



HAL
open science

Méthodes exactes pour les problèmes combinatoires bi-objectif: Application sur les problèmes de tournées de véhicules

Estele Glize

► **To cite this version:**

Estele Glize. Méthodes exactes pour les problèmes combinatoires bi-objectif: Application sur les problèmes de tournées de véhicules. Automatique / Robotique. INSA de Toulouse, 2019. Français. NNT: 2019ISAT0023 . tel-02404217v2

HAL Id: tel-02404217

<https://theses.hal.science/tel-02404217v2>

Submitted on 4 Mar 2020

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

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

Université Fédérale



Toulouse Midi-Pyrénées

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ FÉDÉRALE TOULOUSE MIDI-PYRÉNÉES

Délivré par :

l'Institut National des Sciences Appliquées de Toulouse (INSA de Toulouse)

Présentée et soutenue le *29 novembre 2019* par :

ESTÈLE GLIZE

**Méthodes exactes pour les problèmes combinatoires bi-objectif :
Application aux problèmes de tournées de véhicules**

JURY

M. DOMINIQUE FEILLET	Professeur	Président du jury
M. FABIEN LEHUÉDÉ	Professeur	Rapporteur
M. AZIZ MOUKRIM	Professeur des Universités	Rapporteur
M. FABIEN TRICOIRE	Assistant Professor	Examineur
M. NICOLAS JOZEFOWIEZ	Professeur des Universités	Directeur de thèse
MME SANDRA ULRICH NGUEVEU	Maître de conférence	Co-directrice de thèse

École doctorale et spécialité :

EDSYS : Informatique 4200018

Unité de Recherche :

Laboratoire d'analyse et d'architecture des systèmes

Directeur(s) de Thèse :

Nicolas JOZEFOWIEZ et Sandra Ulrich NGUEVEU

Rapporteurs :

Aziz MOUKRIM et Fabien LEHUEDE

Remerciements

En premier lieu, je voudrais remercier Monsieur L  hu  d   Fabien, professeur des universit  s    l'Ecole des Mines de Nantes, et Monsieur Moukrim Aziz, professeur des universit  s    l'Universit   de Technologie de Compi  gne, pour avoir   t   rapporteurs de cette th  se. J'exprime ma gratitude    Monsieur Feillet Dominique, professeur aux Mines Saint-  tienne et au LIMOS, pour avoir pr  sid   le jury de th  se et    Monsieur Tricoire Fabien, professeur associ      l'*Institute for Production and Logistics Management*, Johannes Kepler University, pour avoir accept   d'  tre examinateur.

Ensuite, je tiens    remercier mes directeurs de th  se, Nicolas Jozefowicz et Sandra Ulrich Ngueveu, pour m'avoir donn   l'opportunit   de mener    bien les recherches de cette th  se. Leur encadrement, accompagn   de conseils et de soutien tout au long de cette th  se, m'a permis de mener    bien cette aventure.

I would like to add a special thank to the Associate Professor Roberto Roberti, researcher at the Vrije Universiteit in Amsterdam, for hosting me during three months in his team, and providing me great advice and ideas. Working with him was really stimulating, and meeting all his colleagues was a great pleasure.

L'  quipe ROC du LAAS-CNRS a   t   un cadre tr  s agr  able pour faire cette th  se et je tiens donc    remercier tous les membres de l'  quipe, permanents et non permanents, pour leur sympathie et leur bonne humeur. En particulier, merci    Simon, acolyte de th  se sans qui je n'aurais pas pu d  verser mon flot de paroles interminable.

Enfin, je n'aurais pas pleinement appr  ci   ces trois ann  es sans Thibault, ma famille et mes amis qui sont pr  sents au quotidien.

Résumé

De nombreux problèmes réels comportent plusieurs critères à considérer simultanément. À titre d'exemple, un trajet peut se caractériser par son coût, son impact écologique, son temps de parcours ou encore sa longueur. Les problèmes mathématiques résultants relèvent de l'optimisation multi-objectif. En général, il n'existe pas de solution réalisable optimisant tous les objectifs. Ainsi, les décideurs veulent analyser le compromis entre tous les objectifs pour pouvoir choisir la solution la plus adéquate. Par conséquent, résoudre un problème multi-objectif consiste à trouver un sous-ensemble de points, dits non dominés, dans l'espace des objectifs. Ces points sont associés à des solutions réalisables pour lesquelles il n'est pas possible d'améliorer un objectif sans en détériorer un autre. Peu de méthodes exactes existent dans la littérature pour traiter les problèmes combinatoires multi-objectif \mathcal{NP} -difficiles, en particulier ceux dont la variante mono-objectif est déjà \mathcal{NP} -difficile.

Cette thèse s'inscrit dans l'étude des méthodes exactes pour de tels problèmes multi-objectif et utilise la classe des problèmes de tournées de véhicules bi-objectif comme référence. Les travaux se concentrent sur une approche basée sur la génération de colonnes et qui vise à énumérer efficacement l'ensemble des points non dominés de ces problèmes. Nous proposons notamment d'analyser diverses techniques d'exploration de l'espace des objectifs et de les améliorer grâce à des propriétés structurelles. Afin d'en démontrer la généralité, l'approche est appliquée à plusieurs variantes bi-objectif des problèmes de tournées de véhicules : le problème de tournées de véhicules avec fenêtre de temps, le problème de tournées couvrantes (Covering Tour Problem) et le problème de course d'orientation par équipes avec fenêtre de temps (Team-Orienteering Problem with Time Windows). Les multiples tests numériques soulignent l'efficacité de la méthode proposée.

Mots-clés : Programme linéaire en nombres entiers - Optimisation multi-objectif - Génération de colonnes - Problème de tournées de véhicules - Méthodes exactes - Optimisation combinatoire

Abstract

Real-world problems involve several different criteria to take into account. For instance, a route may be associated with multiple features such as its cost, its ecological footprint, its duration, or its length. Resulting mathematical problems are addressed by multi-objective optimization. In general, there is no feasible solution able to maximize or minimize all objectives. Thus, decision makers want to examine the trade-off between the objectives in order to select the most suitable solution for them. Solving a multi-objective optimization problem consists of finding a set of points in the objective space, called non dominated points. No point in the objective space is better than a point of this set for all objectives. Few exact methods exist in literature to solve \mathcal{NP} -hard multi-objective combinatorial problems, especially those with a \mathcal{NP} -hard mono-objective variant.

This thesis works on exact methods for such multi-objective problems, and the class of bi-objective vehicle routing problems is used as reference. The manuscript presents a column generation based-approach which aims to efficiently enumerate the set of non dominated points of the problems. We seek the best way to explore the objective space, and we propose different acceleration techniques based on structural properties. To show its generic aspect, the approach is applied to several bi-objective variants of the vehicle routing problem : the vehicle routing problem with time windows, the covering tour problem and the team-orienteeing problem with time windows. Extensive computational experiments highlight the efficiency of the proposed method.

Keywords : Integer linear programming - Multi-objective optimization - Column generation - Vehicle routing problem - Exact methods - Combinatorial optimization

Table des matières

Table des figures	ix
Liste des tableaux	xii
Liste des algorithmes	xiii
Notations	xv
Introduction	1
1 Optimisation combinatoire multi-objectif : état de l'art	5
1.1 Introduction	5
1.2 Notions de base	6
1.2.1 Principe	6
1.2.2 Caractérisation des solutions	7
1.2.3 Bornes supérieures et inférieures	9
1.2.4 Évaluation des fronts	10
1.3 Méthodes exactes	11
1.3.1 Méthode basée la somme pondérée des objectifs	12
1.3.2 Méthode ε -contrainte	15
1.3.3 Méthode ε -contrainte avec changement de l'ordre d'exploration	17
1.3.4 Méthode de séparations et évaluations	17
1.3.5 Méthode basée sur la programmation dynamique	20
1.3.6 Conclusion	20
1.4 Conclusion du chapitre	21
2 Problèmes de tournées de véhicules : état de l'art	23
2.1 Introduction	23
2.2 Formulations mathématiques	24
2.2.1 Formulation de flux de véhicules à deux indices	24
2.2.2 Formulation de flux de marchandises à deux indices	25
2.2.3 Formulation de partitionnement	25
2.3 Génération de colonnes	26
2.3.1 Principe	27
2.3.2 Problème de plus court chemin élémentaire avec contraintes de res- sources	28
2.3.3 Techniques de stabilisation	33
2.4 Méthodes exactes	34
2.4.1 Méthode de séparations et évaluations	34
2.4.2 Méthode de séparations et coupes	35

2.4.3	Méthode de séparations et génération de colonnes	36
2.4.4	Méthode de séparations et coupes et génération de colonnes	36
2.4.5	Énumération explicite des colonnes	36
2.5	Problèmes de tournées de véhicules multi-objectif	36
2.5.1	Réduction de la pollution	37
2.5.2	Équité entre les véhicules	37
2.5.3	Satisfaction des usagers	38
2.5.4	Réduction du risque	39
2.5.5	Collecte d'un profit	39
2.5.6	Conclusion	40
2.6	Conclusion du chapitre	40
3	Méthodes exactes pour les problèmes de tournées de véhicules bi-objectif	41
3.1	Introduction	42
3.2	Formulations mathématiques	43
3.2.1	Formulation bi-objectif	43
3.2.2	Formulation pour la méthode ε -contrainte	44
3.2.3	Formulation de la somme pondérée des objectifs	44
3.2.4	Conclusion	45
3.3	Résolution des problèmes mono-objectif avec la génération de colonnes . . .	46
3.3.1	Étape 1 : Solution optimale de la relaxation linéaire	46
3.3.2	Étape 2 : Solution réalisable du problème entier	48
3.3.3	Étape 3 : Énumération de colonnes	48
3.3.4	Étape 4 : Solution optimale du problème entier	49
3.3.5	Conclusion	51
3.4	Méthodes pour le problème bi-objectif	52
3.4.1	Calcul des points extrêmes	52
3.4.2	Méthode 1 : basée sur l'approche ε -contrainte avec génération de toutes les colonnes	52
3.4.3	Méthode 2 : basée sur l'approche ε -contrainte par étapes	53
3.4.4	Méthode 3 : deux phases avec génération de toutes les colonnes pour les points supportés	53
3.4.5	Méthode 4 : deux phases par étapes	56
3.4.6	Conclusion	61
3.5	Résultats expérimentaux	61
3.5.1	Description des instances	62
3.5.2	Comparaison des méthodes	63
3.5.3	Conclusion	67
3.6	Conclusion du chapitre	67
4	Étude de la méthode ε-contrainte	69
4.1	Introduction	69
4.2	Analyse de la méthode ε -contrainte par étapes	70

4.2.1	Décomposition de la méthode	70
4.2.2	Analyse de l'ensemble des points non dominés	72
4.2.3	Ajout d'inégalités valides	74
4.2.4	Conclusion	77
4.3	Améliorations de la méthode ε -contrainte par étapes	77
4.3.1	Réduction du gap	78
4.3.2	Construction de la borne inférieure	79
4.3.3	Validation des points non dominés	81
4.3.4	Gestion et sauvegarde de colonnes	84
4.3.5	Multiple coûts duaux	86
4.3.6	Exploration bidirectionnelle	88
4.3.7	Conclusion	89
4.4	Nouvelle méthode ε -contrainte	89
4.4.1	Description	90
4.4.2	Résultats expérimentaux	92
4.4.3	Conclusion	93
4.5	Conclusion du chapitre	94
5	Application au problème de la course d'orientation par équipes bi-objectif avec fenêtre de temps	95
5.1	Introduction	95
5.2	État de l'art	96
5.2.1	Problème de la course d'orientation	97
5.2.2	Problème de la course d'orientation par équipes	97
5.2.3	Problème de la course d'orientation par équipes bi-objectif	99
5.3	Méthodes exactes pour le problème de la course d'orientation par équipes bi-objectif	99
5.3.1	Formulations mathématiques	99
5.3.2	Adaptation de SEM	101
5.3.3	Adaptation de ISEM	102
5.3.4	Conclusion	102
5.4	Résolution de la formulation ε -contrainte	102
5.4.1	Algorithme d'étiquetage	102
5.4.2	Règles de dominance	103
5.4.3	Bornes de complétion	103
5.5	Résultats expérimentaux	104
5.5.1	Conclusion	106
5.6	Conclusion du chapitre	106
6	Application au problème de tournées couvrantes	107
6.1	Introduction	107
6.2	État de l'art	110
6.2.1	Problème de la tournée couvrante	110

6.2.2	Problème de tournées couvrantes	111
6.2.3	Problème de la tournée couvrante bi-objectif	112
6.2.4	Problème de tournées couvrantes bi-objectif	113
6.3	Méthode exacte pour le problème de tournées couvrantes	114
6.3.1	Formulation mathématique	114
6.3.2	Étape 1 : Calcul d'une borne inférieure LB	116
6.3.3	Étape 2 : Calcul d'une borne supérieure UB	121
6.3.4	Étape 3 : Énumération des colonnes dans le gap	122
6.3.5	Étape 4 : Calcul de la solution optimale entière	122
6.3.6	Détails d'implémentation	122
6.4	Extension au problème de tournées couvrantes bi-objectif	123
6.4.1	Formulations mathématiques	123
6.4.2	Adaptation de SEM	124
6.4.3	Adaptation de ISEM	125
6.5	Résultats expérimentaux	125
6.5.1	Problème de tournées couvrantes	125
6.5.2	Problème de tournées couvrantes bi-objectif	128
6.5.3	Conclusion	130
6.6	Conclusion du chapitre	130
	Conclusions et perspectives	131
	Liste de publications	135
	A Résultats expérimentaux du problème de tournées de véhicules bi-objectif avec fenêtres de temps	137
A.1	Comparaison des méthodes d'exploration de l'espace des objectifs	137
A.2	Comparaison de SEM et ISEM	137
	B Résultats expérimentaux du problème de la course d'orientation par équipes bi-objectif avec fenêtre de temps	145
	C Résultats expérimentaux du problème de tournées couvrantes	149
C.1	Comparaison des méthodes	149
C.2	Comparaison des bornes de complétion	149
	Bibliographie	155

Table des figures

1.1	Correspondance entre l'espace des solutions et l'espace des objectifs	7
1.2	Caractérisations des points de l'espace des objectifs	8
1.3	Ordre naturel des points non dominés	9
1.4	Exemple d'une borne inférieure pour un problème bi-objectif	11
1.5	Méthode dichotomique pour un problème bi-objectif	13
1.6	Zone de recherche pour deux points non dominés supportés consécutifs. . .	14
1.7	Méthode ε -contrainte pour un problème bi-objectif	16
1.8	Méthode <i>Balanced Box</i> pour un problème de minimisation bi-objectif	18
2.1	Processus de génération de colonnes	28
2.2	Expansion du ng-ensemble sur un graphe de VRP	31
3.1	Première phase de la méthode en deux phases	56
3.2	Première phase de la méthode en deux phases par étapes	60
4.1	Détail des temps d'exécution des fonctions pour certaines instances	73
4.2	Points non dominés des instances <i>R107_C1_25</i> et <i>RC201_C1_25</i>	75
4.3	Points non dominés des instances <i>C102_R1_25</i> et <i>R103_C1_25</i>	76
4.4	Écart relatif entre SEM et SEM avec T_{gap}	79
4.5	Écart relatif entre SEM et SEM avec T_{lb}	81
4.6	Cas d'utilisation de la technique T_{valid}	83
4.7	Écart relatif entre SEM et SEM avec T_{valid}	84
4.8	Écart relatif entre SEM et SEM avec T_{save}	86
4.9	Illustration des bornes inférieures et des gaps obtenus avec T_{mult_dual}	87
4.10	Écart relatif entre SEM et SEM avec T_{mult_dual}	88
4.11	Écart relatif entre SEM et SEM avec T_{bidir}	89
6.1	Exemple du MCTP	109
6.2	Qualités d'une estimation d'un front de Pareto	134

Liste des tableaux

3.1	Caractéristiques des différentes méthodes bi-objectif.	61
3.2	Caractéristiques des instances du BOVRP pour $ V = 25$	63
3.3	Caractéristiques des instances du BOVRP pour $ V = 50$	64
3.4	Nombre d'instances fermées et moyenne des temps de résolution des méthodes selon la classe d'instances $R_$, $RC_$ et $C_$	64
3.5	Résultats expérimentaux détaillés des méthodes 1 et 3.	65
3.6	Résultats expérimentaux détaillés des méthodes 2 et 4.	66
4.1	Détails de la méthode ε -contrainte par type d'instances.	71
4.2	Caractéristiques des ensembles de points non dominés selon le coût c^2 pour $V = 25$	74
4.3	Détails de la méthode ε -contrainte par type d'instances.	77
4.4	Nombre d'instances fermées et moyenne des temps de résolution des méthodes selon la classe d'instances pour $ V = 25$	93
5.1	Nombre d'instances fermées par des méthodes sur les différentes classes d'instances.	105
5.2	Moyenne des résultats expérimentaux des méthodes sur les différentes classes d'instances.	105
6.1	Tableau sur la littérature du BOCTP.	114
6.2	Synthèse des résultats de notre méthode et celle de Hà et al. par classes. . .	126
6.3	Nombre d'instances fermées par notre méthode et celle de Hà et al. par classes. .	126
6.4	Moyennes des résultats de notre méthode avec les bornes de complétion f et g sur des classes d'instances.	127
6.5	Résultats de notre méthode bi-objectif et de SOGA pour trouver la borne inférieure du BOMCTP.	128
6.6	Résultats de notre méthode bi-objectif pour résoudre le BOMCTP.	129
A.1	Résultats expérimentaux des méthodes sur les instances de type R_C et R_{RC} pour $ V = 25$	138
A.2	Résultats expérimentaux des méthodes sur les instances de type RC_C et RC_R pour $ V = 25$	139
A.3	Résultats expérimentaux des méthodes sur les instances de type C_C , C_R et C_{RC} pour $ V = 25$	140
A.4	Résultats expérimentaux des méthodes sur les instances avec $ V = 50$	141
A.5	Résultats expérimentaux détaillés de SEM et de ISEM sur les instances de type R_C et R_{RC} pour $ V = 25$	142
A.6	Résultats expérimentaux détaillés de SEM et de ISEM sur les instances de type RC_C et RC_R pour $ V = 25$	143

A.7	Résultats expérimentaux détaillés de SEM et de ISEM sur les instances de type C_C , C_RC et R_RC pour $ V = 25$.	144
B.1	Résultats expérimentaux des méthodes sur les instances de type $c101$ et $pr01$.	146
B.2	Résultats expérimentaux des méthodes sur les instances de type $r101$ et $rc101$.	147
C.1	Résultats de notre méthode et celle de Hà et al. sur des instances avec $ V = 100$ et $ M = 0$.	150
C.2	Résultats de notre méthode et celle de Hà et al. sur des instances avec $ V = 100$ et $ M > 0$.	151
C.3	Résultats de notre méthode et celle de Hà et al. sur des instances avec $ V = 200$.	152
C.4	Comparaison des bornes de complétion f et g sur des instances avec $ V = 100$ et $ M \geq 0$.	153
C.5	Comparaison des bornes de complétion f et g sur des instances avec $ V = 200$.	154

Liste des algorithmes

1	Algorithme ε -contrainte avec génération de toutes les colonnes	53
2	Algorithme ε -contrainte par étapes	54
3	Algorithme de la méthode en deux phases avec génération de toutes les colonnes pour les points supportés	57
4	Algorithme pour obtenir un point supporté	58
5	Algorithme de la méthode en deux phases par étapes	59
6	Algorithme de la méthode ε -contrainte par étapes améliorée	91

Notations

Acronyme	Signification
B&B	Méthode de séparations et évaluations
B&C	Méthode de séparations et coupes
B&C&P	Méthode de séparations et coupes et génération de colonnes
B&P	Méthode de séparations et génération de colonnes
BOCTP	Problème de la tournée couvrante bi-objectif
BOMCTP	Problème des tournées couvrantes bi-objectif
BOTOP	Problème de la course d'orientation par équipes bi-objectif
BOTOPTW	Problème de la course d'orientation par équipes bi-objectif avec fenêtre de temps
BOVRP	Problème de tournées de véhicules bi-objectif
BOVRPTW	Problème de tournées de véhicules bi-objectif avec fenêtre de temps
CTP	Problème de la tournée couvrante
CVRP	Problème de tournées de véhicules avec capacité
ESPPRC	Problème de plus court chemin élémentaire avec contraintes de ressources
ISEM	Méthode ε -contraintes par étapes améliorée
LB	Borne inférieure
MCTP	Problème des tournées couvrantes
MOP	Problème d'optimisation multi-objectif
ng-SPPRC	Problème de plus court ng-chemin avec contraintes de ressources
OP	Problème de la course d'orientation
PDL	Problème Dual Linéaire
PDLR	Problème Dual Linéaire Restreint
PM	Problème Maître
PML	Problème Maître Linéaire
PMLR	Problème Maître Linéaire Restreint
PMR	Problème Maître Restreint
SEM	Méthode ε -contraintes par étapes
TOP	Problème de la course d'orientation par équipes
TSP	Problème du voyageur de commerce
UB	Borne supérieure
VRP	Problème de tournées de véhicules

Introduction

Contexte

Un problème d'optimisation combinatoire mono-objectif consiste à trouver la meilleure solution parmi un ensemble fini ordonnable de choix. La modélisation nécessite de définir un ensemble de contraintes posées sur les solutions et une fonction objectif représentant le critère à optimiser. Lorsqu'il s'agit d'un problème de minimisation (respectivement maximisation), la meilleure solution, ou solution optimale, est celle qui minimise (respectivement maximise) ce critère tout en respectant les contraintes. Deux approches existent pour résoudre ces problèmes : les méthodes exactes et les méthodes approchées. Les premières garantissent de trouver la solution optimale lorsqu'elles sont menées à terme. Leur temps d'exécution peut être long, voire rédhibitoire si le problème à résoudre est \mathcal{NP} -difficile et que la taille des données est importante. Les méthodes approchées, quant à elles, tentent de produire des solutions de très bonne qualité dans un temps limité, mais sans garantie d'optimalité.

La majorité des problèmes d'optimisation combinatoire sont modélisés avec une unique fonction objectif. Cependant, les problèmes réels qu'ils représentent peuvent comporter plusieurs critères à optimiser simultanément. À titre d'exemple, un trajet peut se caractériser par son coût, son impact écologique, son temps de parcours ou encore sa longueur. De même, un paquet peut être défini par son poids, son volume, le nombre d'éléments à l'intérieur ou son prix. Les problèmes mathématiques résultants relèvent alors de l'optimisation combinatoire multi-objectif. Si le décideur ne donne pas de préférence entre les objectifs, un tel problème consiste à chercher un ensemble de solutions pour lesquelles il n'est pas possible d'améliorer un objectif sans en détériorer un autre. Chacune de ces solutions est associée à un point dit non dominé dans l'espace des objectifs. Avec l'obtention de tous les points non dominés, le décideur peut prendre la meilleure décision possible en visualisant le compromis entre tous les objectifs.

De nombreuses méthodes approchées pour les problèmes combinatoires multi-objectif ont été proposées dans la littérature depuis les années 90. Les méthodes exactes sont moins abordées, notamment pour les problèmes dont la variante mono-objectif est déjà \mathcal{NP} -difficile. Cette thèse contribue à développer des méthodes exactes pour de tels problèmes. Pour ce faire, le problème de tournées de véhicules bi-objectif est étudié tout au long du manuscrit. La plupart des versions mono-objectif de ce problème visent à minimiser le coût des trajectoires effectuées par les véhicules. La présence d'un second objectif permet d'ajouter une dimension écologique, sociale ou de satisfaction des usagers par exemple.

Organisation du manuscrit

Ce manuscrit comporte six chapitres. Le Chapitre 1 donne un aperçu de l'optimisation combinatoire multi-objectif en deux parties. Tout d'abord, les notions de base sont

introduites ainsi que les définitions et les terminologies spécifiques telles que les solutions recherchées et les bornes supérieure et inférieure. La seconde section de ce chapitre est focalisée sur les différentes méthodes exactes proposées dans la littérature.

Le Chapitre 2 présente les problèmes de tournées de véhicules. En premier lieu, il y a une description du problème mono-objectif, ainsi que les principales formulations utilisées pour le résoudre. Une seconde section est dédiée à la technique de génération de colonnes, particulièrement utilisée en transport pour générer des bornes inférieures de qualité et qui sera également employée dans cette thèse. Les principales méthodes exactes de résolution des problèmes de tournées de véhicules mono-objectif sont présentées dans une troisième partie. Enfin, ce chapitre se clôt par la littérature sur les méthodes exactes pour les problèmes de tournées de véhicules multi-objectif.

Le premier chapitre de contribution est le Chapitre 3. Il donne une description détaillée du problème bi-objectif étudié, ainsi qu'une formulation mathématique. Pour pouvoir obtenir l'ensemble des points non dominés du problème bi-objectif, plusieurs méthodes exactes sont présentées. Toutes ces méthodes se basent sur la résolution successive de différentes formulations mono-objectif et diffèrent sur la technique d'exploration de l'espace des objectifs. Toutes les formulations mono-objectif utilisées sont introduites au préalable, ainsi qu'une méthode exacte de résolution. Cette méthode se base sur la génération et l'énumération de colonnes. Toutes les approches d'exploration de l'espace des objectifs sont comparées sur des instances que nous avons construites pour un problème de tournées de véhicules bi-objectif avec fenêtres de temps avec deux coûts entiers différents sur chaque arête.

La meilleure approche du Chapitre 3, nommée SEM, est étudiée en détail dans la première partie du Chapitre 4. Cette analyse amorce l'édification de plusieurs techniques d'amélioration qui sont testées indépendamment. Ces techniques sont génériques car elles sont spécifiques à la méthode utilisée, mais pas au problème étudié. Toutes les techniques performantes sont ajoutées à SEM pour former une nouvelle méthode nommée ISEM. Après de nombreux tests numériques sur les mêmes instances que le Chapitre 3, l'efficacité de ISEM est démontrée.

Les deux derniers chapitres se focalisent sur d'autres problèmes de tournées de véhicules bi-objectif pour observer les limites et les avantages de SEM et ISEM. Le Chapitre 5 présente le problème de la course d'orientation par équipes avec fenêtres de temps. Une courte littérature des méthodes exactes de résolution des variantes mono-objectif et bi-objectif de ce type de problème est donnée. Ensuite, les méthodes exactes précédentes sont facilement appliquées au problème. Les changements des méthodes concernent davantage la partie mono-objectif et, en particulier, la résolution des sous-problèmes de la génération de colonnes. Ces spécificités sont alors brièvement expliquées. Les résultats expérimentaux démontrent l'efficacité de ISEM par rapport à SEM et à une méthode de l'état de l'art.

Le Chapitre 6 traite des problèmes de tournées couvrantes. En premier lieu, un état de l'art est proposé pour le problème dans sa version mono-objectif et bi-objectif. Ensuite, l'application de la méthode mono-objectif du Chapitre 3 est expliquée en détail et son efficacité est démontrée par rapport aux méthodes de l'état de l'art. La version bi-objectif étudiée dans ce chapitre diverge des deux précédents problèmes par l'apparition d'un objectif de type *min-max*. SEM est facilement appliquée et ceci montre la genericité

de la méthode. Cependant, les techniques du Chapitre 4 ne s'appliquent pas sur ce type de problème puisque la structure de la formulation mono-objectif est modifiée. Les résultats expérimentaux montrent la pertinence de SEM lors de la comparaison avec une méthode de l'état de l'art pour l'obtention de la borne inférieure.

Le manuscrit se termine par des conclusions générales et les perspectives de ce travail de recherche.

Optimisation combinatoire multi-objectif : état de l'art

Sommaire

1.1	Introduction	5
1.2	Notions de base	6
1.2.1	Principe	6
1.2.2	Caractérisation des solutions	7
1.2.3	Bornes supérieures et inférieures	9
1.2.4	Évaluation des fronts	10
1.3	Méthodes exactes	11
1.3.1	Méthode basée la somme pondérée des objectifs	12
1.3.2	Méthode ε -contrainte	15
1.3.3	Méthode ε -contrainte avec changement de l'ordre d'exploration	17
1.3.4	Méthode de séparations et évaluations	17
1.3.5	Méthode basée sur la programmation dynamique	20
1.3.6	Conclusion	20
1.4	Conclusion du chapitre	21

1.1 Introduction

Ce chapitre traite de l'optimisation combinatoire multi-objectif où plusieurs fonctions objectif doivent être optimisées simultanément. À la fin du dix-neuvième siècle, l'économiste Vilfredo Pareto (1896) a défini l'optimum dit de Pareto comme un état de société où l'on ne peut améliorer le bien-être d'un individu sans détériorer celui d'un autre. Par la suite, cette définition a amené les notions mathématiques de dominance de Pareto et de front de Pareto utilisées dans l'optimisation multi-objectif.

Depuis lors, cette branche de l'optimisation combinatoire est de plus en plus étudiée puisqu'elle reflète de véritables enjeux industriels. En effet, les problèmes réels sont souvent représentés par un unique objectif, même s'ils possèdent de nombreuses caractéristiques qui seraient mieux représentées par l'utilisation de plusieurs fonctions objectif. Cependant, la présence de multiples objectifs entraîne une complexité accrue par rapport aux modèles mono-objectif.

Ce chapitre introduit les concepts de base de l'optimisation multi-objectif en Section 1.2. Ensuite, les méthodes exactes pour résoudre les problèmes multi-objectif sont présentées en Section 1.3. De multiples méthodes approchées sont proposées dans la littérature (Abdel-Basset et al. (2018)), mais elles ne seront pas traitées dans ce manuscrit. Enfin, la conclusion est en Section 1.4.

1.2 Notions de base

Dans cette section, les définitions et les notions de base sont abordées. En premier lieu, la partie 1.2.1 donne la définition de l'optimisation combinatoire multi-objectif. Ensuite, les caractérisations des solutions d'un problème multi-objectif sont présentées, ainsi que les bornes supérieures et inférieures. Enfin, la dernière partie 1.2.4 aborde la notion d'hypervolume pour évaluer les bornes.

1.2.1 Principe

En optimisation combinatoire mono-objectif, le problème consiste à déterminer l'élément optimisant un critère parmi un ensemble fini ordonnable d'éléments. Cet élément est la solution optimale et la valeur de son critère est l'optimum. Dans le domaine multi-objectif, les problèmes visent à optimiser p objectifs simultanément et ils sont définis comme suit :

$$(MOP) \quad \begin{cases} \min c(x) = (c^1(x), c^2(x), \dots, c^p(x)) \\ t.q. x \in \mathcal{X} \end{cases}$$

Dans cette formulation, les critères sont représentés par un vecteur de fonctions objectif c de taille p qui doit être optimisé. L'ensemble $\mathcal{X} \subseteq \mathbb{N}^n$, où n est le nombre de variables, représente l'**espace des solutions**. Les contraintes et l'espace de définition du problème sont implicitement données par la définition de \mathcal{X} . Tout élément x de \mathcal{X} est une solution du problème.

L'image de \mathcal{X} par c forme l'**espace des objectifs** $\mathcal{Y} \subseteq \mathbb{N}^p$. Tout élément y de \mathcal{Y} est l'image d'une solution $x \in \mathcal{X}$ telle que $y^i = c^i(x)$, $\forall i \in \llbracket 1; p \rrbracket$ et est un point dans l'espace des objectifs. Pour alléger les notations, les coûts des objectifs d'un point $y \in \mathcal{Y}$, associé à la solution $x \in \mathcal{X}$, peuvent être représentés avec les notations y^i , $i \in \llbracket 1; p \rrbracket$ au lieu des coûts de la solution $c^i(x)$, $i \in \llbracket 1; p \rrbracket$. La Figure 1.1 montre un exemple de la correspondance entre \mathcal{X} et \mathcal{Y} .

L'ensemble des solutions réalisables d'un problème multi-objectif combinatoire est un ensemble fini ayant pour image un ensemble, potentiellement non convexe, de points dans l'espace des objectifs. L'optimum est donc un ensemble de points qui peuvent être caractérisés de plusieurs manières.

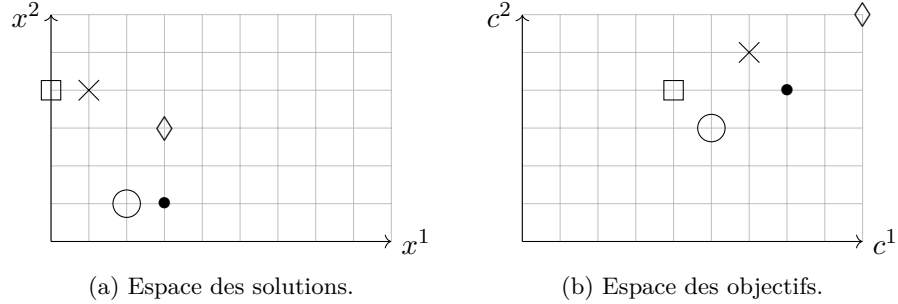


FIGURE 1.1: Correspondance entre l'espace des solutions et l'espace des objectifs dans un problème bi-objectif pour $n = 2$, $p = 2$, $c^1(x) = 2x^1 + x^2$ et $c^2(x) = x^1 + x^2$, $x \in \mathcal{X}$.

1.2.2 Caractérisation des solutions

Pour pouvoir comparer les solutions obtenues en résolvant (MOP), la Définition 1 de dominance de Pareto est utilisée.

Définition 1. *Wicksell et al. (1913)* Soit x et x' deux solutions de \mathcal{X} .

— x domine faiblement x' , noté $x \leq x' \Leftrightarrow c^i(x) \leq c^i(x') \forall i \in \llbracket 1; p \rrbracket$

— x domine x' , noté $x \leq x' \Leftrightarrow \begin{cases} c^i(x) \leq c^i(x') \forall i \in \llbracket 1; p \rrbracket \\ c^i(x) < c^i(x') \exists i \in \llbracket 1; p \rrbracket \end{cases}$

— x domine strictement x' , noté $x < x' \Leftrightarrow c^i(x) < c^i(x') \forall i \in \llbracket 1; p \rrbracket$

Les Définitions 2, 3, 4 et 5 caractérisent les solutions recherchées dans un problème multi-objectif et sont illustrées dans la Figure 1.2.

Définition 2. Une solution $x \in \mathcal{X}$ est dite solution efficace (ou Pareto-optimale) s'il n'existe pas de solution $x' \in \mathcal{X}$, $x' \neq x$, telle que $x' \leq x$.

Une solution $x \in \mathcal{X}$ est dite solution faiblement efficace (ou faiblement Pareto-optimale) s'il n'existe pas de solution $x' \in \mathcal{X}$, $x' \neq x$, telle que $x' < x$.

Définition 3. Un point $y \in \mathcal{Y}$ est dit point non dominé si la solution $x \in \mathcal{X}$ telle que $c(x) = y$ est une solution efficace.

Un point $y \in \mathcal{Y}$ est dit point faiblement non dominé si la solution $x \in \mathcal{X}$ telle que $c(x) = y$ est une solution faiblement efficace.

Définition 4. Un point non dominé $y \in \mathcal{Y}$ est dit supporté s'il est sur l'enveloppe convexe de \mathcal{Y} .

Un point non dominé $y \in \mathcal{Y}$ est dit non supporté s'il est à l'intérieur de l'enveloppe convexe de \mathcal{Y} .

Définition 5. Un point non dominé supporté $y \in \mathcal{Y}$ est dit extrême s'il est un point extrême de l'enveloppe convexe de \mathcal{Y} .

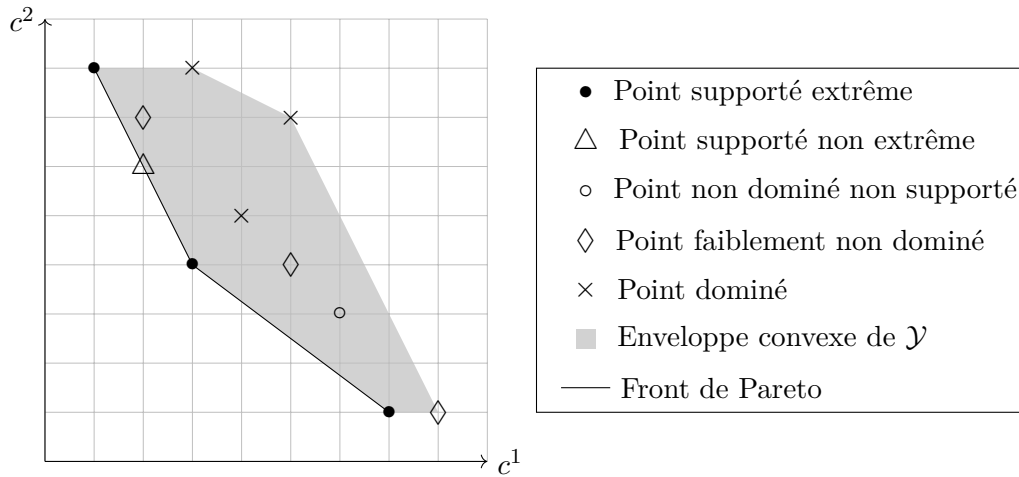


FIGURE 1.2: Différentes caractérisations des points de l'espace des objectifs d'un problème bi-objectif.

Les points non dominés supportés qui ne sont pas des points extrêmes de l'enveloppe convexe de \mathcal{Y} sont appelés non extrêmes.

Dans ce manuscrit, résoudre (MOP) revient à trouver l'**ensemble des points non dominés** du problème, supportés et non supportés, noté \mathcal{Y}_N . Un même point $y \in \mathcal{Y}$ peut être associé à plusieurs solutions distinctes de \mathcal{X} . De ce fait, le nombre de solutions efficaces est plus grand que le nombre de points non dominés. Ainsi, la Définition 6 permet de distinguer deux ensembles complets.

Définition 6. *Hansen (1980) L'ensemble complet minimal de (MOP) est l'ensemble des points non dominés avec au moins une solution efficace associée par point non dominé.*

L'ensemble complet maximal est l'ensemble des points non dominés avec toutes les solutions associées par point.

Note 1. *Dans un problème comportant deux fonctions objectif à minimiser, les points non dominés de \mathcal{Y}_N peuvent être ordonnés selon un ordre naturel qui est la valeur croissante du premier objectif c^1 , pour un problème de minimisation. Deux points non dominés y et y' sont dits consécutifs s'il n'existe pas de point non dominé \hat{y} tel que $y^1 < \hat{y}^1 < y'^1$ (et par conséquent $y^2 > \hat{y}^2 > y'^2$).*

La Figure 1.3 présente l'ordre naturel entre les points non dominés y_i , $i \in \{1, \dots, 5\}$ d'un problème bi-objectif avec minimisation des deux fonctions objectif. Les points y_i et y_{i+1} pour $i \in \{1, \dots, 5\}$ sont dits consécutifs.

La notion de dominance de Pareto permet de définir les solutions d'un problème multi-objectif. Cependant, cette relation de dominance est un ordre partiel. Les points non dominés ne sont donc pas comparables entre eux, s'il n'y a pas de faible dominance. D'autres

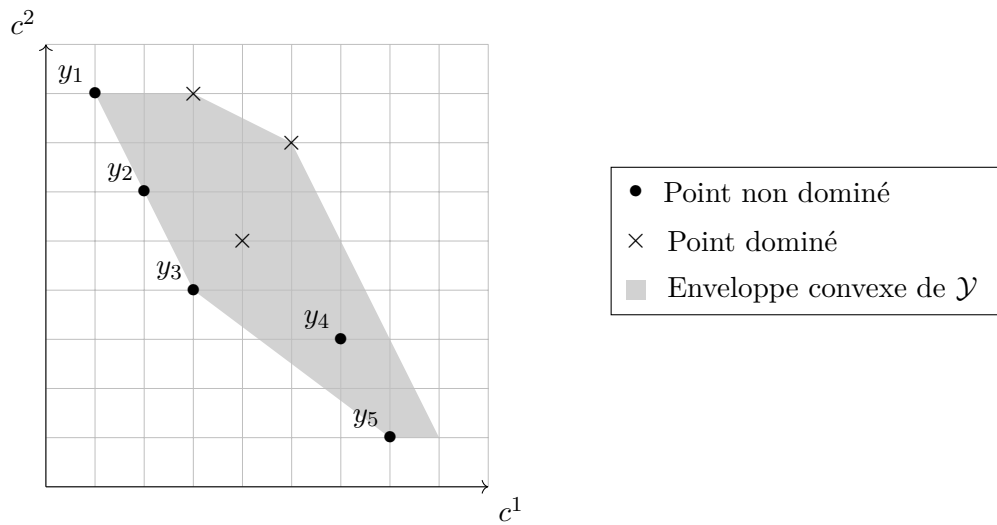


FIGURE 1.3: Ordre naturel des points non dominés d'un problème bi-objectif.

dominances peuvent être utilisées pour établir des préférences entre les solutions comme la dominance lexicographique, présentée en Définition 7.

Définition 7. Soit x et x' deux solutions de \mathcal{X} , x domine x' au sens lexicographique $\Leftrightarrow \exists$ un indice $i^* \leq p$ tel que $c^i(x) = c^i(x')$ pour $i \in \{1, \dots, i^* - 1\}$ et $c^{i^*}(x) < c^{i^*}(x')$.

Une solution réalisable est dite lexicographiquement optimale si elle n'est pas lexicographiquement dominée par une autre solution. Ainsi, sur la Figure 1.3, si nous classons les objectifs par $c = (c^1, c^2)$, alors la solution lexicographiquement optimale est y_1 . Au contraire, si les objectifs sont classés par $c = (c^2, c^1)$, alors la solution lexicographiquement optimale est y_5 . Par la suite, le point y_1 d'objectif c^1 minimal est appelé le point c^1 -extrême et le point y_5 d'objectif c^2 minimal est appelé le point c^2 -extrême. L'ensemble des solutions lexicographiquement optimales est $\{y_1, y_5\}$.

1.2.3 Bornes supérieures et inférieures

Les méthodes d'optimisation mono-objectif reposent sur l'obtention de bornes inférieures et supérieures qui sont des valeurs encadrant le coût de la solution optimale. Dans le cadre de l'optimisation multi-objectif, les bornes ne correspondent pas aux solutions primales et duales, mais à un ensemble de points. Si nous considérons la minimisation du vecteur objectif, une borne inférieure est une relaxation de \mathcal{Y}_N , tandis qu'une borne supérieure est un ensemble de points correspondant à des solutions réalisables et qui ne se dominent pas entre eux.

Les premières bornes ont été définies par deux points spécifiques définissant un rectangle où tous les points non dominés se situent : le point idéal et le point nadir.

Définition 8. Le point idéal y_I de \mathcal{Y}_N est défini par : $y_I^i = \min_{x \in \mathcal{X}} c^i(x)$.

Le point nadir y_N de \mathcal{Y}_N est défini par : $y_N^i = \max_{x \in \mathcal{X}} c^i(x)$.

Nous introduisons également la Définition 9 d'un point idéal local et d'un point nadir local pour définir des zones de recherche précises.

Définition 9. Soit $\hat{\mathcal{Y}} \subseteq \mathcal{Y}$.

Le point idéal local \hat{y}_I de $\hat{\mathcal{Y}}$ est défini par : $\hat{y}_I^i = \min_{y \in \hat{\mathcal{Y}}} y^i$.

