II (16 proc.)		827.39	0.15	67.55	0.27	1132.28
	NSGAED (8 proc.)		827.39	0.15	67.55	0.10	1610.86
	NSGA II + TAPaS		827.39	0.15	67.55	0.11	1665.13
E200-17c	PAR (NEMO)	1028.42	1047.35	1.84	74.78	0.80	1484.48
	NSGA II (16 proc.)		1032.95	0.44	97.57	0.44	1551.50
	NSGAED (8 proc.)		1034.91	0.63	109.06	0.21	1523.94
E121-07c	NSGA II + TAPaS		1029.15	0.07	107.39	0.09	2504.54
	PAR (NEMO)	1291.45	1352.46	4.72	76.60	1.38	1902.64
	NSGA II (16 proc.)		1328.44	2.86	96.61	0.86	3490.14
E101-10c	NSGAED (8 proc.)		1327.89	2.82	97.20	0.58	3746.57
	NSGA II + TAPaS		1303.57	0.93	79.41	0.59	2064.44
	PAR (NEMO)	1042.11	1042.11	0.00	146.67	0.10	2388.30
E101-10c	NSGA II (16 proc.)		1042.11	0.00	146.67	0.07	2296.43
	NSGAED (8 proc.)		1042.11	0.00	146.67	0.07	1649.76
	NSGA II + TAPaS		1042.11	0.00	146.67	0.03	2207.95
E101-10c	PAR (NEMO)	819.56	819.56	0.00	93.43	1.15	1429.90
	NSGA II (16 proc.)		819.56	0.00	93.43	0.21	1324.53
	NSGAED (8 proc.)		819.56	0.00	93.43	0.35	1664.80
	NSGA II + TAPaS		819.56	0.00	93.43	0.24	1233.52

Conclusions et perspectives

Les travaux de cette thèse ont porté sur la résolution de deux problèmes de tournées multi-objectif. Dans ce but, nous avons proposé plusieurs méthodes pour résoudre ces problèmes. Il est en effet apparu que les méthodes issues de la littérature pouvaient ne pas être suffisantes lorsqu'elles sont appliquées à des problèmes académiques difficiles comme le problème d'élaboration de tournées de véhicules. Ainsi, nous avons exploré plusieurs voies : la définition de nouveaux mécanismes, l'utilisation du parallélisme au travers de modèles insulaires et l'utilisation de modèles coopératifs.

Les problèmes de tournées sont très importants aussi bien d'un point de vue académique que d'un point de vue pratique ou industriel. Il est donc naturel de s'intéresser à la définition, l'étude et la résolution de problèmes de tournées multi-objectif. En effet, les problèmes multi-objectif connaissent un intérêt croissant car ils offrent l'opportunité de s'intéresser à des caractéristiques qui ne sont pas forcément évidentes à prendre en compte dans le cadre de l'optimisation classique. D'autre part, ils permettent de définir des problèmes plus proches des problèmes réels pour lesquels plusieurs objectifs peuvent le plus souvent être définis. L'état de l'art que nous avons effectué sur les problèmes de tournées multi-objectif montre que si leur nombre n'est pas trop important pour l'instant, l'intérêt que l'on porte à ces problèmes est de plus en plus significatif. Les deux problèmes étudiés sont illustratifs de l'utilisation que l'on peut faire de l'optimisation multi-objectif. Ainsi, le problème de la tournée couvrante bi-objectif est une généralisation du problème de la tournée couvrante, tandis que le problème d'élaboration de tournées de véhicules avec équilibrage des tournées est une extension du problème d'élaboration de tournées de véhicules. L'intérêt de l'étude de ces problèmes ne se limite pas aux problèmes en eux-mêmes mais aussi au challenge qu'ils représentent du point de vue de la résolution.

Nous nous sommes ainsi intéressé à l'optimisation multi-objectif. Dans ce cadre, nous nous sommes principalement consacré à la définition de méthodes ainsi que de modèles parallèles et coopératifs. Cependant, il peut être utile d'indiquer quelques voies qui semblent intéressantes ou tout simplement manquer par rapport à l'existant dans le domaine tel qu'il est présenté dans le chapitre 1. Dans l'ensemble, le fonctionnement et l'utilisation des méthodes et mécanismes deviennent de plus en plus abstraits lorsqu'il y a plus de deux ou trois objectifs. De plus, il est à regretter que les méthodes exactes ne peuvent pas être appliquées au-delà de deux objectifs. Ceci semble donc une voie très importante à explorer. Enfin, une autre lacune dans l'optimisation multi-objectif porte sur l'évaluation des méthodes. Il existe en effet de nombreuses métriques mais le choix n'est pas aisé car elles possèdent toutes au moins un inconvénient.

Dans cette thèse, nous nous sommes aussi intéressé aux mécanismes pour l'optimisation multi-objectif. Dans cette optique, nous en avons proposé un, la diversification élitiste. Il est à déplorer l'existence d'un grand nombre de mécanismes et d'outils existants qui ont un même rôle sans qu'il soit possible de savoir lequel est le plus efficace ; en effet, leur efficacité dépendant fortement du problème étudié. C'est en cela que le mécanisme proposé, la diversification élitiste, est intéressant car il a été conçu de telle sorte qu'il puisse être utilisé en collaboration avec d'autres mécanismes pour la diversification. Une voie à explorer ici est sans nul doute l'utilisation de plateforme logicielle générique incluant les outils et mécanismes pour l'optimisation multi-objectif. En effet, de telles plateformes permettent de s'abstraire de l'implémentation, parfois difficile, des mécanismes et d'essayer plusieurs combinaisons, tout en laissant la possibilité de se concentrer sur les aspects spécifiques au problème étudié.

Un autre outil utilisé dans cette thèse est le parallélisme et notamment les modèles coopératifs insulaires. Les résultats obtenus montrent que cet outil est capable d'offrir un gain important en termes de qualité des résultats. Il est très intéressant de poursuivre des recherches dans cette voie et cela pour plusieurs raisons. En effet, de nos jours, l'utilisation du parallélisme n'est plus un *luxé* réservé aux possesseurs d'onéreux calculateurs parallèles, car il est possible d'assembler des clusters de stations de travail et il ne faut pas négliger l'apparition du méta-calcul tant au niveau matériel avec la création de grilles de calcul qu'au niveau logiciel avec la création de plateformes qui offrent la possibilité de déployer des applications parallèles de manière facile et efficace.

Le dernier point étudié lors de cette thèse est la coopération entre méthodes. Pour cela, nous avons tout d'abord proposé des stratégies pour guider des méthodes à base de voisinage dans le cas où ces dernières sont utilisées lors d'une phase de post-optimisation après application d'une première approche d'optimisation. Plus particulièrement, nous avons proposé une méthodologie, *Target Aiming Pareto Search*, qui offre de très bons résultats dans le cadre du problème d'élaboration de tournées de véhicules avec équilibrage des tournées. Dans cette thèse, nous l'appliquons sur les résultats d'un algorithme génétique multi-objectif. Il serait cependant intéressant de définir et d'étudier des modèles coopératifs où la transition entre les méthodes serait dynamique, adaptative et réversible. Lors de l'étude du problème de la tournée couvrante bi-objectif, nous avons envisagé un modèle de coopération entre une méta-heuristique et une méthode exacte. Dans cette approche coopérative, des sous-problèmes étaient extraits des résultats obtenus par l'algorithme génétique pour être résolus par un algorithme de séparations et coupes mono-objectif. L'application de ce dernier était possible car la taille des sous-problèmes ne devenait pas trop grande. Les résultats expérimentaux montrent que l'application de la méthode exacte permet une amélioration sensible de la qualité des résultats sans entraîner une pénalisation en termes de temps de calcul. Ce genre de coopération apparaît comme une voie de recherche très prometteuse. Dans le travail effectué, nous avons fait coopérer une méta-heuristique multi-objectif avec une méthode exacte mono-objectif. Dans le cadre de l'optimisation multi-objectif, il serait intéressant d'utiliser des méthodes exactes qui travaillent sur tout le problème multi-objectif. Cependant, comme indiqué précédemment, cette voie est restreinte par l'absence de méthode exacte traitant plus de deux objectifs.