De même, le point nadir local \hat{y}_N de $\hat{\mathcal{Y}}$ est défini par : $\hat{y}_N^i = \max_{y \in \hat{\mathcal{Y}}} y^i$.

Dans un problème bi-objectif avec le vecteur objectif $c = (c^1, c^2)$, le point idéal local \hat{y}_I de deux points non dominés consécutifs selon l'ordre naturel y et y' est tel que $\hat{y}_I = (y^1, y'^2)$. De même, le nadir local \hat{y}_N est tel que $\hat{y}_N = (y'^1, y^2)$.

Ces points donnent une précision sur la zone de l'espace des objectifs où se situent des points non dominés. Cependant, ils présentent certains inconvénients comme leur éloignement de l'ensemble des points non dominés, le calcul du point idéal qui revient à résoudre p problèmes potentiellement \mathcal{NP} -difficiles et la difficulté de trouver le nadir lorsque $p > 2$ (Ehrgott et Tenfelde-Podehl (2003)), même sur un problème \mathcal{P} -difficile (Kirlik et Sayin (2014)).

Villarreal et Karwan (1981) ont défini les bornes comme des ensembles de points encadrant \mathcal{Y}_N . Les points de la borne inférieure doivent être des points non dominés ou qui dominent au moins un point de \mathcal{Y}_N . De plus, chaque point de \mathcal{Y}_N doit être dans la borne inférieure ou dominé par au moins un point de la borne inférieure. La borne supérieure, quant à elle, est un ensemble de points appartenant à \mathcal{Y}_N ou dominé par au moins un point de \mathcal{Y}_N . Les points de la borne supérieure ne doivent pas se dominer entre eux.

Par la suite, cette définition a été généralisée par Ehrgott et Gandibleux (2001; 2007). Ils ont introduit la notion d'enveloppe convexe de la borne inférieure. Elle contient les points supportés obtenus par résolution de la relaxation linéaire de la formulation où les objectifs sont transformés en une somme pondérée avec des poids variables (cf. méthode dichotomique en Section 1.3.1). L'enveloppe convexe de la borne inférieure est donc l'ensemble de ces points reliés entre eux par des segments. Par la suite, Sourd et Spanjaard (2008) ont défini la borne inférieure comme une fonction séparant l'espace des objectifs en deux parties de telle sorte que tous les points non dominés soient réunis. Dans la suite de ce manuscrit, nous utiliserons la définition de Sourd et Spanjaard (2008) comme borne inférieure.

1.2.4 Évaluation des fronts

Dans ce manuscrit, nous allons comparer les qualités des bornes inférieures de deux méthodes pour un problème de minimisation à deux objectifs. Cette partie présente la mesure utilisée bien que d'autres indicateurs existent dans la littérature (Zitzler et al. (2002)). L'hypervolume ou S-métrique a été défini par Zitzler et Thiele (1998) pour comparer la qualité des bornes supérieures. Ainsi, nous introduisons une différence puisque nous comparons la qualité de la borne inférieure comprenant des segments et non uniquement des points.

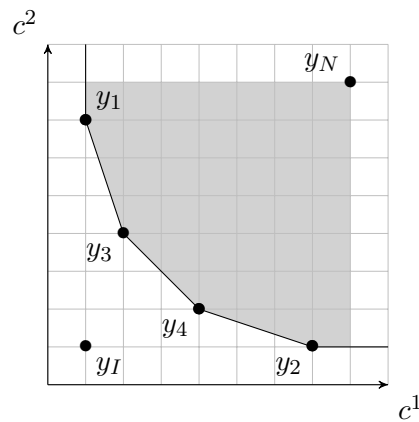


FIGURE 1.4: Exemple d'une borne inférieure pour un problème bi-objectif

L'hypervolume est le pourcentage de la zone de l'espace des objectifs couverte par la borne inférieure, comparé à un point dominé par tous les points non dominés. La Figure 1.4 illustre cette mesure.

Dans cet exemple, le point y_N est dominé par chaque point non dominé de \mathcal{Y}_N et le point y_I est le point idéal qui domine tous les points non dominés. Cela signifie que le front entier est contenu dans le rectangle formé par les points y_I et y_N . Les points y_1 , y_2 , y_3 et y_4 et les lignes entre chaque point successif représentent l'ensemble de la borne inférieure. L'hypervolume de cette borne est l'aire en gris divisée par l'aire du rectangle formé par les points y_I et y_N . Plus la mesure est faible et plus la borne inférieure est précise. Les points y_I et y_N doivent être les mêmes pour les méthodes comparées.

1.3 Méthodes exactes

Bien que la solution d'un problème multi-objectif soit un ensemble, une seule solution sera sélectionnée en pratique par un décideur. Les méthodes peuvent être classifiées selon la phase durant laquelle le décideur intervient :

- Préférence *a priori* où le décideur intervient seulement au début de la résolution pour définir ses préférences entre les objectifs. Ces préférences peuvent être définies indépendamment pour chaque objectif comme l'agrégation des objectifs en une somme pondérée, ou être interactives selon l'ordre des critères (Fouchal et al. (2011)). L'avantage de ces approches est qu'un unique problème mono-objectif est résolu, bien que définir ses attentes en amont de la résolution peut être difficile pour le décideur.
- Préférence *interactive* où le décideur définit ses préférences parmi les objectifs au début de la résolution et intervient également durant le processus de résolution pour la guider. Comme les approches *a priori*, une unique solution est retournée, mais qui est plus fidèle aux attentes du décideur. L'inconvénient majeur de cette approche est le besoin d'intervention du décideur au cours de la résolution qui peut être longue.

- Préférence *a posteriori* où le décideur n'intervient qu'en fin de résolution pour choisir la solution à garder dans l'ensemble proposé. Le décideur connaît alors les différents compromis possibles entre les objectifs avant de prendre une décision. Cependant, cette approche requiert plus de temps de calcul puisque de nombreuses solutions sont obtenues.

Dans ce manuscrit, nous nous intéressons aux approches *a posteriori* qui fournissent un ensemble de solutions efficaces au décideur. Le choix de la solution finale parmi cet ensemble n'est pas abordé. Les méthodes *a posteriori* peuvent paraître inintéressantes par rapport aux deux autres types d'approches par la longueur de la résolution. Toutefois, ce type de méthodes permet au décideur de prendre la meilleure décision en connaissance de tous les compromis qui s'offrent à lui. Prendre une décision en amont de la résolution, sans connaître l'impact sur les autres objectifs du problème peut être risqué.

Dans cette section, nous abordons les différentes méthodes exactes ayant une approche *a posteriori*. Elles peuvent être divisées en deux groupes : les méthodes basées sur l'exploration de l'espace des objectifs (Sections 1.3.1 à 1.3.3) et les méthodes basées sur l'exploration des solutions (Sections 1.3.4 à 1.3.5). Bien que de nombreuses études étendent les méthodes exactes bi-objectif pour des problèmes à plus d'objectifs, nous ne donnerons pas d'explications détaillées puisque seulement des algorithmes bi-objectif sont proposés dans ce manuscrit.

1.3.1 Méthode basée la somme pondérée des objectifs

Toutes les méthodes de cette section se basent sur la formulation (MOP_w) faisant la somme pondérée des objectifs (Geoffrion (1968)).

$$(MOP_w) \left\{ \begin{array}{l} \min c_w(x) = \sum_{i=1}^p w_i c^i(x) \\ t.q. \sum_{i=1}^p w_i = 1 \\ w_i \in [0; 1], i \in \llbracket 1; p \rrbracket \\ x \in \mathcal{X} \end{array} \right.$$

Ainsi, les points non dominés sont obtenus en résolvant une séquence de problèmes mono-objectif, chacun résolu avec la méthode de son choix. Les deux méthodes les plus connues pour résoudre des problèmes bi-objectif via l'utilisation de la somme pondérée sont la méthode dichotomique de Aneja et Nair (1979) et la méthode en deux phases de Ulungu et Teghem (1995). La méthode de recherche perpendiculaire de Chalmet et al. (1986) étend la méthode dichotomique pour trouver l'ensemble complet minimal des points non dominés. Contrairement aux trois méthodes précédentes, la méthode de Sylva et Crema (2004) fixe les poids w et pose des contraintes sur les objectifs pour pouvoir parcourir tout l'espace. Enfin, la dernière méthode présentée se base sur une scalarisation différente des objectifs en utilisant la distance de Tchebycheff.

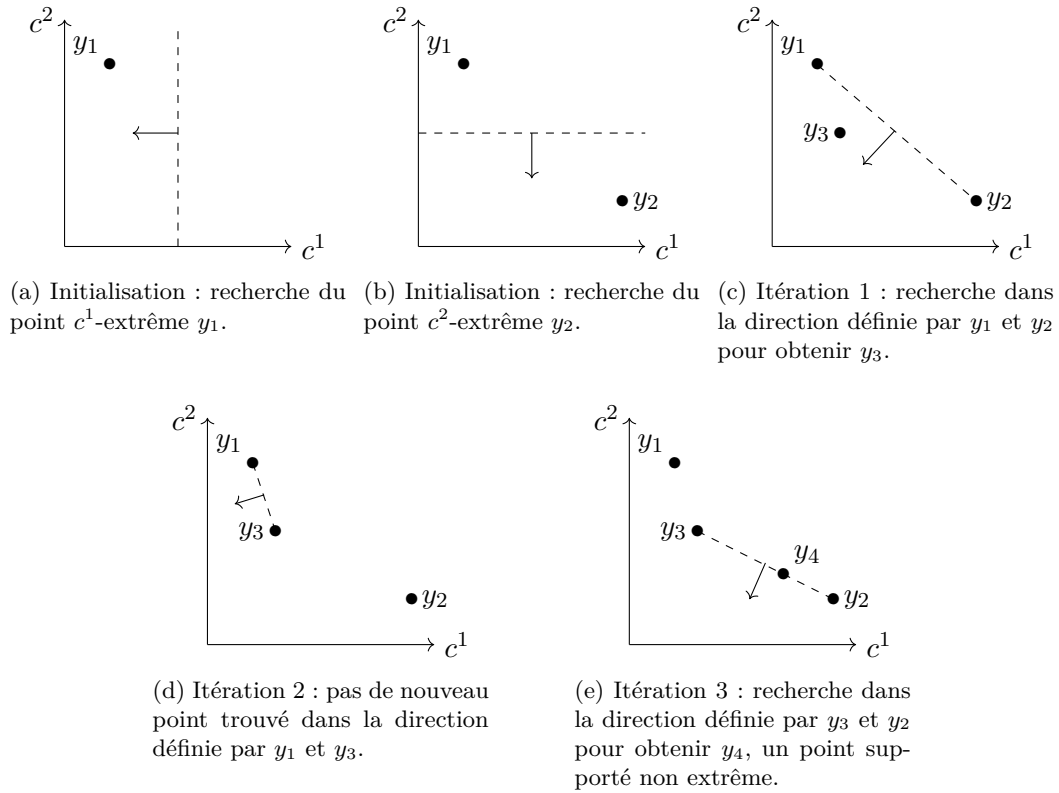


FIGURE 1.5: Méthode dichotomique pour un problème bi-objectif où les points sont des points non dominés supportés et la flèche une direction de recherche dans l'espace des objectifs.

Méthode dichotomique. Aneja et Nair (1979) ont conçu une méthode pour trouver l'ensemble des points non dominés extrêmes et quelques points supportés non extrêmes de problèmes linéaires bi-objectif et l'ont appliquée à des problèmes de transport. La Figure 1.5 illustre la méthode. Premièrement, les points lexicographiquement optimaux y_1 et y_2 des objectifs (c^1, c^2) et (c^2, c^1) sont trouvés. Ensuite, la phase itérative commence. L'espace des objectifs est exploré entre chaque paire de points non dominés consécutifs dans l'ensemble de points courant y et y' , associés aux solutions x et x' . La première paire de points (y, y') est (y_1, y_2) . La direction de recherche \hat{w} pour laquelle $c_{\hat{w}}(x)$ est égal à $c_{\hat{w}}(x')$ est calculée et $(MOP_{\hat{w}})$ est alors résolu. Si la solution \hat{x} de $(MOP_{\hat{w}})$ a un coût inférieur à x et x' , alors un nouveau point non dominé extrême est trouvé. Au contraire, si \hat{x} a un coût égal à x et x' , alors son point non dominé associé est égal à y ou y' ou est un point supporté non extrême.

Przybylski et al. (2010) et Özpeynirci et Köksalan (2010) ont étendu la méthode dichotomique pour qu'elle soit appliquée sur $p > 2$ objectifs. La difficulté de cet algorithme

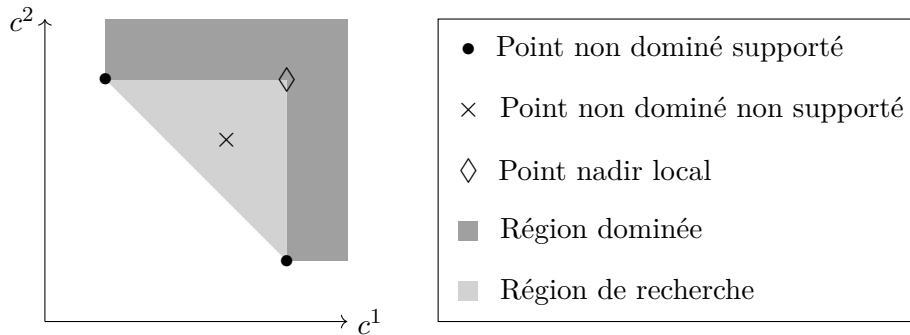


FIGURE 1.6: Zone de recherche pour deux points non dominés supportés consécutifs.

provient alors de la définition de la direction de recherche qui est un hyperplan de dimension $p - 1$.

Méthode en deux phases. [Ulungu et Teghem \(1995\)](#) ont proposé une méthode en deux phases pour la construction de l'ensemble complet minimal d'un problème bi-objectif. La première phase se base sur la méthode dichotomique pour obtenir l'ensemble des points non dominés extrêmes. La seconde phase consiste à trouver tous les points non supportés situés dans les zones de recherche entre deux points supportés consécutifs de la première phase. Une zone est le triangle formé par les deux points consécutifs et leur point nadir local et est illustrée en Figure 1.6. Pour trouver les points non dominés dans chaque triangle indépendamment, [Ulungu et Teghem](#) ont utilisé une méthode d'énumération, mais d'autres techniques peuvent être utilisées comme une méthode de séparations et évaluations ou de programmation dynamique.

Méthode de recherche perpendiculaire. [Chalmet et al. \(1986\)](#) ont introduit une méthode similaire à la méthode dichotomique, excepté que l'ensemble des points non dominés est trouvé. Lors de la phase itérative de recherche entre deux points non dominés consécutifs y et y' , associés aux solutions x et x' telles que $c^1(x) < c^1(x')$, deux contraintes sont rajoutées sur l'espace des objectifs. Une contrainte restreint le coût c^1 de la solution à être inférieur à $c^1(x')$. L'autre contrainte restreint le coût c^2 de la solution à être inférieur à $c^2(x)$.

Méthode de Sylva et Crema. Cette méthode a été présentée par [Sylva et Crema \(2004\)](#), ainsi que [Jahanshahloo et al. \(2005\)](#). Les poids w sont fixés au début de telle sorte que $0 < w_i < 1, \forall i \in \llbracket 1; p \rrbracket$ et $\sum_{i=1}^p w_i = 1$. Le problème (MOP_w) est résolu tant que de nouveaux points non dominés sont trouvés. Lorsqu'un nouveau point est obtenu, des contraintes sur les objectifs sont posées pour éliminer les zones dominées par les points précédemment obtenus et éviter de générer des points déjà obtenus. Bien que le nombre de problèmes mono-objectif traités soit $|\mathcal{Y}_N| + 1$, le problème devient de plus en plus difficile

à résoudre à mesure qu'il est de plus en plus contraint.

Méthode de la somme augmentée de Tchebycheff. Steuer et Choo (1983) ont proposé une méthode qui se base sur la scalarisation des objectifs avec la distance de Tchebycheff par rapport à des points de référence. Des avancées récentes sont proposées par Dächert et al. (2012).

1.3.2 Méthode ε -contrainte

La méthode ε -contrainte a été introduite par Haimès et al. (1971). Elle consiste à conserver un unique objectif k tel que $1 \leq k \leq p$ et à contraindre les autres objectifs. Elle se base sur la formulation suivante :

$$(MOP_\varepsilon) \begin{cases} \min c_\varepsilon(x) = c^k(x) \\ t.q. c^i(x) \leq \varepsilon_i, i \in \llbracket 1; p \rrbracket \text{ et } i \neq k \\ x \in \mathcal{X} \end{cases}$$

L'initialisation de la méthode se fait en trouvant le point c^k -extrême avec des coefficients ε_i , $i \in \llbracket 1; p \rrbracket$ et $i \neq k$, égaux à $+\infty$. Ensuite, la réduction progressive des coefficients ε_i permet de parcourir tout l'espace des objectifs et de trouver l'ensemble complet minimal. La Figure 1.7 illustre le déroulement de la méthode pour un problème minimisant c^1 et contraignant c^2 à être inférieur à une valeur. Le désavantage de cette méthode est que l'ajout de contraintes casse la structure originale du problème. Cela peut rendre sa résolution plus difficile à mesure que les objectifs sont contraints.

Plusieurs axes d'amélioration de la méthode sont envisageables. D'une part, certains auteurs s'intéressent à la technique d'exploration de l'espace des objectifs lorsque $p > 2$ et aux valeurs que prennent les constantes ε_i . En effet, lorsqu'une unique fonction objectif c^2 à valeurs dans \mathbb{N} est contrainte et qu'une nouvelle solution efficace x est trouvée, la constante ε prend la valeur $c^2(x) - 1$. Ce changement n'est pas aussi direct lorsque la fonction objectif contrainte est à valeur dans \mathbb{R} . De plus, le problème se complexifie lorsque plusieurs fonctions objectif sont contraintes. Laumanns et al. (2006) proposent de séparer l'espace des objectifs en une hyper-grille de taille $p - 1$ où chaque case est résolue par une méthode ε -contrainte. Pour les problèmes entiers multi-objectif, Özlen et Azizoğlu (2009) et Özlen et al. (2014) génèrent $\mathcal{Y}_{\mathcal{N}}$ en choisissant des constantes ε les plus précises possibles pour chaque contrainte. Kirlik et Sayın (2014) séparent également l'espace des objectifs en une hyper-grille de dimension $p - 1$. La case de plus grand volume est alors parcourue et les nouveaux points non dominés trouvés forment une nouvelle grille. Leur méthode est plus efficace que celle de Laumanns et al. (2006) et Özlen et al. (2014).

D'autre part, durant la méthode ε -contrainte de Haimès et al. (1971), certains points faiblement non dominés peuvent être générés alors qu'ils ne sont pas intéressants pour un décideur. Éviter de générer ces points permet de réduire le nombre de sous-problèmes

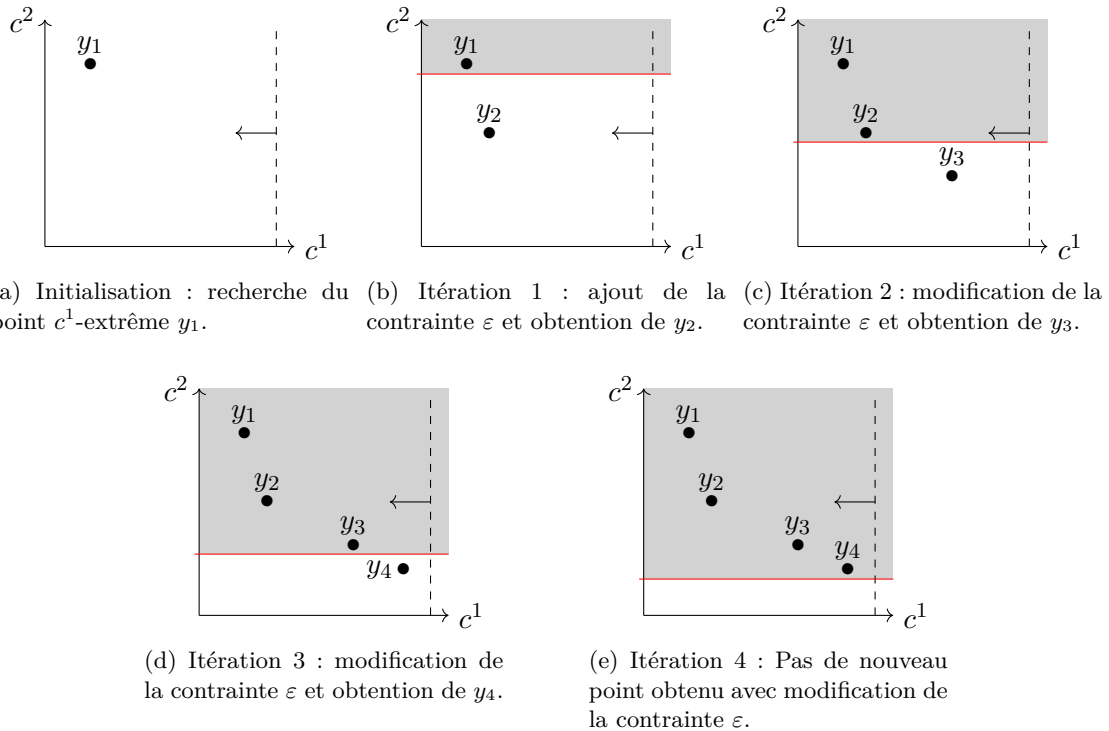


FIGURE 1.7: Méthode ε -contraainte pour un problème bi-objectif où les points sont des points non dominés, la flèche une direction de recherche dans l'espace des objectifs et la droite continue rouge l'ajout d'une contrainte sur l'objectif c^2 .

à résoudre. [Laumanns et al. \(2006\)](#) et [Hamacher et al. \(2007\)](#) ont conçu la méthode ε -contraainte lexicographique qui cherche la meilleure solution pour les objectifs contraints parmi les solutions ayant le même coût c_ε . [Neumayer et Schweigert \(1994\)](#) et [Özlen et Azizoglu \(2009\)](#) ont changé la fonction objectif de (MOP_ε) en y ajoutant les fonctions objectif contraintes pondérées par un petit coefficient, ce qui revient à traiter les objectifs de manière lexicographique. Parmi les solutions de même coût c_ε , cette astuce permet de sélectionner la solution optimisant les autres fonctions objectif. Les coefficients doivent être choisis avec attention pour ne pas éliminer des points non dominés. [Ehrgott et Ryan \(2002\)](#) et [Ehrgott et Ruzika \(2008\)](#) ont également proposé une modification de la fonction objectif de (MOP_ε) qui pénalise les solutions excédant les contraintes sur les fonctions objectif contraintes. [Mavrotas \(2009\)](#) a transformé les inégalités larges des contraintes sur les fonctions objectif de (MOP_ε) en égalité grâce à l'ajout de variables d'écart et a ajouté une combinaison linéaire de ces nouvelles variables à la fonction objectif. Cette méthode ne génère que les points supportés de \mathcal{Y}_N . [Mavrotas et Florios \(2013\)](#) ont ensuite amélioré cette méthode en introduisant un ordre lexicographique entre les objectifs contraints. La dernière amélioration de la méthode de Mavrotas a été faite par [Zhang et Reimann \(2014\)](#) en enlevant les variables d'écart et en proposant des heuristiques permettant d'arrêter

l'exploration de manière anticipée et en améliorant la précision des coefficients ε .

L'approche ε -contrainte est classique dans l'optimisation multi-objectif et a été utilisée dans de nombreux problèmes comme le problème du sac-à-dos multi-objectif (Laumanns et al. (2006), Kirlik et Sayın (2014)), le problème d'affectation multi-objectif (Özlen et Azizoglu (2009), Kirlik et Sayın (2014)), le problème du voyageur de commerce avec profits (Bérubé et al. (2009)) et le problème de l'arbre de Steiner avec collecte de prix (Leitner et al. (2014)).

Une étude complète des applications de la méthode ε -contrainte dans différents problèmes et des techniques d'amélioration proposées dans la littérature est présentée par Sáez-Aguado et Trandafir (2018). Ils ont comparé sept variantes de la méthode ε -contrainte sur des instances du problème de couverture p -médiane (*p-median cover problem*) et la plus efficace est celle de Ehrgott et Ryan (2002).

1.3.3 Méthode ε -contrainte avec changement de l'ordre d'exploration

Certaines méthodes modifient la méthode ε -contrainte par une exploration différente de l'espace des objectifs.

Parallel Partitioning Method (PPM). Lemesre et al. (2007) ont conçu cette méthode pour résoudre des problèmes bi-objectif. Initialement, les deux points lexicographiquement optimaux sont calculés. La première phase consiste à appliquer la méthode ε -contrainte pour des valeurs ε fixes et à trouver un point non dominé pour chacune de ces valeurs. La seconde phase explore chaque rectangle défini par deux points non dominés consécutifs de la première phase, comme pour la méthode en deux phases. La méthode k -PPM de Dhaenens et al. (2010) étend PPM à plus de deux objectifs.

Balanced Box Method. Boland et al. (2015) ont conçu cette méthode pour les problèmes mixtes en nombres entiers bi-objectif. Initialement, les deux points lexicographiquement optimaux y_1 et y_2 sont calculés et définissent un rectangle qui est la première zone de recherche. Ensuite, le processus suivant est itéré sur les régions de recherche suffisamment grandes (i.e. pouvant contenir des points non dominés). Le rectangle défini par (y_1, y_2) est divisé en deux parties égales selon le second objectif c^2 . Soit R_I (respectivement R_S) le rectangle inférieur (respectivement supérieur) comprenant le point y_2 minimisant c^2 (respectivement y_1 minimisant c^1). Sur R_I , le point y_3 minimisant c^1 est obtenu et sur R_S , le point y_4 minimisant c^2 est trouvé. De nouveaux rectangles sont définis en tant que régions de recherche : les rectangles définis par (y_1, y_4) et (y_2, y_3) . La Figure 1.8 illustre l'initialisation et la première itération de la méthode.

1.3.4 Méthode de séparations et évaluations

Les méthodes de séparations et évaluations (*branch-and-bound* = B&B) sont souvent utilisées pour résoudre les problèmes combinatoires \mathcal{NP} -complets sans énumérer toutes les solutions. Pour cela, la séparation divise l'espace des solutions en un recouvrement, et

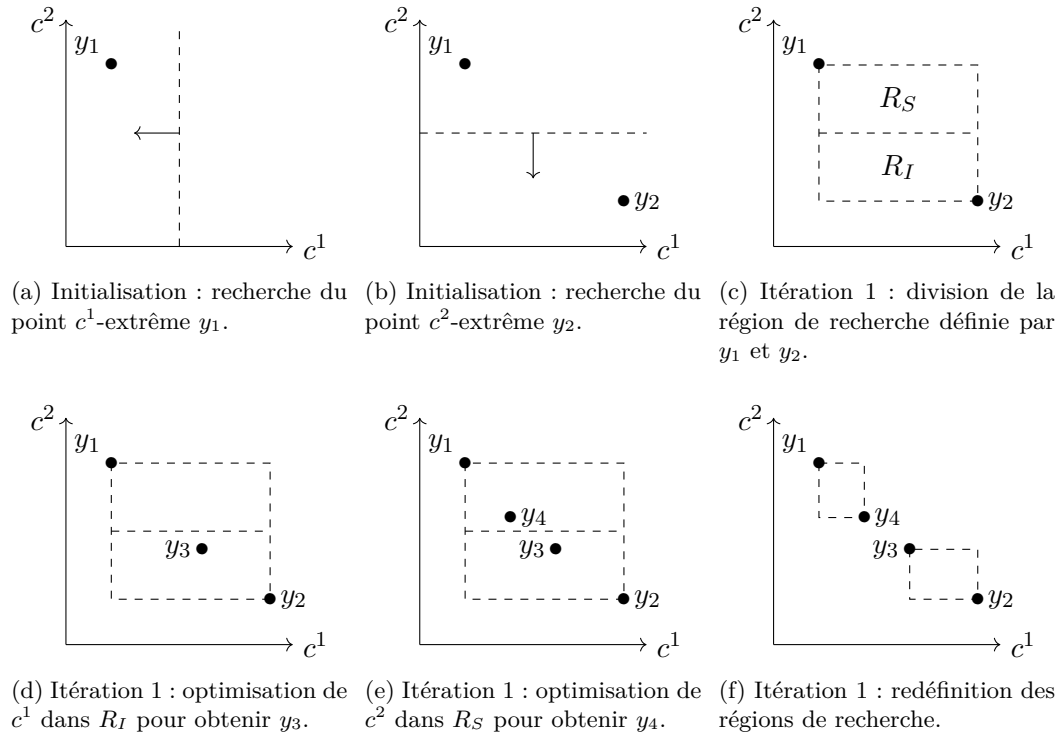


FIGURE 1.8: Initialisation et première itération de la méthode *Balanced Box* appliquée à un problème de minimisation bi-objectif.

idéalement une partition. Cette division de l'espace est représentée sous forme de branches dans un arbre de recherche. Ensuite, l'évaluation explore chaque branche créée par la séparation pour déterminer si elle peut aboutir à une solution optimale et doit être explorée plus en profondeur. En pratique, dans un problème de minimisation, l'évaluation élimine une branche si une borne inférieure du sous-problème courant est supérieure à la meilleure solution réalisable connue. Ce test est appelé un test de dominance. Une description plus détaillée de ces méthodes est disponible en Section 2.4.1.

Przybylski et Gandibleux (2017) proposent un état-de-l'art complet des B&B pour les problèmes linéaires en nombres entiers (ou mixtes) multi-objectif. Contrairement aux problèmes mono-objectif, les B&B appliqués à des problèmes multi-objectif ont pour but d'identifier un ensemble de solutions dites efficaces. De ce fait, les éléments clés de ces méthodes doivent être adaptés.

Définition des bornes inférieures et supérieures. Les bornes inférieures et supérieures doivent borner simultanément plusieurs solutions. Ainsi, les notions de bornes multi-objectif ont été largement étudiées dans la littérature et la section 1.2.3 en fournit une discussion plus détaillée sur leurs définitions.

Séparation. En général, l'étape de séparation statique d'une méthode mono-objectif sélectionne une variable non fixée qui semble la plus prometteuse selon l'objectif considéré et scinde son domaine de validité dans chaque branche de la partition. Sélectionner une variable lors de la séparation est encore moins évident si l'on considère plusieurs objectifs à optimiser simultanément. En effet, il est alors possible de sélectionner la variable la plus prometteuse selon une somme des objectifs, l'apport maximal ou minimal à un objectif... [Jorge \(2010\)](#) a montré qu'il n'y avait pas de consensus entre les différentes stratégies de séparation statiques sur le problème du sac-à-dos tri-objectif.

Dans le cas d'une séparation dynamique en mono-objectif, le choix d'une variable est guidé par la solution unique retournée par la résolution d'une relaxation. Par exemple, dans le cas de la relaxation linéaire, il est possible de sélectionner la variable fractionnaire la plus éloignée d'un entier. Dans le cas multi-objectif, la résolution d'une relaxation engendre plusieurs solutions, ce qui complexifie la procédure de séparation.

Choix de la branche à explorer. Certaines méthodologies qui indiquent l'ordre d'exploration des branches en optimisation mono-objectif ne sont pas directement transposables au cas multi-objectif. Par exemple, il existe la méthodologie visant à explorer la branche ayant la meilleure borne inférieure. Or, en optimisation multi-objectif une borne inférieure LB_1 ne domine une autre borne LB_2 que si l'ensemble des points de la borne LB_2 est dominé et qu'aucun point de la borne LB_1 n'est dominé. Cette notion de dominance est alors faiblement observée en pratique.

Test de dominance. En optimisation multi-objectif, le test de dominance est moins immédiat qu'en mono-objectif. En effet, pour éliminer une branche par dominance, il faut que tous les points de la borne inférieure soient dominés par un point de la borne supérieure.

Division de l'espace des objectifs. Malgré les avancées de la littérature sur les méthodes B&B multi-objectif, les conditions de dominance des branches ne permettent pas de réduire efficacement la taille de l'arbre de recherche. Ainsi, [Visée et al. \(1998\)](#) ont résolu le problème du sac-à-dos bi-objectif avec une méthode en deux phases, décrite en Section 1.3.1, où la seconde étape de la méthode consiste à résoudre des B&B restreints dans l'espace des objectifs. [Stidsen et al. \(2014\)](#) ont proposé de partitionner l'espace des objectifs selon des droites passant par le point $(0,0)$ pour des problèmes bi-objectif et d'appliquer des B&B sur chacun de ces morceaux. Contrairement à la méthode de [Visée et al. \(1998\)](#), les partitions de l'espace des objectifs ne sont pas indépendantes puisqu'un point d'une partition peut être dominé par un point d'une autre partition.

Parmi tous les B&B proposés dans la littérature pour résoudre un problème linéaire en nombres entiers, [Parragh et Tricoire \(2019\)](#) sont les seuls à l'avoir appliqué sur une variante du problème de tournées de véhicules : le problème de la course d'orientation par équipes bi-objectif avec fenêtres de temps. Ce problème est détaillé dans le Chapitre 5 du manuscrit. Dans leur méthode, la borne inférieure est calculée grâce à l'application d'une méthode dichotomique (cf. Section 1.3.1) sur la relaxation linéaire de la formulation où

chaque problème mono-objectif résultant est résolu par génération de colonnes.

L'élément clef de leur méthode repose sur la stratégie de branchement qui permet de passer d'un algorithme travaillant sur l'ensemble de l'espace des objectifs à plusieurs sous-ensembles locaux. En effet, ils ont remarqué que la borne inférieure était souvent partiellement dominée par la borne supérieure. Ainsi, les zones de l'espace des objectifs où elle est dominée ne nécessitent plus d'exploration, tandis que les zones de l'espace des objectifs où elle n'est pas dominée sont encore à explorer. La stratégie de branchement alors utilisée définit chaque branche comme une zone à explorer grâce à l'ajout de deux contraintes linéaires restreignant l'espace des objectifs. Un branchement est alors équivalent à l'ajout de bornes sur les objectifs. Une fois l'espace des objectifs découpé, des stratégies de branchement sur les variables plus classiques sont utilisées.

1.3.5 Méthode basée sur la programmation dynamique

La programmation dynamique est une méthode algorithmique introduite par [Bellman \(1966\)](#) consistant à résoudre un problème grâce aux résultats intermédiaires de sa décomposition en sous-problèmes. Par exemple, pour résoudre des problèmes de plus court chemin multi-objectif d'un nœud source s à un nœud terminal t , la méthode repose sur une énumération des chemins optimaux en partant de s grâce au principe suivant : tout chemin optimal $(i_0 = s, i_1, \dots, i_n = t)$ de s à t est composé de $n - 1$ sous-chemins optimaux de s à i_j , pour tout j de 1 à $n - 1$ ([Martins \(1984\)](#), [Reinhardt et Pisinger \(2011\)](#)). La programmation dynamique a également été utilisée pour résoudre d'autres problèmes multi-objectif tels que le problème du sac-à-dos ([Klamroth et Wiecek \(2000\)](#), [Captivo et al. \(2003\)](#), [Bazgan et al. \(2009\)](#), [Figueira et al. \(2013\)](#)).

1.3.6 Conclusion

Les méthodes basées sur l'exploration de l'espace des objectifs (Sections 1.3.1 à 1.3.3) et qui résolvent une succession de problèmes mono-objectif tirent partie de l'efficacité des solveurs commerciaux de programmes linéaires en nombre entiers, ainsi que de la vaste littérature mono-objectif ([Boland et al. \(2015\)](#)). Cependant, les problèmes multi-objectif \mathcal{NP} -difficiles, dont la variante mono-objectif est déjà \mathcal{NP} -difficile, n'ont été que peu traités par ces méthodes. Par exemple, nous verrons en Section 2.5 que très peu de méthodes exactes ont été appliquées à des problèmes de tournées de véhicules multi-objectif. Les méthodes en deux phases et ε -contraintes restent les méthodes exactes multi-objectif les plus employées ([Ehrgott et Gandibleux \(2000\)](#)) par leur efficacité et leur simplicité de mise en place. [Boland et al. \(2015\)](#) concluent que *Balanced Box method* est la méthode basée sur l'exploration de l'espace des objectifs la plus efficace. Cependant, [Parragh et Tricoire \(2019\)](#) montrent que la méthode ε -contrainte classique surpasse cette méthode sur des problèmes d'emplacement d'installations (*facilities location problem*).

Les méthodes basées sur l'exploration des solutions (Sections 1.3.4 et 1.3.5), quant à elles, se basent sur la structure bi-objectif du problème pour élaborer des propriétés (utilisation des bornes, indicateurs de performance...). Il n'est pas évident, à l'heure actuelle, de tirer une conclusion claire sur la meilleure approche multi-objectif.

1.4 Conclusion du chapitre

De par leur utilité et leur complexité, les problèmes combinatoires multi-objectif ont été largement étudiés depuis une cinquantaine d'années. De plus, l'accroissement des capacités de calcul des ordinateurs a permis de développer des outils d'optimisation multi-objectif de plus en plus performants qui peuvent s'adapter à des problèmes industriels réels. Toutefois, le nombre de méthodes approchées dans la littérature, notamment des métaheuristiques évolutionnaires, reste bien supérieur au nombre de méthodes exactes. De plus, ces dernières sont souvent appliquées à des problèmes ayant un équivalent mono-objectif pouvant être résolu en temps polynomial ou pseudo-polynomial. Très peu de méthodes exactes multi-objectif sont efficaces pour résoudre des problèmes plus difficiles. Ainsi, ce manuscrit se concentre sur les méthodes exactes pour des problèmes combinatoires \mathcal{NP} -difficiles.

Problèmes de tournées de véhicules : état de l'art

Sommaire

2.1	Introduction	23
2.2	Formulations mathématiques	24
2.2.1	Formulation de flux de véhicules à deux indices	24
2.2.2	Formulation de flux de marchandises à deux indices	25
2.2.3	Formulation de partitionnement	25
2.3	Génération de colonnes	26
2.3.1	Principe	27
2.3.2	Problème de plus court chemin élémentaire avec contraintes de ressources	28
2.3.3	Techniques de stabilisation	33
2.4	Méthodes exactes	34
2.4.1	Méthode de séparations et évaluations	34
2.4.2	Méthode de séparations et coupes	35
2.4.3	Méthode de séparations et génération de colonnes	36
2.4.4	Méthode de séparations et coupes et génération de colonnes	36
2.4.5	Énumération explicite des colonnes	36
2.5	Problèmes de tournées de véhicules multi-objectif	36
2.5.1	Réduction de la pollution	37
2.5.2	Équité entre les véhicules	37
2.5.3	Satisfaction des usagers	38
2.5.4	Réduction du risque	39
2.5.5	Collecte d'un profit	39
2.5.6	Conclusion	40
2.6	Conclusion du chapitre	40

2.1 Introduction

Il y a soixante ans, [Dantzig et Ramser \(1959\)](#) introduisaient le problème de tournées de véhicules (*Vehicle Routing Problem* = VRP) pour distribuer de l'essence d'un terminal pétrolier à des stations services avec des routes de coût minimum effectuées par une flotte

de véhicules. Depuis, le VRP a été appliqué à une multitude de problèmes industriels (Toth et Vigo (2014)).

Plus formellement, le VRP consiste à trouver un ensemble de routes sur un réseau, qui passe par tous les points de ce réseau et minimise la somme du coût affecté à chacune des routes. Ce chapitre donne un aperçu des notions de base pour le VRP. Tout d'abord, la Section 2.2 présente les principales formulations mathématiques de ce problème. Une méthode pour trouver la borne inférieure d'un problème linéaire est présentée en Section 2.3. Ensuite, les méthodes de résolution exactes classiques du VRP sont abordées en Section 2.4. Comme ce travail de thèse porte sur les problèmes de tournées de véhicules bi-objectif, une revue de la littérature est faite en Section 2.5 sur l'ensemble des méthodes exactes utilisées pour résoudre des problèmes de tournées de véhicules avec plusieurs objectifs. Enfin, la conclusion est en Section 2.6.

2.2 Formulations mathématiques

Le problème classique des tournées de véhicules est le VRP avec capacité (*Capacitated Vehicle Routing Problem* = CVRP). Le CVRP est défini sur un graphe $G = (V, A)$ orienté. L'ensemble des sommets $V = \{0, \dots, n\}$ contient le dépôt 0 et n clients i , $1 \leq i \leq n$. Soit V' l'ensemble des clients tel que $V = V' \cup \{0\}$. Chaque client possède une demande q_i . L'ensemble des arcs A est défini comme suit $A = \{(i, j) | i, j \in V\}$ et un coût c_{ij} est associé à chaque arc (i, j) du graphe. Un ensemble K de véhicules homogènes de capacité maximale Q est disponible au dépôt. Un chemin réalisable est un chemin élémentaire $ch = (0, i_1, i_2, \dots, i_{v_{ch}})$ effectué par un véhicule qui commence au dépôt et visite un ensemble de clients tel que la demande totale des clients visités par le véhicule n'excède pas sa capacité maximale. Si un chemin réalisable est tel que $i_{v_{ch}} = 0$, alors c'est une route réalisable. L'objectif du CVRP est de définir un ensemble de K routes qui visite tous les clients du graphe et qui minimise le coût c égal à la somme du coût des arcs utilisés.

Dans la littérature, nous pouvons trouver trois principaux types de formulation pour les problèmes de tournées de véhicules : formulation de flux de véhicules (*vehicle flow formulation*), formulation de flux de marchandises (*commodity flow formulation*) et formulation de partitionnement (*set partitioning formulation*) (Toth et Vigo (2014)). Les deux premières sont des formulations à deux ou trois indices, en général. Les variables de la première formulation se basent sur le trajet des véhicules alors que dans la seconde formulation d'autres variables sont rajoutées pour représenter la quantité de marchandises en circulation sur les routes. La dernière formulation est une formulation avec un nombre exponentiel de variables. Ces trois formulations sont présentées ci-après.

2.2.1 Formulation de flux de véhicules à deux indices

Cette formulation a été introduite par Laporte et Nobert (1983). Les variables binaires x_{ij} , $(i, j) \in A$, de la formulation indiquent si l'arc (i, j) est traversé par un véhicule dans la solution ($x_{ij} = 1$) ou non ($x_{ij} = 0$). Dans le modèle, $\delta^+(S)$, $S \subset V$, (respectivement $\delta^-(S)$) est l'ensemble des arcs (i, j) tels que $i \in S$ et $j \in V \setminus S$ (respectivement $i \in V \setminus S$

et $j \in S$).

$$\left\{ \begin{array}{ll} \min \sum_{(i,j) \in A} c_{ij} x_{ij} & (2.1.1) \\ t.q. \sum_{(0,j) \in A} x_{0j} \leq |K| & j \in V' \quad (2.1.2) \\ \sum_{(i,j) \in \delta^+(\{i\})} x_{ij} = 1 & i \in V' \quad (2.1.3) \\ \sum_{(i,j) \in \delta^-(\{i\})} x_{ij} = 1 & i \in V' \quad (2.1.4) \\ \sum_{(i,j) \in \delta^+(S)} x_{ij} \geq \left\lceil \frac{\sum_{i \in S} q_i}{Q} \right\rceil & S \subseteq V' \quad (2.1.5) \\ x_{ij} \in \{0, 1\} & (i, j) \in A \quad (2.1.6) \end{array} \right. \quad (2.1)$$

En premier lieu, les contraintes (2.1.2) limitent le nombre d'arcs quittant le dépôt. Les contraintes (2.1.3) et (2.1.4) assurent que chaque client n'est visité qu'une seule fois. Les contraintes (2.1.5) visent à éliminer les solutions qui utilisent moins de véhicules que nécessaire pour servir un sous-ensemble de clients et évitent également les solutions avec des sous-tours, i.e. des cycles ne passant pas par le dépôt. Enfin, les dernières contraintes donnent le domaine de définition des variables.

2.2.2 Formulation de flux de marchandises à deux indices

Cette formulation a été introduite par [Gavish \(1984\)](#). Les variables continues f_{ij} , $(i, j) \in A$, de la formulation indiquent la quantité de marchandises transportées le long de l'arc (i, j) .

Les contraintes (2.1.5) de la formulation (2.1) sont remplacées par les contraintes suivantes :

$$\left\{ \begin{array}{ll} \sum_{(i,j) \in \delta^-(\{i\})} f_{ij} - \sum_{(i,j) \in \delta^+(\{i\})} f_{ij} = q_i & i \in V' \\ q_j x_{ij} \leq f_{ij} \leq (Q - q_i) x_{ij} & (i, j) \in A \end{array} \right.$$

Les premières contraintes assurent que les clients reçoivent leurs demandes, alors que les dernières bornent les variables f .

2.2.3 Formulation de partitionnement

Cette formulation a été introduite par [Balinski et Quandt \(1964\)](#). Soit R l'ensemble des routes réalisables. Soit a_{ir} un coefficient binaire indiquant si le nœud $i \in V$ est visité par la route $r \in R$. Soit c_r le coût total des arcs traversés par la route r . Les variables binaires x_r de la formulation indiquent si la route r est sélectionnée dans la solution ($x_r = 1$) ou non ($x_r = 0$).

$$\left\{ \begin{array}{l} \min \sum_{r \in R} c_r x_r \quad (2.2.1) \\ t.q. \sum_{r \in R} a_{ir} x_r = 1 \quad i \in V' \quad (2.2.2) \\ \sum_{r \in R} x_r \leq |K| \quad (2.2.3) \\ x_r \in \{0, 1\} \quad r \in R \quad (2.2.4) \end{array} \right. \quad (2.2)$$

Cette formulation présente deux contraintes (2.2.2) et (2.2.3) qui contraignent respectivement tous les clients à être visités et l'utilisation d'au plus $|K|$ véhicules. Le domaine de définition des variables est décrit avec les contraintes (2.2.4).

Si la matrice de coût c respecte l'inégalité triangulaire, la formulation peut être transformée en une formulation d'ensemble couvrant (*set-covering formulation*) :

$$\left\{ \begin{array}{l} (2.2.1) \\ t.q. \sum_{r \in R} a_{ir} x_r \geq 1 \quad i \in V' \quad (2.3.2) \\ (2.2.3) - (2.2.4) \end{array} \right. \quad (2.3)$$

Les solutions optimales des formulations de partitionnement et d'ensemble couvrant ont le même coût c . En effet, toute solution réalisable de (2.2) est une solution réalisable de (2.3). De plus, toute solution réalisable de (2.3) est soit réalisable pour (2.2) ou peut être transformée en une solution de coût c inférieur, réalisable pour (2.2), en enlevant les visites multiples à tous les clients.

La formulation (2.3) est introduite puisque l'espace des solutions du dual est plus petit que celui de la formulation (2.2). En effet, les variables duales associées aux contraintes (2.2.2) ne sont pas contraintes, alors que celles associées aux contraintes (2.3.2) doivent être positives ou nulles.

L'avantage de ces deux formulations en comparaison aux autres formulations de flux de véhicules et de marchandises est la qualité de la relaxation linéaire qui fournit des bornes inférieures proches de l'optimum entier. Toutefois, leur désavantage est le nombre exponentiel de variables. En effet, pour $|V| = 10$, le nombre de variables pour la formulation du flux de véhicules est $10 * 9 = 90$, alors que le nombre de variables de la formulation de partitionnement est $\frac{10!}{2} = 1814400$. Pour résoudre ces formulations, une énumération explicite des variables est à proscrire et des techniques d'optimisation permettant une énumération dynamique des variables sont alors utilisées.

2.3 Génération de colonnes

Cette section décrit la génération de colonnes basique, ainsi que ses améliorations car notre méthode exacte de résolution du VRP bi-objectif l'utilise. La première utilisation de

la génération de colonnes pour résoudre un VRP a été faite par [Appelgren \(1969\)](#). [Feillet \(2010\)](#) et [Toth et Vigo \(2014\)](#) proposent une lecture détaillée sur cette technique.

La formulation d'ensemble couvrant (2.3) pour le VRP est appelée le problème maître (PM). Sa relaxation linéaire est appelée le Problème Maître Linéaire (PML) et est donnée par la formulation suivante :

$$(PML) \left\{ \begin{array}{ll} \min \sum_{r \in R} c_r x_r & (2.4.1) \\ t.q. \sum_{r \in R} a_{ir} x_r \geq 1 & i \in V' \quad (2.4.2) \\ \sum_{r \in R} x_r \leq |K| & (2.4.3) \\ x_r \geq 0 & r \in R \quad (2.4.4) \end{array} \right. \quad (2.4)$$

Note 2. *Le PM considère des variables entières et non binaires pour éviter les contraintes de type $x_r \leq 1$ dans le PML. Comme les coûts sur les arcs sont positifs, la solution optimale du PM ne peut pas contenir de variables $x_r > 1$.*

Soit λ_i , $i \in V'$, les variables duales associées aux contraintes (2.4.2) et λ_0 la variable duale associée à la contrainte (2.4.3). La formulation duale du PML est appelée le Problème Dual Linéaire (PDL) et est la suivante :

$$(PDL) \left\{ \begin{array}{ll} \max |K| \lambda_0 + \sum_{i \in V'} \lambda_i & \\ t.q. \sum_{i \in V} a_{ir} \lambda_i \leq c_r & r \in R \\ \lambda_i \geq 0 & i \in V' \\ \lambda_0 \leq 0 & \end{array} \right. \quad (2.5)$$

Si le PML contient un nombre trop important de variables, l'énumération de toutes les variables est irréalisable en temps raisonnable. Ainsi, la technique de génération de colonnes est utilisée pour trouver la solution optimale du problème originel en travaillant sur un nombre limité de variables. Lors de cette technique, PM, PML et PDL sont résolus sur un nombre restreint de variables, ils sont alors respectivement appelés le PM restreint (PMR), le PML restreint (PMLR) et le PDL restreint (PDLR).

De plus, les variables x_r sont appelées des colonnes puisqu'une colonne de la matrice des contraintes du PM représente les coefficients d'une variable. En VRP, les colonnes correspondent aux routes.

2.3.1 Principe

Le principe de la génération de colonnes est illustré par la Figure 2.1. Deux éléments principaux composent la technique :

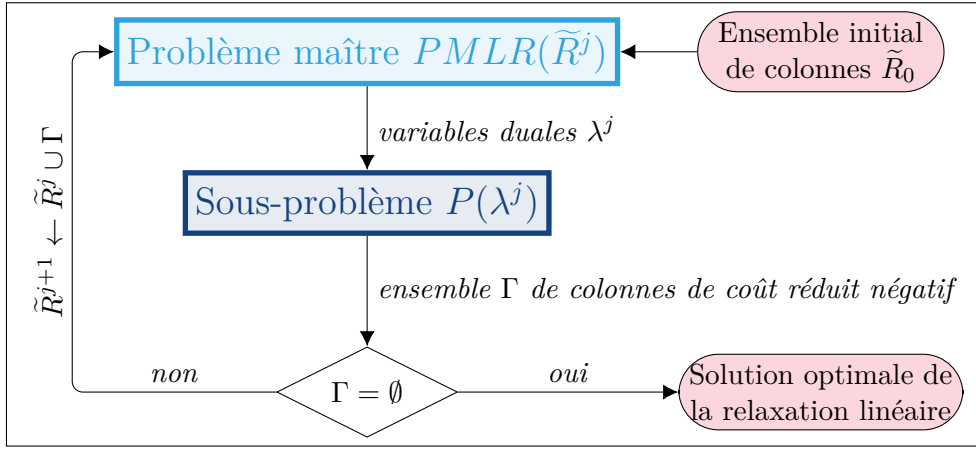


FIGURE 2.1: Processus de génération de colonnes.

- le PMLR permet de générer les solutions optimales courantes primales x^j et duales λ^j du problème comprenant un ensemble restreint de colonnes \tilde{R}^j à l'itération j ;
- le sous-problème P utilise λ^j pour générer un ensemble de colonnes Γ qui peuvent améliorer la solution optimale courante.

Les colonnes Γ sont alors ajoutées dans le PMLR et le processus continue jusqu'à ce qu'il n'existe plus de colonnes pouvant améliorer x^j . À la fin de la génération de colonnes, si $j = j_{fin}$, la solution optimale courante $x^{j_{fin}}$ est la solution optimale du PML.

À l'itération j , le sous-problème résout le problème (2.6) qui cherche les nouvelles routes r dans $R \setminus \tilde{R}^j$ qui ont un coût réduit minimum. Le coût réduit d'une route r est égal à la fonction objectif du problème (2.6). Le sous-problème, quant à lui, ne renvoie que les routes retournées par le problème (2.6) ayant un coût réduit strictement négatif.

$$\min c_r - \sum_{i \in V} a_{ir} \lambda_i^j, \quad r \in R \setminus \tilde{R}^j \quad (2.6)$$

où λ^j est la solution optimale duale à l'itération j . Ce sous-problème est dit de *pricing* et en VRP, il est très souvent une variante du problème de plus court chemin : le problème de plus court chemin élémentaire avec contraintes de ressources.

2.3.2 Problème de plus court chemin élémentaire avec contraintes de ressources

Pour la plupart des VRP, le sous-problème de *pricing* est un problème de plus court chemin élémentaire avec contraintes de ressources (*Elementary Shortest Path Problems with Resource Constraints* = ESPPRC). La résolution de ce problème \mathcal{NP} -difficile est classiquement faite avec un algorithme d'étiquetage décrit ci-dessous.

2.3.2.1 Algorithme d'étiquetage

Une étiquette E représente un chemin réalisable ch qui commence au dépôt et dont les caractéristiques sont données par un tuple $E = (v_E, \bar{c}_E, q_E, C_E)$, où $v_E \in V$ est le dernier client visité de ch , \bar{c}_E est le coût réduit de ch , q_E est la demande cumulée des clients de ch et C_E est l'ensemble des clients visités par ch .

L'étiquette initiale est $E_0 = (0, 0, 0, \{0\})$. L'extension d'une étiquette E de son nœud terminal v_E à un client $j \in V \setminus C_E$ produit une nouvelle étiquette E' via les règles (2.7)-(2.10) :

$$v_{E'} = j \quad (2.7)$$

$$\bar{c}_{E'} = \bar{c}_E + \bar{c}_{v_E j} \quad (2.8)$$

$$q_{E'} = q_E + q_j \quad (2.9)$$

$$C_{E'} \leftarrow C_E \cup \{j\} \quad (2.10)$$

où \bar{c}_{ij} , $(i, j) \in A$, est le coût réduit sur l'arc (i, j) . Les règles de propagation (2.7), (2.8), (2.9) et (2.10) mettent respectivement à jour le nœud terminal, le coût réduit, la charge et les clients visités.

L'efficacité de cet algorithme repose sur sa capacité à éliminer des étiquettes qui ne pourront pas aboutir à des routes de coût réduit minimum. Ainsi, des règles de dominance entre deux étiquettes sont établies. On dit qu'une étiquette E^1 domine une étiquette E^2 si $E^1 \neq E^2$ et que les conditions suivantes sont satisfaites :

$$v_{E^1} = v_{E^2} \quad (2.11)$$

$$C_{E^1} \subseteq C_{E^2} \quad (2.12)$$

$$q_{E^1} \leq q_{E^2} \quad (2.13)$$

$$\bar{c}_{E^1} \leq \bar{c}_{E^2} \quad (2.14)$$

Ces conditions sont valides de par la croissance des attributs C , q et \bar{c} . Les trois premières conditions assurent que tous les clients $i \in V$ qui peuvent être atteints par propagation de l'étiquette E^2 peuvent également être atteints par propagation de l'étiquette E^1 . La dernière condition permet de vérifier que le coût réduit de l'étiquette E^2 est supérieur à celui de E^1 .

Feillet et al. (2004) ont proposé une amélioration de la règle de dominance en remplaçant la condition (2.12) par une condition stipulant que l'ensemble des clients inaccessibles par E^1 doit être compris dans l'ensemble des clients inaccessibles par E^2 . Cette condition renforce la précédente règle de dominance lors de contraintes opérationnelles spécifiques comme les fenêtres de temps.

2.3.2.2 Relaxation de l'élémentarité

Les coûts réduits \bar{c}_{ij} sur les arcs $(i, j) \in A$ pouvant être négatifs, le graphe du ESPPRC peut contenir des cycles de coûts négatifs. Dror (1994) a montré que l'ESPPRC est un problème \mathcal{NP} -difficile. Ainsi, certains auteurs résolvent le problème de plus court chemin avec contraintes de ressources (*Shortest Path Problem with Resource Constraints* = SPPRC) à la place, qui autorise la génération de routes visitant les clients plus d'une fois (Irnich et Desaulniers (2005)) et qui peuvent être résolus en temps pseudo-polynomial (Desrochers et al. (1992)). Toutefois, la qualité de la borne inférieure est moins bonne avec cette relaxation et des compromis entre résoudre des SPPRC et des ESPPRC ont été proposés dans la littérature.

Ng-routes. La relaxation de l'élémentarité la plus efficace proposée dans la littérature est la relaxation ng-route, introduite par Baldacci et al. (2011). Cette technique permet d'obtenir des routes presque élémentaires en des temps raisonnables. Les routes construites peuvent contenir de longs cycles en terme de distance sur les arêtes, mais elle empêche la construction de routes aux cycles courts. Le nouveau sous-problème obtenu pour générer des routes avec un coût réduit négatif est un problème de plus court ng-chemin avec contraintes de ressources (*ng-Shortest Path Problem with Resource Constraints* = ng-SPPRC).

La construction des ng-routes peut être résumée comme suit. Pour chaque client $i \in V'$ du problème, un ensemble N_i de cardinalité Δ est pré-défini. Chaque N_i contient i plus ses $\Delta - 1$ plus proches voisins. Pour chaque chemin $ch = (0, i_1, i_2, \dots, i_{n_{ch}})$, un ng-ensemble $NG_{ch} \subseteq N_i$ est calculé comme suit :

$$NG_{ch} = \{i_k, k \in \llbracket 1; n_{ch}-1 \rrbracket \mid i_k \in \cap_{j=k+1}^{n_{ch}} N_{i_j}\} \cup \{i_{n_{ch}}\}$$

Un ng-ensemble du chemin ch contient les clients que ch ne peut plus visiter puisqu'ils ont déjà été visités et sont considérés trop proches pour y retourner. Un ng-chemin réalisable est un chemin $ch = (0, i_1, i_2, \dots, i_{n_{ch}})$ qui peut être non élémentaire, commence au dépôt, satisfait les contraintes de construction du ng-ensemble et se termine au client $i_{n_{ch}}$. Si un ng-chemin réalisable est tel que $i_{n_{ch}} = 0$, alors c'est une ng-route réalisable.

La Figure 2.2 donne un exemple de calcul du ng-ensemble sur un graphe de VRP avec un dépôt 0 et quatre clients. Dans cet exemple, la ng-route $\{0, 1, 2, 3, 1, 0\}$ est réalisable car le client 1 n'est pas dans le ng-ensemble du chemin $\{0, 1, 2, 3\}$. La ng-route $\{0, 1, 2, 3, 2, 0\}$, quant à elle, n'est pas réalisable car le client 2 est dans le ng-ensemble du chemin $\{0, 1, 2, 3\}$.

Lors de l'application des ng-routes dans une méthode, le paramètre Δ doit être décidé avant le début de la résolution. Plus ce paramètre est grand, plus le ng-SPPRC se rapproche du ESPPRC, mais plus la complexité de l'algorithme d'étiquetage augmente. Baldacci et al. (2011) ont utilisé $\Delta = \{8, 10\}$ pour résoudre des instances de VRP avec fenêtres de temps et de CVRP jusqu'à 100 clients.

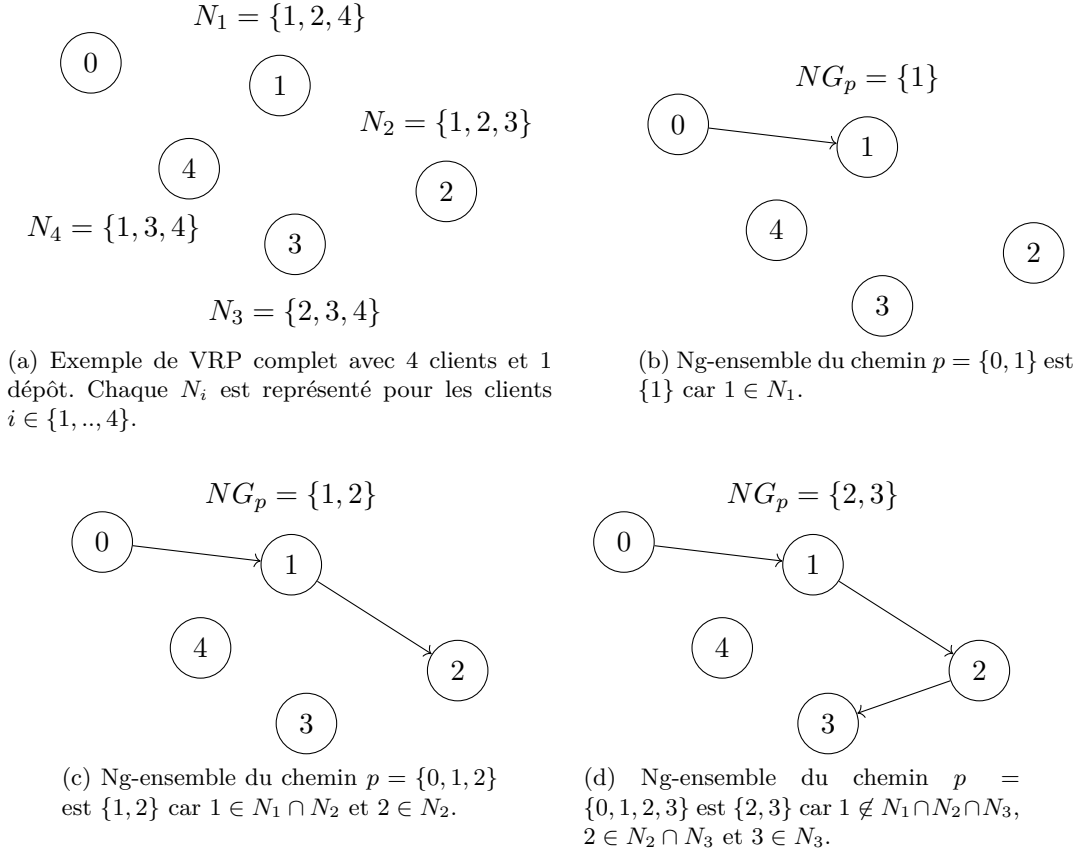


FIGURE 2.2: Exemple d'expansion du ng-ensemble sur un graphe de VRP.

2.3.2.3 Bornes de complétion

Les bornes de complétion permettent d'éliminer les étiquettes qui ne peuvent pas aboutir à des routes de coût réduit négatif. Étant donné l'étiquette E , la borne de complétion de E , notée $F(E)$, est une borne inférieure du coût réduit de toutes les extensions possibles de E vers le dépôt. Ainsi, si le coût réduit de E plus le coût de $F(E)$ est supérieur à 0, alors l'étiquette peut être éliminée.

Il y a plusieurs manières de calculer ces bornes de complétion. Par exemple, le calcul peut reposer sur la matrice du coût réduit courant (Baldacci et al. (2008; 2011)) ou sur la valeur des étiquettes à l'itération précédente du *Decremental State Space Relaxation* (Contardo et Martinelli (2014), Pecin et al. (2017)).

Dans cette section, nous expliquons le principe des bornes de complétion comme utilisé dans Baldacci et al. (2008; 2011) lorsque le sous-problème de *pricing* est utilisé pour générer des ng-routes. Dans ce cas, les bornes de complétion sont utilisées pour éviter de générer des étiquettes qui ne pourront pas aboutir à des ng-routes de coût réduit négatif. Soit l'étiquette $E = (v_E, \overline{c_E}, q_E, NG_E)$ comme défini en Section 2.3.2.1 où l'ensemble des

clients visités C_E est remplacé par le ng-ensemble NG_E . La valeur $F(NG, q, i)$ représente le coût réduit minimal de toutes les extensions jusqu'au dépôt des étiquettes terminant au client $v_E = i$, de capacité $q_E = q$ et de ng-ensemble $NG_E = NG$. Ainsi, toute extension de E aura un coût réduit supérieur à une valeur δ si :

$$\bar{c}_E + F(NG_E, q_E, v_E) > \delta$$

En fixant la valeur de δ à 0 et en calculant le coût réduit des étiquettes grâce aux variables duales associées à la solution optimale du PMLR(\bar{R}), la borne de complétion empêche l'expansion d'étiquettes qui aboutiront à des ng-routes de coût réduit positif.

La borne F se calcule grâce à l'obtention de la borne f sur le coût réduit minimal des étiquettes arrières réduites. Une étiquette arrière réduite e est le tuple (v_e, q_e, NG_e) qui représente un ng-chemin commençant au nœud v_e et terminant au dépôt avec une capacité q_e et un ng-ensemble NG_e . L'expansion de NG_e se fait en partant du dépôt jusqu'à v_e et non pas dans le sens de parcours du chemin. La valeur $F(NG_E, q_E, v_E)$ d'une étiquette E est calculée comme suit :

$$F(NG_E, q_E, v_E) = \min_{\substack{NG' \cap NG_E = v_E \\ 0 \leq q \leq Q - q_E + q_{v_E}}} f(NG, q, v_E)$$

où $f(NG, q, i)$ est le coût réduit minimal des étiquettes arrières réduites commençant au nœud i avec un ng-ensemble NG et de capacité q , Q est la capacité maximale d'un véhicule et $q_i, i \in V'$, est la demande du client i . D'autres ressources que la capacité peuvent être utilisées pour calculer les bornes ; par exemple le temps de parcours le temps de parcours d'un véhicule est limité. Les bornes f et F se calculent par programmation dynamique en début de l'algorithme.

2.3.2.4 Algorithme bi-directionnel

Righini et Salani (2006) ont proposé un algorithme d'étiquetage bi-directionnel où les étiquettes sont propagées dans les deux directions d'une route jusqu'à ce que toutes les parties restantes de la route aient déjà été générées. Cette garantie peut être faite grâce à une borne sur le nombre de ressources consommées. Par exemple, si la ressource critique choisie est la capacité du véhicule, les chemins peuvent être étendus tant que leur capacité est inférieure à $\frac{Q}{2}$, avec Q la charge maximale d'un véhicule. La propagation des chemins dans les deux directions d'une route est efficace lorsque la faisabilité d'une route dépend du sens de parcours comme lorsque le problème définit des fenêtres de temps.

Après la phase de propagation, les étiquettes de sens contraire doivent être concaténées deux à deux. Pour éviter de générer la même route de multiples fois, Baldacci et al. (2008) ont proposé de concaténer les paires d'étiquettes qui diffèrent dans leur valeur de ressource critique consommée du plus petit écart possible. Par exemple, les étiquettes associées aux chemins $\{0, 1, 2\}$ et $\{2, 3, 0\}$ de capacité 2, peuvent être concaténées pour former la route $\{0, 1, 2, 3, 0\}$; alors que les étiquettes associées aux chemins $\{0, 1\}$ et $\{1, 2, 3, 0\}$ de capacité respective 1 et 3 ne le peuvent pas.

Cette technique est forcément plus rapide durant la phase de propagation que l'algorithme mono-directionnel, mais la phase de concaténation est assez coûteuse selon les applications. Contardo et Martinelli (2014) ont ainsi préféré l'algorithme mono-directionnel lors de la résolution de VRP avec contraintes de capacité et de distance maximale sur les routes.

2.3.2.5 Heuristiques

De nombreuses heuristiques ont été développées pour accélérer les premières itérations de la génération de colonnes. En effet, le problème de *pricing* ne doit être résolu à l'optimalité qu'une seule fois pour prouver qu'il n'existe plus de colonnes de coût réduit négatif.

Certains algorithmes réduisent la taille du graphe du sous-problème en éliminant des arcs de coûts réduits non prometteurs (Fukasawa et al. (2006), Desaulniers et al. (2008), Pecin et al. (2017)). D'autres relâchent certaines règles de dominance pour éliminer plus d'étiquettes (Desaulniers et al. (2008)) ou limitent la génération d'étiquettes avec certaines caractéristiques communes (nœud terminal, ressources consommées...) (Fukasawa et al. (2006), Pecin et al. (2017)).

2.3.3 Techniques de stabilisation

D'après Vanderbeck (2005), la technique de génération de colonnes peut être ralentie par trois effets :

- l'orientation (*heading-in*) qui apparaît dans les premières itérations de la technique. Les colonnes dans le PMR ne fournissent pas de bonnes variables duales et les premières bornes inférieures sont très éloignées de la solution optimale ;
- le plateau qui est causé par la dégénérescence du PM - ici le problème de l'ensemble couvrant (*set covering problem*) - et qui entraîne une forte variation des variables duales lors d'itérations proches ;
- l'infléchissement (*tailing-off*) qui est la faible amélioration de la borne inférieure dans les dernières itérations de la génération de colonnes.

L'effet de l'orientation est atténué en choisissant un ensemble de colonnes initial représentatif de l'ensemble des colonnes réalisables du problème. Les problèmes de plateau et d'infléchissement sont tous les deux dus à l'oscillation des variables duales. Pour empêcher ce phénomène, Marsten et al. (1975) ont conçu la méthode BOXSTEP qui définit des espaces de faisabilité pour les variables duales compris dans des boîtes centrées autour de la valeur à la précédente itération. Kim et al. (1995) ont proposé une pénalisation de l'objectif par la distance entre chaque valeur duale courante et celle de la précédente itération. La méthode de *Box Stabilization* de Du Merle et al. (1999) combine les deux idées précédentes pour limiter la distance parcourue par les variables duales dans l'espace du dual.

Parmi les récentes méthodes de stabilisation de la littérature, Pigatti et al. (2005) ont proposé une version simplifiée de la méthode de *Box Stabilization*. Rousseau et al. (2007) ont proposé une méthode nommée *Interior Point Stabilization* pour générer des solutions duales situées au centre du polyèdre des solutions duales optimales par combinaison convexe des points duaux extrêmes. La méthode conçue par Desrosiers et al. (2014) enlève

les contraintes non actives du primal pour réduire la dégénérescence du problème. Enfin, Pessoa et al. (2018) réduisent l'impact de mauvaises solutions duales en prenant en compte les solutions duales des itérations précédentes dans le sous-problème.

Pour éviter les techniques de stabilisation, il existe des heuristiques duales ascendantes (*dual ascent*) qui estiment et affinent progressivement les valeurs des variables duales (Baldacci et al. (2008)). Les variables duales initiales sont améliorées grâce à une stratégie ascendante qui assurent que les variables duales ne se dégradent pas au cours des itérations. Lorsque la solution duale courante est égale à une solution primale réalisable, le théorème de la dualité forte garantit l'optimalité des solutions.

2.4 Méthodes exactes

Le VRP est un problème \mathcal{NP} -difficile comme il peut se réduire au problème du voyageur de commerce (*Traveling Salesman Problem* = TSP) lorsqu'un seul véhicule est utilisé ($|K| = 1$) et que la capacité de ce véhicule est infinie ($Q = +\infty$). Dans cette section, nous nous focalisons sur les principales méthodes exactes pour le VRP et nous invitons le lecteur à se référer au papier de Costa et al. (2019) pour des explications complètes et récentes sur les méthodes exactes pour le VRP.

2.4.1 Méthode de séparations et évaluations

La plupart des méthodes exactes pour le VRP reposent sur la méthode de séparations et évaluations (*branch-and-bound* = B&B) pour explorer implicitement l'espace des solutions (cf. Section 1.3.4). Pour avoir un B&B efficace, il faut avoir une bonne solution réalisable dès le début de l'exécution, ainsi qu'une procédure performante de calcul de bornes inférieures, souvent en résolvant une relaxation des problèmes (linéaire, lagrangienne, agrégée...).

Les B&B appliqués au VRP utilisent différentes politiques de séparation (ou branchement) sur les variables de décision :

- branchement sur les arcs : Desrosiers et al. (1984) ont introduit un B&B où les branchements sont effectués sur les variables de flots sur les arcs x_{ij} , $(i, j) \in A$. Ces branchements sont très utilisés de par leur simplicité mais ont le désavantage d'induire des changements locaux sur la solution fractionnaire courante.
- branchement sur le nombre de véhicules utilisés dans la solution : ce branchement peut être employé si la valeur des variables sur les arcs sortants du dépôt n'est pas entière.
- branchement sur les sous-ensembles de clients $S \subseteq V'$: Augerat et al. (1995) ont introduit ces branchements pour contrôler le flot total de marchandises entrant dans S . Le nombre de branches possibles devient alors exponentiel et des heuristiques de décision des sous-ensembles doivent être mises en place (Augerat et al. (1995), Lysgaard et al. (2004), Fukasawa et al. (2006), Pecin et al. (2017)).

Certaines méthodes (Fukasawa et al. (2006), Pecin et al. (2017)) évaluent l'impact de plusieurs candidats de branchements sur la borne inférieure avant de faire la séparation et

sélectionnent le candidat dont un des sous-problèmes a la plus faible borne inférieure.

De plus, les B&B se différencient par la méthode d'exploration de l'arbre de recherche qui peut notamment être effectuée en largeur, en profondeur ou en privilégiant le meilleur nœud. L'exploration en largeur (*breadth-first search*) consiste à traiter les nœuds non traités selon l'ordre *First In First Out*, c'est-à-dire que les nœuds non traités ayant la distance au nœud origine minimale sont sélectionnés. Au contraire, l'exploration en profondeur (*depth-first search*) traite les nœuds selon l'ordre *Last In First Out*, c'est-à-dire que les nœuds les plus récents sont sélectionnés. Enfin, l'exploration privilégiant le meilleur nœud (*best-first search*) traite le nœud ayant la borne inférieure la plus faible. L'exploration en largeur est rarement choisie car elle utilise trop d'espace mémoire si la borne supérieure initiale n'est pas suffisamment bonne. L'exploration en profondeur limite l'utilisation de l'espace mémoire au nœud courant. La dernière exploration est l'exploration la plus performante car elle donne un bon compromis entre l'utilisation de la mémoire et le temps d'exécution. En effet, la convergence rapide vers de bonnes solutions réalisables élimine des zones de l'espace des solutions efficacement.

Comme la performance du B&B repose fortement sur la qualité des bornes inférieures calculées à chaque nœud de l'arbre et que les formulations peuvent présenter un nombre exponentiel de variables ou de contraintes, cette méthode peut être combinée avec des générations d'inégalités valides pour former les méthodes de séparations et coupes (*branch-and-cut* = B&C), ou avec de la génération de colonnes pour former les méthodes de séparations et génération de colonnes (*branch-and-price* = B&P). Enfin, les deux dernières techniques peuvent également être combinées pour aboutir à des méthodes de séparations et coupes et génération de colonnes (*branch-and-cut-and-price* = B&C&P).

2.4.2 Méthode de séparations et coupes

Les méthodes de coupes ont été introduites par [Gomory et al. \(1958\)](#). Elles simplifient la résolution d'un problème P en optimisant une relaxation de ce problème PR et en rajoutant des contraintes sur PR jusqu'à ce que la solution optimale de PR soit celle de P. Une relaxation sur l'intégrité des variables de décision est souvent choisie dans ce type de méthodes. Si la solution optimale de PR est réalisable pour P, alors c'est la solution optimale recherchée. Si la solution optimale de PR n'est pas réalisable pour P, alors il existe certaines contraintes de P qui ne sont pas respectées par cette solution. Un sous-ensemble de ces contraintes, appelées coupes, sont alors ajoutées à PR pour trouver une nouvelle solution optimale. La procédure se poursuit jusqu'à que la solution optimale courante ne viole aucune contrainte de P.

Dans l'espace des solutions, les solutions d'un programme linéaire sont un polyèdre convexe. Lorsqu'une solution optimale d'un problème linéaire est trouvée, celle-ci correspond à un sommet de ce polyèdre ou peut s'y ramener. Dans le cas d'un problème relâché, le polyèdre des solutions réalisables contient toutes les solutions du problème initial et est donc plus grand. Trouver des coupes revient à couper ce polyèdre de solutions pour le rendre plus petit. Les coupes sont dites valides si elles ne coupent pas le polyèdre du problème initial.

Les méthodes B&C ont longtemps été parmi les plus efficaces pour résoudre le VRP (Laporte et al. (1985), Lysgaard et al. (2004)).

2.4.3 Méthode de séparations et génération de colonnes

Desrosiers et al. (1984) ont introduit la méthode de génération de colonnes pour résoudre le problème de tournées de véhicules avec fenêtres de temps (*Vehicle Routing Problem with Time Windows* = VRPTW) où chaque client autorise un véhicule à le visiter seulement s'il arrive à une certaine période. Depuis lors, les méthodes B&P sont les méthodes les plus efficaces, avec le B&C&P, pour résoudre le VRP. Ces méthodes se basent sur la formulation de partitionnement. À chaque nœud de l'arbre de recherche, la relaxation linéaire de la formulation est résolue grâce à la technique de génération de colonnes (*cf.* Section 2.3).

2.4.4 Méthode de séparations et coupes et génération de colonnes

Ces méthodes combinent le B&B avec la technique d'ajout d'inégalités valides et de la génération de colonnes pour renforcer les bornes inférieures. Elles font partie des méthodes exactes les plus efficaces pour résoudre le VRP, avec les B&P, et font l'objet de nombreuses recherches (Fukasawa et al. (2006), Pecin et al. (2017), Costa et al. (2019)).

Les coupes dites robustes ont été définies par de Aragao et Uchoa (2003) comme des inégalités ne changeant pas la structure du sous-problème de la génération de colonnes et n'augmentant donc pas sa complexité. Pour le VRP, ces coupes concernent souvent les variables de flots sur les arcs. Bien que ces coupes soient performantes sur des instances faciles à résoudre, elles ne sont pas suffisantes pour éliminer l'écart entre les bornes inférieure et supérieure d'optimalité sur des instances plus difficiles.

2.4.5 Énumération explicite des colonnes

Soit LB une borne inférieure et UB une borne supérieure de la formulation considérée. Le gap est la différence de coût entre LB et UB . Pour fermer ce gap, certains auteurs (Fukasawa et al. (2006), Røpke (2012)) utilisent une méthode B&C&P classique. Baldacci et al. (2008) ont introduit l'énumération explicite des variables ayant un coût réduit inférieur à la valeur de ce gap et la résolution de la formulation sur ces variables avec un solveur de programmes linéaires en nombres entiers. Cette méthode a montré son efficacité sur les problèmes ayant un faible gap (Baldacci et al. (2008; 2011), Contardo et Martinelli (2014)).

2.5 Problèmes de tournées de véhicules multi-objectif

Dans cette section, les méthodes exactes présentées dans la littérature pour des problèmes de tournées de véhicules multi-objectif (*Multi-Objective Vehicle Routing Problem* = MOVRP) sont données. Seuls les problèmes considérant plusieurs véhicules et les méthodes générant l'ensemble des points non dominés sont abordés. Tout d'abord, les papiers de Jozefowicz et al. (2008), de Labadie et Prodhon (2014) et de Vega-Mejía et al. (2017) donnent un état de l'art complet sur les méthodes existantes, approchées ou exactes,

pour résoudre les MOVRP. La minimisation du coût total des arêtes empruntées par les véhicules reste un critère dans la majorité des problèmes multi-objectif étudiés, même si d'autres critères sont ajoutés selon la nature du problème. Ainsi, ils permettent de considérer une dimension écologique (Molina et al. (2014), Demir et al. (2014)), sociale (Halvorsen-Weare et Savelsbergh (2016)) ou de satisfaction des usagers (Lehuédé et al. (2014)).

Cette section aborde les problèmes de tournées de véhicules purs, ainsi que les problèmes d'emplacements-tournées (*location-routing problems*) ou d'inventaires-tournées (*Inventory-Routing = IR*). Les problèmes d'emplacements-tournées consistent à trouver des points subsidiaires du réseau pour y implanter des centres, tout en ayant des problèmes de tournées de véhicules entre les différents points subsidiaires ou à partir de ces centres. Dans le domaine de IR, une politique de distribution périodique doit être planifiée pour connaître l'ordre de service des clients, la quantité à délivrer et les routes à emprunter régulièrement. En version mono-objectif, les coûts totaux doivent être minimisés.

2.5.1 Réduction de la pollution

Les problèmes de tournées de véhicules verts (*Green Vehicle Routing Problem*) ou les problèmes de tournées polluantes (*Pollution-Routing Problem*) (Bektaş et Laporte (2011)) multi-objectif visent à réduire l'empreinte écologique des tournées effectuées en considérant les émissions de gaz à effet de serre ou l'énergie utilisée par un véhicule en fonction de sa charge ou de sa vitesse. De nombreuses méthodes approchées cherchent à caractériser au mieux les fonctions représentant cet impact écologique (Molina et al. (2014), Demir et al. (2014)), mais une seule méthode exacte a été appliquée à ce type de problèmes.

Toro et al. (2017) travaillent sur le *Capacitated Location-Routing Problem* où des dépôts annexes doivent être ouverts pour pouvoir servir la demande de tous les clients du graphe avec des véhicules placés dans ces centres. Ils ont formulé le problème avec deux objectifs minimisant les coûts opérationnels (ouverture de centre et coût de trajet), ainsi que la consommation d'essence et d'émission de gaz. Ils ont conçu une méthode exacte avec l'approche ε -contrainte où chaque formulation mono-objectif résultante est résolue par un solveur de Programmation Linéaire en Nombres Entiers (PLNE).

2.5.2 Équité entre les véhicules

D'autres problèmes travaillent sur l'équité entre les véhicules. Cette équité peut se traduire par la différence de distance parcourue par les véhicules ou l'heure d'arrivée au plus tard chez un client.

Reiter et Gutjahr (2012) ont proposé de minimiser le coût total des routes et de maximiser l'équilibre entre la taille des routes pour un problème de tournées de véhicules. Cet équilibre est obtenu en minimisant la différence entre la plus longue et la plus courte des routes empruntées. Ils ont utilisé une méthode B&C dans l'approche ε -contrainte de Laumanns et al. (2006) (cf. Section 1.3.2). Ils ont également utilisé deux algorithmes génétiques, dont un orienté bi-objectif, pour générer des bornes supérieures. Les instances

utilisées contiennent entre 16 et 57 nœuds avec 2 à 9 véhicules autorisés. L'ensemble des points non dominés est obtenu pour 10 instances sur 57 en 8 heures. Dans leur étude expérimentale, ils ont également noté la difficulté de la génération des points non dominés lorsque la valeur ε est très contrainte. Ainsi, ils proposent une heuristique qui ne génère pas la partie la plus contrainte du front de Pareto en soulignant que les entreprises peuvent faire des concessions sur le profit, mais pas vers la valeur extrême maximisant l'équité.

Une étude sur l'effet de quatre critères d'équité entre les routes, combinés à la minimisation du coût total des routes, a été faite par [Halvorsen-Weare et Savelsbergh \(2016\)](#). Ils ont utilisé la programmation dynamique, ainsi qu'une méthode ε -contrainte pour trouver l'ensemble complet minimal. Leur méthode a été testée sur des instances contenant jusqu'à 12 nœuds. Les résultats expérimentaux montrent que la minimisation de la différence entre la plus longue route et la plus courte est un critère d'équité entre les routes qui est robuste pour la plupart des problèmes de tournées de véhicules.

[Matl et al. \(2019\)](#) ont également fait une étude sur le compromis entre l'équité selon plusieurs ressources et le coût sur les solutions du VRP. Pour cela, 18 fonctions d'équité ont été trouvées et analysées à l'optimalité sur des instances de CVRP à 20 clients, ainsi que de manière approchée sur de grandes instances. La méthode exacte est une méthode de scalarisation, mais les temps d'exécution ne sont pas indiqués.

2.5.3 Satisfaction des usagers

[Tricoire et al. \(2012\)](#) ont introduit un problème bi-objectif de tournées couvrantes stochastique qui minimise la distance totale parcourue pour servir les centres de distribution avec le coût d'ouverture de ces centres et les demandes non couvertes attendues. Dans ce problème, les demandes des clients sont incertaines et sont approximées par une répartition empirique. Chaque centre a une capacité maximale de clients qu'il peut couvrir. Ils ont utilisé un algorithme de séparations et coupes intégré dans une méthode ε -contrainte. Ils ont également développé une heuristique qui permet de ne pas résoudre chaque problème à l'optimalité mais de s'arrêter avec un certain gap de tolérance fixe. Des instances réelles de 30 nœuds sont utilisées pour tester leurs méthodes dans un temps limite de trois jours. Le nombre de solutions dans \mathcal{Y}_N varie entre 6 et 112. L'heuristique donne une bonne approximation de \mathcal{Y}_N en des temps raisonnables.

Dans le domaine de la logistique urbaine, [Abounacer et al. \(2014\)](#) ont introduit un problème de location-transport qui cherche les meilleurs emplacements pour des centres de distribution d'aide humanitaire dans des zones sinistrées. Chaque centre utilise ses propres véhicules pour distribuer des biens de première nécessité à la population. Le problème minimise le temps total des routes, le nombre de centres ouverts et les zones non couvertes par les routes. Ils ont utilisé une méthode ε -contrainte pour résoudre le problème à l'optimalité. Ils ont généré 40 instances dont les plus grandes, contenant 4 centres et 45 habitants, sont résolues en plus de 8 heures. Les plus petites instances, contenant 3 centres et 15 habitants, sont résolues en 22 minutes en moyenne.

[Gutjahr et Dzibur \(2016\)](#) ont résolu le problème d'emplacements de centres de distribution bi-objectif minimisant le coût d'ouverture des centres et les demandes non couvertes

par ces centres. Ils ont développé une technique d'optimisation à deux niveaux intégrée dans l'approche ε -contrainte adaptative de Laumanns et al. (2006). Pour une valeur ε fixée, le problème mono-objectif est résolu grâce à une méthode de séparations et évaluations où les décisions de branchement sont effectuées sur les centres à ouvrir. Des instances réelles sont utilisées pour tester leur méthode.