Bibliographie

- [1] F. B. Abdelaziz, S. Krichen et J. Chaouchi. *Metaheuristics : Advances and trends in local search paradigms for optimization*, chapitre A hybrid heuristic for multiobjective knapsack problems, pages 205–212. Kluwer Academic Press, 1999.
- [2] A. S. Alfa, S. S. Heragu et M. Chen. A 3-opt based simulated annealing algorithm for vehicle routing problems. *Computers and Industrial Engineering*, 21 : 635–639, 1991.
- [3] R. Allenson. Genetic algorithms with gender for multi-function optimisation. Rapport technique EPCC-SS92-01, Edinburgh Parallel Computing Center, Edimbourg, Ecosse, 1992.
- [4] T. Arslan, E. Ozdemir, M. S. Bright et D. H. Horrock. Genetic synthesis techniques for low-power digital signal processing circuits. Dans *IEEE Colloquium on Digital Synthesis*, pages 71–75, Londres, Royaume Uni, 1996. IEEE service center.
- [5] T. Bäck. *Evolutionary algorithm in theory and practice*. Oxford University Press, New York, 1996.
- [6] B. M. Baker et M. A. Ayechev. A genetic algorithm for the vehicle routing problem. *Computers and Operations Research*, 30 : 787–800, 2003.
- [7] J. E. Baker. Reducing bias and inefficiency in the selection algorithm. Dans J. J. Grefenstette, éditeur, *Proceedings of the Second International Conference on Genetic Algorithms and their Application*, pages 14–21, Hillsdale, New Jersey, USA, 1987. Lawrence Erlbaum Associates.
- [8] E. Balas. The prize collecting traveling salesman problem. *Networks*, 19 : 621–636, 1989.
- [9] E. Balas et A. Ho. Set covering algorithm using cutting planes, heuristics, and subgradient optimization : A computational study. *Mathematical Programming*, 12 : 37–60, 1980.
- [10] S. Baluja. Population-based incremental learning : A method for integrating genetic search based function optimization and competitive learning. Rapport technique CMU-CS-94-163, Carnegie Mellon University, Pittsburgh, PA, USA, 1994.
- [11] S. Baluja et R. Caruana. Removing the genetics from the standard genetic algorithm. Dans A. Prieditis et S. Russel, éditeurs, *The International Conference on Machine Learning 1995*, pages 38–46. Morgan Kaufmann Publisher, 1995.

-
- [12] B. Baràn et M. Schaerer. A multiobjective ant colony system for vehicle routing problem with time windows. Dans *IASTED International Conference on Applied Informatics*, 2003.
- [13] G. Barbarosoglu et D. Ozgur. A tabu search algorithm for the vehicle routing problem. *Computers and Operations Research*, 26 : 255–270, 1999.
- [14] M. Basseur, F. Seynhaeve et E-G. Talbi. Design of multi-objective evolutionary algorithms : Application to the flow-shop. Dans *Congress on Evolutionary Computation (CEC'2002)*, pages 1151–1156, Honolulu, Hawaii, USA, 2002. IEEE Service Center.
- [15] J. E. Beasley. Route-first cluster-second methods for vehicle routing. *Omega*, 11 : 403–408, 1983.
- [16] J. E. Beasley et P. C. Chu. A genetic algorithm for the set covering tour problem. *European Journal of Operational Research*, 94 : 392–404, 1996.
- [17] J. L. Blanton et R. L. Wainwright. Multiple vehicle routing with time and capacity constraints using genetic algorithms. Dans S. Forrest, éditeur, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 452–459, San Mateo, CA, 1993. Morgan Kaufmann.
- [18] B. Boffey. Multiobjective routing problems. *Top*, 3 : 167–220, 1995.
- [19] R. Bowerman, B. Hall et P. Calamai. A multi-objective optimization approach to urban school bus routing : Formulation and solution method. *Transportation Research Part A*, 29 : 197–123, 1995.
- [20] I-M. Chao. A tabu search method for the truck and trailer problem. *Computers and Operations Research*, 29 : 22–51, 2002.
- [21] Y. L. Chen et C. C. Liu. Multiobjective VAR planning using the goal attainment method. *IEEE Proceedings on Generation, Transmission, and Distribution*, 141 : 227–232, 1994.
- [22] N. Christofides et S. Eilon. An algorithm for the vehicle dispatching problem. *Operational Research Quarterly*, 20 : 309–318, 1969.
- [23] N. Christofides, A. Mingozzi, P. Toth et C. Sandi, éditeurs. *Combinatorial optimization*, chapitre 11. John Wiley, Chichester, 1979.
- [24] G. Clarke et J. V. Wright. Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12 : 568–581, 1964.
- [25] C. A. Coello Coello. Using a min-max method to solve multiobjective optimization problems with genetic algorithms. Dans *IBERAMIA'98*, Lecture notes in computer science, pages 303–314, Lisbon, Portugal, 1998. Springer-Verlag.
- [26] C. A. Coello Coello, A. H. Aguirre et B. P. Buckles. Evolutionary multiobjective design of combinatorial logic circuits. Dans J. Lohn, A. Stoica, D. Keymeulen et S. Colombano, éditeurs, *Proceedings of the second NASA/DoD workshop on evolvable hardware*, pages 161–170, Los Alamitos, California, USA, 2000. IEEE Computer Society.
- [27] C. A. Coello Coello et G. Toscano Pulido. A micro-genetic algorithm for multiobjective optimization. Dans E. Zitzler, K. Deb, L. Thiele, C. A. Coello Coello et D. Corne,

- éditeurs, *Proceedings of the First International Conference on Evolutionary Multi-criterion Optimization*, volume 1993 de *Lecture Notes in Computer Science*, pages 126–140. Springer-Verlag, 2001.
- [28] C. A. Coello Coello, D. A. Van Veldhuizen et G. B. Lamont, éditeurs. *Evolutionary algorithms for solving multi-objective problems*. Kluwer Academic Press, 2002.
- [29] A. Corberan, E. Fernandez, M. Laguna et R. Marti. Heuristic solutions to the problem of routing school buses with multiple objectives. *Journal of the Operational Research Society*, 53 : 427–435, 2002.
- [30] J.-F. Cordeau, M. Gendreau et G. Laporte. A tabu search heuristic for periodic and multi-depot vehicle problems. *Networks*, 30 : 105–119, 1997.
- [31] J.-F. Cordeau, G. Laporte et A. Mercier. A unified tabu search heuristic for vehicle routing problem with time windows. *Journal of the Operational Research Society*, 52 : 928–942, 2001.
- [32] D. W. Corne, N. R. Jerram, J. D. Knowles et M. J. Oates. PESA-II : Region-based selection in evolutionary multiobjective optimization. Dans L. Spector, E. D. Goodman, A. Wu, W. Langdon, H.-M. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M. H. Garzon et E. Burke, éditeurs, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001)*, pages 283–290. Morgan Kaufmann Publisher, 2001.
- [33] D. W. Corne, J. D. Knowles et M. J. Oates. The pareto envelope-based selection algorithm for multiobjective optimization. Dans M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo et H.-P. Schwefel, éditeurs, *Proceedings of Parallel Problem Solving from Nature VI*, volume 1917 de *Lecture Notes in Computer Science*, pages 839–848. Springer-Verlag, 2000.
- [34] J. R. Current. *Multiobjective design of transportation networks*. Thèse de doctorat, Department of Geography and Environmental Engineering, John Hopkins University, USA, 1981.
- [35] J. R. Current et D. A. Schilling. The covering salesman problem. *Transportation Science*, 23 : 208–213, 1989.
- [36] J. R. Current et D. A. Schilling. The median tour and maximal covering problems. *European Journal of Operational Research*, 73 : 114–126, 1991.
- [37] P. Czyzak et A. Jaszkievicz. Pareto simulated annealing - a metaheuristic technique for multiple-objective combinatorial optimization. *Journal of Multi-Criteria Decision Analysis*, 7 : 34–47, 1998.
- [38] G. B. Dantzig, D. R. Fulkerson et S. M. Johnson. Solution of a large-scale traveling-salesman problem. *Operations Research*, 2 : 393–410, 1954.
- [39] G. B. Dantzig et J. H. Ramser. The truck dispatching problem. *Management Science*, 6, 1959.
- [40] C. Darwin. *On the origin of species*. John Murray, Londres, Angleterre, 1859.
- [41] I. Das et J. Dennis. A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. *Structural Optimization*, 14 : 63–69, 1997.