Kovacs et al. (2015) ont étudié le *Multi-Objective Generalized Consistent Vehicle Routing Problem* où les clients doivent être servis plusieurs fois sur un échelon de temps. Les critères sont de minimiser le coût total des trajets des véhicules, maximiser la régularité du chauffeur visitant un client et de son temps d'arrivée. Pour résoudre ce problème tri-objectif, ils ont développé une méthode approchée utilisant la recherche locale. Deux méthodes exactes basées sur l'approche ε -contrainte sont également implémentées pour évaluer la qualité de leur heuristique. Chaque problème mono-objectif résultant est résolu par une méthode de séparations et de coupes. Ils concluent que 70% de régularité dans les temps d'arrivée des chauffeurs en plus peut être gagnée avec une perte de moins de 4% de coût total.

2.5.4 Réduction du risque

Certains problèmes traitent de la gestion de marchandises dangereuses et visent à réduire les risques de transport et de stockage de ces marchandises. Le risque peut également être pour les véhicules lors de trajets sur des routes abîmées après des événements climatiques catastrophiques tels qu'un tsunami.

Yu et Solvang (2016) ont résolu le *Bi-Objective Hazardous Waste Location-Routing Problems* qui consiste à sélectionner des endroits où placer des déchets toxiques qui minimisent les coûts de transport et les risques. Les risques se traduisent par la quantité de déchets déposée et transportée par rapport à la population avoisinante. L'approche ε -contrainte est utilisée et chaque problème mono-objectif est résolu par un solveur de PLNE.

Arab et al. (2018) ont étudié un problème de IR où les objectifs sont la minimisation des coûts et la minimisation du risque sur les routes empruntées. Ils ont développé une métaheuristique évolutionnaire, ainsi qu'une méthode exacte avec l'approche ε -contrainte où chaque formulation mono-objectif est résolue par un solveur de PLNE. La méthode exacte valide la qualité de la méthode approchée considérée.

2.5.5 Collecte d'un profit

Parragh et Tricoire (2019) ont proposé un B&B générique pour les problèmes entiers bi-objectif (cf. Section 1.3.4). Ils ont appliqué leur méthode sur trois problèmes différents dont le problème de la course d'orientation par équipes (*team-orienteeing problem*) bi-objectif qui minimise le coût total des routes, ainsi que le profit total collecté auprès des clients visités. L'efficacité de leur méthode a été montrée comparé à des méthodes de l'état de l'art pour les problèmes bi-objectif : la méthode ε -contrainte, la *balanced box method* et l'approche bi-directionnelle de la méthode ε -contrainte présentée par Boland et al. (2015).

2.5.6 Conclusion

Nous pouvons observer que peu de méthodes exactes ont été appliquées au problème de tournées de véhicules multi-objectif. La plupart de ces méthodes ont été appliquées soit pour valider la qualité des solutions retournées par une heuristique, soit pour faire une étude sur le compromis entre les objectifs. Elles appliquent alors une méthode de scalarisation, notamment la méthode ε -contrainte qui est facile à mettre en place et efficace pour les problèmes de tournées, où les problèmes mono-objectif résultants sont résolus avec un solveur PLNE. En conclusion, peu de papiers visent à améliorer l'efficacité des méthodes exactes s'appliquant à ces problèmes.

2.6 Conclusion du chapitre

Dans ce chapitre, nous avons vu les différentes méthodes de résolution exactes du VRP. Deux méthodes populaires pour le résoudre sont les méthodes de branchement et génération de colonnes, ainsi que la programmation dynamique. Le succès de ces méthodes repose sur la qualité de la borne inférieure obtenue, même si le nombre exponentiel de variables impose la résolution d'un sous-problème également \mathcal{NP} -complet. Une partie de ce chapitre propose de montrer les techniques pour améliorer la résolution de ces sous-problèmes et le nombre de travaux démontre l'intérêt de ces méthodes dans la résolution du VRP. Enfin, nous avons abordé l'état de l'art du MOVRP pour toutes les variantes de problème de tournées et d'objectifs à optimiser. Cela montre la faible quantité de méthodes exactes qui ont été appliquées à ce problème au vu de sa complexité. Ainsi, fournir une méthode exacte efficace pour le MOVRP permettrait non seulement d'étoffer les techniques d'optimisation combinatoire multi-objectif, mais également de proposer une méthode d'évaluation de la qualité des nombreuses méthodes approchées de la littérature.

Méthodes exactes pour les problèmes de tournées de véhicules bi-objectif

Sommaire

3.1	Introduction	42
3.2	Formulations mathématiques	43
3.2.1	Formulation bi-objectif	43
3.2.2	Formulation pour la méthode ε -contrainte	44
3.2.3	Formulation de la somme pondérée des objectifs	44
3.2.4	Conclusion	45
3.3	Résolution des problèmes mono-objectif avec la génération de colonnes	46
3.3.1	Étape 1 : Solution optimale de la relaxation linéaire	46
3.3.2	Étape 2 : Solution réalisable du problème entier	48
3.3.3	Étape 3 : Énumération de colonnes	48
3.3.4	Étape 4 : Solution optimale du problème entier	49
3.3.5	Conclusion	51
3.4	Méthodes pour le problème bi-objectif	52
3.4.1	Calcul des points extrêmes	52
3.4.2	Méthode 1 : basée sur l'approche ε -contrainte avec génération de toutes les colonnes	52
3.4.3	Méthode 2 : basée sur l'approche ε -contrainte par étapes	53
3.4.4	Méthode 3 : deux phases avec génération de toutes les colonnes pour les points supportés	53
3.4.5	Méthode 4 : deux phases par étapes	56
3.4.6	Conclusion	61
3.5	Résultats expérimentaux	61
3.5.1	Description des instances	62
3.5.2	Comparaison des méthodes	63
3.5.3	Conclusion	67
3.6	Conclusion du chapitre	67

3.1 Introduction

Dans ce chapitre, nous considérons un problème de tournées de véhicules bi-objectif (*Bi-Objective Vehicle Routing Problem* = BOVRP) générique. Le problème étudié consiste à trouver un ensemble de routes qui passe par tous les points d'un réseau et minimise deux sommes de coûts entiers indépendants affectés à chacune des routes. Ces deux coûts peuvent représenter plusieurs caractéristiques des problèmes de tournées de véhicules comme la distance parcourue par un véhicule, le temps de parcours, l'émission de polluants atmosphériques ou le profit collecté.

Le problème est défini sur un graphe $G = (V, E)$ non orienté. L'ensemble des sommets $V = \{0, \dots, n\}$ contient le dépôt 0 et n clients i , $1 \leq i \leq n$. Soit V' l'ensemble des clients tel que $V = V' \cup \{0\}$. Chaque client i a une demande q_i . L'ensemble des arêtes E est défini comme suit $E = \{(i, j) | i, j \in V, i < j\}$ et deux coûts entiers c_{ij}^1 et c_{ij}^2 , correspondants aux objectifs, sont associés à chaque arête (i, j) du graphe. Un ensemble K de véhicules homogènes est disponible au dépôt. Chaque véhicule effectue une route qui part du dépôt, visite un ensemble de clients et retourne au dépôt. La somme des demandes des clients visités par un véhicule ne doit pas dépasser la capacité maximale du véhicule Q . L'objectif du problème de tournées de véhicules bi-objectif est de définir un ensemble de routes qui visite tous les clients du graphe et qui minimise les deux coûts c^1 et c^2 indépendamment. Le problème peut également contenir d'autres contraintes opérationnelles comme les fenêtres de temps, ou une cardinalité maximale des routes, qui ne changent pas la structure des sous-problèmes.

Le but de ce chapitre est de présenter des méthodes exactes permettant de trouver l'ensemble minimum complet du BOVRP. Ainsi, tous les points non dominés associés à au moins une solution optimale présentant un compromis "intéressant" entre c^1 et c^2 doivent être obtenus. Ces méthodes se basent sur la même adaptation d'une méthode exacte de résolution des tournées de véhicules mono-objectif, mais sur des explorations différentes de l'espace des objectifs. Au total, quatre méthodes différentes sont présentées.

Dans la Section 3.2, le BOVRP est formulé, ainsi que des formulations mono-objectif utiles à la résolution du BOVRP. Une méthode de résolution exacte permettant de résoudre les formulations avec un seul objectif est présentée en Section 3.3. Cette méthode mono-objectif est utilisée pour créer des méthodes exactes de résolution du BOVRP en Section 3.4. Enfin, les différentes méthodes de résolution bi-objectif sont comparées dans la Section 3.5 sur un problème comportant des fenêtres de temps.

Contributions

La première partie présente des formulations de partitionnement, introduites en 1964 par Balinski et Quandt (1964), appliquées à notre problème. La seconde partie présente une adaptation d'une méthode exacte de résolution des problèmes de tournées de véhicules mono-objectif basée sur la génération de colonnes (Baldacci et al. (2008)) appliquée aux nouvelles formulations. Dans la troisième partie, nous proposons de combiner des approches classiques en optimisation bi-objectif comme la méthode ε -contrainte de Haimès et al. (1971)

et la méthode en deux phases de [Sylva et Crema \(2004\)](#) avec la génération de colonnes. Quatre combinaisons sont définies et comparées.

Les contributions de ce chapitre sont donc l'adaptation d'une méthode exacte efficace aux problèmes de tournées de véhicules bi-objectif, ainsi que la combinaison entre des méthodes exactes qui travaillent sur l'espace des objectifs et de la génération de colonnes.

Publications

Une partie du contenu de ce chapitre a été présentée dans des conférences internationales (ISCO, Odysseus, VEROLOG) et nationales (ROADEF, MOPGP). La conférence ISCO a donné lieu à une publication dans les *Lecture Notes in Computer Science*.

3.2 Formulations mathématiques

En premier lieu, cette section présente la formulation de partitionnement pour le BOVRP. Ensuite, trois formulations mono-objectif permettant de travailler sur une partie restreinte de l'espace des objectifs sont présentées.

3.2.1 Formulation bi-objectif

Pour résoudre le BOVRP, une formulation de partitionnement (ou *set-partitioning formulation*) est introduite. Soit R l'ensemble des routes réalisables. Soit a_{ir} un coefficient binaire indiquant si le client $i \in V'$ est visité par la route $r \in R$. Soit c_r^1 et c_r^2 les coûts totaux de la route r . Les variables binaires x_r de la formulation indiquent si la route r est sélectionnée dans la solution ($x_r = 1$) ou non ($x_r = 0$). Le BOVRP peut être formulé comme suit :

$$\left\{ \begin{array}{ll} \min (c^1(x), c^2(x)) = \left(\sum_{r \in R} c_r^1 x_r, \sum_{r \in R} c_r^2 x_r \right) & (3.1.1) \\ t.q. \sum_{r \in R} a_{ir} x_r \geq 1 & i \in V' \quad (3.1.2) \\ \sum_{r \in R} x_r \leq |K| & (3.1.3) \\ x_r \in \{0, 1\} & r \in R \quad (3.1.4) \end{array} \right. \quad (3.1)$$

Cette formulation présente deux contraintes (3.1.2) et (3.1.3) qui contraignent respectivement tous les clients à être visités et l'utilisation d'au plus $|K|$ véhicules. L'ensemble de définition des variables est décrit avec la contrainte (3.1.4). Les deux objectifs (3.1.1) minimisent la somme des deux coûts des arêtes utilisées. La résolution de cette formulation permet l'obtention de l'ensemble des points non dominés. Dans cette formulation, une route est également appelée une colonne du programme linéaire.

Pour pouvoir résoudre ce problème, nous allons utiliser des méthodes basées sur l'exploration de l'espace de recherche des objectifs. Le principe de ces méthodes consiste à résoudre une séquence de problèmes mono-objectifs qui se restreignent sur une partie spécifique de l'espace des objectifs et dont la réunion des espaces visités recrée la totalité de l'espace des

objectifs du problème bi-objectif initial. La formulation bi-objectif est donc transformée en formulations mono-objectif grâce à l'ajout de contraintes spécifiques comme décrit en Section 3.2.2, ou à la linéarisation des objectifs comme présenté en Section 3.2.3.

3.2.2 Formulation pour la méthode ε -contrainte

Dans cette formulation, une constante ε est introduite pour borner l'un des objectifs. Pour une valeur spécifique de ε , un unique point non dominé de la formulation (3.1) est trouvé. Ce point peut être un point faiblement non dominé. Par conséquent, pour trouver le front minimal complet de la formulation (3.1), la valeur ε devra varier en cours de résolution.

$$\left\{ \begin{array}{ll} \min \sum_{r \in R} c_r^1 x_r & (3.2.1) \\ t.q. \sum_{r \in R} a_{ir} x_r \geq 1 & i \in V' \quad (3.2.2) \\ \sum_{r \in R} x_r \leq |K| & (3.2.3) \\ \sum_{r \in R} c_r^2 x_r \leq \varepsilon & (3.2.4) \\ x_r \in \{0, 1\} & r \in R \quad (3.2.5) \end{array} \right. \quad (3.2)$$

L'objectif (3.2.1) minimise le seul objectif c^1 . Les contraintes (3.2.2) et (3.2.3) sont identiques aux contraintes (3.1.2) et (3.1.3). La définition du domaine des variables est également identique à la formulation (3.1). La nouvelle contrainte (3.2.4) permet de restreindre le second objectif c^2 de la formulation bi-objectif à être inférieur à la valeur ε . Nous pouvons remarquer que la résolution de la formulation donne la solution de coût c^1 minimal pour $\varepsilon = +\infty$.

3.2.3 Formulation de la somme pondérée des objectifs

Cette formulation consiste à faire la somme pondérée des objectifs de la formulation (3.1). Le reste des contraintes de la formulation (3.1) reste inchangé. Pour une valeur spécifique de w , $0 \leq w \leq 1$, un ou deux sommets de l'enveloppe convexe des points non dominés sont trouvés. Aucun point non supporté ne peut être atteint avec la formulation suivante (Geoffrion (1968)).

$$\left\{ \begin{array}{ll} \min w \sum_{r \in R} c_r^1 x_r + (1 - w) \sum_{r \in R} c_r^2 x_r & (3.3.1) \\ t.q. \sum_{r \in R} a_{ir} x_r \geq 1 & i \in V' \quad (3.3.2) \\ \sum_{r \in R} x_r \leq |K| & (3.3.3) \\ x_r \in \{0, 1\} & r \in R \quad (3.3.4) \end{array} \right. \quad (3.3)$$

Les contraintes (3.3.2), (3.3.3) et (3.3.4) sont équivalentes aux contraintes (3.1.2), (3.1.3) et (3.1.4). La formulation (3.3) présente un unique objectif étant la somme pondérée des deux objectifs de la formulation (3.1). Nous pouvons remarquer que la résolution de la formulation donne la solution de coût c^2 ou c^1 minimal pour respectivement $w = 0$ ou $w = 1$.

Pour pouvoir atteindre des points non dominés non supportés, la formulation (3.3) est généralisée pour restreindre l'espace des objectifs atteignable. Deux constantes L^1 et L^2 sont ajoutées dans ce but.

$$\left\{ \begin{array}{l} \min w \sum_{r \in R} c_r^1 x_r + (1-w) \sum_{r \in R} c_r^2 x_r \quad (3.4.1) \\ t.q. (3.3.2) - (3.3.4) \\ \sum_{r \in R} c_r^1 x_r \leq L^1 \quad (3.4.5) \\ \sum_{r \in R} c_r^2 x_r \leq L^2 \quad (3.4.6) \end{array} \right. \quad (3.4)$$

Les contraintes (3.4.5) et (3.4.6) restreignent les coûts c^1 et c^2 , respectivement, à être inférieurs à des valeurs prédéfinies. La variation des valeurs w , L^1 et L^2 dans la formulation (3.4) permet de trouver l'ensemble minimum complet. De plus, la formulation (3.2) est un cas particulier de la formulation (3.4) puisqu'elle est retrouvée pour les valeurs suivantes : $w = 1$, $L^1 = +\infty$ et $L^2 = \varepsilon$.

Soit $\lambda_i \geq 0$ ($i \in V'$), $\lambda_0 \leq 0$, $\alpha_1 \leq 0$ et $\alpha_2 \leq 0$ les variables duales respectivement associées aux contraintes (3.3.2), (3.3.3), (3.4.5) et (3.4.6), alors le dual de la relaxation linéaire de la formulation (3.4) est la formulation (3.5).

$$\left\{ \begin{array}{l} \max \sum_{i \in V'} \lambda_i + |K| \lambda_0 + L^1 \alpha_1 + L^2 \alpha_2 \\ t.q. \sum_{i \in V'} a_{ir} \lambda_i + \lambda_0 + c_r^1 \alpha_1 + c_r^2 \alpha_2 \leq w c_r^1 + (1-w) c_r^2 \quad r \in R \\ \lambda_i \geq 0 \quad i \in V' \\ \lambda_0 \leq 0 \\ \alpha_1 \leq 0 \\ \alpha_2 \leq 0 \end{array} \right. \quad (3.5)$$

Pour $\alpha_1 = 0$ et $\alpha_2 = 0$, nous retrouvons le dual de la relaxation linéaire de la formulation (3.3). Pour $\alpha_2 = 0$ et $w = 1$ et $L^1 = \varepsilon$, nous retrouvons le dual de la relaxation linéaire de la formulation (3.2).

3.2.4 Conclusion

Toutes les formulations précédemment introduites sont basées sur la formulation de partitionnement et contiennent un nombre exponentiel de variables. L'approche classique de résolution de ce type de formulation est de résoudre la relaxation linéaire des formulations

grâce à la génération de colonnes et ensuite d'appliquer une méthode de branchement pour atteindre la solution optimale entière. La suite de ce chapitre propose une méthode exacte de résolution de la formulation générale (3.4).

3.3 Résolution des problèmes mono-objectif avec la génération de colonnes

Dans cette section, nous expliquons une approche pour résoudre un problème de tournées de véhicules mono-objectif à l'optimalité. Nous considérons que la formulation (3.4) est le problème maître (PM) et sa relaxation linéaire le problème maître linéaire (PML). Le problème maître et sa relaxation linéaire restreints sur un ensemble de routes \bar{R} sont notés $\text{PMR}(\bar{R})$ et $\text{PMLR}(\bar{R})$. Les formulations (3.2) et (3.3) étant des cas particuliers de (3.4), la même méthode avec les paramètres adéquats est utilisée pour les résoudre.

La méthode exacte utilisée pour résoudre les problèmes mono-objectif est basée sur la méthode exacte de Baldacci et al. (2008) appliquée au problème de tournées de véhicules avec capacités (*Capacitated Vehicle Routing Problem* = CVRP). Cette méthode repose sur quatre étapes :

1. Calculer une borne inférieure LB , optimum du PML, et une solution duale associée λ par génération de colonnes.
2. Calculer une borne supérieure UB du problème maître PM.
3. Générer l'ensemble \bar{R} des routes réalisables ayant un coût réduit, calculé par rapport à λ , inférieur ou égal au gap, $UB - LB$.
4. Résoudre $\text{PMR}(\bar{R})$ à l'optimalité avec un solveur de programmes linéaires en nombres entiers.

Dans la suite de cette section, les notations suivantes sont utilisées :

- R est l'ensemble des routes réalisables du problème maître ;
- \bar{R} est l'ensemble des routes dans le gap et générées à l'étape 3 ;
- R_{ng} est l'ensemble des ng-routes (cf. Section 2.3.2.2) réalisables du problème maître ;
- \bar{R}_{ng} est l'ensemble des ng-routes générées pour la borne inférieure à l'étape 1.

Dans la suite de ce manuscrit, nous utilisons le terme *gap* pour désigner la différence de coût entre une borne inférieure et une borne supérieure. Les quatre étapes de cette méthode exacte sont décrites en détail dans les Sections 3.3.1 à 3.3.4.

3.3.1 Étape 1 : Solution optimale de la relaxation linéaire

La borne inférieure LB est calculée par génération de colonnes. Le sous-problème correspond à un problème de plus court chemin élémentaire avec contraintes de ressources (*Elementary Shortest Path Problem with Resource Constraints* = ESPPRC) qui est connu comme étant un problème NP-difficile (Dror (1994)) et sa résolution prend donc du temps. Pour l'accélérer, il est d'usage d'utiliser des règles de dominance, des bornes de complétion et une relaxation de l'élémentarité des routes pour énumérer efficacement les colonnes intéressantes.

3.3. Résolution des problèmes mono-objectif avec la génération de colonnes 47

La relaxation de l'élémentarité des routes choisie est la ng-relaxation expliquée en Section 2.3.2.2 pour le VRP. Aucun changement n'est appliqué pour le BOVRP. L'ensemble R de routes réalisables de la formulation considérée est maintenant remplacé par l'ensemble R_{ng} de ng-routes réalisables. Les bornes de complétion sont les mêmes que celles de la littérature pour le VRP et sont décrites en Section 2.3.2.3.

La génération de colonnes débute par l'initialisation d'un ensemble de ng-routes \tilde{R}_{ng} avec les ng-routes réalisables de cardinalité une et deux et avec une route artificielle qui visite l'ensemble des clients et qui a un coût \tilde{c}^1 suffisamment grand pour ne faire partie d'aucune solution optimale. D'autres ng-routes de coût réduit négatif vont ensuite être ajoutées à l'ensemble \tilde{R}_{ng} par l'algorithme d'étiquetage suivant.

3.3.1.1 Algorithme d'étiquetage

Résoudre le ng-SPPRC consiste à générer dynamiquement les colonnes de coût réduit négatif avec un algorithme d'étiquetage (*labeling algorithm*). Le coût réduit d'une colonne est calculé avec les variables duales associées à la solution optimale du PMLR(\tilde{R}_{ng}). Une étiquette E est un ng-chemin réalisable qui commence au dépôt et termine au client v_E avec un ng-ensemble NG_E , deux coûts c_E^1 et c_E^2 et un coût réduit \overline{c}_E . Soit le coût réduit \overline{c}_{ij} d'une arête $(i, j) \in E$:

$$\overline{c}_{ij} = (w - \alpha_1)c_{ij}^1 + (1 - w - \alpha_2)c_{ij}^2 - \frac{(\lambda_i + \lambda_j)}{2}$$

L'extension d'une étiquette E de son nœud terminal v_E à un client $j \in V \setminus NG_E$ produit une nouvelle étiquette E' via les règles (3.6)-(3.10) :

$$v_{E'} = j \tag{3.6}$$

$$NG_{E'} = (NG_E \cap N_j) \cup \{j\} \tag{3.7}$$

$$\overline{c}_{E'} = \overline{c}_E + \overline{c}_{v_E j} \tag{3.8}$$

$$c_{E'}^1 = c_E^1 + c_{v_E j}^1 \tag{3.9}$$

$$c_{E'}^2 = c_E^2 + c_{v_E j}^2 \tag{3.10}$$

où $N_j, j \in V$, est le ng-ensemble pré-défini pour le client j . Ces règles de propagation (3.6), (3.7) et (3.8) sont couramment employées en génération de colonnes pour la plupart des problèmes de tournées de véhicules. Elles mettent respectivement à jour le nœud terminal, le ng-ensemble et le coût réduit. Les règles (3.9) et (3.10) sont spécifiques au BOVRP et mettent à jour les coûts c^1 et c^2 courants de la route.

3.3.1.2 Règles de dominance

Les règles de dominance suivantes évitent de générer des labels qui ne pourront pas aboutir à des ng-routes de coût réduit minimum. Une étiquette E^1 domine une étiquette E^2

si les conditions suivantes sont satisfaites :

$$v_{E^1} = v_{E^2} \quad (3.11)$$

$$NG_{E^1} \subseteq NG_{E^2} \quad (3.12)$$

$$c_{E^1}^1 \leq c_{E^2}^1 \quad (3.13)$$

$$c_{E^1}^2 \leq c_{E^2}^2 \quad (3.14)$$

$$\overline{c_{E^1}} \leq \overline{c_{E^2}} \quad (3.15)$$

Les quatre premières conditions assurent que tous les clients $i \in V$ qui peuvent être atteints par propagation de l'étiquette E^2 peuvent également être atteints par propagation de l'étiquette E^1 . La dernière condition permet de vérifier que le coût réduit de l'étiquette E^2 est supérieur à celui de E^1 . Pour des raisons d'efficacité de l'algorithme, la règle (3.12) est vérifiée en dernière car c'est la règle la plus coûteuse.

3.3.2 Étape 2 : Solution réalisable du problème entier

À la fin de la génération de LB , un ensemble $\tilde{R}_{ng} \subseteq R_{ng}$ de ng-routes est disponible. Un sous-ensemble \tilde{R}_{elem} de routes réalisables élémentaires peut être facilement obtenu en enlevant toutes les routes non élémentaires de \tilde{R}_{ng} . Une borne supérieure UB du PM est calculée en résolvant $PMR(\tilde{R}_{elem})$.

3.3.3 Étape 3 : Énumération de colonnes

Dans cette partie, le coût réduit d'une route est calculé avec la solution duale de l'étape une. Lors de la troisième étape de la méthode exacte, un algorithme d'étiquetage mono-directionnel ou bi-directionnel est utilisé pour générer toutes les routes de coût réduit inférieur ou égal à la différence de coût entre UB et LB . Cette différence de coût est le gap γ défini par (3.16).

$$\gamma = w |c^1(UB) - c^1(LB)| + (1 - w) |c^2(UB) - c^2(LB)| \quad (3.16)$$

Dans cet algorithme de programmation dynamique, une étiquette E est un chemin réalisable élémentaire qui commence au dépôt, visite les clients C_E et se termine au client v_E avec un coût réduit $\overline{c_E}$, un coût c_E^1 et un coût c_E^2 . Les règles d'extension de l'étiquette E pour obtenir l'étiquette E' sont définies par (3.6) et (3.8)-(3.10), ainsi que la mise à jour de l'ensemble des clients visités décrit par la règle (3.17).

$$C_{E'} = C_E \cup \{v_E\} \quad (3.17)$$

La règle de dominance éliminant des étiquettes non prometteuses est la suivante. Une étiquette E^1 domine une étiquette E^2 si les conditions (3.11), (3.13), (3.14) et (3.15) sont respectées. On ajoute aussi la condition (3.18) suivante :

3.3. Résolution des problèmes mono-objectif avec la génération de colonnes 49

$$C_{E^1} \subseteq C_{E^2} \quad (3.18)$$

Les mêmes bornes de complétion que celles pour le calcul de la borne inférieure sont utilisées. À la fin de l'énumération des colonnes dans le gap, l'ensemble \bar{R} contient les routes de la solution optimale entière de la formulation considérée.

3.3.4 Étape 4 : Solution optimale du problème entier

Lorsque les colonnes dans le gap ont été générées, la solution optimale du PMR(\bar{R}) est obtenue grâce à un solveur de programmes linéaires en nombres entiers. Le Théorème 1 montre que la solution optimale d'un programme linéaire général est bien obtenue avec la méthode de Baldacci et al. (2008).

Théorème 1. *Soit la formulation suivante :*

$$\left\{ \begin{array}{ll} \min c^T x & (3.19.1) \\ t.q. A^T x \geq b & (3.19.2) \\ D^T x \leq e & (3.19.3) \\ x \geq 0 & (3.19.4) \end{array} \right. \quad (3.19)$$

où c et x sont des vecteurs de taille n , b est un vecteur de taille m_1 , e un vecteur de taille m_2 , A une matrice de taille (m_1, n) et D une matrice de taille (m_2, n) .

Le dual de la formulation (3.19) est présenté en formulation (3.20). La solution duale associée aux contraintes (3.19.2) est un vecteur y de taille m_1 et la solution duale associée aux contraintes (3.19.3) est un vecteur z de taille m_2 .

$$\left\{ \begin{array}{ll} \max b^T y + e^T z & (3.20.1) \\ t.q. Ay + Dz \leq c & (3.20.2) \\ y \geq 0 & (3.20.3) \\ z \leq 0 & (3.20.4) \end{array} \right. \quad (3.20)$$

Soit x^{LB} la solution optimale de la formulation (3.19) et x^{UB} une solution réalisable de la formulation (3.19) avec des variables x entières. Soit y^{LB} et z^{LB} les solutions optimales de la formulation duale (3.20). La somme des coûts réduits des colonnes i telles que $x_i^{UB} > 0$ est inférieure ou égale à la différence de coût c entre x^{UB} et x^{LB} . Cette différence de coût est appelée le gap γ . L'égalité est vérifiée si ($y^{LB} = 0$ ou $\sum_{i=1}^n a_{ij} x_i^{UB} = b_j, \forall j \in \llbracket 1; m_1 \rrbracket$) et ($z^{LB} = 0$ ou $\sum_{i=1}^n d_{ij} x_i^{UB} = e_j, \forall j \in \llbracket 1; m_2 \rrbracket$).

Démonstration. Soit a_{ij} (respectivement d_{ij}) la valeur de la i^{eme} ligne et j^{eme} colonne de la matrice A (respectivement D). Soit y_i la i^{eme} valeur du vecteur y . Le gap est décrit par l'équation (3.21), ainsi que le coût réduit d'une colonne x_i en équation (3.22).

$$\gamma = \sum_{i=1}^n c_i x_i^{UB} - \sum_{i=1}^n c_i x_i^{LB} \quad (3.21)$$

$$\bar{c}_i = c_i - \sum_{j=1}^{m_1} a_{ij} y_j^{LB} - \sum_{j=1}^{m_2} d_{ij} z_j^{LB} \quad \forall i \in \llbracket 1; n \rrbracket \quad (3.22)$$

Par (3.19.3), $\forall j \in \llbracket 1; m_2 \rrbracket$, $\sum_{i=1}^n d_{ij} x_i \leq e_j \Rightarrow \sum_{i=1}^n d_{ij} x_i z_j^{LB} \geq e_j z_j^{LB}$ car $z_j^{LB} \leq 0$

$$\text{Ainsi, } \sum_{j=1}^{m_2} e_j z_j^{LB} \leq \sum_{j=1}^{m_2} \left(\sum_{i=1}^n d_{ij} x_i \right) z_j^{LB} \quad (3.23)$$

$$\text{De même, par (3.19.2) et (3.20.3) : } \sum_{j=1}^{m_1} b_j y_j^{LB} \leq \sum_{j=1}^{m_1} \left(\sum_{i=1}^n a_{ij} x_i^{UB} \right) y_j^{LB} \quad (3.24)$$

Ainsi, par le théorème de la dualité forte, on sait qu'à l'optimalité :

$$\begin{aligned} & \sum_{i=1}^n c_i x_i^{LB} = \sum_{j=1}^{m_1} b_j y_j^{LB} + \sum_{j=1}^{m_2} e_j z_j^{LB} \\ \Rightarrow & \sum_{i=1}^n c_i x_i^{LB} \leq \sum_{j=1}^{m_1} \left(\sum_{i=1}^n a_{ij} x_i^{UB} \right) y_j^{LB} + \sum_{j=1}^{m_2} \left(\sum_{i=1}^n d_{ij} x_i^{UB} \right) z_j^{LB} \quad \text{par (3.23) et (3.24)} \\ \Rightarrow & \sum_{i=1}^n c_i x_i^{UB} - \sum_{j=1}^{m_1} \left(\sum_{i=1}^n a_{ij} x_i^{UB} \right) y_j^{LB} - \sum_{j=1}^{m_2} \left(\sum_{i=1}^n d_{ij} x_i^{UB} \right) z_j^{LB} \leq \sum_{i=1}^n c_i x_i^{UB} - \sum_{i=1}^n c_i x_i^{LB} \\ \Rightarrow & \sum_{i=1}^n \left(c_i - \sum_{j=1}^{m_1} a_{ij} y_j^{LB} - \sum_{j=1}^{m_2} d_{ij} z_j^{LB} \right) x_i^{UB} \leq \gamma \quad \text{par (3.21)} \\ \Rightarrow & \sum_{i=1}^n \bar{c}_i x_i^{UB} \leq \gamma \quad \text{par (3.22)} \end{aligned}$$

Par conséquent, la somme du coût réduit des colonnes i telles que $x_i^{UB} > 0$ est inférieure ou égale au gap. \square

L'énumération des colonnes dans le gap est une méthode qui est applicable à un problème linéaire général et donc au BOVRP. Dans le cas du BOVRP, nous déduisons de la preuve du théorème que la somme des coûts réduits d'une solution UB est égale à la

3.3. Résolution des problèmes mono-objectif avec la génération de colonnes 51

différence de coût entre UB et la borne inférieure LB si :

$$((\lambda_i = 0 \quad \text{ou} \quad \sum_{r \in R} a_{ir} x_r^{UB} = 1, \forall i \in V') \quad (3.25)$$

$$\text{et } (\lambda_0 = 0 \quad \text{ou} \quad \sum_{r \in R} x_r^{UB} = |K|) \quad (3.26)$$

$$\text{et } (\alpha_1 = 0 \quad \text{ou} \quad \sum_{r \in R} c_r^1 x_r^{UB} = L^1) \quad (3.27)$$

$$\text{et } (\alpha_2 = 0 \quad \text{ou} \quad \sum_{r \in R} c_r^2 x_r^{UB} = L^2) \quad (3.28)$$

La condition (3.25) stipule que chaque client est servi strictement plus d'une fois dans LB ou que la solution réalisable sert exactement une seule fois chaque client. La première partie n'est jamais vérifiée, alors que la seconde partie est peut toujours être vérifiée dans un graphe qui respecte l'inégalité triangulaire. Cette première condition est donc toujours vraie dans un graphe respectant l'inégalité triangulaire. Ensuite, la condition (3.26) est vérifiée si la borne inférieure n'utilise pas tous les véhicules disponibles ou si la borne supérieure utilise tous les véhicules disponibles. Si la borne inférieure utilise tous les véhicules disponibles, alors la borne supérieure également et la deuxième partie est vérifiée. Cette condition est donc toujours vraie. Enfin, la condition (3.27) (respectivement (3.28)) retourne vraie si la borne inférieure n'est pas située sur la contrainte du coût c^1 (respectivement c^2) ou si la borne supérieure est située sur cette contrainte. La première partie n'est jamais vérifiée pour L^1 (respectivement L^2) inférieur au coût c^1 du point c^1 -extrême (respectivement c^2 du point c^2 -extrême). La deuxième partie n'est pas toujours vraie.

Ainsi, la propriété peut être simplifiée aux conditions (3.27) et (3.28). Si ces deux conditions ne sont pas vérifiées, alors la somme des coûts réduits de UB est strictement inférieure au gap.

3.3.5 Conclusion

La résolution des formulations (3.2), (3.3) et (3.4) se fait grâce à la méthode exacte décrite dans cette section. Les règles de dominance doivent être adaptées selon les formulations, ainsi que le calcul du coût réduit avec les paramètres adéquats. Cette méthode mono-objectif va être utilisée plusieurs fois au cours des méthodes bi-objectif présentées en Section 3.4. Dans la suite de ce chapitre, lorsque la relaxation linéaire d'une formulation est résolue pour donner une borne inférieure LB , ce sera avec la méthode présentée en Section 3.3.1. De même, une borne réalisable UB sera calculée avec la méthode en Section 3.3.2. Enfin, l'énumération des colonnes de coût réduit inférieur au gap, décrite en Section 3.3.3, sera faite via l'appel à la fonction $genroute(w, L^1, L^2, LB, UB, \lambda)$, avec w , L^1 et L^2 les paramètres de la formulation (3.4) et λ une solution optimale de la formulation duale. Lorsque le problème maître est la formulation (3.2) (respectivement (3.3)), l'appel à cette fonction sera $genroute(\varepsilon, LB, UB, \lambda)$ (respectivement $genroute(w, LB, UB, \lambda)$).

3.4 Méthodes pour le problème bi-objectif

Cette section présente des méthodes exactes bi-objectif de résolution de la formulation (3.1). Elles sont basées sur l’exploration de l’espace des objectifs via l’utilisation des formulations (3.2), (3.3) ou (3.4), présentées en Section 3.2, et de la méthode exacte de la Section 3.3. Nous introduisons différentes notations selon les formulations :

- la formulation (3.2) pour le paramètre ε sera notée $\text{PM2}(\varepsilon)$, sa relaxation linéaire $\text{PML2}(\varepsilon)$ et la formulation duale du $\text{PML2}(\varepsilon)$, $\text{PDL2}(\varepsilon)$. La formulation (3.2) et sa relaxation linéaire sur un nombre restreint de colonnes \bar{R} seront notées $\text{PMR2}(\varepsilon, \bar{R})$ et $\text{PMLR2}(\varepsilon, \bar{R})$.
- la formulation (3.3) pour le paramètre w sera notée $\text{PM3}(w)$, sa relaxation linéaire $\text{PML3}(w)$ et la formulation duale du $\text{PML3}(w)$, $\text{PDL3}(w)$. La formulation (3.3) et sa relaxation linéaire sur un nombre restreint de colonnes \bar{R} seront notées $\text{PMR3}(w, \bar{R})$ et $\text{PMLR3}(w, \bar{R})$.
- la formulation (3.4) pour les paramètres w , L^1 et L^2 sera notée $\text{PM4}(w, L^1, L^2)$, sa relaxation linéaire $\text{PML4}(w, L^1, L^2)$ et la formulation duale du $\text{PML4}(w, L^1, L^2)$, $\text{PDL4}(w, L^1, L^2)$. La formulation (3.4) et sa relaxation linéaire sur un nombre restreint de colonnes \bar{R} seront notées $\text{PMR4}(w, L^1, L^2, \bar{R})$ et $\text{PMLR4}(w, L^1, L^2, \bar{R})$.

3.4.1 Calcul des points extrêmes

En début de chaque méthode bi-objectif, les points extrêmes vont être calculés pour définir l’espace des objectifs à explorer. Pour calculer le point c^1 -extrême E_1 , la méthode exacte de la Section 3.3 est utilisée sur la formulation (3.4) avec les paramètres suivants : $w = 1$, $L^1 = +\infty$ et $L^2 = +\infty$. Pour le point c^2 -extrême E_2 les paramètres $w = 0$, $L^1 = +\infty$ et $L^2 = +\infty$ sont appliqués dans la formulation (3.4). La fonction `calcul_extreme()` retourne les deux points extrêmes E_1 et E_2 .

3.4.2 Méthode 1 : basée sur l’approche ε -contrainte avec génération de toutes les colonnes

Cette première méthode est basée sur l’approche ε -contrainte introduite par [Haimès et al. \(1971\)](#) et utilise la formulation (3.2). Elle consiste en un unique appel à la fonction `genroute` qui génère toutes les routes nécessaires au problème bi-objectif et à l’application directe de la méthode ε -contrainte sur cet ensemble réduit de routes.

Avant la génération de routes, les deux points extrêmes E_1 et E_2 sont calculés comme indiqué dans la Section 3.4.1. Soit LB l’optimum du $\text{PML2}(+\infty)$ et λ une solution duale associée. L’ensemble de routes \bar{R} est produit via l’appel à la fonction `genroute`($\varepsilon, LB, E_2, \lambda$). Ainsi, toutes les colonnes des solutions optimales associées aux points non dominés du problème sont générées lors de cet appel. La méthode ε -contrainte est appliquée sur $\text{PMR2}(\varepsilon, \bar{R})$ pour ε variant entre $c^2(E_1) - 1$ et $c^2(E_2) + 1$. Le pseudo-code de cette première méthode est donné par l’Algorithme 1.

Les avantages de cette méthode sont de n’utiliser que deux solutions réalisables et deux bornes inférieures en entrée, ainsi que de ne faire qu’un unique appel à la fonction

Algorithme 1 Algorithme ε -contrainte avec génération de toutes les colonnes

Entrée : Le graphe G représentant le BOVRP

Sortie : L'ensemble Y des points non dominés

```

 $Y \leftarrow \emptyset$ 
 $LB_1, \lambda \leftarrow$  optimum du PML2(+ $\infty$ ) et solution optimale du PDL2(+ $\infty$ )
 $E_1, E_2 \leftarrow$  calcul_extreme()
 $\varepsilon = c^2(E_1) - 1$ 
 $\bar{R} \leftarrow$  genroute( $\varepsilon, LB_1, E_2, \lambda$ )
tant que  $\varepsilon > c^2(E_2)$  faire
   $O \leftarrow$  solution optimale du PMR2( $\varepsilon, \bar{R}$ )
   $Y \leftarrow Y \cup O$ 
   $\varepsilon = c^2(O) - 1$ 
fin tant que
retourne  $Y$ 

```

de génération de routes. Cependant, cela signifie que toutes les colonnes intéressantes du problème bi-objectif sont générées en une fois et que les problèmes maîtres entiers sont résolus autant de fois que le nombre de points non dominés sur toutes ces routes.

3.4.3 Méthode 2 : basée sur l'approche ε -contrainte par étapes

Cette méthode est également basée sur l'approche ε -contrainte de [Haines et al. \(1971\)](#) et la formulation (3.2). Pour améliorer son efficacité, on cherche à réduire le nombre de colonnes générées à chaque appel de la fonction *genroute*. En premier lieu, les points extrêmes E_1 et E_2 sont calculés pour pouvoir cerner la zone de recherche. À chaque itération i pour une certaine valeur de ε , la méthode calcule une borne inférieure LB_i et supérieure UB_i et génère l'ensemble des routes \bar{R}_i avec un appel à *genroute*. La formulation est résolue sur \bar{R}_i . Si la borne supérieure est telle que $c^1(UB_i) > c^1(E_2)$, alors UB_i est remplacé par E_2 et les PMR des itérations suivantes sont résolus sur \bar{R}_i sans rappel à la fonction *genroute*. L'algorithme est décrit dans l'Algorithme 2.

L'avantage de cette méthode est de générer moins de routes que la première méthode pour le calcul de chaque point non dominé. Cependant, les opérations de résolution du PML2 et de calcul de solutions réalisables sont répétées pour chaque point non dominé plus un (pour l'arrêt de la méthode ε -contrainte). De plus, les routes peuvent être générées plusieurs fois et l'ajout de la contrainte (3.2.4) rend la formulation plus difficile à résoudre à mesure que la valeur de ε est petite et donc contraignante.

3.4.4 Méthode 3 : deux phases avec génération de toutes les colonnes pour les points supportés

Cette méthode se divise en deux phases. La première phase permet de générer l'ensemble des points non dominés situés dans le triangle formé par les deux points extrêmes et le point idéal I , grâce à la formulation (3.3). La deuxième phase se base sur la formulation (3.4)

Algorithme 2 Algorithme ε -contrainte par étapes

Entrée : Le graphe G représentant le BOVRP

Sortie : L'ensemble Y des points non dominés

```

 $Y \leftarrow \emptyset$ 
 $E_1, E_2 \leftarrow \text{calcul\_extreme}()$ 
 $\varepsilon = c^2(E_1) - 1$ 
 $\text{regenerere} = \text{true}$ 
tant que  $\varepsilon > c^2(E_2)$  faire
    si  $\text{regenerere}$  alors
         $LB, \lambda \leftarrow \text{optimum du PML2}(\varepsilon)$  et solution optimale du PDL2( $\varepsilon$ )
         $UB \leftarrow \text{solution réalisable du PM2}(\varepsilon)$ 
        si  $c^1(UB) > c^1(E_2)$  alors
             $UB = E_2$ 
        fin si
         $\bar{R} \leftarrow \text{genroute}(\varepsilon, LB, UB, \lambda)$ 
    fin si
    si  $c^1(UB) \geq c^1(E_2)$  alors
         $\text{regenerere} = \text{false}$ 
    fin si
     $O \leftarrow \text{solution optimale du PMR2}(\varepsilon, \bar{R})$ 
     $Y \leftarrow Y \cup O$ 
     $\varepsilon = c^2(O) - 1$ 
fin tant que
retourne  $Y$ 

```

qui restreint l'espace des objectifs pour pouvoir explorer le triangle formé par deux points non dominés consécutifs et le nadir local, si besoin.