-
- [42] K. Deb. *Multi-objective optimization using evolutionary algorithms*. John Wiley and sons, 2001.
- [43] K. Deb et T. Goel. A hybrid multi-objective evolutionary approach to engineering shape design. Dans E. Zitzler, K. Deb, L. Thiele, C. A. Coello Coello et D. Corne, éditeurs, *Evolutionary Multi-criterion Optimization*, volume 1993 de *Lecture Notes in Computer Science*, pages 385–399, Zurich, Suisse, 2001. Springer-Verlag.
- [44] K. Deb, A. Pratap, S. Agarwal et T. Meyarvan. A fast and elitist multiobjective genetic algorithm : NSGA II. *IEEE Transactions on Evolutionary Computation*, 6 : 182–197, 2002.
- [45] J. L. Deneubourg et S. Goss. Collective patterns and decision-making. *Ethology and Evolution*, pages 295–311, 1989.
- [46] J. L. Deneubourg, J. M. Pasteels et J. C. Verhaeghe. Probabilistic behaviour in ants : A strategy of errors ? *Journal of Theoretical Biology*, 105 : 259–271, 1983.
- [47] K. Doerner, M. Gronalt, R. F. Hartl, M. Reimann, C. Strauss et M. Stummer. SavingAnts for the vehicle routing problem. Dans S. Cagioni et al., éditeurs, *Evo-Workshops 2002*, volume 2279 de *Lecture Notes in Computer Science*, pages 11–20. Springer-Verlag, 2002.
- [48] K. Doerner, R. F. Hartl, G. Kiechle, M. Lucka et M. Reimann. Parallel ant systems for the capacitated vehicle routing problem. Dans J. Gottlieb et G. R. Raidl, éditeurs, *Proceedings of EvoCOP 2004*, volume 3004 de *Lecture Notes in Computer Science*, pages 72–83. Springer-Verlag, 2004.
- [49] M. Dorigo. *Learning and natural algorithms*. Thèse de doctorat, Dipartimento di Elettronica e Informatica, Politecnico di Milano, Italie, 1992.
- [50] M. Dorigo, V. Maniezzo et A. Colorni. Positive feedback as a search strategy. Rapport technique 910116, Dipartimento di Elettronica e Informatica, Politecnico di Milano, Italie, 1991.
- [51] M. Dorigo, V. Maniezzo et A. Colorni. The ant system : optimization by a colony of cooperating agents. *IEEE Transactions on System, Man, and Cybernetics Part B : Cybernetics*, 26 : 29–41, 1996.
- [52] F. Y. Edgeworth. *Mathematical physics*. P. Keagan, Londres, Angleterre, 1881.
- [53] M. Ehrgott. Approximation algorithms for combinatorial multicriteria optimization problems. *International Transactions in Operational Research*, 7 : 5–31, 2000.
- [54] N. H. Eklund et M. J. Embrecht. GA-based multi-objective optimization of visible spectra for lamp design. Dans C. H. Dagli, A. L. Buczak, J. Ghosh, M. J. Embrecht et O. Ersoy, éditeurs, *Smart engineering system design : neural networks, fuzzy logic, evolutionary programming, data mining and complex systems*, pages 451–456, New York, NY, USA, 1999. ASME Press.
- [55] W. El Moudani, C. A. N. Cosenza, M. de Coligny et F. Mora-Camino. A bi-criterion approach for the crew rostering problem. Dans E. Zitzler, K. Deb, L. Thiele, C. A. Coello Coello et D. Corne, éditeurs, *First International Conference on Evolutionary Multi-Criterion Optimization*, volume 1993 de *Lecture Notes in Computer Science*, pages 486–500. Springer-Verlag, 2001.

-
- [56] N. El-Sherbeny. *Resolution of a vehicle routing problem with multi-objective simulated annealing method*. Thèse de doctorat, Faculté Polytechnique de Mons, Mons, Belgique, 2001.
- [57] M. Erickson, A. Mayer et J. Horn. The niched pareto genetic algorithm 2 applied to the design of groundwater remediation systems. Dans E. Zitzler, K. Deb, L. Thiele, C. A. Coello Coello et D. Corne, éditeurs, *First International Conference on Evolutionary Multi-criterion Optimization*, volume 1993 de *Lecture Notes in Computer Science*, pages 681–695. Springer-Verlag, 2001.
- [58] D. Feillet, P. Dejax et M. Gendreau. Traveling salesman problem with profits. *Transportation Science*, 2003. (to be published).
- [59] M. Fischetti, J. J. Salazar et P. Toth. A branch-and-cut algorithm for the symmetric generalized traveling salesman problem. *Operations Research*, 45 : 378–394, 1997.
- [60] M. Fischetti et P. Toth. An additive approach for the optimal solution of the prize-collecting traveling salesman problem. Dans B. L. Golden et A. A. Assad, éditeurs, *Vehicle routing : Methods and studies*, pages 319–343, Amsterdam, Holland, 1988.
- [61] M. L. Fisher et R. Jaikumar. A generalized assignment heuristic for the vehicle routing problem. *Networks*, 11 : 109–124, 1981.
- [62] J. Fogel, J. Owens et J. Walsh. *Artificial Intelligence through simulated evolution*. John Wiley, 1966.
- [63] C. M. Fonseca et P. J. Flemming. Genetic algorithms for multi-objective optimization : Formulation, discussion and generalization. Dans S. Forrest, éditeur, *Proceedings of the fifth International Conference on Genetic Algorithms*, pages 416–423, San Mateo, Californie, USA, 1993. Morgan Kaufmann Publishers.
- [64] M. P. Fourman. Compaction of symbolic layout using genetic algorithms. Dans *Proceedings of the first international conference on genetic algorithms*, pages 141–153, 1985.
- [65] L. Gacôgne. Multiple objective optimization of fuzzy rules for obstacles avoiding by an evolution algorithm with adaptative operators. Dans *Proceedings of the fifth international Mendel conference on soft computing*, pages 236–242, Brno, République Tchèque, 1999.
- [66] C. A. Garrett, J. Huang, M. N. Goltz et G. B. Lamont. Parallel real-valued genetic algorithms for bioremediation of TCE-contaminated groundwater. Dans *1999 Congress on Evolutionary Computation*, pages 2183–2189, Washington, D.C., USA, 1999. IEEE Service Center.
- [67] M. J. Geiger. Genetic algorithms for multiple objective routing. Dans *Metaheuristic International Conference 2001 (MIC'2001)*, pages 348–353, 2001.
- [68] M. Gendreau, A. Hertz et G. Laporte. New insertion and postoptimization procedures for the traveling salesman problem. *Operations Research*, 40 : 1086–1094, 1992.
- [69] M. Gendreau, A. Hertz et G. Laporte. A tabu search heuristic for the vehicle routing problem. *Management Science*, 40 : 1276–1290, 1994.
- [70] M. Gendreau, G. Laporte et F. Semet. The covering tour problem. *Operations Research*, 45 : 568–576, 1997.

- [71] F. Glover. Heuristics for integer programming using surrogate constraints. *Decision Sciences*, 8 : 156–166, 1977.
- [72] F. Glover. Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, 13 : 533–549, 1986.
- [73] D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, Massachusetts, USA, 1989.
- [74] D. E. Goldberg et R. Lingle. Alleles, loci and the traveling salesman problem. Dans J. J. Grefenstette, éditeur, *Proceedings of the First International Conference on Genetic Algorithms*, pages 154–159. Lawrence Erlbaum, 1985.
- [75] D. E. Goldberg et J. Richardson. Genetic algorithms with sharing for multi-modal function optimization. Dans *Second International Conference on Genetic Algorithms*, pages 41–49, 1987.
- [76] M. Hachicha, M. J. Hodgson, G. Laporte et F. Semet. Heuristics for the multi-vehicle covering tour problem. *Computers and Operations Research*, 27 : 29–42, 2000.
- [77] M. P. Hansen. Tabu search for multiobjective optimization : MOTS. Dans *Multiple Criteria Decision Making*, 1997.
- [78] P. Hansen. The steepest ascent mildest descent heuristic for combinatorial programming. Dans *Congress on Numerical Methods in Combinatorial Optimization*, 1986.
- [79] M. J. Hodgson, G. Laporte et F. Semet. A covering tour model for planning mobile health care facilities in Suhum district, Ghana. *Journal of Regional Science*, 38 : 621–638, 1998.
- [80] J. Holland. *Adaptation in natural and artificial systems*. University of Michigan Press, 1975.
- [81] S-C. Hong et Y-B. Park. A heuristic for a bi-objective vehicle routing with time window constraints. *International Journal of Production Economics*, 62 : 249–258, 1999.
- [82] J. Horn et N. Nafpliotis. Multiobjective optimization using the Niched Pareto Genetic Algorithm. IlliGAL Report 93005, Illinois Genetic Algorithm Laboratory, University of Illinois at Urbana-Champaign, Illinois, USA, 1993.
- [83] J. Horn, N. Nafpliotis et D. E. Goldberg. A Niched Pareto Genetic Algorithm for multiobjective optimization. Dans *Proceedings of the first Conference on Evolutionary Computation*, pages 82–87. IEEE Service Center, 1994.
- [84] H. Ishibuchi et T. Murata. A multi-objective genetic local search algorithm and its application to flowshop scheduling. *IEEE Transactions on Systems, Man, and Cybernetics - Part C : Applications and Reviews*, 28 : 392–403, 1998.
- [85] H. Ishibuchi, T. Yoshida et T. Murata. Balance between genetic search and local search in hybrid evolutionary multi-criterion optimization algorithms. Dans E. Cantu-Paz et al., éditeurs, *Genetic and Evolutionary Computation Conference*, pages 1301–1308. Morgan Kaufmann, 2002.
- [86] A. Jaszkiwicz. Genetic local search for multi-objective combinatorial optimization. *European Journal of Operational Research*, 137 : 50–71, 2002.

-
- [87] H. Kawamura, M. Yamamoto, T. Mitamura, K. Suzuki et A. Ohuchi. Cooperative search on pheromone communication for vehicle routing problems. *IEEE Transactions on Fundamentals*, 81 : 1089–1096, 1998.
- [88] C. P. Keller. *Multiobjective routing through space and time : The MVP and TD-VRP problems*. Thèse de doctorat, Department of Geography, University of Western Ontario, London, Ontario, Canada, 1985.
- [89] C. P. Keller et M. Goodchild. The multiobjective vending problem : A generalization of the traveling salesman problem. *Environment and Planning B : Planning and Design*, 15 : 447–460, 1988.
- [90] J. Kennedy et R. C. Eberhart. Particle swarm optimization. Dans *Proceedings of the 1995 IEEE International Conference on Neural Networks*, pages 1942–1948, 1995.
- [91] S. Kirkpatrick, D. C. Gelatt et M. P. Vecchi. Optimization by simulated annealing. *Science*, 220 : 671–680, 1983.
- [92] J. Knowles et D. Corne. On metrics for comparing nondominated sets. Dans *Congress on Evolutionary Computation (CEC'2002)*, volume 1, pages 711–716. IEEE Service Center, 2002.
- [93] J. D. Knowles. *Local-search and hybrid evolutionary algorithms for Pareto optimization*. Thèse de doctorat, Université de Reading, Reading, Angleterre, 2002.
- [94] J. D. Knowles et D. W. Corne. Approximating the nondominated front using the Pareto Archived Evolution Strategy. *Evolutionary Computation*, 8 : 149–172, 2000.
- [95] J. R. Koza. *Genetic programming : On the programming of computers by natural selection*. MIT Press, Cambridge, MA, 1992.
- [96] F. Kurwase. *A variant of evolution strategies for vector optimisation*. Thèse de doctorat, Vanderbilt University, Nashville, Tennessee, USA, 1984.
- [97] P. Lacomme, C. Prins et M. Sevaux. Multiobjective capacitated arc routing problem. Dans C. M. Fonseca et al., éditeurs, *Evolutionary Multi-criterion Optimization*, volume 2632 de *Lecture Notes in Computer Science*, pages 550–564. Springer-Verlag, 2003.
- [98] G. Laporte et S. Martello. The selective traveling salesman problem. *Discrete Applied Mathematics*, 26 : 193–207, 1990.
- [99] D. Lee. Multiobjective design of a marine vehicle with aid of design knowledge. *International Journal for Numerical Methods in Engineering*, 40 : 2665–2677, 1997.
- [100] L. H. Lee, K. C Tan, K. Ou et Y. H. Chew. Vehicle capacity planning system (VCPS) : A case study on vehicle routing problem with time windows. *IEEE Transaction on Systems, Man and Cybernetics : Part B*, 2003. (A paraître).
- [101] T-R. Lee et J-H. Ueng. A study of vehicle routing problem with load balancing. *International Journal of Physical Distribution and Logistics Management*, 29 : 646–648, 1998.
- [102] J. Lemesre, C. Dhaenens et E-G. Talbi. A parallel exact method for a bicriteria permutation flow-shop problem. Dans *Project Management and Scheduling (PMS)*, pages 359–362, Nancy, France, Avril 2004.

- [103] J. K. Lenstra et A. H. G. Rinnooy Kan. Complexity of vehicle routing and scheduling problem. *Networks*, 11 : 221–227, 1981.
- [104] S. Lin. Computer solutions of the traveling salesman problem. *Bell System Technical Journal*, 44 : 2245–2269, 1965.
- [105] J. Lis et A. Eiben. A multi-sexual genetic algorithm for multi-objective optimization. Dans T. Fukuda et T. Furuhashi, éditeurs, *Proceedings of the International Conference on Genetic Algorithms*, pages 59–64, Nagoya, Japan, 1996.
- [106] V. Maniezzo, R. Baldacci, M. Boschetti et M. Zamboni. Scatter search methods for the covering tour problem. Rapport technique, Science dell’Informazione, Université de Bologne, Italie, 1999.
- [107] S. Mardle, S. Pascoes et M. Tamiz. An investigation of genetic algorithm for the optimisation of multi-objective fisheries bioeconomic models. *International Transaction of Operations Research*, 7 : 33–49, 2000.
- [108] H. Meunier. *Algorithmes évolutionnaires parallèles pour l’optimisation multi-objectif de réseaux de télécommunications mobiles*. Thèse de doctorat, Laboratoire d’Informatique Fondamentale de Lille, Université des Sciences et Technologies de Lille, Villeneuve d’Ascq, France, Juin 2002.
- [109] J. N. Morse. Reducing the size of the non-dominated set : Pruning by clustering. *Computers and Operations Research*, 7 : 55–66, 1980.
- [110] L. Motta, L. S. Ochi et C. Martinhon. GRASP metaheuristics for the generalized covering tour problem. Dans 4th *Metaheuristic International Conference (MIC’2001)*, pages 387–391, Porto, Portugal, 2001.
- [111] M. Mourgaya. *The periodic vehicle routing problem : planning before routing*. Thèse de doctorat, Laboratoire de Mathématiques Appliquées de Bordeaux, Université de Bordeaux 1, Bordeaux, France, Juillet 2004.
- [112] I. Or. *Traveling salesman-type combinatorial problems and their relation to the logistics of blood banking*. Thèse de doctorat, Department of Engineering and Management Science, Northwestern University, Evanston, IL, 1976.
- [113] I. H. Osman. Metastrategy simulated annealing and Tabu search algorithms for the vehicle routing problem. *Annals of Operations Research*, 41 : 421–451, 1993.
- [114] A. Osyczka. Multicriteria optimization for engineering design. Dans J. S. Gero, éditeur, *Design optimization*, pages 193–227. Academic Press, 1985.
- [115] J. Pacheco et R. Marti. Tabu search for a multi-objective routing problem. Rapport technique TR09-2003, Université de Valence, 2003.
- [116] C. H. Papadimitriou. *The complexity of combinatorial optimization problems*. Thèse de doctorat, Princeton, 1976.
- [117] C. H. Papadimitriou et K. Steiglitz. *Combinatorial optimization : Algorithms and complexity*. Prentice-Hall, 1982.
- [118] L. Paquete et T. Stützle. A two-phase local search for the bi-objective traveling salesman problem. Dans C. M. Fonseca et al., éditeurs, *Evolutionary Multi-criterion Optimization*, volume 2632 de *Lecture Notes in Computer Science*. Springer-Verlag, 2003.