En début de la première phase, les deux points extrêmes E_1 et E_2 sont calculés comme expliqué dans la Section 3.4.1. Une direction de recherche w_0 pour la fonction objective est calculée avec la formule (3.29) où $P_1 = E_1$ et $P_2 = E_2$.

$$w_0 = \frac{c^1(P_2) - c^1(P_1)}{c^2(P_1) - c^2(P_2) + c^1(P_2) - c^1(P_1)} \quad (3.29)$$

Cette formule qui retourne la direction de recherche entre deux points P_1 et P_2 est appelée dans les algorithmes suivants via la fonction $\text{direction_recherche}(P_1, P_2)$.

Soit LB_0 l'optimum du PML3(w_0) et λ^0 une solution optimale du PDL2(w_0). L'appel à la fonction $\text{genroute}(w_0, +\infty, +\infty, LB_0, E_1, \lambda_0)$ retourne un ensemble de routes \bar{R}_0 de coût réduit, calculé à partir des variables duales λ_0 , qui est inférieur ou égal à la différence de coût entre E_1 et LB_0 . Ensuite, la méthode ε -contrainte est appelée sur cet ensemble \bar{R}_0 . À la fin de cette première phase, tous les points non dominés du triangle formé par les deux points extrêmes E_1 et E_2 et le point idéal I sont générés.

Lors de la deuxième étape, les points non dominés de coût c^1 consécutifs P_i et P_{i+1} , trouvés précédemment, sont examinés pour savoir si le triangle formé par ces deux points

et le nadir local N_i est situé dans la zone déjà observée, i.e. le triangle (E_1, E_2, I) . Deux cas différents peuvent se produire :

- Si N_i est tel que $w c^1(N_i) + (1 - w) c^2(N_i) \leq w c^1(E_1) + (1 - w) c^2(E_1)$, cela signifie que tous les points non dominés situés dans le triangle (P_i, P_{i+1}, N_i) ont été trouvés par la méthode ε -contrainte de la première phase. En effet, ce résultat est immédiat car la solution réalisable considérée lors du calcul du gap a une valeur de la fonction objective supérieure ou égale à la valeur de la fonction objective de N_i .
- Si N_i est tel que $w c^1(N_i) + (1 - w) c^2(N_i) > w c^1(E_1) + (1 - w) c^2(E_1)$, certains points non dominés peuvent ne pas avoir été générés. La formulation (3.4) est alors utilisée en posant $L^1 = c^1(P_{i+1}) - 1$, $L^2 = c^2(P_i) - 1$ et w la direction de recherche calculée avec la formule (3.29) où $P_1 = P_i$ et $P_2 = P_{i+1}$. Les deux contraintes (3.4.5) et (3.4.6) limitent l'espace des objectifs à un triangle où des points non dominés non supportés peuvent se trouver. Avec ces deux contraintes, il est possible de ne pas obtenir de solution réalisable. Soit LB_i l'optimum du $PML4(w, L^1, L^2)$ et λ_i la solution optimale du $PDL4(w, L^1, L^2)$. L'appel à la fonction $genroute(w, L^1, L^2, LB_i, N_i, \lambda_i)$ retourne l'ensemble de routes \bar{R}_i . Ensuite, la méthode ε -contrainte est appelée sur cet ensemble \bar{R}_i avec ε variant entre $c^2(P_i) - 1$ et $c^2(P_{i+1} + 1)$.

L'Algorithme 3 résume cette méthode et la Figure 3.1 permet d'illustrer son fonctionnement. Sur le schéma de gauche, les points extrêmes E_1 et E_2 sont indiqués, ainsi que la borne inférieure LB_0 et la direction de recherche w_0 . Tous les points non dominés situés sous la droite reliant E_1 et E_2 peuvent être trouvés avec la méthode ε -contrainte de la première phase. Dans cet exemple, ces points sont P_2 , P_3 et P_4 . Le point P_3 n'est pas situé sur l'enveloppe convexe des points non dominés et n'est donc pas un point supporté. À droite, les nadirs locaux entre deux points non dominés successifs sont représentés et, lors de la deuxième phase, tous ces triangles sont explorés. Les deux cas définis précédemment sont illustrés dans ce schéma. Le triangle (P_2, P_3, N_2) a déjà été exploré car N_2 est situé sous la droite $(E_1 E_2)$. À l'inverse, tous les points non dominés situés dans la partie du triangle (E_1, P_2, N_1) au-dessus de la droite $(E_1 E_2)$ n'ont pas encore été générés.

Dans cette méthode, un unique appel à la fonction $genroute$ est nécessaire pour obtenir les points supportés et la plupart des points non dominés. Cependant, ce premier appel à $genroute$ génère un nombre important de routes qui balayent tout le triangle formé par les deux points extrêmes et le point idéal I . Le problème maître restreint doit être résolu sur ce grand nombre de routes pour, au moins, tous les points supportés. De plus, rajouter des contraintes sur l'espace atteignable des objectifs complique la formulation (3.3). Le nombre de solutions réalisables à obtenir est limité à deux, pour le calcul des points extrêmes. Le nombre de bornes inférieures calculées est également faible puisqu'il est égal à trois plus le nombre de fois où le nadir local entre deux points non-dominés consécutifs est situé au-dessus de la droite $(E_1 E_2)$.

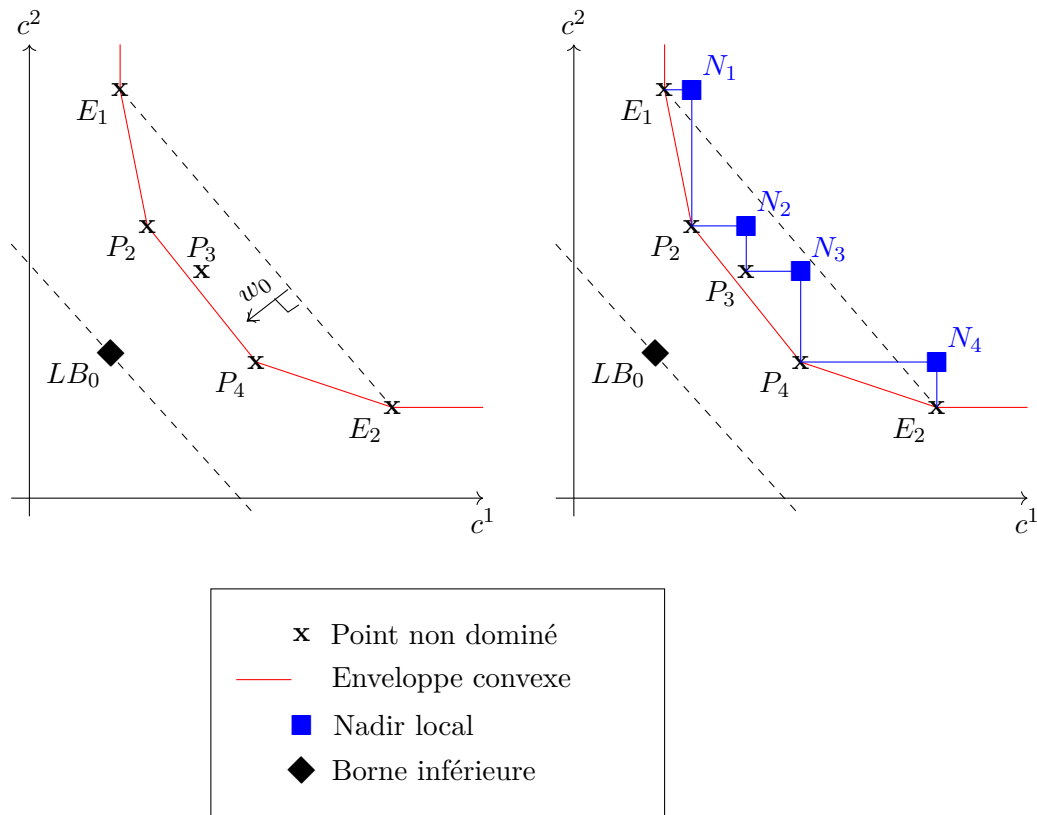


FIGURE 3.1: Figure illustrant les points non dominés durant la première phase de la méthode en deux phases (à gauche) et les points de nadir local entre deux points non dominés successifs (à droite).

3.4.5 Méthode 4 : deux phases par étapes

Cette méthode présente également deux phases et reprend la méthode précédente en diminuant le premier gap calculé pour diminuer le nombre de colonnes générées avec un appel à la fonction *genroute*. La première phase permet de ne générer que les points non dominés supportés grâce à la formulation (3.3). La deuxième phase se base sur la formulation (3.4) pour explorer l'espace entre deux points supportés consécutifs.

L'initialisation de la méthode calcule les deux points extrêmes E_1 et E_2 de la formulation (3.1). Posons $P_1 = E_1$ et $P_2 = E_2$. Une itération de la première phase se passe comme suit pour les points P_1 et P_2 . Tout d'abord, une nouvelle direction w_i de la fonction objective est calculée avec la formule (3.29). Notons que lors de la première itération la direction de recherche w_0 est celle de la méthode 3. De plus, quel que soit P_1 et P_2 dans \mathcal{Y} , $w c^1(P_1) + (1 - w) c^2(P_1) = w c^1(P_2) + (1 - w) c^2(P_2)$. Soit LB l'optimum de $LMP3(w)$, λ une solution optimale du $PDL3(w)$ et UB une solution réalisable du $PM3(w)$. L'appel à la fonction $genroute(w, LB, UB, \lambda)$ retourne un ensemble de routes \bar{R} de coût réduit, calculé à partir des variables duales λ , qui est inférieur ou égal à la différence de coût entre LB et UB . Le point non dominé O , solution optimale du $PMR3(w, \bar{R})$ est alors

Algorithme 3 Algorithme de la méthode en deux phases avec génération de toutes les colonnes pour les points supportés

Entrée : Le graphe G représentant le BOVRP

Sortie : L'ensemble Y des points non dominés

```

 $W \leftarrow \emptyset$ 
 $E_1, E_2 \leftarrow \text{calcul\_extreme}()$ 
 $w_0 = \text{direction\_recherche}(E_1, E_2)$ 
 $LB_0, \lambda_0 \leftarrow \text{optimum du PML3}(w_0)$  et solution optimale du PDL3( $w_0$ )
 $\bar{R}_0 \leftarrow \text{genroute}(w_0, LB_0, E_1, \lambda_0)$ 
 $\varepsilon = c^2(E_1) - 1$ 
tant que  $\varepsilon > c^2(E_2)$  faire
   $O \leftarrow \text{solution optimale du PMR2}(\varepsilon, \bar{R}_0)$ 
   $W \leftarrow W \cup O$ 
   $\varepsilon = c^2(O) - 1$ 
fin tant que
 $Y \leftarrow W$ 
pour  $i \in 1..|W| - 1$  faire
   $N_i = (c^1(W_{i+1}) - 1, c^2(W_i) - 1)$ 
  si  $w c^1(N_i) + (1 - w) c^2(N_i) > w c^1(E_1) + (1 - w) c^2(E_1)$  alors
     $w = \text{direction\_recherche}(W_i, W_{i+1})$ 
     $LB, \lambda \leftarrow \text{optimum du PML4}(w, c^1(N_i), c^2(N_i))$  et solution optimale du
    PDL4( $w, c^1(N_i), c^2(N_i)$ )
     $\bar{R} \leftarrow \text{genroute}(w, c^1(N_i), c^2(N_i), LB, N_i, \lambda)$ 
     $\varepsilon = c^2(N_i)$ 
    tant que  $\varepsilon > c^2(W_{i+1})$  faire
       $O \leftarrow \text{solution optimale du PMR2}(\varepsilon, \bar{R})$ 
       $Y \leftarrow Y \cup O$ 
       $\varepsilon = c^2(O) - 1$ 
    fin tant que
  fin si
fin pour
retourne  $Y$ 

```

obtenu et les deux cas suivants peuvent se produire :

- Si $w c^1(O) + (1 - w) c^2(O) = c^1(P_1) + (1 - w) c^2(P_1)$, alors le point O se situe sur le segment $[P_1, P_2]$. Notons que O peut être égal à P_1 ou P_2 . Dans ce cas, cela signifie qu'il n'existe plus de point non dominé dans le triangle formé par P_1 , P_2 et le point idéal local.
- Si $w c^1(O) + (1 - w) c^2(O) < c^1(P_1) + (1 - w) c^2(P_1)$, alors le point O est un nouveau point non dominé supporté extrême. Dans ce cas, il existe peut-être de nouveaux points non dominés supportés extrêmes. L'itération de la première phase est donc répétée pour les couples de points (P_1, O) et (O, P_2) .

À la fin de la première phase, tous les points non dominés supportés sont générés.

Algorithme 4 *calcul_supporte* : Algorithme pour obtenir un point supporté

Entrée : Les points P_1 et P_2 , l'ensemble de points Y

```

 $w = \text{direction\_recherche}(P_1, P_2)$ 
 $LB, \lambda \leftarrow \text{optimum du PML3}(w)$  et solution optimale du PDL3( $w$ )
 $UB \leftarrow \text{solution réalisable du PM3}(w)$ 
 $\bar{R} \leftarrow \text{genroute}(w, LB, UB, \lambda)$ 
 $O \leftarrow \text{solution optimale du PMR3}(w, \bar{R})$ 
si  $O \neq P_1$  and  $O \neq P_2$  alors
     $Y \leftarrow Y \cup O$ 
     $\text{calcul\_supporte}(P_1, O)$ 
     $\text{calcul\_supporte}(P_2, O)$ 
fin si
    
```

Il faut alors explorer l'espace des objectifs pour atteindre les points non dominés non supportés. Ces points se trouvent dans le triangle formé par deux points supportés consécutifs, notés S_i et S_{i+1} , ainsi que le nadir local N_i . La formulation (3.4) est alors utilisée en posant $L_i^1 = c^1(S_{i+1}) - 1$, $L_i^2 = c^2(S_i) - 1$ et w_i la direction de recherche entre S_i et S_{i+1} . Les deux contraintes (3.4.5) et (3.4.6) limitent l'espace des objectifs à ce triangle où des points non dominés non supportés peuvent se trouver. Soit LB_i l'optimum du PML4(w_i, L_i^1, L_i^2) et λ_i une solution optimale du PDL4(w_i, L_i^1, L_i^2). L'appel à la fonction *genroute*($w_i, L_i^1, L_i^2, LB_i, N_i, \lambda_i$) retourne un ensemble de routes \bar{R}_i de coût réduit, calculé à partir des variables duales λ_i , qui est inférieur ou égal à la différence de coût entre LB_i et N_i . Ensuite, la méthode ε -contrainte est appelée sur cet ensemble \bar{R}_i avec ε variant entre $c^2(S_i) - 1$ et $c^2(S_{i+1} + 1)$.

L'Algorithme 5 résume les deux phases décrites précédemment. Il utilise la fonction récursive *calcul_supporte*, présentée dans l'Algorithme 4, pour calculer le point supporté associé à une solution optimale dans la direction de recherche de la droite entre les deux points donnés en argument. Il fait également un appel à la fonction *trier* qui prend en argument un ensemble de points et qui le trie selon la valeur du premier objectif croissant.

La Figure 3.2 permet d'illustrer le fonctionnement de la première phase de la méthode. Les trois premiers schémas représentent cinq itérations de la première phase et le dernier schéma en bas à droite montre les zones de recherche pour la seconde phase. Sur le schéma en haut à gauche, on pose $P_1 = E_1$ et $P_2 = E_2$, avec E_1 et E_2 les points extrêmes. La borne inférieure LB_0 , la borne supérieure UB_0 et la direction de recherche w_0 de la première itération sont indiqués. L'appel à *genroute* permet de générer toutes les routes qui peuvent aboutir à une solution réalisable située sous la droite passant par UB_0 . En résolvant la formulation (3.3) sur ces routes, nous obtenons le point non dominé P_4 . Sur le schéma en haut à droite, les itérations 2 et 3 sont illustrées. L'espace des objectifs est exploré dans la direction w_1 définie par E_1 et P_4 et w_2 définie par P_4 et E_2 . Pour w_2 , le point non dominé trouvé est P_4 , ainsi, il n'y a plus de points supportés situés sous la droite passant par P_4 et E_2 . Pour w_1 , un nouveau point non-dominé est trouvé en $UB_1 = P_2$. D'autres itérations de la phase 1 sont donc effectuées et sont illustrées sur le schéma en bas à gauche. L'espace

Algorithme 5 Algorithme de la méthode en deux phases par étapes

Entrée : Le graphe G représentant le BOVRP

Sortie : L'ensemble Y des points non dominés

```

 $W \leftarrow \emptyset$ 
 $E_1, E_2 \leftarrow \text{calcul\_extreme}()$ 
 $\text{calcul\_supporte}(E_1, E_2, W)$ 
 $\text{trier}(W)$ 
 $Y \leftarrow W$ 
pour  $i \in 1..|W| - 1$  faire
   $N_i = (c^1(W_{i+1}) - 1, c^2(W_i) - 1)$ 
   $w = \text{direction\_recherche}(W_i, W_{i+1})$ 
   $LB, \lambda \leftarrow$  optimum du PML4( $w, c^1(N_i), c^2(N_i)$ ) et solution optimale du
  PDL4( $w, c^1(N_i), c^2(N_i)$ )
   $\bar{R} \leftarrow \text{genroute}(w, c^1(N_i), c^2(N_i), LB, N_i, \lambda)$ 
   $\varepsilon = c^2(N_i)$ 
  tant que  $\varepsilon > c^2(W_{i+1})$  faire
     $O \leftarrow$  solution optimale du PMR2( $\varepsilon, \bar{R}$ )
     $Y \leftarrow Y \cup O$ 
     $\varepsilon = c^2(O) - 1$ 
  fin tant que
fin pour
retourne  $Y$ 

```

des objectifs est exploré dans la direction w_3 définie par E_1 et P_2 et w_4 définie par P_2 et P_4 . Aucun nouveau point non dominé n'est obtenu et la phase 1 est donc terminée. Sur le schéma en bas à droite, les nadirs locaux entre deux points supportés successifs sont représentés et tous ces triangles sont explorés dans la seconde phase.

Dans cette méthode, les points supportés sont obtenus rapidement sans imposer de contraintes supplémentaires à la formulation (3.3). Autant de solutions réalisables que de points supportés sont à calculer. De plus, un appel à *genroute* par nombre de points supportés est nécessaire plus le nombre de points supportés moins un pour visiter les triangles supérieurs entre deux points supportés consécutifs. Le grand désavantage de cette méthode réside dans le nombre de problèmes entiers à résoudre pour prouver que les zones sont toutes entièrement visitées.

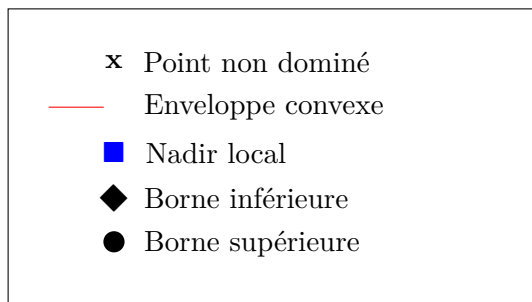
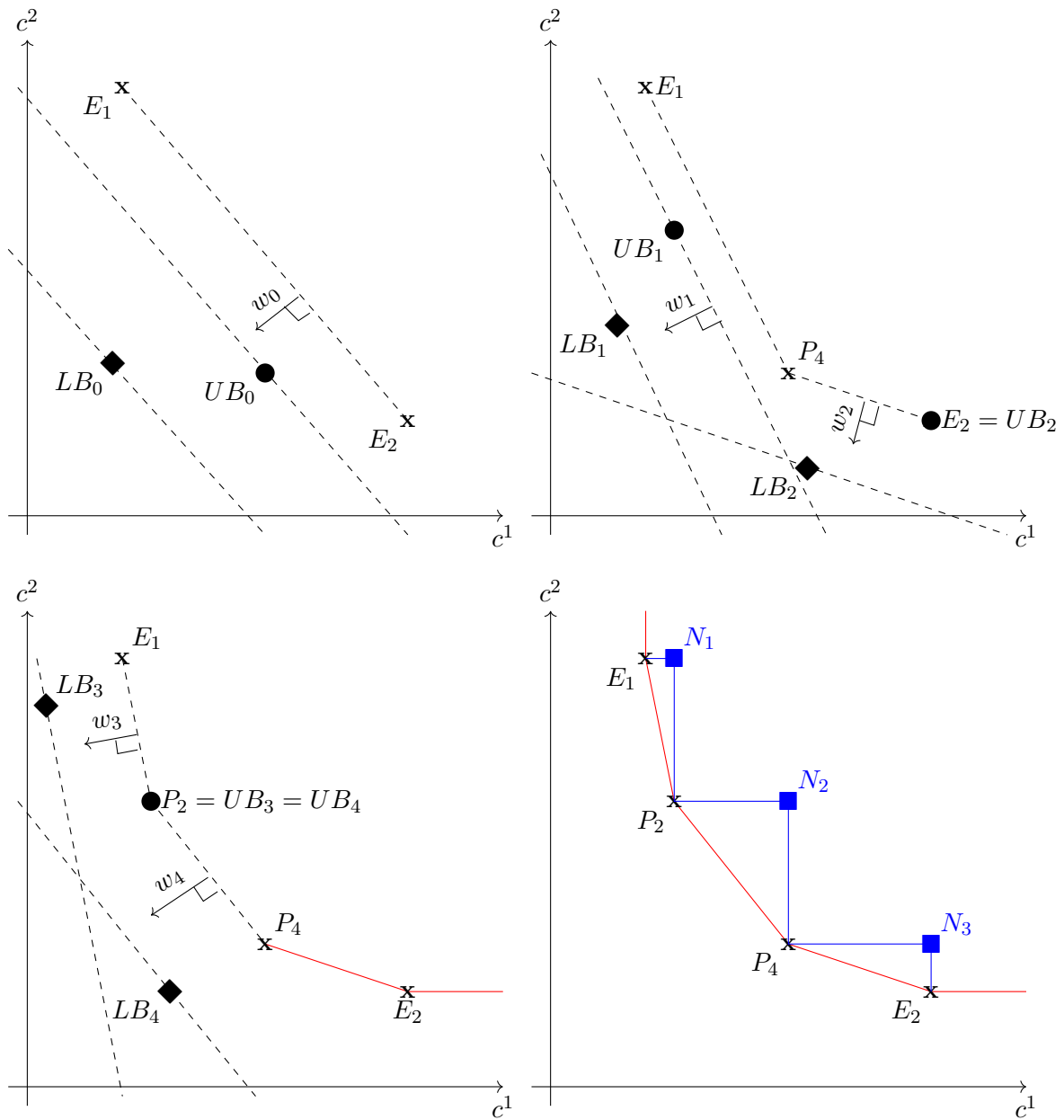


FIGURE 3.2: Figure illustrant les points non dominés trouvés durant la première phase de la méthode en deux phases par étapes.

TABLE 3.1: Caractéristiques des différentes méthodes bi-objectif.

Caractéristiques	M1	M2	M3	M4
Genroute	3	$ ND + 1$	$ ND_2 + 3$	$3 S - 2$
<i>LB</i>	3	$ ND + 1$	$ ND_2 + 3$	$3 S - 1$
<i>UB</i>	2	$ ND + 1$	2	$2 S - 1$
Contrainte c^1	0	0	$ ND_2 $	$ S - 1$
Contrainte c^2	$ ND $	$ ND $	$ ND_2 + ND $	$ ND $
Résolution RMP	$ ND + 1$	$ ND + 1$	$ ND_2 + ND $	$ ND + 2 S - 1$

3.4.6 Conclusion

Les quatre méthodes présentées sont nommées comme suit : M1 pour la méthode ε -contrainte global (Section (3.4.2)), M2 pour la méthode ε -contrainte par étapes (Section (3.4.3)), M3 pour la méthode en deux phases global (Section (3.4.4)) et M4 pour la méthode en deux phases par étapes (Section (3.4.5)). Elles explorent l'espace des objectifs selon différentes manières. Le tableau 3.1 récapitule les caractéristiques de chaque méthode avec *genroute* le nombre d'appel à la fonction *genroute*, *LB* le nombre de résolution de la relaxation linéaire d'une formulation, *UB* le nombre de bornes supérieures à générer, *contrainte c^1* et *contrainte c^2* le nombre de contraintes sur c^1 et sur c^2 qui sont rajoutées et *résolution RMP* le nombre de problèmes entier restreint résolu pour obtenir un point non dominé. Soit $|ND|$ le nombre final de points non dominés, $|S|$ le nombre de points supportés et $|ND_2|$ le nombre de fois où le nadir entre deux points non dominés sous la droite reliant les points extrêmes et étant successifs, est situé au-dessus de cette même droite.

Nous pouvons donc remarquer que la méthode 2 est celle qui requiert le plus de calculs de borne inférieure et d'appels à la fonction *genroute*. De plus, cette méthode est la plus dépendante de la qualité de l'heuristique pour calculer les bornes supérieures. Selon les formes des fronts de Pareto des instances étudiées, les méthodes 3 et 4 contraignent le plus l'espace des objectifs et résolvent plus de problèmes maîtres restreints.

3.5 Résultats expérimentaux

En premier lieu, les instances sont décrites en partie 3.5.1 et les résultats expérimentaux des méthodes de la Section 3.4 sont présentés en Section 3.5.2.

Toutes les expériences sont conduites sur un CPU Xeon E5-2695 avec 2.30GHz de fréquence et 4Go de mémoire allouée et sur un unique cœur. L'implémentation est faite en langage C++ et tous les problèmes linéaires et entiers sont résolus avec la version 7.1 de Gurobi. Le temps limite de résolution d'une instance est quatre heures. Lors de la présentation de tous les résultats, si une méthode n'a pas convergé durant le temps limite pour une instance, celle-ci sera présentée avec un tiret (-).

3.5.1 Description des instances

Nous présentons de nouvelles instances pour le problème de tournées de véhicules bi-objectif avec fenêtres de temps (*Bi-Objective Vehicle Routing Problem with Time Windows* = BOVRPTW). Dans ces problèmes, chaque client $i \in V'$ a une fenêtre de temps $[b_i; e_i]$ durant laquelle le véhicule doit visiter le client et un temps de service s_i . Chaque instance est le résultat de la concaténation de deux instances de Solomon¹. La première instance de Solomon nommée A fournit le coût c^1 sur les arcs qui sera également le temps de parcours des véhicules, les fenêtres de temps, les temps de service et les demandes de chaque client. La seconde instance nommée B ne fournit que les seconds coûts c^2 sur les arcs. Chaque nouvelle instance est marquée A_B_X , avec X le nombre de nœuds dans le graphe.

Lors du lancement de chaque méthode, la taille des ng-ensembles pré-définis pour la relaxation des ng-routes (Section 2.3.2.2) est telle que $|N_i| = 8$, $i \in V'$. Le tableau 3.2 introduit les caractéristiques de chaque type d'instances pour le BOVRPTW avec 25 nœuds dans le graphe. La colonne *Type* décrit le type d'instances de Solomon, $|ND|$ est le nombre moyen de points non dominés dans \mathcal{Y}_N et $|S|$ est le nombre moyen de points non dominés supportés. La colonne *MGap* représente le gap moyen entre la borne inférieure et la solution optimale à chaque étape de la méthode ε -contrainte par étapes en pourcentage, pour chaque valeur ε . La formule du gap G pour une valeur ε fixée est :

$$G = 100 - \frac{c^1(LB)*100}{c^1(O)} (\%);$$

Enfin, la colonne *Fermé* est le nombre d'instances fermées pour chaque type d'instances par une des quatre méthodes au moins et la colonne *Total* est le nombre total d'instances de chaque type.

Nous pouvons observer que les instances utilisant de grandes fenêtres de temps (R2, RC2, C2) sont plus difficiles à résoudre que les instances avec de petites fenêtres de temps (R1, RC1, C1); 7 instances fermées sur 81 contre 82 instances fermées sur 87, respectivement. Ceci s'explique car la génération de colonnes est particulièrement efficace sur les instances avec de petites fenêtres de temps, même sur des problèmes de tournées de véhicules mono-objectif. Le nombre de points non dominés varie entre 11 et 65 et est égal à 34 en moyenne. Parmi ces points, 10 points en moyenne sont supportés; soit 29% de \mathcal{Y}_N . Les problèmes considérés ont donc des objectifs bien contradictoires et intéressants d'un point de vue bi-objectif puisque le nombre de points non supportés, qui sont les plus difficiles à obtenir, est important. Nous pouvons également noter que le nombre de points non dominés est plus grand, en moyenne, dans les instances de type R que celles de type C. Au contraire, la moyenne des gaps moyens est plus grande pour les instances de type C et plus faible pour celles de type R. Les instances combinant C1 et C2 sont celles avec les plus petits ensembles de points non dominés et ceci s'explique par la construction des instances. Par exemple, en mixant les instances de Solomon C1 et C2, 16 points du graphe sur 25 ont les mêmes coordonnées. Ainsi, les différences de coûts entre c^1

1. Instances de Solomon disponibles sur <https://www.sintef.no/nearp>

TABLE 3.2: Caractéristiques des instances du BOVRP pour $|V| = 25$.

Type	$ ND $	$ S $	MGap	Fermé	Total
R1_C1_25	37	11	3.0	12	12
R1_C2_25	45	12	3.6	12	12
R1_RC_25	53	12	2.9	12	12
R2_C1_25	-	-	-	0	11
R2_C2_25	63	12	2.0	1	11
R2_RC_25	48	16	2.4	1	11
RC1_C1_25	18	7	3.9	8	8
RC1_C2_25	22	9	4.4	8	8
RC1_R_25	38	11	5.5	8	8
RC2_C1_25	37	7	3.5	1	8
RC2_C2_25	41	11	4.2	2	8
RC2_R_25	65	15	4.1	1	8
C1_C2_25	11	4	6.3	8	9
C1_R_25	35	10	6.7	6	9
C1_RC_25	20	6	6.4	8	9
C2_C1_25	-	-	-	0	8
C2_R_25	-	-	-	0	8
C2_RC_25	14	6	5.2	1	8
Moyenne	34	10	4.4		
Total				89	168

et c^2 se forment grâce aux 9 clients restants.

Le tableau 3.3 introduit les caractéristiques des 16 instances sur 168 qui ont convergé pour des graphes de 50 nœuds.

Le nombre de points non dominés des instances à 50 nœuds est environ quatre fois plus important que pour les instances à 25 nœuds, sauf pour celles de type $C1_C2$. Le nombre de points non supportés est toujours élevé et témoigne de la difficulté des instances résolues. Cependant, les gaps moyens des instances résolues est faible en comparaison avec les instances de 25 nœuds. Ceci s'explique par le fait que les instances avec des gaps moyens plus élevés n'ont pas pu être résolues dans le temps imparti.

3.5.2 Comparaison des méthodes

Le tableau 3.4 présente le nombre d'instances fermées et les temps CPU moyens en secondes des quatre méthodes pour les instances qui ont été résolues en quatre heures. Le détail des résultats pour chaque instance est donné dans l'Annexe A.1.

En terme d'instances fermées, la méthode 2 est la meilleure car elle résout 88 instances sur les 168 avec 25 nœuds, contre 76 pour la méthode 4, 71 pour la méthode 3 et 50 pour la méthode 1. Cependant, les instances $R108_C2_25$ et $R108_RC1_25$ sont uniquement résolues par la méthode 4. Au contraire, 16 instances sont uniquement résolues par la

TABLE 3.3: Caractéristiques des instances du BOVRP pour $|V| = 50$.

Type	$ ND $	$ S $	MGap
R101_C1_50	119	26	0.7
R101_C2_50	113	20	1.1
R102_C1_50	189	31	1.7
R102_C2_50	155	28	0.8
R103_C1_50	177	26	1.5
R103_C2_50	192	26	1.8
R105_C1_50	154	20	2.0
R105_C2_50	169	24	1.4
R106_C1_50	212	28	1.4
R109_C1_50	177	23	1.6
R109_C2_50	162	20	1.8
C101_C2_50	7	4	3.1
C101_R1_50	157	17	1.9
C102_C2_50	24	8	3.2
C107_C2_50	15	6	2.6
C108_C2_50	42	10	1.9
Moyenne	129	20	1.8

TABLE 3.4: Nombre d'instances fermées et moyenne des temps de résolution des méthodes selon la classe d'instances $R_$, $RC_$ et $C_$.

Classe d'instance	$ V $	M1		M2		M3		M4	
		Fermées	Temps	Fermées	Temps	Fermées	Temps	Fermées	Temps
$R_$	25	22	3991.7	36	102.5	27	2755.9	36	262.8
$RC_$	25	20	1681.1	28	44.8	25	909.1	24	155.0
$C_$	25	8	5693.6	24	31.9	19	634.4	16	254.8
$R_$, $C_$ et $RC_$	50	4	2246.1	16	586.2	6	1775.8	7	1561.9

méthode 2. En moyenne, nous pouvons établir que la méthode la plus rapide en terme de temps de résolution est la méthode 2, suivie de la méthode 4, puis de la méthode 3 et enfin, la méthode la moins performante est la méthode 1. Cependant, ces résultats sont variables selon les instances. De plus, les méthodes 1 et 3, qui travaillent sur la génération de beaucoup de colonnes avec un appel à *genroute*, ont tendance à avoir des résultats similaires, i.e. des performances acceptables sur des instances faciles où peu de colonnes sont générées au total et des performances médiocres sur les instances plus difficiles.

La méthode 1 est la plus rapide sur quatre instances dont $R101_C1_25$. Au contraire, elle est surpassée par les trois autres méthodes sur 31 instances comme $R111_C1_25$. La méthode 2 est la plus rapide en moyenne et est très efficace sur la plupart des instances comme $RC201_C2_25$ et $C108_C2_25$, i.e. le meilleur temps sur 70 instances. Cependant, la méthode 2 est la plus lente sur 6 instances faciles comme $R101_C1_25$, où le temps maximum de résolution est de 265 secondes. Quelque fois, la méthode 2 est surpassée par la méthode 3 ($RC201_C1_25$) ou par la méthode 4 ($R107_RC1_25$). La méthode 3 est la méthode la plus rapide sur 11 instances et la plus lente sur 10 instances.

TABLE 3.5: Résultats expérimentaux détaillés des méthodes 1 et 3.

Instance	Méthode 1				Méthode 3								
	LB	Col	MP	Temps	Phase 1				Phase 2				Temps
					LB	Col	MP	Temps	Triangle	Col	MCol	Temps	
R101_C1_25	2	734	28	2.1	3	580	24	1.9	0	0	-	0.0	2.4
R102_C1_25	2	25926	62	729.6	3	25157	64	1573.8	0	0	-	0.0	1575.7
R107_RC1_25	-	-	-	-	-	-	-	-	-	-	-	-	-
R111_C1_25	2	131929	26	13808.6	6	71100	26	4857.1	3	1680	560	3.7	4974.9
RC108_R1_25	2	198937	32	19983.5	5	174726	28	10697.5	2	220	110	1.5	10701.0
RC201_C1_25	-	-	-	-	22	10463	37	70.2	19	159475	8393	331.4	404.6
RC201_C2_25	-	-	-	-	-	-	-	-	-	-	-	-	-
C108_C2_25	-	-	-	-	9	26937	16	119.5	6	22537	3756	180.7	303.9

Par exemple, les instances *RC201_C1_25* et *R102_C1_25* montrent la différence de performance de la méthode. Enfin, la méthode 4 est la plus rapide sur 13 instances contre 3 où elle est la moins performante. L'instance *C108_C2_25* illustre le fait que les méthodes 2 et 3 peuvent surpasser la méthode 4.

Toutes les instances que nous venons d'évoquer vont être analysées grâce aux tableaux 3.5 et 3.6 qui détaillent le comportement des méthodes 1 et 3 et des méthodes 2 et 4 pour ces instances. Les en-têtes suivants sont utilisés pour les colonnes des tableaux :

- Phase 1 / Phase 2 : pour les méthodes 3 et 4, désigne si les données se réfèrent à la phase 1 ou la phase 2 uniquement ;
- |LB| : nombre de bornes inférieures calculées ;
- Col : nombre de colonnes totales générées lors de l'énumération des colonnes avec *genroute* ;
- MCol : nombre moyen de colonnes retournées avec un appel à la fonction *genroute* ;
- |MP| : nombre de problèmes entiers restreints résolus pour obtenir un point non dominé ;
- Temps : temps CPU en secondes ;
- Triangle : pour la méthode 3, désigne le nombre de fois où le nadir local entre deux point non dominés successifs trouvés en phase 1 est situé sur la droite des extrêmes. Pour la méthode 4, nombre de fois où il y a eu exploration entre deux points supportés consécutifs.

Pour chaque instance, les observations suivantes peuvent s'appliquer :

- *R101_C1_25* : la méthode la plus rapide est la méthode 1, suivie par la méthode 3 (respectivement résolue en 2.1 et 2.4 secondes). Cela s'explique par le fait que le nombre de colonnes totales est très faible (moins de 1000 colonnes). Ainsi, résoudre les problèmes maîtres restreints sur toutes ces colonnes n'est pas coûteux en temps et ces deux méthodes évitent les multiples appels à la fonction *genroute*.
- *R102_C1_25* : cette instance est rapidement résolue par les méthodes 2 et 4 (moins de 260 secondes), mais prend six fois plus de temps pour la méthode 3. Ceci s'explique

TABLE 3.6: Résultats expérimentaux détaillés des méthodes 2 et 4.

Instance	Méthode 2					Méthode 4							
	LB	MP	Col	MCol	Temps	Phase 1				Phase 2			
						LB	MCol	MP	Temps	Triangle	MCol	Temps	Temps
R101_C1_25	24	22	2066	94	8.8	26	0	8	11.4	8	164	3.6	7.0
R102_C1_25	55	55	113609	2066	252.4	45	1074	15	102.6	14	1889	40.6	66.1
R107_RC1_25	51	50	248812	4976	9619.0	40	2641	12	254.0	13	6473	469.5	540.1
R111_C1_25	24	24	95126	3964	448.9	17	7954	5	357.5	5	17432	2189.3	2344.4
RC201_C1_25	38	34	415105	12209	1491.3	-	-	-	-	-	-	-	-
RC201_C2_25	27	25	143532	5741	252.9	-	-	-	-	-	-	-	-
C108_C2_25	17	15	71108	4741	162.1	7	0	1	10.8	2	114093	12100.4	12112.1

par le fait que les problèmes maîtres restreints sont tous résolus sur un ensemble de plus de 25000 routes, contre 2000 en moyenne pour les méthodes 2 et 4.

- *R107_RC1_25* : cette instance peut être considérée comme difficile puisque les méthodes 1 et 3 générant de grands ensembles de colonnes ne convergent pas dans le temps imparti. Ici, la méthode 2 est surpassée par la méthode 4. Ceci s’explique car la méthode 2 génère au total deux fois plus de colonnes que la méthode 4. Même si le nombre moyen des ensembles de colonnes sont plus grands lors de la phase 2 de la méthode 4, seulement 13 appels à *genroute* ont été nécessaires pour obtenir les points non supportés, contre 36 pour la méthode 2.
- *R111_C1_25* : la méthode 1 est la plus lente car elle génère près de deux fois plus de colonnes que la méthode 3. Lors de la résolution de cette instance avec la méthode 1, l’unique appel à *genroute* n’est pas suffisant pour diminuer le temps de calcul. En effet, les problèmes maîtres sont résolus 26 fois sur plus de 130000 colonnes contre 24 fois sur seulement 4000 colonnes en moyenne pour la méthode 2.
- *RC201_C1_25* : les méthodes 1 et 4 ne convergent pas dans le temps imparti pour cette instance. De plus, la méthode 3 surpasse la méthode 2. Ceci s’explique car la taille des ensembles de routes générés lors de la méthode 3 est plus petite que celle des ensembles de la méthode 2. Cette instance est particulière puisque seulement 10000 colonnes sont générées lors de la phase 1 de la méthode 3 et couvrent pourtant 36 points non dominés.
- *RC201_C2_25* : la méthode 2 semble indiquée pour cette instance puisque c’est l’unique méthode pouvant la résoudre. Le nombre moyen de colonnes générées est raisonnable (environ 6000) pour la méthode 2. Le fait que les autres méthodes ne convergent pas s’explique par le nombre trop important de colonnes générées pour la méthode 1 et 3 lors du premier appel à *genroute* et lors de l’exploration d’un triangle formé par deux points non dominés consécutifs et le nadir local pour la méthode 4.
- *C108_C2_25* : les méthodes 2 et 3 sont plus efficaces que la méthode 4 sur cette instance. Ceci s’explique car l’exploration entre les points supportés successifs génèrent plus de 100000 colonnes à chaque fois et la résolution des problèmes maîtres restreints est donc très coûteuse.

3.5.3 Conclusion

Les résultats expérimentaux mettent en avant l'efficacité de la méthode ε -contrainte par étapes, en comparaison avec les autres explorations de l'espace des objectifs. L'efficacité de cette méthode repose sur le fait que la résolution des problèmes maîtres restreints sur un grand nombre de colonnes (plus de 30000) ralentit considérablement la résolution d'une instance. De ce fait, les méthodes ε -contrainte global et des deux phases globales sont moins efficaces que les méthodes proposant des gaps plus réduits. Lors de la méthode en deux phases par étapes, l'exploration des triangles supérieurs entre deux points supportés consécutifs peut s'avérer critique dans certaines instances et freine la résolution, même si l'obtention de points supportés avec la formulation 3.3 est rapide.

Le type d'instances joue également un rôle important pour déterminer l'efficacité d'une méthode. Ainsi, le nombre de points non dominés, la forme du front de Pareto, le nombre de colonnes pouvant être générées selon les zones d'exploration... sont autant d'informations non accessibles a priori mais qui font varier les performances d'une méthode.

3.6 Conclusion du chapitre

Dans ce chapitre, la méthode exacte de [Baldacci et al. \(2008\)](#) est adaptée aux tournées de véhicules bi-objectif pour pouvoir générer l'ensemble complet minimal des points non dominés. Quatre pistes d'exploration de l'espace des objectifs ont été abordées. Des nouvelles instances pour le BOVRP ont été créées par combinaison d'instances existantes pour les tournées de véhicules mono-objectif. Cela a permis d'obtenir de nombreux résultats expérimentaux et de comparer les performances des différentes méthodes. La méthode exacte bi-objectif la plus efficace est une méthode qui combine la génération de colonnes et la méthode ε -contrainte pour obtenir des zones intermédiaires d'exploration les plus petites possibles. Il s'agit de la méthode ε -contrainte par étapes. La suite de cette thèse va se focaliser sur l'amélioration de cette méthode et sur l'application de la méthode sur d'autres problèmes de tournées de véhicules.