-
- [119] V. Pareto. *Cours d'économie politique*. Rouge, Lausanne, Suisse, 1896.
- [120] Y. Park et C. Koelling. A solution of vehicle routing problems in multiple objective environment. *Engineering Costs and Production Economics*, 10 : 121–132, 1986.
- [121] Y. Park et C. Koelling. An interactive computerized algorithm for multicriteria vehicle routing problems. *Computers and Industrial Engineering*, 16 : 477–490, 1989.
- [122] J.-Y. Potvin et S. Bengio. The vehicle routing problem with time windows part II : Genetic search. *INFORMS Journal on Computing*, 8 : 339–370, 1996.
- [123] R. C. Prim. Shortest connection networks and some generalizations. *Bell System Technical Journal*, 36 : 1389–1401, 1957.
- [124] C. Prins. A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers and Operations Research*, 31 : 1985–2002, 2004.
- [125] V. M. Pureza et P. M. França. Vehicle routing problems via tabu search metaheuristic. Rapport technique CRT-347, Centre de Recherche sur les Transports, Montréal, Canada, 1991.
- [126] M. Rahoual et W. Djoukhdjoukh. Métaheuristique hybride Pareto pour le problème de tournées de véhicules avec fenêtres horaires. Dans *Evolution Artificielle 2003*, pages 380–395, 2003.
- [127] M. Rahoual, B. Kitoun, M. Mabed, V. Bachelet et F. Benameur. Multicriteria genetic algorithms for the vehicle routing problem with time windows. Dans *Metaheuristic International Conference (MIC'2001)*, pages 527–532, 2001.
- [128] I. Rechenberg. *Evolutions strategie : Optimierung technischer systeme nach prinzipien der biologischen evolution*. Frommann-Holzboog, Stuttgart, Allemagne, 1973.
- [129] C. Rego et C. Roucairol. A parallel tabu search algorithm using ejection chains for the vehicle routing problem. Dans I. H. Osman et J. P. Kelly, éditeurs, *Meta-heuristics : Theory and applications*, pages 661–675. Kluwer, Boston, MA, USA, 1996.
- [130] M. Reimann, K. Doerner et R. F. Hartl. D-Ants : Savings based Ants divide and conquer the vehicle routing problem. *Computers and Operations Research*, 31 : 563–591, 2004.
- [131] M. Reimann, M. Stummer et K. Doerner. A savings based ant system for the vehicle routing problem. Dans W. B. Langdon, éditeur, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2002)*, San Francisco, CA, 2002. Morgan Kaufmann.
- [132] R. Ribeiro et H. R. Lourenço. A multi-objective model for a multi-period distribution management problem. Dans *Metaheuristic International Conference 2001 (MIC'2001)*, pages 91–102, 2001.
- [133] B. J. Ritzel, J. W. Eheart et S. Ranjithan. Using genetic algorithms to solve a multiple objective groundwater pollution containment problem. *Water Resources Research*, 30 : 1589–1603, 1994.
- [134] F. Robusté, C. F. Daganzo et R. Souleyrette. Implementing vehicle routing models. *Transportation Research B*, 24 : 263–286, 1990.
- [135] Y. Rochat et E. D. Taillard. Probabilistic diversification and intensification in local search for vehicle routing. *Journal of Heuristics*, 1 : 147–167, 1995.

- [136] J. L. Rogers. A parallel approach to optimum actuator selection with a genetic algorithm. Dans *AIAA Guidance, Navigation, and Control Conference*, Denver, Colorado, USA, 2000.
- [137] R. S. Rosenberg. *Simulation of genetic populations with biochemical properties*. Thèse de doctorat, University of Michigan, Ann Harbor, Michigan, USA, 1967.
- [138] D. Schaffer. Multiple objective optimization with vector evaluated genetic algorithm. Dans *First International Conference on Genetic Algorithms*, pages 93–100, 1985.
- [139] L. J. Schmitt. *An empirical computational study of genetic algorithms to solve order based problems : An emphasis on TSP and VRPTC*. Thèse de doctorat, Fogelman College of Business Economics, University of Memphis, TN, USA, 1994.
- [140] L. J. Schmitt. An evaluation of genetic algorithmic approach to the vehicle routing problem. Working paper, Department of Information Technology Management, Christian Brothers University, Memphis, TN, USA, 1995.
- [141] J. R. Schott. Fault tolerant design using single and multicriteria genetic algorithm optimization. Master's thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, 1995.
- [142] H-P. Schwefel. *Numerical optimization for computer models*. John Wiley & Sons, Chichester, 1981.
- [143] P. Serafini. Simulated annealing for multiple objective optimization problems. Dans *The Tenth International Conference on Multiple Criteria Decision Making*, pages 87–96, 1992.
- [144] W. Sessomboon, K. Watanabe, T. Irohara et K. Yoshimoto. A study on multi-objective vehicle routing problem considering customer satisfaction with due-time (the creation of Pareto optimal solutions by hybrid genetic algorithm). *Transaction of the Japan Society of Mechanical Engineering*, 1998.
- [145] B. W. Silverman. *Density estimation for statistics and data analysis*. Chapman and Hall, Londres, Angleterre, 1986.
- [146] N. Srinivas et K. Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2 : 221–248, 1994.
- [147] T. Starkweather, S. McDaniel, K. Mathias, C. Whitley et D. Whitley. A comparison of genetic sequencing operators. Dans R. Belew et L. Booker, éditeurs, *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 69–76, San Mateo, CA, 1991. Morgan Kaufmann Publishers.
- [148] R. E. Steuer. *Multiple criteria optimization : Theory, computation, and application*. Krieger Pub Co, 1986.
- [149] P. D. Surry et N. J. Radcliffe. The COMOGA method : constrained optimization by multiobjective genetic algorithms. *Control and Cybernetics*, 26 : 391–412, 1997.
- [150] C. Sutcliffe et J. Board. Optimal solution of a vehicle routing problem : Transporting mentally handicapped adults to an adult training centre. *Journal of the Operational Research Society*, 41 : 61–67, 1990.
- [151] E. D. Taillard. Parallel iterative search methods for vehicle routing problem. *Networks*, 23 : 661–673, 1993.

-
- [152] E-G. Talbi, M. Rahoual, M. H. Mabed et C. Dhaenens. A hybrid evolutionary approach for multicriteria optimization problems : Application to the flow shop. Dans E. Zitzler, K. Deb, L. Thiele, C. A. Coello Coello et D. Corne, éditeurs, *Evolutionary Multi-criterion Optimization*, volume 1993 de *Lecture Notes in Computer Science*, pages 416–428, Zurich, Suisse, 2001. Springer-Verlag.
- [153] H. Tamaki, H. Kita et S. Kobayashi. Multi-objective optimization by genetic algorithms : A review. Dans *IEEE International Conference on Evolutionary Computation*, pages 517–522, 1996.
- [154] H. Tamaki, M. Mori, M. Anaki, Y. Mishima et H. Ogai. Multicriteria optimization by genetic algorithms : A case of scheduling in hot rolling process. Dans *Proceeding of the third asian-pacific operational research societies conference*, pages 374–381. World Scientific, 1995.
- [155] K. C. Tan, T. H. Lee, Y. H. Chew et L. H. Lee. A hybrid multiobjective evolutionary algorithm for solving truck and trailer vehicle routing problems. Dans *Proceedings of the 2003 Congress on Evolutionary Computation (CEC'2003)*, pages 2134–2141, Canberra, Australia, 2003. IEEE Press.
- [156] K. C. Tan, T. H. Lee, Y. H. Chew et L. H. Lee. A multiobjective evolutionary algorithm for solving vehicle routing problem with time windows. Dans *IEEE International Conference on Systems, Man, and Cybernetics*, pages 361–366. IEEE Service Center, 2003.
- [157] S. R. Thangiah. Vehicle routing with time windows using genetic algorithms. Rapport technique SRU-CpSc-TR-93-23, Slippery Rock University, Slippery Rock, PA, USA, 1993.
- [158] S. R. Thangiah. An adaptive clustering method using a geometric shape for vehicle routing problems with time windows. Dans L. J. Eshelman, éditeur, *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 536–543. Morgan Kaufmann, 1995.
- [159] P. Toth et D. Vigo, éditeurs. *The vehicle routing problem*, volume 9 de *SIAM Monographs on Discrete Mathematics and Applications*. SIAM, 2001.
- [160] P. Toth et D. Vigo. The granular tabu search and its application to the vehicle routing problem. *INFORMS Journal on Computing*, 15 : 333–346, 2003.
- [161] E. L. Ulungu. *Optimisation combinatoire multicritère : Détermination de l'ensemble des solutions efficaces et méthodes interactives*. Thèse de doctorat, Université de Mons-Hainaut, Mons, Belgique, 1993.
- [162] E. L. Ulungu et J. Teghem. The two phases method : An efficient procedure to solve bi-objective combinatorial optimization problems. *Foundation of Computing and Decision Science*, 20 : 149–156, 1995.
- [163] E. L. Ulungu, J. Teghem, P. Fortemps et D. Tuyttens. MOSA method : a tool for solving moco problems. *Journal of Multi-Criteria Decision Analysis*, 8 : 221–236, 1999.
- [164] A. Van Breedam. An analysis of the effect of local improvement operators in genetic algorithms and simulated annealing for the vehicle routing problem. RUCA Working Paper 96/14, Université d'Anvers, Anvers, Belgique, 1996.

- [165] D. A. Van Veldhuizen. *Multiobjective evolutionary algorithms : classifications, analyses, and new innovations*. Thèse de doctorat, Graduate School of Engineering, Air Force Institute of Technology, Wright Patterson AFB, Ohio, USA, Janvier 1999.
- [166] D. A. Van Veldhuizen et G. B. Lamont. Multiobjective optimization with messy genetic algorithms. Dans *Proceedings of the 2000 ACM Symposium on Applied Computing*, pages 470–476. ACM, 2000.
- [167] D. Whitley, T. Starkweather et D'A Fuquay. Scheduling problems and traveling salesman : The genetic edge recombination operator. Dans J. Schaffer, éditeur, *Proceedings of the Third International Conference on Genetic Algorithms*, pages 133–140, San Mateo, CA, 1989. Morgan Kaufmann Publishers.
- [168] P. B. Wienke, C. Lucasius et G. Kateman. Multicriterion target optimization of analytical procedures using a genetic algorithm. *Analytical Chimica Acta*, 265 : 211–225, 1992.
- [169] J. A. G. Willard. Vehicle routing using r-optimal tabu search. Master's thesis, The Management School, Imperial College, Londres, Angleterre, 1989.
- [170] J. Xu et J. P. Kelly. A network flow-based tabu search heuristic for the vehicle routing problem. *Transportation Science*, 30 : 379–393, 1996.
- [171] Y. Zhenyu, L. Zhang, K. Lishan et L. Guangming. A new MOEA for multi-objective TSP and its convergence property analysis. Dans C. M. Fonseca et al., éditeurs, *Evolutionary Multi-criterion Optimization*, volume 2632 de *Lecture Notes in Computer Science*. Springer-Verlag, 2003.
- [172] E. Zitzler. *Evolutionary algorithms for multiobjective optimization : Methods and applications*. Thèse de doctorat, Ecole Polytechnique Fédérale (ETH), Zurich, Suisse, November 1999.
- [173] E. Zitzler, M. Laumanns et S. Bleuler. A tutorial on evolutionary multiobjective optimization. Dans *Workshop on Multiple Objective Metaheuristics (MOMH 2002)*. Springer-Verlag, 2004.
- [174] E. Zitzler, M. Laumanns et L. Thiele. SPEA2 : Improving the strength pareto evolutionary algorithm. Rapport technique 03, Computer Engineering and Networks Laboratory (TIK), Institut Fédéral de Technologie de Zurich (ETH), Zurich, Suisse, 2001.
- [175] E. Zitzler et L. Thiele. Multiobjective evolutionary algorithms : A comparative case study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 3 : 257–271, 1999.
- [176] K. G. Zografos et K. N. Androustopoulos. A heuristic algorithm for solving hazardous materials distribution problems. *European Journal of Operational Research*, 152 : 507–519, 2004.
- [177] J. B. Zydallis, D. A. Van Veldhuizen et G. B. Lamont. A statistical comparison of multiobjective evolutionary algorithm including the MOMGA-II. Dans E. Zitzler, K. Deb, L. Thiele, C. A. Coello Coello et D. Corne, éditeurs, *Proceedings of the First International Conference on Evolutionary Multi-criterion Optimization*, volume 1993 de *Lecture Notes in Computer Science*, pages 226–240. Springer-Verlag, 2001.

Annexe A

Résultats détaillés de la comparaison entre l'algorithme génétique et la méthode exacte sur le PTC bi-objectif

Les différentes colonnes des tableaux correspondent aux informations suivantes :

NB : Nombre de solutions Pareto optimales.

temps ϵ : Temps d'exécution en secondes de la méthode exacte.

Max % : Ratio maximal de solutions Pareto optimales trouvées par l'algorithme génétique.

Moy % : Ratio moyen de solutions Pareto optimales trouvées par l'algorithme génétique.

Min % : Ratio minimal de solutions Pareto optimales trouvées par l'algorithme génétique.

e.t. % : Ecart type des ratios de solutions Pareto optimales trouvées par l'algorithme génétique.

Max DG : Distance générationnelle maximale entre une approximation générée par l'algorithme génétique et l'ensemble Pareto optimal.

Moy DG : Moyenne des distances générationnelles entre les approximations générées par l'algorithme génétique et l'ensemble Pareto optimal.

Min DG : Distance générationnelle minimale entre une approximation générée par l'algorithme génétique et l'ensemble Pareto optimal.

e.t. DG : Ecart type entre les différentes valeurs de la mesure distance générationnelle entre une approximation générée par l'algorithme génétique et l'ensemble Pareto optimal.

Temps AG : Temps moyen d'exécution en secondes de l'algorithme génétique.

TAB. A.1 – Résultats por $|V| = 50$.