Étude de la méthode ε -contrainte

Sommaire

4.1	Introduction	69
4.2	Analyse de la méthode ε-contrainte par étapes	70
4.2.1	Décomposition de la méthode	70
4.2.2	Analyse de l'ensemble des points non dominés	72
4.2.3	Ajout d'inégalités valides	74
4.2.4	Conclusion	77
4.3	Améliorations de la méthode ε-contrainte par étapes	77
4.3.1	Réduction du gap	78
4.3.2	Construction de la borne inférieure	79
4.3.3	Validation des points non dominés	81
4.3.4	Gestion et sauvegarde de colonnes	84
4.3.5	Multiplés coûts duaux	86
4.3.6	Exploration bidirectionnelle	88
4.3.7	Conclusion	89
4.4	Nouvelle méthode ε-contrainte	89
4.4.1	Description	90
4.4.2	Résultats expérimentaux	92
4.4.3	Conclusion	93
4.5	Conclusion du chapitre	94

4.1 Introduction

D'après le chapitre précédent, la méthode la plus efficace pour résoudre les instances de BOVRP est la méthode ε -contrainte par étapes, présentée en Section 3.4.3. Cela confirme l'utilité de l'approche ε -contrainte qui est la méthode multi-objectif la plus utilisée pour la résolution exacte de tournées de véhicules multi-objectif (*cf.* Section 2.5). Dorénavant, la méthode ε -contrainte par étapes ou *Stepwise ε -constraint Method*, notée SEM, est la méthode de référence.

Dans ce chapitre, les résultats obtenus par SEM pour les instances de BOVRPTW sont analysés en Section 4.2 pour pouvoir proposer des améliorations en Section 4.3. Ainsi, une nouvelle version de SEM est proposée en Section 4.4 et est appelée ISEM pour *Improved*

Stepwise ε -constraint Method. Son efficacité est montrée grâce à d'intensives expérimentations en comparaison avec SEM. Enfin, une conclusion sur la méthode et les perspectives d'améliorations sont présentées en Section 4.5.

Les notations du chapitre sont les suivantes. La formulation (3.2) avec le paramètre ε est le problème maître (PM(ε)) et sa relaxation linéaire le problème maître linéaire (PML(ε)). Le problème maître et sa relaxation linéaire restreints sur un ensemble de colonnes \bar{R} seront notés PMR(ε, \bar{R}) et PMLR(ε, \bar{R}). Le dual du problème maître linéaire est noté PDL(ε) et utilise les variables duales suivantes : $\lambda_i^\varepsilon \geq 0$ ($i \in V'$), $\lambda_0^\varepsilon \leq 0$ et $\alpha^\varepsilon \leq 0$ respectivement associées aux contraintes (3.2.2), (3.2.3) et (3.2.4). De plus, le coût réduit d'une solution UB de PM(ε), calculé à partir des variables duales $\lambda^\varepsilon = (\lambda_i^\varepsilon, \lambda_0^\varepsilon, \alpha^\varepsilon)$, est noté $\sum_{r \in R} \bar{c}_r(\lambda^\varepsilon) x_r^{UB}$.

Contributions

Les contributions de ce chapitre sont de proposer des théorèmes sur la formulation ε -contrainte appliquée au BOVRP et d'en déduire de nouvelles règles d'exploration de l'espace des objectifs. Les techniques des Sections 4.3.2, 4.3.3, 4.3.4 et 4.3.5 n'ont jamais été évoquées dans la littérature. Une nouvelle méthode améliorant l'efficacité de l'approche ε -contrainte classique en est déduite.

Publications

Toutes les techniques d'amélioration de la Section 4.3 et la méthode décrite en Section 4.4 sont présentées dans un article soumis à *Computers & Operations Research* (C&OR) et intitulé '*An ε -constraint column generation-and-enumeration algorithm for bi-objective vehicle routing problems*'. Une partie de ce travail a également été exposé à la conférence internationale sans actes VEROLOG 2019.

4.2 Analyse de la méthode ε -contrainte par étapes

Cette section présente les avantages et les inconvénients de la méthode ε -contrainte par étapes (SEM), décrite en Section 3.4.3, lors de la résolution sur les instances de BOVRPTW. Tout d'abord, les temps de résolution dans chaque étape de la méthode sont analysés. Nous verrons d'une part quelles fonctions sont les plus coûteuses en terme de temps et d'autre part à quel niveau de résolution de l'ensemble des points non dominés la méthode est la plus lente. Ensuite, les formes des ensembles de points non dominés et des bornes inférieures obtenues sont décrites. Dans la dernière partie, des coupes classiques de résolution des tournées de véhicules mono-objectif sont ajoutées pour tenter de renforcer la borne inférieure.

4.2.1 Décomposition de la méthode

SEM est une méthode combinant la génération de colonnes et la méthode ε -contrainte qui s'est révélée prometteuse dans les résultats expérimentaux présentés en Section 3.5. Ces résultats ont mis en évidence que les méthodes d'exploration de l'espace des objectifs se focalisant sur des petites zones étaient plus efficaces. Cependant, ces avantages et ces

TABLE 4.1: Détails de la méthode ε -contrainte par type d'instances.

	Instances faciles			Instances difficiles		
	R_	RC_	C_	R_	RC_	C_
Temps LB (%)	16	30	12	2	9	1
Temps <i>genroute</i> (%)	37	41	27	72	76	34
Temps autre (%)	47	29	61	26	15	65
Temps total (s)	189	86	293	4278	1279	7512
Pire Col	7719	6680	31837	101173	64103	144298

inconvenients n'ont pas été détaillés. Par exemple, nous ne savons pas si les fonctions intermédiaires prenant le plus de temps sont pour la génération des bornes inférieures, pour la résolution des problèmes maîtres réduits ou pour l'énumération des routes dans le gap (*genroute*), ou si leurs temps d'exécution sont équilibrés.

Lors de cette analyse, les 88 instances fermées sont scindées en deux groupes : les faciles (75 instances) et les difficiles (13 instances), résolues respectivement en moins et en plus de 1000 secondes avec SEM. Le tableau 4.1 donne les pourcentages des temps de chaque fonction intermédiaire, selon les types d'instances (instances de Solomon¹ de type *R*, *RC* ou *C*) et selon la difficulté des instances. Nous observons le temps d'obtention des bornes inférieures (*Temps LB*), le temps de l'énumération des colonnes dans le gap (*Temps genroute*) et le temps restant qui consiste en la mise en place des modèles avec Gurobi, la résolution des problèmes maîtres restreints et divers traitements de données (*Temps autre*). L'opération la plus coûteuse dans ce traitement est la résolution des problèmes maîtres restreints à l'optimalité. La ligne *Temps total* représente la moyenne des temps de résolution en secondes et *Pire Col* est la moyenne du plus grand ensemble de routes générées par *genroute* en un seul appel.

Le nombre de calculs de bornes inférieures est élevé puisque la méthode calcule une borne par point non dominé. Cependant, le temps de résolution de toutes ces bornes est relativement faible. Plus le problème est difficile et plus l'impact du calcul des bornes inférieures dans le temps total est minimal. Selon les types d'instances, les opérations les plus coûteuses fluctuent entre *genroute* (*R_* et *RC_*) et la résolution des PMR à l'optimalité (*C_*). Cependant, nous pouvons voir que les temps de résolution des instances sont directement liés aux plus grands nombres de routes générées en un seul appel, comme cela est visible entre les instances faciles et difficiles, notamment pour les instances de type *C_*. Le lien entre temps de résolution et nombre maximal de routes générées s'explique car plus le nombre de routes à énumérer est grand, plus la fonction *genroute* prend du temps et plus la résolution du PMR résultant est longue.

Pour le groupe des instances difficiles, certaines instances représentatives sont sélectionnées pour être analysées en détail : les instances *R107_C1_25*, *RC201_C1_25* et

1. Instances de Solomon disponibles sur <https://www.sintef.no/nearp>

C102_R1_25 et *R103_C1_50*. Pour chacune de ces instances, la Figure 4.1 donne les temps des fonctions intermédiaires appelées en secondes. En première ligne, un point vert représente un point non dominé trouvé. En seconde ligne, *Genroute* représente les appels à la fonction *genroute*. En troisième ligne, *LB* représente le temps passé à résoudre des bornes inférieures et la dernière ligne représente le temps pour générer les points c^1 et c^2 -extrêmes.

En premier lieu, nous pouvons observer que le temps de génération des deux points extrêmes est toujours négligeable. Pour *R107_C1_25*, le début des points non dominés (jusqu'à 439s) est généré facilement avec de multiples appels à *genroute* et *LB*. À partir de 439 secondes, seulement deux appels à *genroute* sont faits et durent plus de 400 secondes à chacun. Cette instance est donc ralentie avec ces deux appels. Pour *RC201_C1_25*, deux appels à *genroute* sont longs (plus de 400s) mais, contrairement à *R107_C1_25*, ils se situent au milieu de la génération des points non dominés. La génération des bornes inférieures est rapide. Contrairement aux deux premières instances, *C102_R1_25* effectue des appels à *genroute* qui ont presque la même durée au cours de l'exécution. Le temps de résolution est ralenti lors de la résolution des problèmes restreints à l'optimalité sur les routes générées par *genroute*. Ceci se remarque puisque c'est le laps de temps entre la fin d'une fonction *genroute* et l'obtention d'un point non dominé. Enfin, l'instance *R103_C1_50* est difficile notamment par le nombre de points non dominés qui est clairement plus élevé que les autres instances. Les temps des fonctions intermédiaires dans cette instance sont équilibrés.

4.2.2 Analyse de l'ensemble des points non dominés

La connaissance des caractéristiques des points non dominés des instances est importante pour comprendre SEM. Le tableau 4.2 donne les informations des ensembles des points non dominés \mathcal{Y}_N et de SEM selon le second objectif c^2 . Ainsi, chaque \mathcal{Y}_N est divisé en quatre parties de coût c^2 égales entre les deux points extrêmes. Pour chaque partie, le tableau 4.2 donne le pourcentage de temps de résolution *Temps*, le nombre de points non dominés $|ND|$ et le nombre de points supportés $|S|$. La ligne *MGap* représente le gap moyen entre la borne inférieure et la solution optimale à chaque étape de SEM en pourcentage, pour chaque valeur ε . La formule du gap G pour une valeur ε fixée est :

$$G = 100 - \frac{c^1(LB)*100}{c^1(O)} (\%);$$

Plus la valeur de ε décroît et donc que la méthode passe du premier au quatrième quart, plus les temps de résolution sont longs. Cela est lié d'une part au fait que le plus grand nombre de points non dominés (supportés ou non supportés) se situe dans ce dernier quart. D'autre part, cette expansion du temps peut également être expliquée par l'accroissement des gaps entre les bornes inférieures et les coûts des solutions optimales.

Pour illustrer ces observations, les Figures 4.2 et 4.3 représentent \mathcal{Y}_N (en vert), les bornes inférieures (en bleu) et supérieures (en rouge) pour les mêmes instances que sur la Figure 4.1 : *R107_C1_25*, *RC201_C1_25* et *C102_R1_25* et *R103_C1_50*. À partir de ces figures, nous observons que les derniers points non dominés ont de mauvaises bornes inférieures associées. Au milieu des points non dominés, nous voyons que beaucoup de

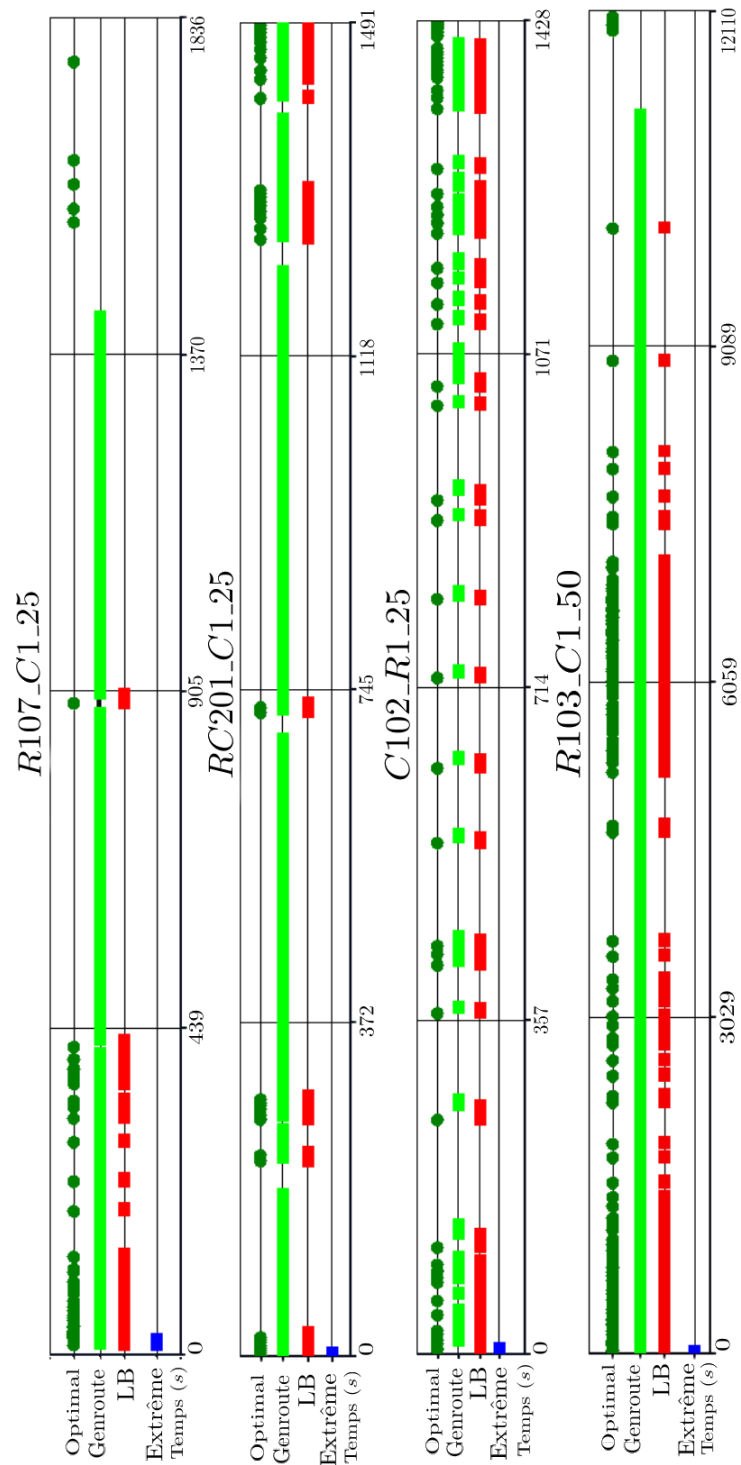


FIGURE 4.1: Détail des temps d'exécution des fonctions appelées pour résoudre les instances *R107_C1_25*, *RC201_C1_25*, *C102_R1_25* et *R103_C1_50*.

TABLE 4.2: Caractéristiques des ensembles de points non dominés selon le coût c^2 pour $V = 25$.

Caractéristiques	Type	Premier quart	Second quart	Troisième quart	Quatrième quart
Temps (%)	R_	2	5	6	87
	RC_	7	9	50	34
	C_	3	11	12	74
MGap	R_	1.3	2.3	2.5	3.9
	RC_	1.5	2.7	4.9	6.2
	C_	3.1	5.1	7.2	6.9
ND	R_	6	8	10	21
	RC_	4	5	7	12
	C_	4	3	5	8
S	R_	2	1	2	6
	RC_	2	1	2	4
	C_	2	1	0	3

points non supportés sont éloignés de l’enveloppe convexe de \mathcal{Y}_N . Nous pouvons également observer que la dernière borne supérieure a souvent un coût c^1 plus élevé que le point c^2 -extrême E_2 . Dans ce cas-là, la méthode de référence prend le point E_2 en tant que borne supérieure jusqu’à la fin de la résolution. Comme $c^1(E_2)$ est assez grand comparé au coût c^1 de la borne inférieure finale, cela explique les valeurs du gap moyen du dernier quart de \mathcal{Y}_N et le temps passé pour l’énumération des colonnes. Cette mauvaise qualité de la borne supérieure peut perturber les conclusions sur les améliorations de la méthode. Ainsi, dans la suite de ce chapitre, les bornes supérieures pour toutes les méthodes sont les solutions optimales, si connues, et sont données dans un fichier en entrée de l’algorithme. Si on ne connaît pas la solution optimale et que la borne supérieure donnée en entrée est moins bonne que celle de l’heuristique préalablement utilisée (Section 3.3.2), cette dernière est utilisée.

4.2.3 Ajout d’inégalités valides

Pour rapprocher la borne inférieure de l’enveloppe convexe de \mathcal{Y}_N et générer moins de routes lors d’un appel à la fonction *genroute*, une piste a été étudiée : l’ajout de coupes classiques mono-objectif des tournées de véhicules à chaque étape de SEM.

Les inégalités valides que nous considérons s’appellent les inégalités *subset-row*, notées SR-inégalités et sont des coupes Chvatal-Gomory de rang 1 (Chvatal (1973)) ajoutées à la formulation (3.2). Ces coupes ont été introduites par Jepsen et al. (2008) dans le cadre du VRP. Elles permettent d’éviter l’utilisation de trop de véhicules lors du service d’un sous-ensemble de clients. Elles sont définies comme suit :

$$\sum_{r \in R} \lfloor \frac{1}{k} \sum_{i \in S} a_{ir} \rfloor x_r \leq \lfloor \frac{|S|}{k} \rfloor, \text{ pour } S \subseteq V' \text{ et } 1 \leq k \leq |S|.$$

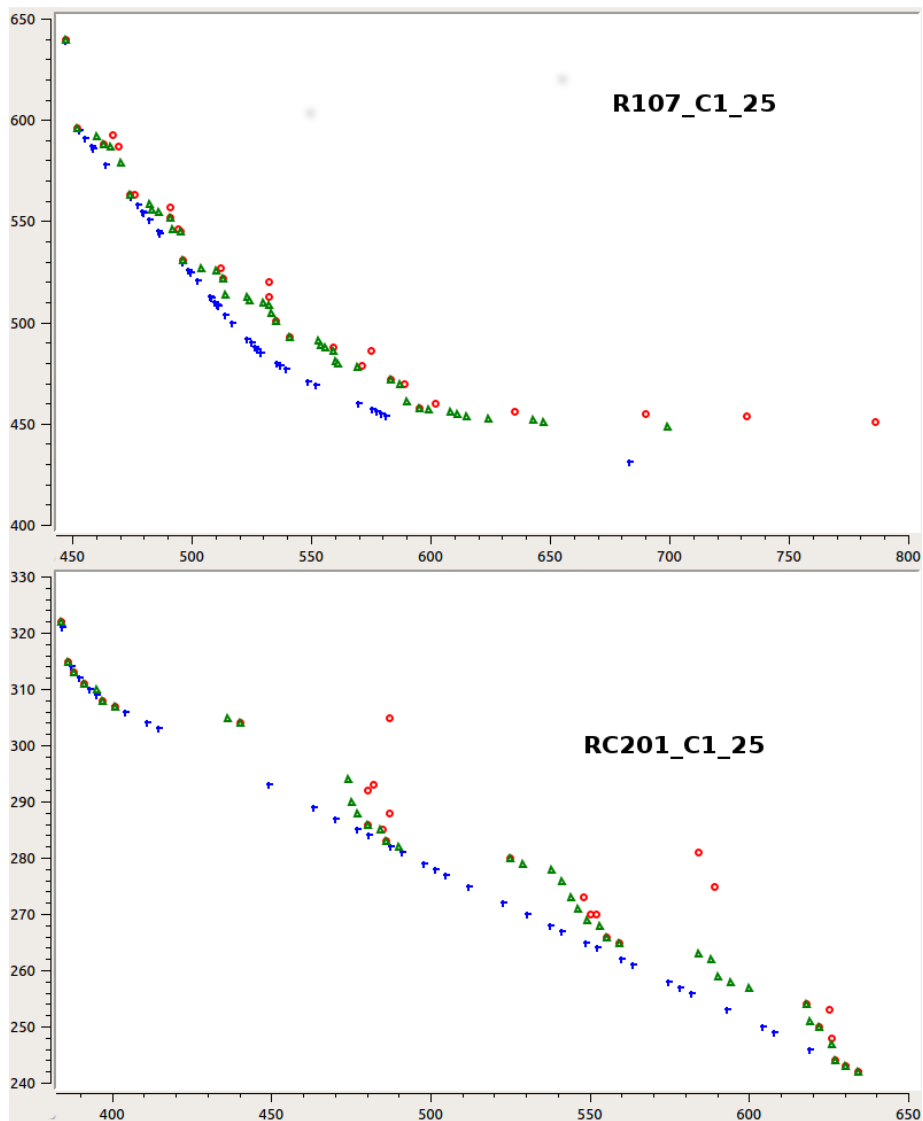


FIGURE 4.2: Ensemble des points non dominés des instances $R107_C1_25$ et $RC201_C1_25$ - en triangle vert, les points non dominés - en croix bleue, les points de la borne inférieure - en cercle rouge, les bornes supérieures.

Le problème de séparation des SR-inegalités est NP-complet (Jepsen et al. (2008)). Il est donc coûteux de trouver de telles inegalités et de les ajouter dans la formulation. De plus, l'ajout de telles inegalités augmente la difficulté de résolution du sous-problème puisqu'une ressource par inegalité est ajoutée lors de la génération de colonnes. De ce fait, la règle de dominance est également affaiblie. Cependant, la performance de ces coupes a été démontrée par Jepsen et al. (2008) lors de leur ajout dans une méthode de branchement et de génération de colonnes (*Branch-and-Price*) pour résoudre les instances VRPTW de Solomon. La plupart de ces instances ont été fermées dans le nœud racine de la méthode

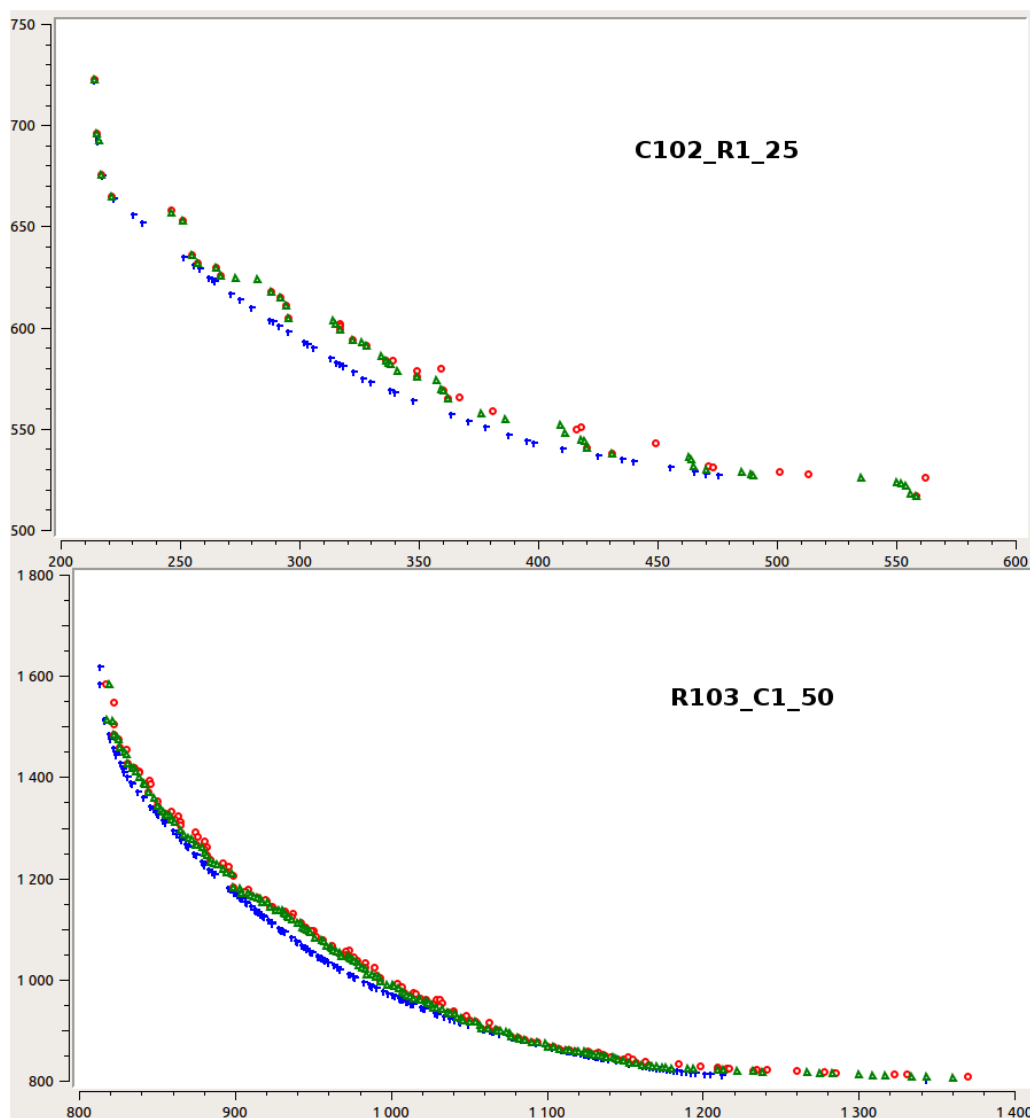


FIGURE 4.3: Ensemble des points non dominés des instances $C102_R1_25$ et $R103_C1_25$ - en triangle vert, les points non dominés - en croix bleue, les points de la borne inférieure - en cercle rouge, les bornes supérieures.

de branchement.

Pour vérifier si l'ajout des SR-inegalités dans la formulation (3.2) réduit le gap moyen du BOVRP, nous ajoutons toutes ces inegalités pour $|S| = 3$. Le tableau 4.3 montre l'impact de l'ajout de ces inegalités pour trois instances difficiles et deux valeurs de ε pour lesquelles le gap est élevé. Les colonnes LB , OPT , GAP , LB_{sr} et GAP_{sr} donnent respectivement la borne inférieure initiale, la solution optimale, le gap entre LB et OPT , la borne inférieure après ajout de toutes ces inegalités et le gap entre LB_{sr} et OPT . Cependant, nous ne relançons pas d'iteration de la generation de colonnes et il est donc important de noter

que LB_{sr} n'est pas une borne inférieure valide. LB_{sr} est seulement une borne supérieure sur la meilleure borne inférieure que nous pourrions obtenir en ajoutant certaines inégalités valides.

TABLE 4.3: Détails de la méthode ε -contrainte par type d'instances.

Instance	ε	LB	OPT	GAP	LB_{sr}	GAP_{sr}
25_R107_C1	477	539.621	583	7.4	549.126	5.8
	454	581.446	615	5.5	587.636	4.4
25_RC201_C1	306	403.955	436	7.3	404.373	7.2
	303	414.388	474	12.6	414.800	12.5
25_C201_R1	664	222.016	246	9.7	222.091	9.7
	554	370.778	409	9.3	386.938	5.4

La meilleure amélioration du gap obtenue est de 4.3% pour l'instance *C201_R1_25*, mais pour une autre valeur de ε dans cette même instance, le gap n'est pas amélioré. De plus, pour les deux autres instances testées, *R207_C1_25* et *RC201_C1_25*, le gap est peu amélioré (1.4% au maximum). Ainsi, nous voyons qu'ajouter les coupes classiques de tournées de véhicules n'est pas la meilleure piste à exploiter. En effet, cela augmente les temps de résolution sans une grande réduction du gap moyen et avec une règle de dominance des étiquettes moins efficace.

4.2.4 Conclusion

La performance de SEM repose sur le nombre de colonnes maximum qui doivent être générées pour trouver un point non dominé. Plus ce nombre est grand et plus la fonction *genroute* et la résolution des problèmes maîtres restreints sont longues. Nous avons également observé que plus la valeur de ε contraint le problème et plus ce nombre de colonnes générées est grand. Enfin, l'ajout d'inégalités valides usuelles pour les problèmes de tournées de véhicules ne diminuerait pas beaucoup le gap et d'autres pistes doivent être exploitées.

4.3 Améliorations de la méthode ε -contrainte par étapes

Cette section présente les améliorations de SEM suivantes :

- T_{gap} : une réduction du gap en Section 4.3.1 ;
- T_{lb} : une génération des bornes inférieures en Section 4.3.2 ;
- T_{valid} : une validation des points non dominés sans énumération des colonnes dans le gap en Section 4.3.3 ;
- T_{save} : une sauvegarde de l'état de l'énumération des colonnes en cours pour une certaine valeur de ε et une réutilisation pour d'autres valeurs en Section 4.3.4 ;
- T_{mult_dual} : un changement de la règle de dominance et un calcul de multiples bornes de complétion pour réduire le nombre de colonnes créées durant la génération de colonnes en Section 4.3.5 ;

- T_{bidir} : une variation de l'exploration de l'espace des objectifs en Section 4.3.6.

Chaque technique est insérée une à une dans SEM pour analyser son impact sur le temps de résolution. Les conditions d'expérimentation sont les mêmes que celles de la Section 3.5, hormis les bornes supérieures données en entrée. Enfin, les instances utilisées dans cette section sont les 168 instances avec 25 nœuds décrites dans la Section 3.5.1.

Dans les sous-sections de 4.3.1 à 4.3.6, chacune des 6 techniques T est ajoutée à SEM. Cette nouvelle méthode est alors comparée à SEM sans aucune technique. Nous comparons le nombre d'instances fermées, ainsi que la moyenne des temps de résolution des instances fermées par les deux méthodes. Ensuite, nous indiquons le nombre d'instances où le temps de résolution est de plus de 10 secondes entre les deux méthodes. La différence de 10 secondes permet de concentrer l'analyse sur des instances où T a un impact. Pour chaque technique T , 3 groupes d'instances sont définies :

- *Toutes*_ T , notée To : instances fermées par SEM et SEM avec l'ajout de T ;
- *Dégradées*_ T , notée D : instances où SEM est plus rapide que SEM avec l'ajout de T de plus de 10 secondes ;
- *Améliorées*_ T , notée A : instances où SEM avec l'ajout de T est plus rapide que SEM de plus de 10 secondes.

Enfin, chaque sous-section présente une figure représentant l'écart relatif de différentes caractéristiques, en pourcentages, entre SEM et SEM avec T selon les classes d'instances. Ces caractéristiques peuvent être le temps de résolution ou le nombre de colonnes générées. Par exemple, en notant $t(SEM)$ et $t(SEM_T)$ la moyenne des temps de résolution pour SEM et SEM avec T , l'écart relatif e est calculée comme suit :

$$e = 100 * \frac{t(SEM_T) - t(SEM)}{t(SEM)}$$

Par exemple, si e est négatif (resp. positif), SEM avec T est plus rapide (resp. plus lent). Si $e = -66\%$, alors SEM avec T a une perte de 66% du temps de résolution et est donc 3 fois plus rapide.

4.3.1 Réduction du gap

La méthode SEM est une méthode itérative qui, à chaque valeur de ε , calcule LB la solution optimale du PML(ε) et UB une solution réalisable du PM(ε) et génère toutes les routes de coût réduit positif inférieur au gap γ , différence de coût c^1 entre LB et UB . Cette technique, notée T_{gap} , réduit la valeur de γ d'une unité à chaque itération. Ainsi, la nouvelle valeur du gap γ est :

$$\gamma = c_{UB}^1 - c_{LB}^1 - 1 \quad (4.1)$$

Avec ce changement, il est possible de ne pas trouver de solution réalisable avec la méthode mono-objectif présentée en Section 3.3 ou de trouver une solution de coût

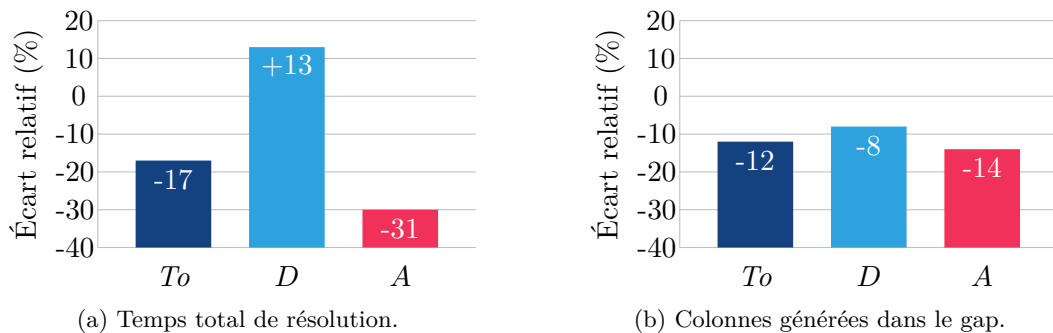


FIGURE 4.4: Écart relatif entre SEM et SEM avec T_{gap} selon les classes d'instances $Toutes_T_{gap}$ (T_0), $Dégradées_T_{gap}$ (D) et $Améliorées_T_{gap}$ (A).

supérieur à la borne supérieure UB . Dans ces cas, le point non dominé retourné est UB . Cette technique est viable car le coût c^1 est entier.

Résultats expérimentaux. Autant d'instances sont fermées par SEM avec ou sans T_{gap} , soit 100 instances sur 168. La moyenne des temps de résolution des instances fermées de SEM est de 1019.1 secondes et la moyenne du nombre de colonnes générées dans le gap est de 152942. Avec T_{gap} , la moyenne des temps de résolution des instances fermées est de 874.4 secondes et la moyenne du nombre de colonnes générées dans le gap est de 134983. SEM est plus rapide d'au moins 10 secondes que SEM avec T_{gap} sur 10 instances et SEM avec T_{gap} est plus rapide d'au moins 10 secondes que SEM sur 27 instances. La Figure 4.4 représente les différences de performance entre SEM et SEM avec T_{gap} pour le temps de résolution et le nombre de colonnes générées.

Nous observons que le nombre de colonnes générées est réduit par T_{gap} d'environ 10% en moyenne quel que soit la classe d'instances. Plus le nombre de colonnes générées est réduit, même légèrement, et plus l'impact de T_{gap} est positif. En théorie, cette technique ne peut qu'améliorer le temps de résolution car les bornes supérieures sont données au solveur utilisé en amont de la résolution. Cependant, certaines instances sont en pratique plus longues à résoudre avec la technique lors de la résolution du PMR à l'optimalité par Gurobi. Dans certains cas, il semble que la réduction du nombre de colonnes empêche de trouver des bornes inférieures de qualité au début de l'arbre de recherche du B&B. En effet, la séparation des coupes à la racine de l'arbre de recherche dans Gurobi est heuristique. Cela entraîne une explosion de la taille de l'arbre de recherche.

4.3.2 Construction de la borne inférieure

Soit E_1 le point c^1 -extrême du PM. Lorsque ε est fixée à une valeur ε_1 telle que $\varepsilon_1 < c^2(E_1)$, la contrainte (3.2.4) du PML est saturée et par conséquent, la solution duale α^{ε_1} est strictement négative. La valeur α^{ε_1} représente également l'augmentation (ou la diminution) de la valeur de la fonction objectif si le coefficient ε_1 augmente (ou dimi-

nue) d'une unité. Le domaine de validité du coût marginal α^{ε_1} est donné par une analyse de sensibilité et définit un intervalle $[\Delta_a^1; \Delta_b^1]$ dans lequel la valeur de α^{ε_1} ainsi que la base optimale ne changent pas.

Note 3. *Plus la valeur de ε décroît dans $PML(\varepsilon)$, plus la valeur de α^ε sera faible.*

Soit LB_1 l'optimum du $PML(\varepsilon_1)$, α^{ε_1} la variable duale associée à la contrainte (3.2.4) et $[\Delta_a^1; \Delta_b^1]$ le domaine de validité de α^{ε_1} . Pour ε_2 tel que $\Delta_a^1 \leq \varepsilon_2 \leq \Delta_b^1$, nous pouvons déduire la valeur de la solution optimale du $PML(\varepsilon_2)$, LB_2 :

$$c_{LB_2}^1 = c_{LB_1}^1 - (\varepsilon_1 - \varepsilon_2)\alpha^{\varepsilon_1} \quad (4.2)$$

D'après la note 3, dans le cas où un coefficient ε_3 est tel que $\varepsilon_3 < \Delta_a^1$, alors on a $\alpha^{\varepsilon_1} > \alpha^{\varepsilon_3}$. La valeur de l'optimum du $PML(\varepsilon_3)$, LB_3 , peut donc être bornée par :

$$c_{LB_1}^1 - (\varepsilon_1 - \varepsilon_3)\alpha^{\varepsilon_1} \leq c_{LB_3}^1 \leq c_{LB_1}^1 - (\varepsilon_1 - \varepsilon_3)\alpha^{\varepsilon_3} \quad (4.3)$$

De plus, dans la domaine de validité de la variable α^{ε_1} , la base optimale ne change pas. Ainsi, les routes utilisées par la solution LB_1 composent également une solution valide pour LB_2 . Par conséquent, en théorie, une seule génération de colonnes est nécessaire pour calculer les bornes inférieures associées à des ε ayant des variables duales α^ε dans le même domaine de validité. En pratique, l'ensemble des colonnes \tilde{R} générées pour trouver la solution optimale du $PML(\varepsilon_1)$ est utilisé pour trouver la solution optimale du $PML(\varepsilon_2)$ en résolvant une unique fois $PMLR(\varepsilon_2, \tilde{R})$, si ε_1 et ε_2 sont dans le même domaine de validité. Cette technique de génération de borne inférieure est notée T_{lb} .

Résultats expérimentaux. Huit instances de moins sont fermées avec SEM plus T_{lb} en comparaison avec SEM, soit 92 instances sur 168. Sur l'ensemble des instances fermées par les deux méthodes, la moyenne des temps de résolution est de 665.1 et 958.9 secondes pour SEM sans et avec T_{lb} , respectivement. SEM est plus rapide d'au moins 10 secondes que SEM avec T_{lb} sur 23 instances et SEM avec T_{lb} est plus rapide d'au moins 10 secondes que SEM sur 25 instances. De plus, T_{lb} est utilisée sur 60% des points de la borne inférieure en moyenne. La Figure 4.5 représente les différences de performance entre SEM et SEM avec T_{lb} pour le temps de génération de la borne inférieure, le temps de résolution total et le nombre de colonnes générées dans le gap. La partie (b) de la figure représente le pourcentage d'utilisation de T_{lb} sur le nombre de borne inférieure calculée dans SEM.

Nous observons que les temps de résolution sont en moyenne plus rapides lorsque T_{lb} n'est pas utilisée; même si T_{lb} est appliqué sur un grand nombre de bornes inférieures (environ 60% des fois). Ce résultat est surprenant vu que la technique diminue le nombre d'itérations de génération de colonnes pour générer des bornes inférieures et que le temps de génération des bornes inférieures est réduit dans tous les cas. Ceci s'explique car les variables duales obtenues avec l'utilisation de la technique sont moins bonnes et elles entraînent un plus grand temps de calcul pour la génération des routes de coût réduit inférieur au gap.

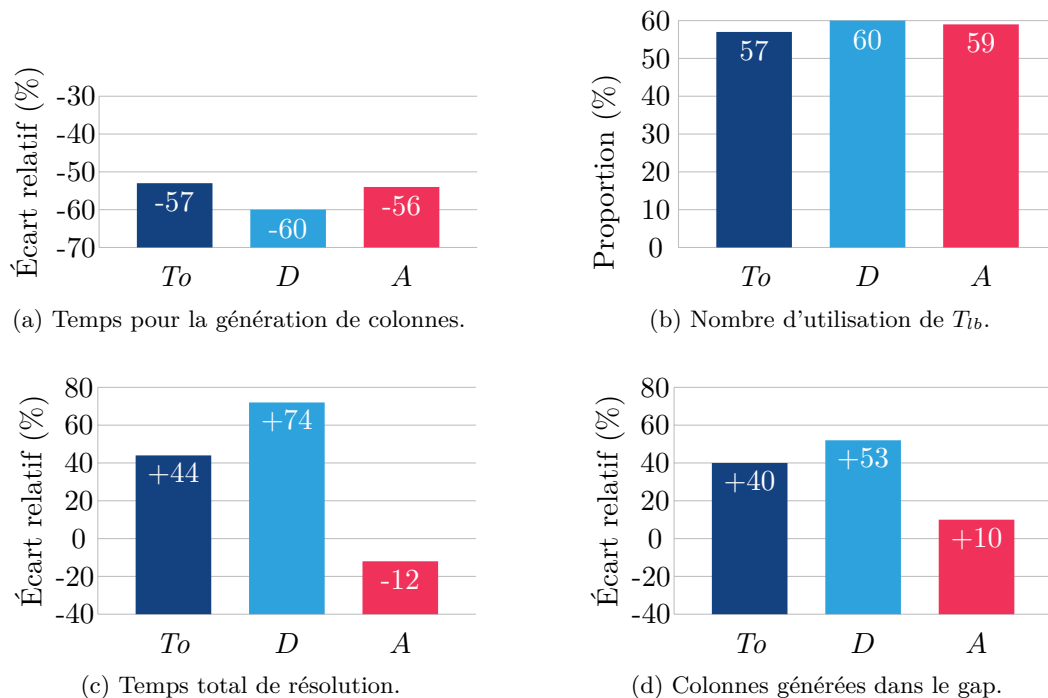


FIGURE 4.5: Écart relatif entre SEM et SEM avec T_{lb} selon les classes d'instances $Toutes_T_{lb}$ (T_o), $Dégradées_T_{lb}$ (D) et $Améliorées_T_{lb}$ (A).

4.3.3 Validation des points non dominés

Dans SEM, à chaque itération pour une valeur ε_1 , une borne inférieure et une borne supérieure sont calculées et un ensemble de routes \bar{R}_1 est généré. Cet ensemble contient les routes de coût réduit inférieur à la différence de coût c^1 entre les deux bornes. La solution O_1 est obtenue en résolvant $PMR(\varepsilon_1, \bar{R}_1)$. À l'itération suivante $\varepsilon_2 = c^2(O_1) - 1$, la solution du $PMR(\varepsilon_2, \bar{R}_1)$ est O_2 . Cependant, il n'existe aucune preuve que cette solution est la solution optimale du $PM(\varepsilon_2)$. Dans cette partie, nous donnons les conditions pour que O_2 soit prouvée comme solution optimale pour $PM(\varepsilon_2)$.

Soit λ^{ε_1} (respectivement λ^{ε_2}) la solution optimale du $PDL(\varepsilon_1)$ (respectivement $PDL(\varepsilon_2)$) avec $\varepsilon_1 > \varepsilon_2$. Nous pouvons noter qu'en général $\lambda_i^{\varepsilon_1} \neq \lambda_i^{\varepsilon_2}, \forall i \in V'$, même si $\alpha^{\varepsilon_1} = \alpha^{\varepsilon_2}$. Le Théorème 2 donne les conditions pour lesquelles la somme des variables duales associées aux contraintes (3.2.2) et (3.2.3) est égale pour λ^{ε_1} et λ^{ε_2} . Si la condition est respectée, le Théorème 3 peut être appliqué. Il prouve que la somme des coûts réduits d'une solution réalisable UB_2 , calculés avec λ^{ε_1} , est de valeur égale à la somme des coûts réduits de UB_2 , calculés avec λ^{ε_2} .

Théorème 2. $\sum_{i \in V'} \lambda_i^{\varepsilon_1} + K \lambda_0^{\varepsilon_1} \leq \sum_{i \in V'} \lambda_i^{\varepsilon_2} + K \lambda_0^{\varepsilon_2}$.

De plus, l'égalité a lieu si $\alpha^{\varepsilon_1} = \alpha^{\varepsilon_2}$.