$ T $	$ W $	NB	temps ϵ	Max %	Moy %	Min %	e.t. %	Max DG	Moy DG	Min DG	e.t. DG	temps AG
1	50	52	28	0.90	0.89	0.87	0.01	1.85	0.80	0.23	0.55	61
1	50	47	15	0.98	0.98	0.98	0.00	0.00	0.00	0.00	0.00	37
1	50	36	39	0.94	0.93	0.89	0.02	1.25	0.25	0.00	0.50	51
1	50	40	23	0.95	0.93	0.93	0.01	2.26	0.81	0.38	0.73	47
1	50	63	65	0.95	0.93	0.89	0.02	0.67	0.18	0.00	0.25	93
1	100	73	86	0.90	0.85	0.71	0.07	1.71	1.52	1.15	0.23	150
1	100	40	46	0.95	0.95	0.95	0.00	0.00	0.00	0.00	0.00	38
1	100	47	28	0.91	0.90	0.87	0.02	3.94	3.54	3.41	0.20	56
1	100	94	137	0.86	0.82	0.72	0.05	2.84	0.90	0.08	0.99	178
1	100	41	34	0.98	0.95	0.90	0.02	1.66	0.33	0.00	0.66	56
1	150	55	60	0.96	0.96	0.95	0.01	0.11	0.02	0.00	0.04	79
1	150	50	39	1.00	0.97	0.90	0.03	1.27	0.51	0.00	0.48	81
1	150	44	33	0.95	0.93	0.89	0.03	4.43	4.40	4.38	0.02	61
1	150	85	97	0.85	0.77	0.70	0.04	2.53	2.03	0.66	0.71	228
1	150	60	69	0.93	0.89	0.85	0.03	1.25	1.17	1.11	0.05	95
5	50	46	26	0.93	0.91	0.87	0.02	1.58	1.39	0.77	0.31	131
5	50	30	11	0.97	0.88	0.83	0.05	4.07	3.81	3.03	0.39	68
5	50	49	41	1.00	0.97	0.94	0.02	0.12	0.07	0.00	0.06	111
5	50	45	22	0.82	0.77	0.71	0.04	3.88	2.74	2.09	0.61	124
5	50	25	5	0.96	0.93	0.92	0.02	1.72	1.38	0.00	0.69	30
5	100	34	21	0.94	0.91	0.91	0.01	0.00	0.00	0.00	0.00	76
5	100	33	10	1.00	0.98	0.94	0.02	0.00	0.00	0.00	0.00	75
5	100	31	16	0.90	0.90	0.87	0.01	0.00	0.00	0.00	0.00	40
5	100	29	18	1.00	0.99	0.97	0.02	1.65	0.33	0.00	0.66	61
5	100	33	10	0.94	0.93	0.91	0.01	0.82	0.33	0.00	0.40	39
5	150	38	18	0.97	0.96	0.95	0.01	0.00	0.00	0.00	0.00	91
5	150	29	7	1.00	0.96	0.93	0.02	1.16	0.46	0.00	0.57	44
5	150	38	49	0.97	0.96	0.94	0.01	0.13	0.03	0.00	0.05	101
5	150	29	7	1.00	1.00	1.00	0.00	0.00	0.00	0.00	0.00	26
5	150	65	37	1.00	0.97	0.94	0.02	0.47	0.11	0.00	0.18	181
10	50	17	3	1.00	1.00	1.00	0.00	0.00	0.00	0.00	0.00	74
10	50	19	4	1.00	1.00	1.00	0.00	0.00	0.00	0.00	0.00	78
10	50	16	12	0.75	0.69	0.63	0.04	4.26	4.25	3.78	0.71	83
10	50	23	8	1.00	0.94	0.87	0.06	8.54	3.42	0.00	4.10	127
10	50	22	6	0.95	0.95	0.95	0.00	2.03	2.03	2.03	0.00	70
10	100	21	5	1.00	0.99	0.95	0.02	0.00	0.00	0.00	0.00	93
10	100	12	2	1.00	1.00	1.00	0.00	0.00	0.00	0.00	0.00	72
10	100	29	17	0.97	0.97	0.97	0.00	0.00	0.00	0.00	0.00	102
10	100	29	10	0.86	0.86	0.86	0.00	0.00	0.00	0.00	0.00	74
10	100	20	8	1.00	1.00	1.00	0.00	0.00	0.00	0.00	0.00	64
10	150	19	4	1.00	1.00	1.00	0.00	0.00	0.00	0.00	0.00	82
10	150	18	3	1.00	0.92	0.83	0.08	0.99	0.40	0.00	0.48	96
10	150	27	8	1.00	1.00	1.00	0.00	0.00	0.00	0.00	0.00	47
10	150	19	5	0.90	0.90	0.90	0.00	0.00	0.00	0.00	0.00	60
10	150	39	17	0.85	0.84	0.79	0.02	0.48	0.28	0.09	0.12	144

TAB. A.2 – Résultats pour $|V| = 75$.

$ T $	$ W $	NB	temps ϵ	Max	Moy	Min	e.t.	Max	Moy	Min	e.t.	temps
				%	%	%	%	DG	DG	DG	DG	AG
1	75	74	368	0.89	0.86	0.84	0.02	1.17	0.76	0.60	0.21	128
1	75	82	804	0.79	0.73	0.67	0.04	3.95	2.42	1.23	0.96	183
1	75	80	542	0.94	0.91	0.82	0.03	4.95	4.82	4.72	0.08	199
1	75	92	962	0.83	0.79	0.76	0.02	1.74	1.24	0.33	0.52	274
1	75	96	629	0.89	0.80	0.71	0.06	2.11	1.73	1.44	0.26	243
1	150	68	457	0.88	0.87	0.82	0.02	3.10	2.82	2.68	0.15	123
1	150	81	1859	0.69	0.67	0.60	0.03	4.01	2.71	1.70	0.80	267
1	150	98	1449	0.87	0.83	0.81	0.02	3.22	2.46	0.76	0.92	508
1	150	105	2606	0.69	0.66	0.60	0.03	2.79	2.19	1.71	0.35	336
1	150	82	752	0.80	0.72	0.66	0.05	1.84	1.48	1.22	0.27	171
1	225	124	7701	0.55	0.51	0.48	0.03	2.42	2.03	1.76	0.24	526
1	225	100	1874	0.83	0.78	0.70	0.05	2.93	1.64	0.99	0.71	305
1	225	67	292	0.90	0.87	0.85	0.01	1.42	0.71	0.26	0.50	105
1	225	103	912	0.83	0.76	0.71	0.04	0.86	0.53	0.31	0.20	268
1	225	108	2493	0.76	0.70	0.65	0.04	2.88	1.74	1.24	0.59	350
7	75	58	196	0.97	0.91	0.83	0.05	2.52	0.99	0.00	0.85	258
7	75	72	164	0.63	0.56	0.47	0.07	4.36	3.38	2.36	0.72	464
7	75	91	5027	0.71	0.63	0.55	0.05	3.02	2.15	1.20	0.58	697
7	75	60	173	0.85	0.84	0.83	0.01	0.93	0.42	0.19	0.26	269
7	75	44	74	1.00	0.93	0.82	0.07	4.86	1.29	0.00	1.82	190
7	150	63	144	0.92	0.85	0.79	0.04	2.10	1.56	1.10	0.36	287
7	150	40	39	0.58	0.50	0.43	0.05	1.40	1.08	0.66	0.28	83
7	150	56	427	0.98	0.87	0.82	0.06	2.34	0.76	0.00	0.85	265
7	150	26	19	1.00	0.99	0.96	0.02	0.00	0.00	0.00	0.00	60
7	150	62	546	0.89	0.87	0.85	0.02	0.87	0.54	0.16	0.32	160
7	225	45	84	1.00	0.98	0.98	0.01	4.49	1.82	0.00	2.18	183
7	225	52	323	0.58	0.56	0.51	0.02	6.31	5.86	5.57	0.25	134
7	225	42	93	0.98	0.96	0.93	0.02	1.60	0.68	0.18	0.52	176
7	225	64	1611	0.75	0.71	0.63	0.04	2.64	2.28	2.03	0.23	395
7	225	77	805	0.78	0.72	0.69	0.03	6.27	4.71	3.83	0.85	560
15	75	14	4	1.00	1.00	1.00	0.00	0.00	0.00	0.00	0.00	85
15	75	33	56	0.88	0.76	0.70	0.07	0.39	0.30	0.12	0.11	205
15	75	29	14	1.00	1.00	1.00	0.00	0.00	0.00	0.00	0.00	146
15	75	43	123	0.65	0.65	0.63	0.01	5.11	3.94	2.78	0.75	319
15	75	28	21	0.79	0.76	0.75	0.01	1.97	1.91	1.83	0.46	147
15	150	19	12	1.00	0.92	0.79	0.09	3.18	1.24	0.00	1.52	116
15	150	34	54	0.91	0.88	0.85	0.02	1.99	1.26	0.37	0.72	221
15	150	28	59	0.89	0.86	0.82	0.03	0.69	0.40	0.21	0.23	156
15	150	25	215	0.92	0.92	0.92	0.00	0.48	0.48	0.48	0.00	141
15	150	36	60	0.94	0.92	0.89	0.03	1.64	1.35	0.91	0.24	182
15	225	19	6	1.00	1.00	1.00	0.00	0.00	0.00	0.00	0.00	103
15	225	25	22	0.88	0.82	0.76	0.04	4.62	3.43	2.93	0.61	215
15	225	50	85	0.72	0.68	0.64	0.03	5.77	5.38	4.71	0.36	354
15	225	38	56	0.97	0.94	0.89	0.03	4.27	4.01	3.94	0.13	272
15	225	39	43	1.00	0.97	0.95	0.02	2.27	0.92	0.00	1.10	149

TAB. A.3 – Résultats pour $|V| = 100$.

$ T $	$ W $	NB	temps ϵ	Max %	Moy %	Min %	e.t. %	Max DG	Moy DG	Min DG	e.t. DG	temps AG
1	100	129	29776	0.62	0.60	0.57	0.02	4.95	3.28	2.15	0.99	623
1	100	98	6592	0.59	0.59	0.58	0.01	6.07	5.49	4.75	0.57	415
1	100	129	35743	0.60	0.57	0.55	0.02	2.76	2.01	1.49	0.51	740
1	100	127	6758	0.78	0.73	0.69	0.04	2.03	1.76	0.90	0.43	446
1	100	141	19242	0.63	0.59	0.54	0.04	3.74	3.12	2.44	0.51	572
1	200	128	28448	0.73	0.63	0.55	0.06	3.81	2.96	1.69	0.76	703
1	200	159	39325	0.54	0.53	0.52	0.01	4.41	3.08	2.29	0.78	842
1	200	112	15690	0.83	0.76	0.66	0.07	4.91	3.56	1.99	1.09	534
1	200	116	6995	0.78	0.74	0.69	0.04	3.94	2.32	1.12	0.94	447
1	200	142	19426	0.61	0.56	0.50	0.04	2.65	2.12	1.81	0.30	803
1	300	143	15347	0.62	0.52	0.43	0.08	3.35	2.72	1.96	0.58	1414
1	300	98	8750	0.79	0.73	0.68	0.04	2.44	1.83	0.73	0.61	500
1	300	102	6335	0.82	0.77	0.67	0.06	2.49	1.04	0.31	0.91	389
1	300	123	37622	0.76	0.73	0.69	0.02	2.57	2.14	1.94	0.23	773
1	300	146	41065	0.56	0.54	0.50	0.02	4.30	2.98	2.01	0.76	1121
10	100	63	3193	0.73	0.69	0.65	0.03	3.02	2.63	1.42	0.61	414
10	100	50	157	0.50	0.56	0.48	0.04	9.34	7.39	6.36	1.10	176
10	100	70	410	0.80	0.79	0.77	0.01	4.56	3.04	1.58	0.98	543
10	100	38	103	0.97	0.97	0.97	0.00	0.00	0.00	0.00	0.00	157
10	100	86	896	0.80	0.78	0.76	0.02	3.15	1.97	0.85	0.83	856
10	200	60	495	0.98	0.79	0.62	0.13	4.23	2.02	0.00	1.79	400
10	200	61	492	0.84	0.82	0.75	0.03	5.98	2.80	0.48	1.87	447
10	200	48	555	0.90	0.77	0.73	0.06	3.37	2.55	0.07	1.25	375
10	200	72	2582	0.81	0.76	0.72	0.03	1.95	1.23	0.74	0.40	678
10	200	42	556	0.95	0.90	0.85	0.03	5.42	2.12	0.00	2.06	287
10	300	60	649	0.95	0.91	0.85	0.04	0.61	0.49	0.43	0.07	415
10	300	72	1897	0.88	0.80	0.74	0.05	1.84	1.24	0.55	0.43	633
10	300	107	17554	0.63	0.59	0.56	0.02	2.19	1.79	1.44	0.29	1086
10	300	98	7426	0.63	0.58	0.52	0.04	3.61	3.39	3.12	0.16	1089
10	300	75	2020	0.84	0.80	0.76	0.03	3.61	0.94	0.21	1.34	714
20	100	44	362	0.82	0.75	0.61	0.08	4.16	2.48	1.68	0.90	710
20	100	12	4	1.00	1.00	1.00	0.00	0.00	0.00	0.00	0.00	152
20	100	48	149	0.60	0.56	0.48	0.05	4.00	2.87	1.43	0.94	700
20	100	12	6	0.50	0.50	0.50	0.00	5.90	5.25	4.43	0.62	519
20	100	41	157	0.88	0.77	0.61	0.10	3.73	3.08	2.50	0.46	574
20	200	16	13	0.94	0.94	0.94	0.00	0.00	0.00	0.00	0.00	135
20	200	49	323	0.67	0.63	0.59	0.03	8.56	8.30	8.11	0.15	632
20	200	44	144	0.61	0.59	0.55	0.03	5.20	4.70	4.55	0.25	355
20	200	71	1507	0.66	0.62	0.55	0.04	4.12	3.07	2.28	0.62	1500
20	200	41	400	0.68	0.67	0.66	0.01	6.39	6.36	6.31	0.04	324
20	300	38	98	0.79	0.70	0.58	0.09	7.01	2.87	1.15	2.11	835
20	300	35	812	0.80	0.75	0.69	0.04	4.97	3.82	2.31	0.95	557
20	300	29	390	0.38	0.38	0.38	0.00	13.11	8.40	6.83	2.36	470
20	300	54	148	0.59	0.57	0.54	0.02	2.28	1.65	1.13	0.47	526
20	300	16	17	0.88	0.88	0.88	0.00	1.03	1.03	1.03	0.00	240

TAB. A.4 – Résultats pour $|V| = 120$.

$ T $	$ W $	NB	temps ϵ	Max %	Moy %	Min %	e.t. %	Max DG	Moy DG	Min DG	e.t. DG	temps AG
12	120	50	609	0.88	0.80	0.70	0.07	4.94	1.12	0.04	1.92	569
12	120	69	1814	0.75	0.70	0.64	0.05	3.58	2.47	1.95	0.58	489
12	120	19	53	1.00	0.97	0.84	0.06	0.08	0.02	0.00	0.03	98
12	120	47	2072	0.91	0.85	0.81	0.05	3.96	1.98	0.66	1.19	368
12	120	76	24776	0.72	0.64	0.54	0.06	5.12	3.90	2.02	1.03	673
12	240	85	5192	0.69	0.67	0.65	0.02	5.28	4.64	2.94	0.88	1129
12	240	70	777	0.83	0.77	0.73	0.03	3.57	1.16	0.25	1.23	502
12	240	88	11446	0.72	0.64	0.57	0.05	2.73	2.48	2.22	0.20	963
12	240	89	9467	0.75	0.69	0.66	0.03	7.33	3.76	1.65	2.10	1424
12	240	84	32239	0.81	0.72	0.62	0.08	1.75	1.40	0.77	0.36	1149
12	360	99	11154	0.39	0.35	0.31	0.03	4.82	4.42	3.94	0.35	1233
12	360	102	34350	0.55	0.46	0.35	0.06	3.30	2.17	1.19	0.76	1881
12	360	117	8083	0.53	0.49	0.42	0.04	5.59	4.29	2.66	1.06	1746
12	360	82	10009	0.55	0.52	0.49	0.02	2.57	2.79	2.29	1.77	744
12	360	68	13811	0.71	0.65	0.57	0.05	4.44	3.89	3.49	0.32	613
24	120	46	566	0.87	0.83	0.80	0.02	7.87	4.53	0.28	2.78	590
24	120	60	829	0.77	0.72	0.68	0.03	3.71	2.49	2.04	0.62	1135
24	120	9	29	0.11	0.11	0.11	0.00	10.85	10.62	9.71	4.53	182
24	120	38	85	0.89	0.83	0.68	0.08	6.91	4.04	1.48	2.40	365
24	120	23	163	0.91	0.87	0.83	0.03	1.40	0.32	0.05	0.54	229
24	240	35	257	0.86	0.72	0.66	0.07	4.43	2.38	0.91	1.67	437
24	240	57	1498	0.95	0.89	0.74	0.08	3.42	1.59	0.37	1.12	748
24	240	59	12264	0.71	0.62	0.54	0.06	1.55	1.26	0.86	0.30	1038
24	240	48	194	0.71	0.68	0.65	0.03	8.16	7.72	7.39	0.29	634
24	240	64	1354	0.41	0.40	0.39	0.01	7.04	5.70	4.76	0.83	1145
24	360	59	2557	0.61	0.58	0.54	0.02	3.57	3.47	3.35	0.08	1276
24	360	62	839	0.82	0.76	0.68	0.05	1.81	1.43	1.07	0.32	1208
24	360	30	1204	1.00	0.99	0.97	0.02	0.55	0.11	0.00	0.22	407
24	360	48	406	0.77	0.68	0.64	0.05	3.08	1.93	1.33	0.62	744
24	360	51	313	0.88	0.83	0.76	0.05	1.83	0.89	0.38	0.52	572