Démonstration. Soit LB_1 (respectivement LB_2) la solution optimale du PML(ε_1) (respectivement PML(ε_2)) avec $\varepsilon_1 > \varepsilon_2$. Avec les remarques de la section 4.3.2, nous obtenons :

$$\begin{aligned}
& c_{LB_1}^1 - (c_{LB_1}^2 - c_{LB_2}^2)\alpha^{\varepsilon_1} = c_{LB_2}^1 \\
\Rightarrow & \sum_{r \in R} c_r^1 x_r^{LB_1} - (\varepsilon_1 - \varepsilon_2)\alpha^{\varepsilon_1} = \sum_{r \in R} c_r^1 x_r^{LB_2} \\
\Rightarrow & \sum_{r \in R} c_r^1 x_r^{LB_1} - \varepsilon_1 \alpha^{\varepsilon_1} \leq \sum_{r \in R} c_r^1 x_r^{LB_2} - \varepsilon_2 \alpha^{\varepsilon_2} \text{ [par Note 3, } \alpha^{\varepsilon_2} \leq \alpha^{\varepsilon_1} \leq 0\text{]} \\
\Rightarrow & \sum_{i \in V'} \lambda_i^{\varepsilon_1} + K \lambda_0^{\varepsilon_1} \leq \sum_{i \in V'} \lambda_i^{\varepsilon_2} + K \lambda_0^{\varepsilon_2} \text{ [théorème de dualité forte]}
\end{aligned}$$

□

Théorème 3. Soit UB_2 une solution réalisable du PM(ε_1) avec $c_{UB_2}^2 = \varepsilon_2$. Si $\lambda_0^{\varepsilon_1} = \lambda_0^{\varepsilon_2}$ et $\alpha^{\varepsilon_1} = \alpha^{\varepsilon_2}$, la somme des coûts réduits de UB_2 par rapport aux variables duales λ^{ε_2} peut être définie comme suit :

$$\sum_{r \in R} \bar{c}_r(\lambda^{\varepsilon_2}) x_r^{UB_2} = \sum_{r \in R} (c_r^1 - \sum_{i \in V} a_{ir} \lambda_i^{\varepsilon_1} - c_r^2 \alpha^{\varepsilon_1}) x_r^{UB_2}$$

Démonstration. Comme $\lambda_0^{\varepsilon_1} = \lambda_0^{\varepsilon_2}$ et $\alpha^{\varepsilon_1} = \alpha^{\varepsilon_2}$, par le Théorème 2, nous pouvons déduire $\sum_{i \in V'} \lambda_i^{\varepsilon_1} = \sum_{i \in V'} \lambda_i^{\varepsilon_2}$. Ainsi :

$$\begin{aligned}
\sum_{r \in R} \bar{c}_r(\lambda^{\varepsilon_2}) x_r^{UB_2} &= \sum_{r \in R} (c_r^1 - \sum_{i \in V} a_{ir} \lambda_i^{\varepsilon_2} - c_r^2 \alpha^{\varepsilon_2}) x_r^{UB_2} \\
&= \sum_{r \in R} c_r^1 x_r^{UB_2} - \sum_{i \in V} (\sum_{r \in R} a_{ir} x_r^{UB_2}) \lambda_i^{\varepsilon_2} - \sum_{r \in R} c_r^2 \alpha^{\varepsilon_2} x_r^{UB_2} \\
&= \sum_{r \in R} c_r^1 x_r^{UB_2} - \sum_{i \in V'} \lambda_i^{\varepsilon_2} - \sum_{r \in R} x_r^{UB_2} \lambda_0^{\varepsilon_2} - \sum_{r \in R} c_r^2 \alpha^{\varepsilon_2} x_r^{UB_2} \\
&= \sum_{r \in R} c_r^1 x_r^{UB_2} - \sum_{i \in V'} \lambda_i^{\varepsilon_1} - \sum_{r \in R} x_r^{UB_2} \lambda_0^{\varepsilon_1} - \sum_{r \in R} c_r^2 \alpha^{\varepsilon_1} x_r^{UB_2} \\
&= \sum_{r \in R} (c_r^1 - \sum_{i \in V} a_{ir} \lambda_i^{\varepsilon_1} - c_r^2 \alpha^{\varepsilon_1}) x_r^{UB_2}
\end{aligned}$$

□

Cette technique de validation de points non dominés, notée T_{valid} , insérée dans SEM permet de réduire le nombre d'appels à *genroute* et donc le nombre total de routes générées. Ainsi, une fois \bar{R}_1 généré et le nouvel ε_2 déterminé, une borne inférieure LB_2 et une borne supérieure UB_2 sont calculées. La Figure 4.6 illustre deux cas possibles.

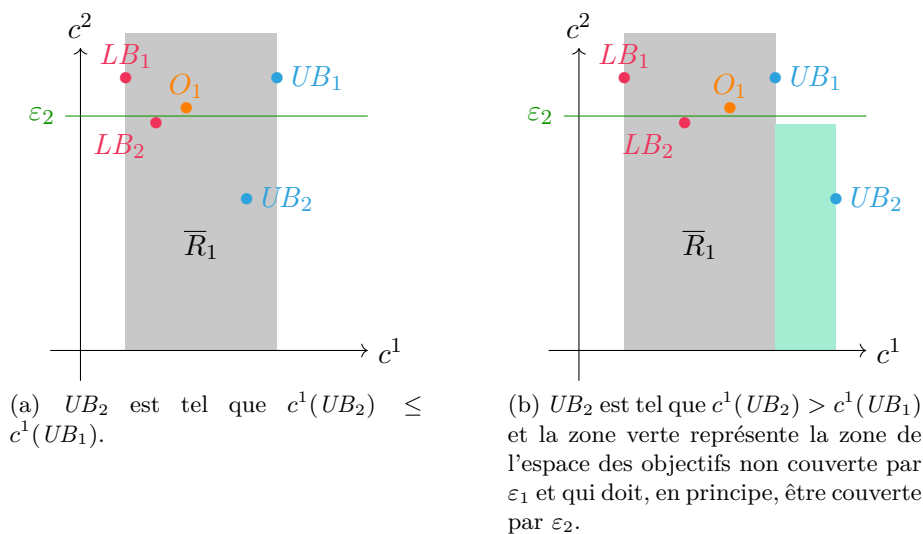


FIGURE 4.6: Itérations possibles de la méthode ε -contrainte où LB_i , UB_i et O_i sont respectivement les bornes inférieures, supérieures et solution optimale calculées pour ε_i et \bar{R}_1 est l'ensemble des routes de coût réduit inférieur ou égal au gap entre LB_1 et UB_1 .

Si UB_2 est tel que $c^1(UB_2) \leq c^1(UB_1)$ (à gauche), alors il est immédiat de noter que l'ensemble de routes \bar{R}_1 peut être utilisé pour trouver le point non dominé suivant O_2 . Dans l'autre cas où $c^1(UB_2) > c^1(UB_1)$ (à droite), les théorèmes précédents donnent des conditions pour lesquelles \bar{R}_2 peut ne pas être généré. Ainsi, si $\lambda_0^{\varepsilon_1} = \lambda_0^{\varepsilon_2}$, $\alpha^{\varepsilon_1} = \alpha^{\varepsilon_2}$ et $c^1(UB_1) - c^1(LB_1) \geq c^1(UB_2) - c^1(LB_2)$, alors la solution optimale O_2 du $\text{PMR}(\varepsilon_2, \bar{R}_1)$ est un point non dominé. L'itération suivante est effectuée pour $\varepsilon_3 = c^2(O_2) - 1$ sans appel à *genroute* pour ε_2 .

Pour éviter d'avoir des routes inutiles dans les problèmes entiers, un tri de l'ensemble \bar{R}_1 est effectué avant de résoudre $\text{PMR}(\varepsilon_2, \bar{R}_1)$. Ce tri enlève les routes de coût réduit, calculé avec λ^{ε_2} , supérieur à $c^1(UB_2) - c^1(LB_2)$.

Résultats expérimentaux. Autant d'instances sont fermées par SEM avec et sans T_{valid} , soit 100 instances sur 168. Sur l'ensemble des instances fermées par les deux méthodes, la moyenne des temps de résolution est de 1019.1 et 938.0 secondes pour SEM sans et avec T_{valid} , respectivement. SEM est plus rapide d'au moins 10 secondes que SEM avec T_{valid} sur 14 instances et SEM avec T_{valid} est plus rapide d'au moins 10 secondes que SEM sur 19 instances. De plus, la technique est utilisée sur 30% des points non dominés en moyenne. La Figure 4.7 représente les différences de performance entre SEM et SEM avec T_{valid} pour le temps de *genroute* et le nombre de colonnes générées dans le gap. La partie (c) de la figure représente le pourcentage d'utilisation de T_{valid} sur le nombre de points non dominés.

Nous observons que toutes les instances ont une réduction d'environ 25% de colonnes

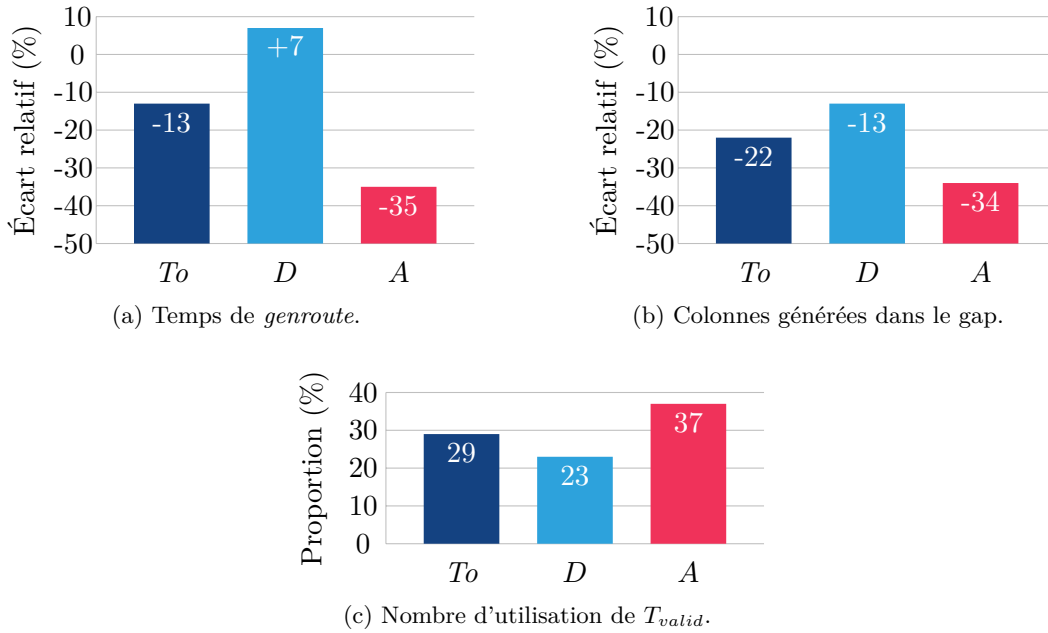


FIGURE 4.7: Écart relatif entre SEM et SEM avec T_{valid} selon les classes d'instances $Toutes_T_{lb}$ (T_o), $Dégradées_T_{valid}$ (D) et $Améliorées_T_{valid}$ (A).

générées dans le gap et que la technique est utilisée sur environ 30% des points non dominés. Malgré cela, les temps de génération des colonnes de coût réduit inférieur au gap de SEM avec T_{valid} peuvent être équivalents à ceux de SEM sur certaines instances, voire plus lents. Ceci s'explique car certains points non dominés de $\mathcal{Y}_{\mathcal{N}}$ vont être plus durs à atteindre et demandent la génération de beaucoup de colonnes. Pour ces points là, la méthode n'améliore pas le temps de résolution car les conditions pour l'appliquer ne sont pas remplies. Ainsi, sur certaines instances, le gain de temps engendré par la technique ne compense pas les traitements supplémentaires de gestion des colonnes.

4.3.4 Gestion et sauvegarde de colonnes

Dans la méthode de référence, de nombreuses routes sont générées à de multiples reprises pour plusieurs valeurs de ε et des routes non réalisables (fenêtres de temps, charge du véhicule...) vont également être vérifiées de nombreuses fois. Cette technique, notée T_{save} , a pour but de limiter la génération d'une même route.

Cette technique sauvegarde les routes précédemment générées et éliminées à de précédentes itérations. Contrairement à SEM, l'énumération des colonnes dans le gap (Section 3.3.3) ne se fait pas avec un algorithme d'étiquetage bidirectionnel, mais avec un algorithme mono-directionnel. En effet, lors d'une énumération bidirectionnelle, plus de chemins doivent être sauvegardés et les performances de la méthode se détériorent. Lors

d'une énumération mono-directionnelle, les étiquettes peuvent être classées en quatre catégories :

- \overline{R} : les étiquettes formant des routes réalisables de coût réduit inférieur au gap ;
- \overline{Er} : les étiquettes formant des routes réalisables dominées ou de coût réduit strictement supérieur au gap ;
- \overline{Ec} : les étiquettes formant des chemins réalisables dominés, ou des chemins irréalisables par rapport au coût c^2 , ou des chemins réalisables éliminés avec les bornes de complétion ;
- \overline{I} : les étiquettes formant des chemins ou des routes irréalisables avec des ressources autres que le coût associé à la contrainte ε .

Les étiquettes de \overline{I} sont irréalisables quel que soit la valeur de ε puisqu'elles correspondent à des routes ne respectant pas les contraintes de faisabilité d'une route et ne sont donc pas conservées. Au contraire, les étiquettes des ensembles \overline{R} , \overline{Er} et \overline{Ec} peuvent être valides pour d'autres itérations de SEM. Soit LB_i la solution optimale du PML(ε_i), UB_i une solution réalisable du PM(ε_i) et λ^{ε_i} la solution optimale du PDL(ε_i). Lors de l'itération i de SEM où $\varepsilon = \varepsilon_i$, l'ensemble des routes \overline{R}_i obtenu avec *genroute* est initialisé avec les ensembles \overline{R}_{i-1} et \overline{Er}_{i-1} . Toutes les routes de ces deux ensembles ayant un coût réduit, calculé avec λ^{ε_i} , inférieur au gap formé par LB_i et UB_i sont ajoutées dans \overline{R}_i . Si une route a un coût réduit supérieur au gap, elle est placée dans \overline{Er}_i . L'extension des étiquettes est alors effectué à partir de l'ensemble \overline{Ec}_{i-1} et les ensembles \overline{R}_i , \overline{Er}_i et \overline{Ec}_i sont remplis en conséquence. Lors de l'initialisation tous les ensembles sont vides excepté \overline{Ec}_0 qui contient le chemin qui reste au dépôt.

Pour éviter de garder des très grands ensembles de routes, les ensembles sont re-initialisés si $\lambda_0^{\varepsilon_i} \neq \lambda_0^{\varepsilon_{i-1}}$ ou $\alpha^{\varepsilon_i} \neq \alpha^{\varepsilon_{i-1}}$. Cela revient à re-initialiser lorsque l'on change de domaine de validité des solutions optimales du PML. De plus, lorsque T_{save} est testée, nous mettons également T_{valid} décrite en Section 4.3.3. En effet, il est inutile de relancer la fonction *genroute* si le gap de l'itération $i - 1$ est plus grand que celui à l'itération i et que les conditions sur les variables duales sont respectées.

Résultats expérimentaux. SEM avec T_{save} ferme 102 instances sur 168, alors que SEM en ferme 100. Cependant, une instance est uniquement fermée par SEM. Sur l'ensemble des instances fermées par les deux méthodes, la moyenne des temps de résolution est de 917.6 et 482.8 secondes pour SEM sans et avec T_{save} , respectivement. SEM est plus rapide d'au moins 10 secondes que SEM avec T_{save} sur 11 instances et SEM avec T_{save} est plus rapide d'au moins 10 secondes que SEM sur 35 instances. De plus, T_{save} est utilisée sur 27% des points non dominés en moyenne. La Figure 4.8 représente les différences de performance entre SEM et SEM avec T_{save} pour le temps de résolution. Les parties (b) et (c) de la figure représente le pourcentage d'utilisation de T_{valid} et de T_{save} sur le nombre de points non dominés.

Cette technique est donc performante sur la majorité des instances, mais elle peut freiner la résolution de certaines instances comme expliqué en Section 4.3.3. Nous remarquons également que les deux techniques combinées s'utilisent sur plus de la moitié des points non dominés en moyenne.

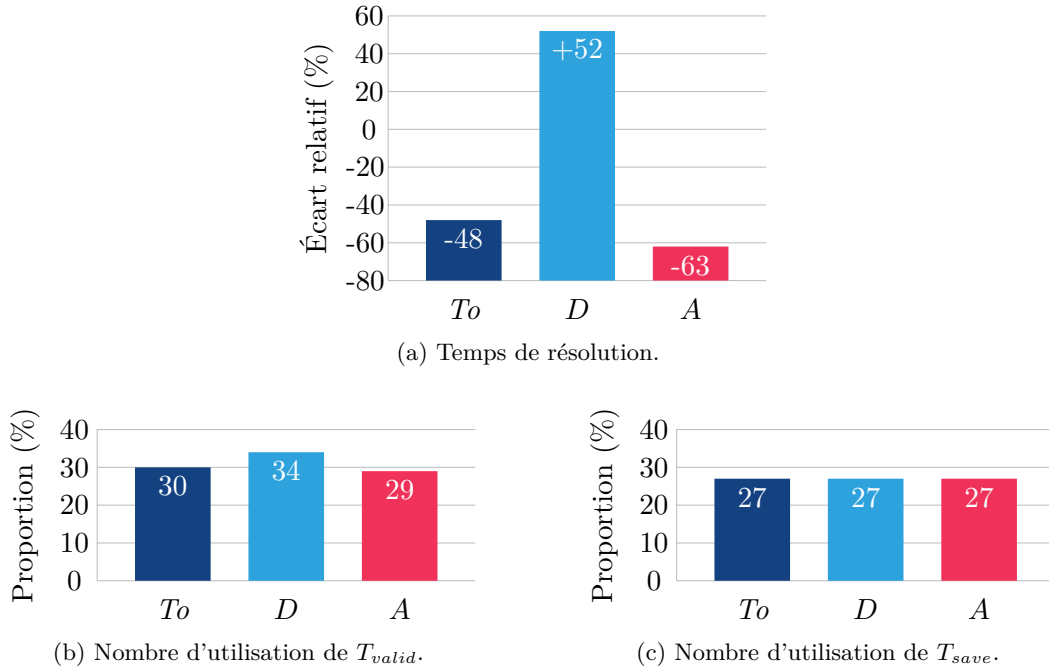
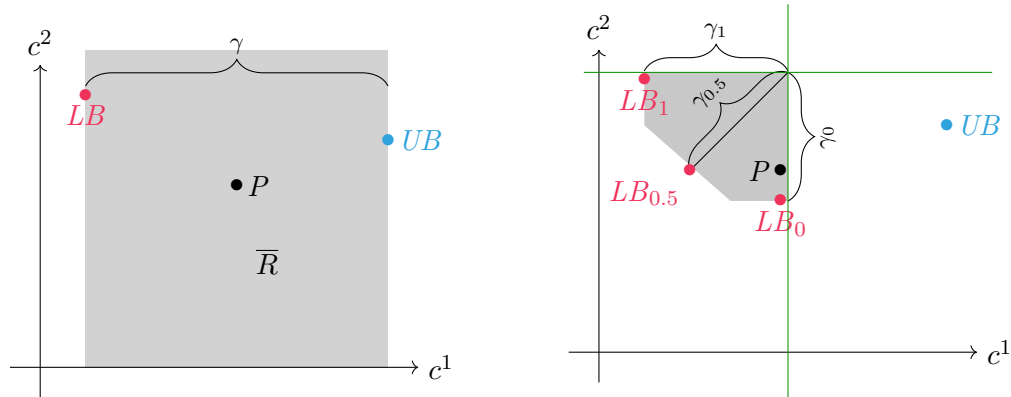


FIGURE 4.8: Écart relatif entre SEM et SEM avec T_{save} selon les classes d'instances $Toutes_T_{save}$ (T_o), $Dégradées_T_{save}$ (D) et $Améliorées_T_{save}$ (A).

4.3.5 Multiples coûts duaux

Lors de l'énumération des colonnes de coût réduit inférieur au gap, l'ensemble \bar{R} de colonnes générées peut être grand, et de ce fait, le problème maître peut devenir long à résoudre. Pour empêcher cette situation, \bar{R} est limité à une taille fixe MAX_COL . Si la limite n'est pas atteinte, le point P associé à la solution optimale du $PMR(\varepsilon, \bar{R})$ est un point non dominé. Au contraire, si la limite est atteinte, P n'est pas forcément un point non dominé et si $PMR(\varepsilon, \bar{R})$ n'a pas de solution optimale, alors P est remplacé par UB . Pour l'itération suivante, la valeur de ε est mise à $c^2(P) - 1$.

Lorsque ε a atteint la borne inférieure sur le coût c^2 , il faut continuer l'énumération des points non dominés dans les rectangles formés par LB et P pour toutes les valeurs de ε où la limite MAX_COL a été atteinte. Ce cas, où P n'est pas forcément un point non dominé, est illustrée dans la Figure 4.9. Pour explorer le rectangle entre LB et P , nous utilisons la relaxation linéaire de la formulation (3.4) pour les paramètres suivants : w quelconque, $L^1 = c^1(P) - 1$ et $L^2 = c^2(LB) - 1$. Cette formulation va être résolue pour plusieurs valeurs de w entre 0 et 1 inclus : LB_0 ($w = 0$), LB_1, \dots, LB_W ($w = 1$). Par exemple, dans la Figure 4.9, $W = \{0, 0.5, 1\}$ et trois bornes inférieures LB_1 , $LB_{0.5}$ et LB_0 sont obtenues. Nous ne calculons pas de bornes supérieures car l'exploration de la zone se fait jusqu'au point de nadir local $N = (c^1(P) - 1, c^2(LB) - 1)$. Ensuite, des gaps sont calculés entre chaque LB_i , $i \in 0, \dots, W$ et le point $N_i = (c^1(P) - 1, c^2(LB) - 1)$.



(a) Les colonnes \bar{R} générées dans le gap γ entre LB et UB sont limitées par MAX_COL . L'optimum P du $\text{PMR}(\bar{R})$ n'est pas forcément un point non dominé.

(b) Dans le cas (a), la technique T_{mult_dual} est utilisée. L'ensemble \bar{R}' de colonnes simultanément dans les gaps γ_1 , $\gamma_{0.5}$ et γ_0 est généré. L'optimum P du $\text{PMR}(\bar{R}')$ est un point non dominé.

FIGURE 4.9: Illustration des bornes inférieures et des gaps obtenus avec T_{mult_dual} pour $W = \{0, 0.5, 1\}$. Les droites continues vertes représentent des contraintes sur les objectifs c^1 et c^2 .

L'énumération des routes est alors faite simultanément pour tous les LB_i , $i \in 0, \dots, W$, les gaps et les variables duales associés. Une route possède donc plusieurs coûts réduits qui seront des ressources distinctes dans l'algorithme d'étiquetage. Cette route peut être éliminée si au moins l'un de ses coûts réduits est supérieur au gap associé. Le nombre de colonnes générées pour prouver que P est un point non dominé est ainsi plus faible qu'une énumération des colonnes pour chaque LB_i pris en compte séparément. Cette technique, notée T_{mult_dual} , vise donc à réduire le nombre de routes générées dans des zones difficiles. Une technique pour éliminer les routes selon plusieurs solutions duales réalisables a été proposée par [Baldacci et al. \(2011\)](#) pour le VRP mono-objectif.

Résultats expérimentaux. SEM avec T_{mult_dual} ferme 117 instances sur 168, alors que SEM en ferme 100. Cependant, une instance est uniquement fermée par SEM. Sur l'ensemble des instances fermées par les deux méthodes, la moyenne des temps de résolution est de 917.6 et 355.4 secondes pour SEM sans et avec T_{mult_dual} , respectivement. SEM est plus rapide d'au moins 10 secondes que SEM avec T_{mult_dual} sur 12 instances et SEM avec T_{mult_dual} est plus rapide d'au moins 10 secondes que SEM sur 21 instances. De plus, T_{mult_dual} est utilisée sur seulement 1.6% des points non dominés en moyenne. La Figure 4.10 représente les différences de performance entre SEM et SEM avec T_{mult_dual} pour le temps de résolution et le nombre de colonnes générées dans le gap.

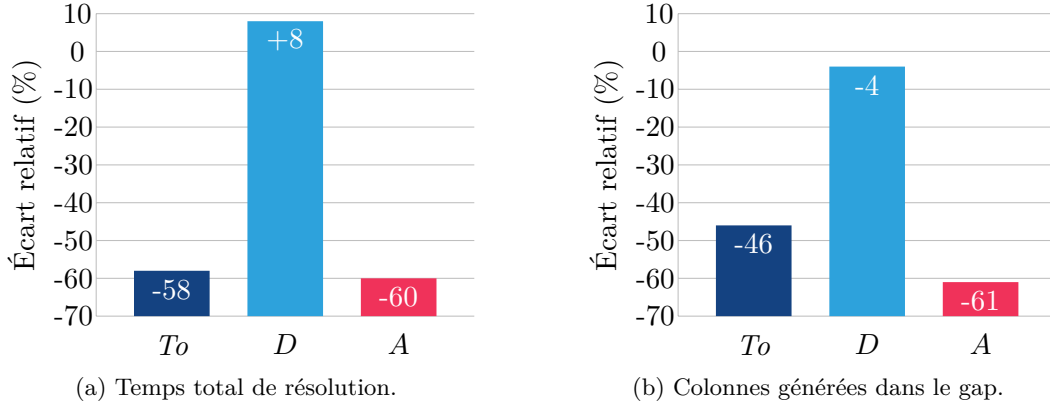


FIGURE 4.10: Écart relatif entre SEM et SEM avec T_{mult_dual} selon les classes d'instances $Toutes_T_{mult_dual}$ (T_o), $Dégradées_T_{mult_dual}$ (D) et $Améliorées_T_{mult_dual}$ (A).

Nous observons que les temps de résolution sont en moyenne plus rapides pour SEM avec T_{mult_dual} ; malgré le nombre minime de fois où cette astuce est utilisée. Ainsi, cela démontre que SEM peut ne pas converger à cause d'un unique point non dominé qui demande la génération d'un trop grand nombre de colonnes.

4.3.6 Exploration bidirectionnelle

Dans la Section 4.2, nous avons observé que la fin de l'ensemble des points non dominés est plus dur à obtenir sur certaines instances car l'objectif est très contraint. Ainsi, nous proposons une technique bidirectionnelle d'exploration de l'espace des objectifs, notée T_{bidir} . Soit E_1 et E_2 les points c^1 et c^2 extrêmes. Le coût c^2 situé au milieu de \mathcal{Y}_N a pour valeur $m = c^2(E_1) + \frac{c^2(E_2) - c^2(E_1)}{2}$. Ainsi, \mathcal{Y}_N peut être divisé en deux parties égales par rapport au coût c^2 . Les points non dominés de coût c^2 entre m et $c^2(E_2)$ sont trouvés par la technique ε -contrainte qui minimise le coût c^1 pour ε variant entre $c^2(E_2) - 1$ et m . Le dernier point non dominé trouvé O est tel que $c^2(O) \leq m$. Les points de coût c^2 entre $c^2(E_1)$ et m sont trouvés par la méthode ε -contrainte qui minimise le coût c^2 pour ε variant entre $c^1(E_1) - 1$ et $c^1(O)$.

Résultats expérimentaux. Dix instances de moins sont fermées avec SEM plus T_{bidir} en comparaison avec SEM, soit 90 instances sur 168. Sur l'ensemble des instances fermées par les deux méthodes, la moyenne des temps de résolution est de 486.0 et 447.3 secondes pour SEM sans et avec T_{bidir} , respectivement. SEM est plus rapide d'au moins 10 secondes que SEM avec T_{bidir} sur 30 instances et SEM avec T_{bidir} est plus rapide d'au moins 10 secondes que SEM sur 10 instances. La Figure 4.11 représente les différences de performance entre SEM et SEM avec T_{mult_dual} pour le temps de résolution et le nombre de colonnes générées dans le gap.

Nous observons que les temps de résolution sont aggravés par l'ajout de la technique et

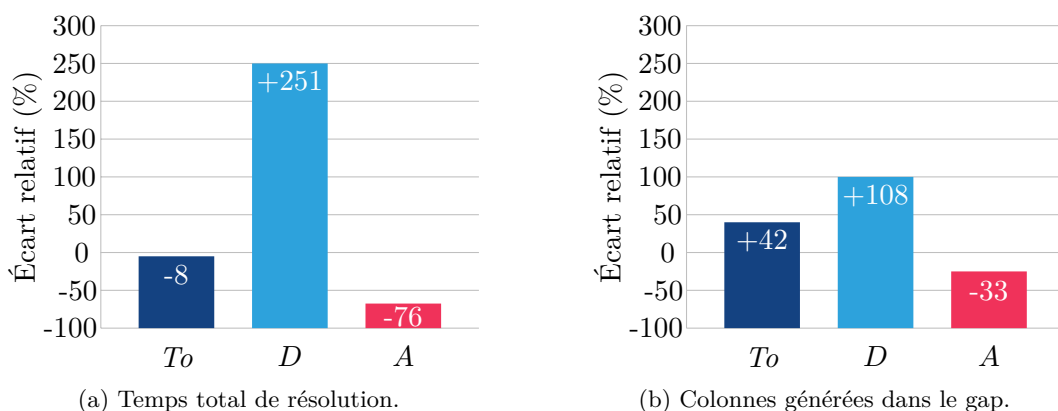


FIGURE 4.11: Écart relatif entre SEM et SEM avec T_{bidir} selon les classes d'instances $Toutes_T_{bidir}$ (T_o), $Dégradées_T_{bidir}$ (D) et $Améliorées_T_{bidir}$ (A).

que le nombre de colonnes générées est augmenté de 50% en moyenne.

4.3.7 Conclusion

Dans cette section, nous avons vu six techniques d'amélioration de la méthode ε -contrainte par étapes. Les résultats expérimentaux effectués mettent en évidence les bonnes performances de T_{gap} (Section 4.3.1), de T_{valid} (Section 4.3.3), de T_{save} (Section 4.3.4) et de T_{mult_dual} (Section 4.3.5). Toutefois, T_{gap} (Section 4.3.1) a peu d'impact sur la moyenne des résultats et est surtout utile lorsque les bornes supérieures sont de mauvaise qualité. La technique T_{lb} (Section 4.3.2) n'est pas très efficace pour SEM, mais peut être appliquée sur des méthodes ne désirant que la borne inférieure. Enfin, T_{bidir} (Section 4.3.6) est performante sur certaines instances puisqu'elle permet de pallier la difficulté de génération de la fin de l'ensemble des points non dominés, comme observé en Section 4.2. Cependant, elle est dépendante des objectifs du problème puisque si la génération des colonnes est moins performante sur l'un des objectifs, cette technique sera directement impactée. De plus, T_{mult_dual} reprend les avantages de T_{bidir} lorsque les points non dominés deviennent difficiles à obtenir, sans en avoir l'inconvénient précédemment décrit.

4.4 Nouvelle méthode ε -contrainte

Pour créer une nouvelle méthode ε -contrainte par étapes, nous utilisons les améliorations proposées en Section 4.3. Cette nouvelle méthode est décrite en Section 4.4.1 et des résultats expérimentaux sont proposés en Section 4.4.2. Elle est notée ISEM pour *Improved Stepwise ε -constraint Method*.

4.4.1 Description

La nouvelle méthode ε -contrainte ISEM est une extension de SEM qui contient les techniques T_{gap} , T_{valid} , T_{save} et T_{mult_dual} . Nous expliquons son fonctionnement pour un problème bi-objectif minimisant c^1 et c^2 . L'objectif c^2 est celui ajouté en contrainte dans la formulation ε -contrainte, alors que le coût c^1 est minimisé. ISEM est présentée dans l'Algorithme 6 où les notations suivantes sont utilisées :

- Y est l'ensemble des points non dominés trouvés ;
- I est une liste d'indice des points qui n'ont pas été prouvés non dominés ;
- λ et λ_{old} sont des solutions optimales duales du PDL2 à l'itération courante et précédente ;
- γ et γ_{old} sont les gap à l'itération courante et précédente ;
- \bar{R} et \bar{R}_{old} sont les routes réalisables de coût réduit inférieur à γ et γ_{old} à l'itération courante et précédente ;
- \bar{Er} et \bar{Er}_{old} sont les routes réalisables dominées ou de coût réduit supérieur à γ et γ_{old} à l'itération courante et précédente ;
- \bar{Ec} et \bar{Ec}_{old} sont les chemins réalisables dominés ou éliminés par des bornes de complétion ou par le coût c^2 à l'itération courante et précédente.

L'Algorithme 6 emploie les fonctions suivantes :

- *calcul_extreme* : décrite en Section 3.4.1, retourne les deux solutions c^1 et c^2 -extrêmes.
- *genroute*($\varepsilon, LB, UB, \lambda, E$) : génère l'ensemble des colonnes de coût réduit, calculé avec les variables duales λ pour la formulation (3.2) avec ε , inférieur à la différence de coût entre la borne supérieure UB et inférieure LB . L'énumération des routes se fait à partir des chemins contenus dans l'ensemble E . Si le seul chemin de E est celui de rester au dépôt, cette fonction est similaire au *genroute* défini en Section 3.3. De plus, l'ensemble retourné doit être de taille maximale MAX_COL.
- *genroute_multiple_duals*($\varepsilon, LB_0, \dots, LB_{|W|}, UB, \lambda_0, \dots, \lambda_{|W|}$) : retourne un ensemble de routes ayant tous leurs coûts réduits, calculés en fonction de $\lambda_0, \dots, \lambda_{|W|}$, inférieurs à chaque différence de coût entre UB et les bornes inférieures $LB_0, \dots, LB_{|W|}$.
- *dual_different*($\lambda^{\varepsilon_1}, \lambda^{\varepsilon_2}$) : retourne vrai si $\lambda_0^{\varepsilon_1} \neq \lambda_0^{\varepsilon_2}$ ou $\alpha^{\varepsilon_1} \neq \alpha^{\varepsilon_2}$, faux sinon.
- *trier*($\lambda, LB, UB, \bar{R}_{old}, \bar{Er}_{old}$) : retourne dans \bar{R} toutes les routes de \bar{R}_{old} et de \bar{Er}_{old} qui sont de coût réduit, calculé avec λ , inférieur à la différence de coût entre UB et LB . Autrement, les routes sont mises dans un ensemble \bar{Er} .

La méthode ISEM commence par calculer les points E_1 et E_2 , respectivement c^1 et c^2 -extrêmes. Tant que la valeur courante de ε est strictement supérieure à $c^2(E_2)$, le processus suivant est itéré. L'optimum LB , une borne supérieure UB et λ une solution optimale duale sont obtenues. Si la borne supérieure est telle que $c^1(UB) > c^1(E_2)$, alors UB est remplacé par E_2 . Ensuite, nous comparons les variables duales λ et λ_{old} pour savoir si elles respectent les conditions de T_{valid} .

Algorithme 6 ISEM(G)**Entrée** : Graphe G représentant le BOVRP**Sortie** : Ensemble Y des points non dominés du BOVRP

```

 $E_1, E_2 \leftarrow \text{calcul\_extreme}()$ 
 $Y = \{E_1\} \cup \{E_2\}$ 
 $\varepsilon = c_{E_1}^2 - 1$ 
tant que  $\varepsilon > c^2(E_2)$  faire
   $LB, \lambda \leftarrow$  solutions optimales du PML2( $\varepsilon$ ) et du PDL2( $\varepsilon$ )
   $UB \leftarrow$  solution réalisable du PM2( $\varepsilon$ )
  si  $c^1(UB) \geq c^1(E_2)$  alors
     $UB = E_2$ 
  fin si
  si  $\text{dual\_different}(\lambda, \lambda_{old})$  ou  $|\overline{R}_{old}| > \text{MAX\_COL}$  alors
     $\overline{R}, \overline{Er}, \overline{Ec} \leftarrow \text{genroute}(\varepsilon, LB, UB, \lambda, \{0\})$ 
  sinon
     $\overline{R}, \overline{Er} \leftarrow \text{trier}(\lambda, LB, UB, \overline{R}_{old}, \overline{Er}_{old})$ 
    si  $c^1(UB) - c^1(LB) > \gamma_{old}$  alors
       $\overline{R}, \overline{Er}, \overline{Ec} \leftarrow \text{genroute}(\varepsilon, LB, UB, \lambda, \overline{Ec}_{old})$ 
    fin si
  fin si
   $O \leftarrow$  solution optimale du PMR( $\varepsilon, \overline{R}$ )
   $Y \leftarrow Y \cup \{O\}$ 
  si  $|\overline{R}| \geq \text{MAX\_COL}$  alors
     $I \leftarrow I \cup \{Y\}$ 
  fin si
   $\varepsilon = c_O^2 - 1$ 
   $\gamma_{old} = c^1(UB) - c^1(LB)$ ,  $\lambda_{old} \leftarrow \lambda$ ,  $\overline{R}_{old} \leftarrow \overline{R}$ ,  $\overline{Er}_{old} \leftarrow \overline{Er}$ ,
   $\overline{Ec}_{old} \leftarrow \overline{Ec}$ ,  $\overline{R} \leftarrow \emptyset$ ,  $\overline{Er} \leftarrow \emptyset$ ,  $\overline{Ec} \leftarrow \emptyset$ 
fin tant que
pour  $i \in I$  faire
   $\varepsilon = c^2(Y[i-1]) - 1$ 
   $N = (c^1(Y[i]) - 1, c^2(Y[i-1]) - 1)$ 
  pour  $w \in W$  faire
     $LB_w$  solution optimale du PML4( $w, c^1(N), c^2(N)$ )
     $\lambda_w \leftarrow$  solution optimale du PDL4( $w, c^1(N), c^2(N)$ )
  fin pour
   $\overline{R} \leftarrow \text{genroute\_multiple\_dual}(\varepsilon, LB_0, \dots, LB_{|W|}, N, \lambda_0, \dots, \lambda_{|W|})$ 
  tant que  $\varepsilon > c^2(Y[i])$  faire
     $O =$  solution optimale du PMR( $\varepsilon, \overline{R}$ )
     $Y \leftarrow Y \cup \{O\}$ 
     $\varepsilon = c_O^2 - 1$ 
  fin tant que
fin pour
retourne  $Y$ 

```

Si elles ne les respectent pas, alors les ensembles \bar{R} , \bar{Er} et \bar{Ec} sont générés avec un appel classique à *genroute* qui se limite à la génération de MAX_COL routes dans \bar{R} . Si elles respectent les conditions, alors les anciens ensembles de routes \bar{R}_{old} , \bar{Er}_{old} sont triés pour enlever les routes de coût réduit supérieur au gap courant et sont mis dans les nouveaux ensembles de routes \bar{R} et \bar{Er} . Deux cas peuvent alors se produire. D'une part, si le gap courant est plus petit que le gap de l'itération précédente γ_{old} , alors la technique T_{valid} est employée et il n'y a pas de générations de routes. D'autre part, si le gap courant est strictement plus grand que γ_{old} , alors \bar{Ec}_{old} est utilisé dans *genroute* pour ne pas recommencer une extension des labels depuis le dépôt. Une fois encore, \bar{R} est limité à la taille MAX_COL.

Dans chacun des cas, un ensemble \bar{R} est disponible pour l'itération courante et $\text{PMR}(\bar{R})$ est résolu pour obtenir O qui est ajouté dans Y . Si \bar{R} est plus grand que MAX_COL, alors il n'y a pas de preuves que O est un point non dominé. Ainsi, l'indice de O dans Y est ajouté dans l'ensemble d'indices I pour un traitement ultérieur. Enfin, ε est mis à $c^2(O) - 1$. À la fin du processus, certains points, référencés par I , n'ont pas été prouvés non dominés. Ainsi, la technique T_{mult_dual} est employée comme décrit dans la Section 4.3.5.

4.4.2 Résultats expérimentaux

Dans cette partie, nous comparons SEM et ISEM. Lors de ces expérimentations, les bornes supérieures des deux méthodes sont données dans un fichier en entrée des algorithmes et correspondent aux points non dominés s'ils sont connus, ou à des bornes supérieures valides.

Comme dans le chapitre précédent, les expériences sont conduites sur un CPU Xeon E5-2695 avec 2,30GHz de fréquence et sur un unique cœur. L'implémentation est faite en langage C++ et tous les problèmes linéaires et entiers sont résolus avec la version 7.1 de Gurobi. Le temps limite de résolution d'une instance est quatre heures. Lors de la présentation de tous les résultats, si une méthode n'a pas convergé durant le temps limite pour une instance, celle-ci sera présentée avec un tiret (-).

Les mêmes instances du BOVRPTW que dans le Chapitre 3 sont utilisées. Lors du lancement de chaque méthode, la taille des ng-ensembles pour la relaxation des ng-routes (Section 2.3.2.2) est telle que $|N_i| = 8$, $i \in V'$. Le paramètre MAX_COL (Section 4.3.5) est mis à 30000 pour les deux méthodes et $W = \{0; 0, 25; 0, 5; 0, 75; 1\}$ (Section 4.3.5).

Le tableau 4.4 présente le nombre d'instances fermées et les temps CPU moyens en secondes des deux méthodes pour chaque classe d'instances et pour toutes les instances qui ont été résolues en quatre heures par les deux méthodes. Les colonnes T_{valid} , T_{save} et T_{mult_dual} indiquent le pourcentage d'utilisation de chaque technique sur l'ensemble des points non dominés. Le détail des résultats pour chaque instance est donné dans l'Annexe A.2.

En terme d'instances fermées, ISEM est la meilleure car elle résout 124 instances sur les 168 avec 25 nœuds, contre 100 pour SEM. Cependant, l'instance *C104_RC1*, qui n'a

TABLE 4.4: Nombre d'instances fermées et moyenne des temps de résolution des méthodes selon la classe d'instances pour $|V| = 25$.

Classe d'instance	SEM		ISEM				
	Fermées	Temps	Fermées	Temps	T_{valid}	T_{save}	T_{mult_dual}
$R_$	39	874.3	46	218.6	27	32	4
$RC_$	31	631.7	35	222.3	36	20	13
$C_$	30	1281.5	43	1066.7	42	19	30
Toutes	100	917.6	124	468.2	35	26	15

pas été fermée par SEM avec T_{save} et T_{mult_dual} , est uniquement fermée par SEM. En moyenne, ISEM est la plus rapide ; mais ces résultats sont variables selon les instances. Les techniques T_{valid} , T_{save} et T_{mult_dual} sont respectivement utilisées sur 35%, 26% et 15% des points non dominés.

Pour les instances de type R_C et R_RC , 46 instances au total sont fermées par ISEM contre 39 pour SEM. Le temps moyen de résolution des instances fermées par les deux méthodes est amélioré de 600 secondes par ISEM. SEM bat la nouvelle méthode sur six instances dont deux seulement de plus de 10 secondes. Ces deux instances sont $R111_C1_25$ et $R205_RC1_25$.

Pour les instances de type RC_C et RC_R , 35 instances sont fermées par ISEM contre 31 pour SEM. Le temps moyen de résolution est divisé par trois pour ISEM, ce qui prouve son efficacité. Cependant, 16 instances sont résolues plus rapidement avec SEM ; mais toutes ces instances sont rapidement résolues par les deux méthodes (moins de 100 secondes) et avec moins de 20 secondes d'écart. Ceci s'explique car les temps de traitement supplémentaire des colonnes peuvent alourdir la résolution sur les instances "faciles".

Enfin, pour les instances de type C_C , C_R et C_RC , 43 instances sont fermées par ISEM contre 30 pour SEM. Cependant, sur les instances fermées par les deux méthodes, les moyennes des temps de résolution sont proches et SEM est plus efficace sur 15 instances, dont 8 avec plus de 100 secondes d'écart. Dans chacune de ces 8 instances, T_{mult_dual} est utilisée plusieurs fois (entre 2 et 38 fois) et requiert le calcul de plusieurs bornes inférieures supplémentaires (entre 5 et 79). Sur les instances du type C , les bornes inférieures sont plus lentes à obtenir. Cependant, comme vu en Section 4.3.5, cette technique permet de fermer de nombreuses instances et d'améliorer les performances de la plupart des instances. De plus, le paramètre MAX_COL est fixé avant la résolution et une valeur plus élevée pourrait permettre d'améliorer le temps de résolution en limitant le nombre d'appels à T_{mult_dual} .

4.4.3 Conclusion

Dans cette section, nous avons agrégé les améliorations prometteuses données en Section 4.3 en une seule méthode, ISEM, qui a ensuite été comparée à SEM, la méthode de référence. Les résultats montrent que toutes les techniques d'amélioration ont été activées et ont donc eu un impact dans les bonnes performances de la nouvelle méthode. Au total, 24 instances en plus sont fermées et les temps de résolution sont meilleurs avec ISEM.

4.5 Conclusion du chapitre

Ce chapitre a tout d'abord mis en évidence que la différence entre la borne inférieure et supérieure joue un rôle primordial dans l'efficacité des méthodes d'exploration lorsqu'elles sont combinées avec de la génération de colonnes. De plus, nous avons vu que les pistes traditionnelles d'amélioration de la borne inférieure pour les problèmes de tournées de véhicules ne sont pas forcément performantes dans un cadre bi-objectif.

Ensuite, ce chapitre a permis d'approfondir la méthode ε -contrainte et de proposer des améliorations inédites reposant sur les propriétés structurelles de la formulation ε -contrainte. Nous avons également développé des améliorations basées sur la génération de colonnes et l'énumération des colonnes dans le gap selon plusieurs coûts duaux.

Enfin, d'intensives expérimentations ont validé chaque technique unitairement, tout en proposant une analyse de leurs avantages et de leurs inconvénients. La méthode ε -contrainte par étapes présentée dans le Chapitre 3 a été complétée avec toutes les techniques performantes. La méthode finale permet de résoudre la plupart des instances à 25 clients dans un laps de temps de 4 heures.

Les méthodes présentées dans ce chapitre sont applicables à toute sorte de problèmes de tournées de véhicules bi-objectif. Les prochains chapitres vont permettre de valider cette approche en les utilisant pour résoudre des problèmes de profit collecté en chapitre 5, tout en découvrant les limites d'application des techniques sur des problèmes de type min-max en chapitre 6.

Application au problème de la course d'orientation par équipes bi-objectif avec fenêtre de temps

Sommaire

5.1	Introduction	95
5.2	État de l'art	96
5.2.1	Problème de la course d'orientation	97
5.2.2	Problème de la course d'orientation par équipes	97
5.2.3	Problème de la course d'orientation par équipes bi-objectif	99
5.3	Méthodes exactes pour le problème de la course d'orientation par équipes bi-objectif	99
5.3.1	Formulations mathématiques	99
5.3.2	Adaptation de SEM	101
5.3.3	Adaptation de ISEM	102
5.3.4	Conclusion	102
5.4	Résolution de la formulation ε-contrainte	102
5.4.1	Algorithme d'étiquetage	102
5.4.2	Règles de dominance	103
5.4.3	Bornes de complétion	103
5.5	Résultats expérimentaux	104
5.5.1	Conclusion	106
5.6	Conclusion du chapitre	106

5.1 Introduction

Le problème de la course d'orientation par équipes (*Team-Orienteering Problem* = TOP) a été introduit par Butt et Cavalier (1994) en tant que *Multiple Tour Maximum Collection Problem* (MTMCP). Chao et al. (1996) ont introduit le terme de TOP pour la première fois pour désigner un problème de tournées de véhicules où tous les nœuds du graphe ne peuvent pas être visités à cause de restrictions sur les tournées (fenêtres de temps ou capacité) et où le but est de collecter un profit maximal sur les nœuds visités.

Le problème de la course d'orientation par équipes bi-objectif avec fenêtre de temps (*Bi-Objective Team-Orienteering Problem with Time Windows* = BOTOPTW) est défini sur un graphe $G = (V, E)$ non orienté. L'ensemble des sommets du graphe $V = \{0, \dots, n\}$ contient le dépôt 0 et n clients i , $1 \leq i \leq n$. Soit V' l'ensemble des clients tel que $V = V' \cup \{0\}$. Chaque client i a une fenêtre de temps $[a_i; b_i]$, un temps de service s_i et un profit p_i . L'ensemble des arêtes E est défini comme suit $E = \{(i, j) | i, j \in V, i < j\}$ et un temps de trajet t_{ij} est associé à chaque arête $(i, j) \in E$. Un ensemble K de véhicules homogènes est disponible au dépôt. Chaque véhicule effectue une route qui part du dépôt, visite un ensemble de clients et retourne au dépôt. Pour qu'une route serve un client, il faut que le temps de trajet du véhicule arrivant au client soit situé dans sa fenêtre de temps. Si le véhicule arrive avant le début de la fenêtre de temps, il faut qu'il attende ce début. Pour servir un client et collecter son profit, un véhicule doit rester au client i durant le temps de service s_i . Le BOTOPTW vise à minimiser la somme du temps de trajet de tous les véhicules sans le temps d'attente aux différents clients, ainsi que maximiser le profit collecté en visitant les clients.

Tout d'abord, ce chapitre propose un état de l'art du problème d'orientation. Ensuite, la Section 5.3 présente une formulation bi-objectif et la formulation ε -contrainte pour le problème étudié, ainsi que l'application des méthodes ε -contrainte par étapes et améliorée. La Section 5.4 montre les spécificités de la génération de colonnes pour résoudre la formulation ε -contrainte. Les résultats expérimentaux des méthodes bi-objectif sont présentés en Section 5.5 et démontrent l'efficacité des méthodes proposées contre une méthode de l'état de l'art. Enfin, la Section 5.6 résume les conclusions et les perspectives de ce chapitre.

Contributions

Ce chapitre met en exergue la généralité de la méthode ε -contrainte améliorée, ainsi que son efficacité non seulement contre la méthode ε -contrainte de référence, mais aussi contre une méthode exacte bi-objectif de l'état de l'art pour le BOTOPTW.

Publication

La méthode exacte pour résoudre le BOTOPTW est celle définie au Chapitre 4 et les résultats expérimentaux de cette section sont ajoutés à l'article soumis à *Computers & Operations Research* (C&OR) et intitulé '*An ε -constraint column generation-and-enumeration algorithm for bi-objective vehicle routing problems*'. Ils ont également été présentés lors de la conférence internationale sans actes VEROLOG 2019.

5.2 État de l'art

Les problèmes d'orientation doivent leur nom à leur application originelle : un sport où une équipe doit se déplacer d'une origine à une destination en un minimum de temps et en collectant un score en passant à certains endroits (Tsiligirides (1984)). Cependant, ils peuvent s'appliquer à d'autres domaines comme la logistique urbaine ou le tourisme. Pour la logistique urbaine, Bouly et al. (2008) et Tang et Miller-Hooks (2005) ont utilisé le TOP pour planifier des tâches d'inspection et de réparation. Chaque tâche a un profit

correspondant à l'urgence de son exécution et la limitation en matériel et main d'œuvre disponibles devient un enjeu crucial. Dans le domaine du tourisme, Souffriau et al. (2008), Vansteenwegen et al. (2011) et Gavalas et al. (2014) planifient des séjours personnalisés en fonction de l'intérêt des touristes pour les lieux proposés et de la durée du séjour.

La littérature sur les méthodes exactes pour les problèmes d'orientation est divisée en trois parties : le problème de la course d'orientation mono-objectif (*Orienteering Problem* = OP) en Section 5.2.1, le TOP (*Team-Orienteering Problem* = TOP) en Section 5.2.2 et le BOTOP en Section 5.2.3.

5.2.1 Problème de la course d'orientation

Le problème de la course d'orientation mono-objectif (*Orienteering Problem* = OP) a été introduit en 1984 par Tsiligirides (1984) pour modéliser un sport où un compétiteur doit visiter des points de contrôle ayant des scores alloués pour collecter le score maximal possible dans un temps limite. Ce problème a été prouvé NP-complet par Laporte et Martello (1990).

La première méthode exacte a été conçue par Ramesh et al. (1992) qui utilise la relaxation lagrangienne dans une approche de séparations et évaluations (*Branch-and-Bound*). Les résultats expérimentaux montrent que la méthode résout des instances contenant jusqu'à 150 points de contrôle.

Plusieurs autres méthodes exactes ont été développées comme la méthode de séparations et coupes (*Branch-and-Cut* = B&C) de Fischetti et al. (1998) qui a permis de résoudre des instances allant jusqu'à 500 nœuds dans le graphe en quelques heures.

Gendreau et al. (1998) ont également conçu un B&C pour une variante de l'OP où certains nœuds doivent être obligatoirement visités : le voyageur de commerce sélectif (*Selective Traveling Salesman Problem*). Des instances de 300 nœuds ont pu être résolues en quelques heures.

Enfin, Righini et Salani (2009) ont proposé de résoudre l'OP avec des fenêtres de temps à chaque point de contrôle. Pour cela, ils ont utilisé différentes stratégies de guidage pour la *Decremental State-Space Relaxation* (DSSR) combinée à de la programmation dynamique.

Une étude sur ce problème a été conduite par Vansteenwegen et al. (2011) et plus récemment par Gunawan et al. (2016).

5.2.2 Problème de la course d'orientation par équipes

Le problème de la course d'orientation par équipes (*Team-Orienteering Problem* = TOP) est une variante de l'OP, introduite en 1996 par Chao et al. (1996), où plusieurs équipes (ou véhicules) sont utilisées pour collecter le score (ou le profit) à chaque nœud du graphe. Plusieurs contraintes de tournées de véhicules peuvent être considérées comme la capacité du véhicule ou une distance maximale sur les routes. Les 387 instances introduites par Chao et al. sont utilisées dans tous les papiers de méthodes exactes de TOP.

La première méthode exacte pour résoudre ce problème est celle de Butt et Ryan (1999) pour une variante du TOP où les véhicules sont hétérogènes. Ils s'appuient sur une formulation d'ensemble couvrant (*set covering formulation*) résolue avec une méthode de sépa-

rations et génération de colonnes (*Branch-and-Price* = B&P). Ils ont fermé des instances allant jusqu'à 102 nœuds en moins de 12 heures.

Ensuite, Boussier et al. (2007) ont proposé un B&P générique pour les problèmes d'orientation. Ils ont validé l'efficacité de leur méthode sur les instances de Chao et al. (1996) et ont pu résoudre des instances de 102 nœuds et 4 véhicules en moins de deux heures.

Une variante du TOP proposant une capacité maximale par véhicule a été résolue par Archetti et al. (2009) avec une adaptation du B&P de Boussier et al. Trois métaheuristiques (recherche locale et recherche Tabou) ont également été proposées. Des instances de 51 à 200 nœuds résolues avec un temps limite d'une heure ont permis de mettre en avant l'efficacité de l'heuristique. La méthode exacte a également résolu à l'optimalité 60 instances sur 90.

Poggi et al. (2010) ont étendu le B&P de Boussier et al. en une méthode de séparations et coupes et génération de colonnes (*Branch-and-Cut-and-Price* = B&C&P). Plusieurs inégalités valides ont été développées comme le *min-cut* et les *triangle clique*. Les bornes supérieures des instances les plus grandes de Chao et al. ont été améliorées.

Une autre méthode de B&C a été développée par Dang et al. (2013) en ajoutant de nouvelles classes d'inégalités et des règles de dominance. Son efficacité a été prouvée en comparaison avec la méthode de Boussier et al. De plus, 29 nouvelles instances de Chao et al. ont été fermées dans un temps limite de deux heures.

Keshtkaran et al. (2016) ont proposé une méthode de B&P pour le TOP où les problèmes de *pricing* sont résolus par programmation dynamique avec un algorithme bi-directionnel utilisant la technique de DSSR et des règles de dominance en deux phases. Ils ont également développé une méthode de B&C&P où les inégalités *subset-row* sont utilisées en tant que coupes. Les résultats expérimentaux mettent en avant l'efficacité de la méthode puisque 17 nouvelles instances de Chao et al. sont fermées en deux heures comparé aux méthodes de Boussier et al. (2007) et de Dang et al. (2013).

El-Hajj et al. (2016) ont formulé le TOP avec un nombre polynomial de variables. En premier lieu, les contraintes d'élimination de sous-tours sont relâchées. Elles sont alors progressivement renforcées grâce à l'ajout successif de coupes. D'autres inégalités valides introduites par Dang et al. (2013) sont utilisées. Avec cette méthode, 12 nouvelles instances de Chao et al. ont pu être fermées en moins de deux heures.

Enfin, Park et al. (2017) ont étendu le B&P de Boussier et al. pour pouvoir résoudre des instances du TOP avec capacité et fenêtres de temps (*Capacitated Team-Orienteering Problem with Time Windows* = CTOPTW). Les instances de Solomon¹ et de Cordeau² ont été modifiées pour pouvoir être appliquées au CTOPTW avec deux véhicules. Toutes les instances de Solomon avec 100 nœuds ont été résolues en deux heures. Seule une instance de Cordeau modifiée avec 48 nœuds a été résolue à l'optimalité. Ceci s'explique car les fenêtres de temps sont plus larges dans les instances de Cordeau et sont moins faciles à résoudre avec une approche de type génération de colonnes.

1. Instances de Solomon disponibles sur <https://www.sintef.no/nearp>

2. Instances de Cordeau disponibles sur <https://neo.lcc.uma.es/vrp/vrp-instances/description-for-files-of-cordeaus-instances/>

5.2.3 Problème de la course d'orientation par équipes bi-objectif

En matière de méthode exacte, le BOTOP a été résolu une seule fois par Parragh et Tricoire (2019) en 2019. Ils ont utilisé une méthode de B&P générique pour les problèmes d'optimisation en nombre entiers (cf. Section 1.3.4). Leur méthode est testée sur plusieurs problèmes dont le BOTOPTW. De ce fait, nos résultats sont comparés à ceux de Parragh et Tricoire sur les mêmes instances.

5.3 Méthodes exactes pour le problème de la course d'orientation par équipes bi-objectif

Dans cette section, la formulation du BOTOPTW est introduite, ainsi que les formulations ε -contrainte associées. Ensuite, les méthodes ε -contrainte par étapes et améliorée sont appliquées au BOTOPTW. Certaines spécificités comme le calcul des points extrêmes ou la technique des multiples coûts duaux sont décrites dans les deux dernières parties.

5.3.1 Formulations mathématiques

Formulation bi-objectif. Pour résoudre le BOTOPTW, une formulation de partitionnement (*set partitioning formulation*) est introduite. Soit R l'ensemble des routes réalisables en matière d'élémentarité et de fenêtres de temps. Soit a_{ir} un coefficient binaire indiquant si le client $i \in V'$ est visité par la route $r \in R$. Soit c_r le coût total de la route r et p_r le profit total collecté par r . Les variables binaires x_r de la formulation indiquent si la route r est sélectionnée dans la solution ($x_r = 1$) ou non ($x_r = 0$). Le BOTOPTW peut être formulé comme suit :

$$\left\{ \begin{array}{ll} \min c(x) = \sum_{r \in R} c_r x_r & (5.1.1) \\ \max p(x) = \sum_{r \in R} p_r x_r & (5.1.2) \\ t.q. \sum_{r \in R} a_{ir} x_r \leq 1 & i \in V' \quad (5.1.3) \\ \sum_{r \in R} x_r \leq |K| & (5.1.4) \\ x_r \in \{0, 1\} & r \in R \quad (5.1.5) \end{array} \right. \quad (5.1)$$

Cette formulation a deux objectifs (5.1.1) et (5.1.2) qui minimisent la somme des coûts des routes et qui maximisent le profit collecté par tous les véhicules. Les contraintes (5.1.3) empêchent de visiter un client plus d'une fois et la contrainte (5.1.4) limite le nombre de véhicules. Enfin, le domaine de définition des variables est défini par la contrainte (5.1.5).

Pour transformer le problème bi-objectif en séquences de problèmes mono-objectif, nous utilisons les formulations pour la méthode ε -contrainte décrite ci-dessous. Pour le BOTOPTW, chacun des objectifs peut être transformé en contrainte, ainsi deux

formulations sont définies.

Formulation ε -contrainte minimisant le coût. Dans cette formulation, une constante ε est introduite pour borner le profit collecté minimum. Ainsi, la formulation est équivalente au *Prize-Collecting Vehicle Routing Problem*.

$$\left\{ \begin{array}{ll} \min c(x) = \sum_{r \in R} c_r x_r & (5.2.1) \\ t.q. \sum_{r \in R} a_{ir} x_r \leq 1 & i \in V' \quad (5.2.2) \\ \sum_{r \in R} x_r \leq |K| & (5.2.3) \\ \sum_{r \in R} p_r x_r \geq \varepsilon & (5.2.4) \\ x_r \in \{0, 1\} & r \in R \quad (5.2.5) \end{array} \right. \quad (5.2)$$

L'objectif (5.2.1) minimise le coût c . Les contraintes (5.2.2), (5.2.3) et (5.2.5) sont identiques aux contraintes (5.1.3), (5.1.4) et (5.1.5). La nouvelle contrainte (5.2.4) oblige le profit total collecté p à être supérieur à la valeur ε . Nous pouvons remarquer que lorsque ε est inférieur au plus petit profit des clients du graphe, la résolution converge vers le point c -extrême. Pour rappel, il s'agit du point ayant le coût c le plus faible.

Soit $\lambda_i \leq 0$ ($i \in V'$), $\lambda_0 \leq 0$, $\alpha \geq 0$ les variables duales respectivement associées aux contraintes (5.2.2), (5.2.3) et (5.2.4), alors le dual de la relaxation linéaire de la formulation (5.2) est la formulation (5.3).

$$\left\{ \begin{array}{ll} \max \sum_{i \in V'} \lambda_i + |K| \lambda_0 + \varepsilon \alpha & \\ t.q. \sum_{i \in V'} a_{ir} \lambda_i + \lambda_0 + p_r \alpha \leq c_r & r \in R \\ \lambda_i \leq 0 & i \in V' \\ \lambda_0 \leq 0 & \\ \alpha \geq 0 & \end{array} \right. \quad (5.3)$$

Formulation ε -contrainte maximisant le profit. Dans cette formulation, une constante ε est introduite pour borner le coût total maximum. Ainsi, la formulation est

équivalente au TOP.

$$\left\{ \begin{array}{l} \max p(x) = \sum_{r \in R} p_r x_r \quad (5.4.1) \\ t.q. \sum_{r \in R} a_{ir} x_r \leq 1 \quad i \in V' \quad (5.4.2) \\ \sum_{r \in R} x_r \leq |K| \quad (5.4.3) \\ \sum_{r \in R} c_r x_r \leq \varepsilon \quad (5.4.4) \\ x_r \in \{0, 1\} \quad r \in R \quad (5.4.5) \end{array} \right. \quad (5.4)$$

L'objectif (5.4.1) maximise le profit collecté p . Les contraintes (5.4.2), (5.4.3) et (5.4.5) sont identiques aux contraintes (5.1.3), (5.1.4) et (5.1.5). La nouvelle contrainte (5.4.4) borne le coût total d'une solution par ε . Nous pouvons remarquer que lorsque $\varepsilon = +\infty$, la résolution converge vers le point p -extrême. Pour rappel, il s'agit du point ayant le profit p le plus élevé.

Soit $\lambda_i \leq 0$ ($i \in V'$), $\lambda_0 \leq 0$, $\alpha \leq 0$ les variables duales respectivement associées aux contraintes (5.4.2), (5.4.3) et (5.4.4), alors le dual de la relaxation linéaire de la formulation (5.4) est la formulation (5.5).

$$\left\{ \begin{array}{l} \max \sum_{i \in V'} \lambda_i + |K| \lambda_0 + \varepsilon \alpha \\ t.q. \sum_{i \in V'} a_{ir} \lambda_i + \lambda_0 + c_r \alpha \leq -p_r \quad r \in R \\ \lambda_i \leq 0 \quad i \in V' \\ \lambda_0 \leq 0 \\ \alpha \leq 0 \end{array} \right. \quad (5.5)$$

5.3.2 Adaptation de SEM

La méthode ε -contrainte par étapes (SEM) est définie dans le Chapitre 3 en Section 3.4.3 pour le BOVRP et utilise la génération de colonnes combinée avec l'approche ε -contrainte. Elle se base sur la formulation (5.2) où la valeur ε est initialement fixée à 1 pour obtenir le point c -extrême et est augmentée à chaque itération jusqu'à dépasser le profit du point p -extrême.

Le point p -extrême est obtenu en résolvant la formulation (5.4) pour $\varepsilon = +\infty$. Soit LB_p la solution optimale de la relaxation linéaire de la formulation (5.4) pour $\varepsilon = +\infty$. Si LB_p a un profit égal à la somme des profits de tous les clients du graphe, ce n'est pas performant d'obtenir la solution optimale entière avec la génération de colonnes. En effet, dans ce cas-là, le dual associé à chaque client λ_i , $i \in V'$, est égal au profit de ce client et λ_0 et α sont égaux à zéro. Toutes les routes réalisables ont alors un coût réduit de zéro et sont générées durant l'énumération des colonnes. Pour éviter ce problème lorsque LB_p a un profit égal à la somme des profits de tous les clients, nous résolvons un VRP classique.

De plus, lorsque le point p -extrême a un profit égal à la somme de tous les profits des

clients du graphe, nous savons que le point non dominé adjacent a un profit égal à la somme de tous les profits des clients du graphe moins le plus petit profit des clients. Lorsque ce point est trouvé, la méthode peut s'arrêter.

5.3.3 Adaptation de ISEM

Toutes les techniques présentées dans le Chapitre 4 en Section 4.3 sont applicables à la formulation ε -contrainte pour le BOTOPTW. Lors de la technique des multiples coûts duaux T_{mult_dual} , les deux formulations (5.2) et (5.4) sont résolues avec la génération de colonne en parallèle. De plus, la technique de construction de la borne inférieure, T_{lb} , est performante pour le problème étudié et elle est donc rajoutée à la méthode ISEM.

5.3.4 Conclusion

Cette section présente l'application de SEM et ISEM au BOTOPTW. Cela montre que très peu de modifications sont nécessaires dans le cadre bi-objectif et ces méthodes sont donc génériques aux problèmes de tournées de véhicules. Toutefois, quelques adaptations doivent être faites dans la méthode de résolution mono-objectif et c'est le sujet de la section suivante.

5.4 Résolution de la formulation ε -contrainte

La méthode exacte pour résoudre la formulation ε -contrainte (5.2) est décrite en Section 3.3. Elle est composée de quatre étapes : 1) le calcul d'une borne inférieure 2) le calcul d'une borne supérieure 3) la génération de toutes les routes ayant un coût réduit entre les deux bornes 4) la résolution de la formulation sur cet ensemble de routes. La borne inférieure est calculée par génération de colonnes et quelques spécificités du BOTOPTW sont abordées dans cette section.

Le sous-problème de *pricing* des formulations étudiées sont des ESPPRC qui sont résolus à l'aide d'un algorithme d'étiquetage défini en Section 5.4.1. Ensuite, les règles de dominance, les bornes de complétion et une heuristique pour améliorer l'efficacité de l'algorithme sont présentées en Sections 5.4.2 et 5.4.3.

5.4.1 Algorithme d'étiquetage

Pour cet algorithme d'étiquetage, une étiquette E est un ng-chemin réalisable qui commence au dépôt, visite un ensemble de client C_E et termine au point de service v_E avec un ng-ensemble NG_E , un temps de trajet t_E , un coût c_E , un profit p_E et un coût réduit \bar{c}_E . Soit le coût réduit \bar{c}_{ij} d'une arête $(i, j) \in E$:

$$\bar{c}_{ij} = c_{ij} - \frac{(\lambda_i + \lambda_j)}{2} - \alpha \frac{(p_i + p_j)}{2} \quad \forall i, j \in V \quad (5.6)$$

L'extension d'une étiquette E de son nœud terminal v_E à un client $i \in V \setminus NG_E$ produit une nouvelle étiquette E' via les règles (5.7)-(5.11) :

$$v_{E'} = i \quad (5.7)$$

$$t_{E'} = \max(t_E + c_{ij}, a_i) + s_i \quad (5.8)$$

$$p_{E'} = p_E + p_i \quad (5.9)$$

$$\overline{c_{E'}} = \overline{c_E} + \overline{c_{v_E i}} \quad (5.10)$$

$$NG_{E'} = (NG_E \cap N_i) \cup \{i\} \quad (5.11)$$

où N_i , $i \in V$, est le ng-ensemble pré-défini pour le client i , a_i est le début de la fenêtre de temps de i et s_i le temps de service de i .

Les règles de propagation (5.7), (5.10) et (5.11) sont utilisées dans beaucoup de problèmes de tournées de véhicules. Au contraire, la règle (5.8) est utilisée pour les problèmes avec fenêtres de temps et met à jour le temps de trajet. De plus, la règle (5.9) est spécifique aux problèmes avec collecte de profit et met à jour le profit.

5.4.2 Règles de dominance

Les règles de dominance (5.12)-(5.16) sont utilisées. Une étiquette E^1 domine une étiquette E^2 différente de E^1 si les conditions suivantes sont satisfaites :

$$v_{E^1} = v_{E^2} \quad (5.12)$$

$$t_{E^1} \leq t_{E^2} \quad (5.13)$$

$$NG_{E^1} \subseteq NG_{E^2} \quad (5.14)$$

$$p_{E^1} \geq p_{E^2} \quad (5.15)$$

$$\overline{c_{E^1}} \leq \overline{c_{E^2}} \quad (5.16)$$

Les trois premières conditions assurent que tous les points de service $i \in O \cup M$ qui peuvent être atteints par propagation de l'étiquette E^2 peuvent également être atteints par propagation de l'étiquette E^1 . La règle (5.15) permet de ne pas éliminer de points non dominés respectant la contrainte bornant le profit à ε . Enfin, la condition (5.16) vérifie que les coûts réduits de l'étiquette E^2 et de toutes ses propagations sont supérieurs à ceux de E^1 et de ses propagations associées.

5.4.3 Bornes de complétion

L'application des bornes de complétion de la littérature sur le problème du TOP est équivalente au BOVRP de la Section 3.3.1.

5.5 Résultats expérimentaux

Dans cette section, les résultats des méthodes bi-objectif des Sections 5.3.2 et 5.3.3 pour le BOTOPTW sont présentés. Les expérimentations ont été conduites sur un processeur Xeon E5-2695 avec un CPU de 2.30GHz de fréquence d'horloge, sur un unique cœur. Le code a été implémenté en C++ et les problèmes linéaires et entiers sont résolus avec Gurobi 7.1.

La méthode la plus performante dans la littérature pour le BOTOPTW est présentée dans le papier de [Parragh et Tricoire \(2019\)](#); c'est pourquoi nous comparons nos méthodes avec la leur dans les tableaux de résultats suivants. Comme le CPU utilisé par Parragh et Tricoire est différent du nôtre, nous utilisons le *passmark CPU score*³ avec un seul cœur pour avoir une comparaison équitable. Ils utilisent un CPU Intel Xeon E5-2650v2 avec une fréquence d'horloge de 2.6GHz. La note du CPU utilisé par Parragh et Tricoire est 1704 et le nôtre est 1628. Ainsi, les temps CPU sont comparables.

Les instances du BOTOPTW sont celles utilisées par Parragh et Tricoire. Elles sont nommées $X-K-V$ où X est le nom de l'instance du VRP qui est utilisée, K est le nombre maximal de véhicules autorisés et V est le nombre de nœud dans le graphe. Au total, 80 instances sont testées. Les paramètres suivants ont été utilisés : $\Delta = 8$ (*cf.* Section 2.3.2.2) et $\text{MAX_COL} = 30000$ (*cf.* Section 4.3.5). Le temps limite est de deux heures pour toutes les méthodes. Si la méthode ne converge pas dans le temps limite, la ligne correspondante est notée avec un tiret (-). Les meilleures méthodes en terme de temps CPU sont notées en gras.

Le tableau 5.1 représente le nombre d'instances fermées par la méthode de [Parragh et Tricoire \(2019\)](#) ($B\&B$), SEM, SEM avec T_{valid} et ISEM. Le tableau 5.2 présente les caractéristiques des instances par classe avec le nombre moyen de points non dominés $|ND|$, le nombre moyen de points non supportés $|NS|$ et le gap moyen entre la borne inférieure et la solution optimale à chaque étape de SEM en pourcentage, pour chaque valeur ε . La formule du gap G pour une valeur ε fixée est :

$$G = 100 - \frac{c(LB)*100}{c(O)} (\%);$$

De plus, le tableau indique les moyennes des temps CPU en secondes pour chaque méthode. Pour SEM avec T_{valid} et ISEM, les colonnes T_{lb} , T_{valid} , T_{save} et T_{mult_dual} montrent le pourcentage moyen respectif d'utilisation de chaque technique par rapport au nombre de points non dominés. Le détail pour chaque instance est donné en Annexe B.

Au total, 69 instances sont fermées par la méthode de [Parragh et Tricoire \(2019\)](#) et 74 instances sont fermées par SEM, SEM avec T_{valid} et ISEM. Sur toutes les instances fermées, la moyenne des temps CPU de résolution en secondes est de 289.2, 160.2, 87.3 et 47.1 pour la méthode de Parragh et Tricoire, SEM, SEM avec T_{valid} et ISEM. Ainsi, ISEM est six fois

3. Données disponibles sur le site https://www.cpubenchmark.net/cpu_list.php

TABLE 5.1: Nombre d'instances fermées par des méthodes sur les différentes classes d'instances.

Classe	B&B	SEM	SEM + T_{valid}	ISEM	Total
c101	12	16	16	16	20
pr01	17	19	19	19	20
r101	20	20	20	20	20
rc101	20	20	20	20	20
Toutes	69	75	75	75	80

TABLE 5.2: Moyenne des résultats expérimentaux des méthodes sur les différentes classes d'instances.

Classe	Description			B&B	SEM	SEM + T_{valid}	ISEM					
	$ ND $	$ NS $	MGap	Temps	Temps	Temps	T_{valid}	Temps	T_{lb}	T_{valid}	T_{save}	T_{mult_dual}
c101	26	13	20.2	759.2	97.8	51.3	63.1	97.2	60.1	61.0	3.1	4.8
pr01	78	9	10.9	491.3	520.7	269.8	78.1	77.7	83.4	77.0	8.3	0.2
r101	43	8	14.3	7.3	52.5	33.6	69.1	24.3	79.7	67.4	17.0	0.2
rc101	31	11	15.4	137.7	37.3	25.6	70.0	16.9	75.0	70.1	9.9	0.0
Toutes	44	10	14.9	289.2	160.9	87.3	70.3	47.1	75.6	69.2	10.5	1.0

plus performante que la méthode de l'état de l'art en moyenne. Cependant, cette conclusion dépend du type d'instances résolu.

Pour les 20 instances *c101*, SEM avec T_{valid} est la méthode la plus performante (51.3 secondes de temps de résolution en moyenne et 16 instances fermées), suivie de ISEM (97.2 secondes de temps de résolution en moyenne et 16 instances fermées) et SEM (97.8 secondes en moyenne et 16 instances fermées). Enfin, la méthode de branchement de Parragh et Tricoire est moins performante puisque seulement 12 instances sont fermées et la moyenne des temps de résolution est de 759.2 secondes, soit près de huit fois moins rapide que ISEM. Dans ce type d'instances, ISEM est moins performante que SEM avec T_{valid} car l'astuce de multiples coûts duaux n'est pas appropriée. En effet, sur les instances où cette technique est utilisée pour plus de 20% des points non dominés, ISEM est moins performante. Ceci s'explique car même si plus de 100000 routes sont générées pour obtenir des points non dominés, la génération de ces routes et la résolution des problèmes maîtres restreints sont rapides. Ainsi, le calcul de nouvelles bornes inférieures pour ces points rajoute du temps de traitement. Enfin, SEM avec T_{valid} est la méthode la plus performante sur les instances de type *c101*. Cette technique est utilisée sur plus de 60% des points non dominés et est très utile sur ce type d'instance car le temps de résolution est presque divisé par deux, comparé à SEM et ISEM.

Pour les 20 instances *pr01*, ISEM, SEM avec T_{valid} et SEM ferment 19 instances contre 17 pour la méthode de Parragh et Tricoire. Sur ces 17 instances, ISEM est la plus performante (77.7 secondes de temps de résolution en moyenne), suivie de SEM avec T_{valid}

(269.8 secondes), de la méthode de branchement (491.3 secondes) et de SEM (520.7 secondes). ISEM est moins performante que la méthode de branchement sur 5 instances, mais aucune n'a plus de 50 secondes de différence.

Les 20 instances *r101* sont fermées par toutes les méthodes en moins de 160 secondes. Il s'agit donc d'instances très faciles sur lesquelles la méthode de Parragh et Tricoire est plus de 3 fois plus performante que ISEM. La méthode de Parragh et Tricoire résout toutes les instances en moins de 40 secondes. Pour ISEM, l'instance la plus dure est fermée en 69 secondes. En moyenne, ISEM améliore le temps de résolution de SEM par deux sur ce jeu d'instances.

Enfin, les 20 instances *rc101* sont également fermées par toutes les méthodes en moins de 947 secondes. Contrairement aux instances *r101*, ISEM est la méthode la plus performante puisque le temps moyen de résolution est de 16.9 secondes contre 137.7 pour la méthode de Parragh et Tricoire, 25.6 pour SEM avec T_{valid} et 37.3 pour SEM. Les instances les plus difficiles de ce groupe (résolues en plus de 10 secondes par la méthode de branchement) sont toutes résolues plus rapidement par ISEM. Au contraire, les 5 instances les plus faciles sont résolues plus rapidement par la méthode de branchement, ce qui confirme les résultats obtenus pour les instances de type *r101* qui sont faciles à résoudre.

5.5.1 Conclusion

Les résultats expérimentaux ont mis en évidence l'efficacité de ISEM, basée sur la méthode ε -contrainte et la génération de colonnes, ainsi que sa généralité. En effet, cette méthode est en moyenne plus efficace que SEM et que la méthode de l'état de l'art. De plus, toutes les techniques sont employées durant la résolution car T_{lb} est employée pour 75.6% des points non dominés en moyenne, T_{valid} sur 70% des points, T_{save} sur 10% des points et T_{mult_dual} sur 1% des points non dominés.

5.6 Conclusion du chapitre

Dans ce chapitre, nous avons proposé une méthode exacte pour résoudre le BOTOPTW. Elle est basée sur la formulation de partitionnement avec une contrainte bornant l'un des deux objectifs et utilise la relaxation ng-route pour le problème de *pricing*. Nous décrivons toutes les spécificités de la génération de colonnes pour le BOTOPTW comme l'extension d'une étiquette, les règles de dominance et les bornes de complétion. L'application de la méthode ε -contrainte améliorée du Chapitre 4 est directe pour le problème étudié, ce qui souligne sa généralité. Enfin, l'efficacité de la nouvelle méthode ε -contrainte est démontrée par d'intensifs résultats expérimentaux comparés avec la méthode ε -contrainte de référence et une méthode de l'état de l'art pour le BOTOPTW.

Application au problème de tournées couvrantes

Sommaire

6.1	Introduction	107
6.2	État de l'art	110
6.2.1	Problème de la tournée couvrante	110
6.2.2	Problème de tournées couvrantes	111
6.2.3	Problème de la tournée couvrante bi-objectif	112
6.2.4	Problème de tournées couvrantes bi-objectif	113
6.3	Méthode exacte pour le problème de tournées couvrantes	114
6.3.1	Formulation mathématique	114
6.3.2	Étape 1 : Calcul d'une borne inférieure LB	116
6.3.3	Étape 2 : Calcul d'une borne supérieure UB	121
6.3.4	Étape 3 : Énumération des colonnes dans le gap	122
6.3.5	Étape 4 : Calcul de la solution optimale entière	122
6.3.6	Détails d'implémentation	122
6.4	Extension au problème de tournées couvrantes bi-objectif	123
6.4.1	Formulations mathématiques	123
6.4.2	Adaptation de SEM	124
6.4.3	Adaptation de ISEM	125
6.5	Résultats expérimentaux	125
6.5.1	Problème de tournées couvrantes	125
6.5.2	Problème de tournées couvrantes bi-objectif	128
6.5.3	Conclusion	130
6.6	Conclusion du chapitre	130

6.1 Introduction

Le problème de la tournée couvrante (*Covering Tour Problem* = CTP) a été introduit par [Current et Schilling \(1989\)](#) comme une généralisation du problème du voyageur de commerce (*Traveling Salesman Problem* = TSP) où le voyageur ne doit pas passer dans toutes les villes du réseau. Cependant, toutes les villes non visitées par le voyageur doivent

être situées à moins d'une certaine distance pré-déterminée d'une ville visitée. Dans le problème de tournées couvrantes (*Multi-Vehicle Covering Tour Problem* = MCTP), plusieurs véhicules sont disponibles au dépôt pour servir les points subsidiaires. Le MCTP vise donc à trouver un ensemble de routes de coût minimum dans un réseau de telle sorte que les critères de chaque nœud soient remplis. Certains nœuds doivent être visités par les véhicules, alors que d'autres doivent être couverts. On dit qu'un nœud est couvert s'il est situé à moins d'une certaine distance pré-déterminée d'un nœud visité.

Le MCTP est défini sur un graphe $G = (V, E)$ non orienté. L'ensemble des sommets du graphe V est défini par $V = \{0\} \cup M \cup O \cup C$, où 0 est le dépôt, M et O sont respectivement les ensembles de points de services obligatoires et optionnels (i.e. nœuds à visiter) et C est l'ensemble des clients (i.e. nœuds à couvrir). L'ensemble des arêtes est $E = \{(i, j) | i, j \in V, i < j\}$. Une distance d_{ij} est associée à chaque arête $(i, j) \in E$. Un ensemble K de véhicules homogènes est disponible au dépôt. Chaque véhicule effectue une route qui part du dépôt, visite un ensemble de points de service et retourne au dépôt. Tous les points de services obligatoires doivent être visités, alors que les points de service optionnels peuvent être visités, si besoin. Une visite à un point de service optionnel $i \in O$ permet de couvrir tous les clients $j \in C$ tels que $d_{ij} \leq \eta$, où η est la distance de couverture maximale. Si une route r visite un point de service $i \in O$ qui couvre le client $j \in C$, on dit que la route r couvre le client j . Tous les clients du graphe doivent être couverts par au moins une route. Les routes ne peuvent pas visiter plus de \bar{n} points de service et la somme des distances des arêtes traversées par une route ne peut pas excéder la distance maximale \bar{d} . Le but du MCTP est de trouver un ensemble de routes qui minimise la distance des routes, visite les points de service obligatoires et couvre tous les clients du graphe. Le MCTP est un problème \mathcal{NP} -difficile puisqu'il peut être réduit au *Distance Constrained Vehicle Routing Problem* (Toth et Tramontani (2008)) si $V = M$ et les contraintes de cardinalité sont relâchées, ou au *Vehicle Routing Problem with unit demands* (Godinho et al. (2008)) si $V = M$ et les contraintes de distance maximale sont relâchées.

La Figure 6.1 illustre un exemple de graphe du MCTP (à gauche) et une solution réalisable correspondante (à droite). Le dépôt est le nœud 0 , les points de service sont $\{1, 2, 3, 4\}$ et les clients sont $\{a, b, c, d\}$. Une flèche bleue en pointillé relie un point de service optionnel $i \in O$ et un client $j \in C$ si j est couvert par i (si i est visité). Sur le graphe de droite, les flèches noires représentent une solution composée de deux routes $0 \rightarrow 1 \rightarrow 0$ et $0 \rightarrow 3 \rightarrow 4 \rightarrow 0$. Cette solution est réalisable car le point de service obligatoire 4 est visité et tous les clients sont couverts par les routes.

Le problème de tournées couvrantes bi-objectif (*Bi-Objective Multi-vehicle Covering Tour Problem* = BOMCTP) que nous considérons dans ce chapitre vise à minimiser la distance parcourue par les véhicules en premier objectif et à minimiser la distance maximale de couverture η comme second critère. Toutes les autres données en entrée restent similaires au MCTP. Ces deux objectifs sont contradictoires car plus la valeur de η augmente, moins de points de service devront être visités et, par conséquent, plus la distance parcourue par les véhicules sera petite.

Tout d'abord, ce chapitre propose un état de l'art des problèmes étudiés. Ensuite, la

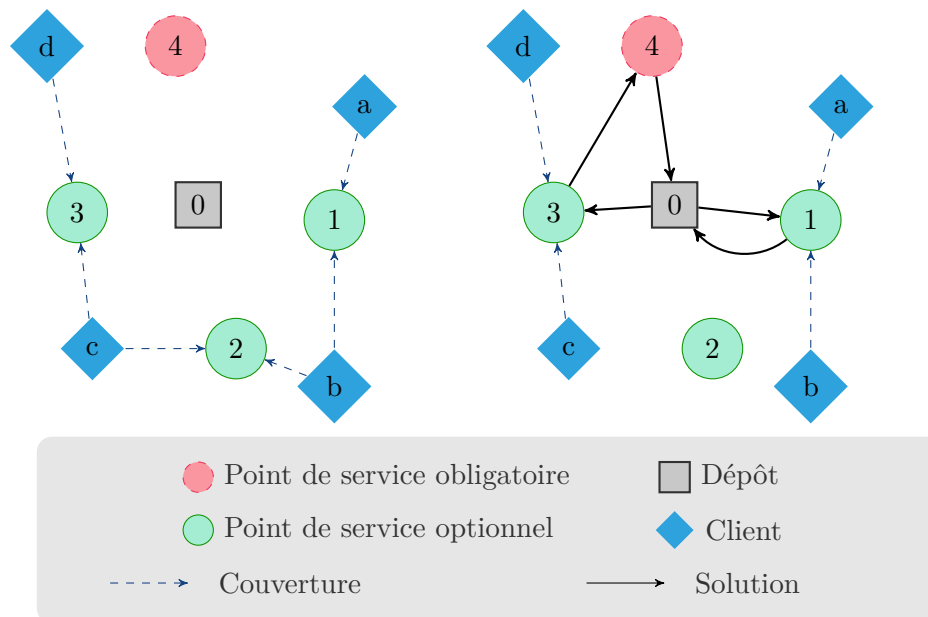


FIGURE 6.1: Exemple du MCTP avec un point de service obligatoire, trois points de services optionnels et quatre clients : le graphe correspondant sur la gauche et une solution réalisable sur la droite.

Section 6.3 présente une formulation mathématique du MCTP, une méthode exacte pour la résoudre basée sur l'énumération de colonnes. La Section 6.4 explique comment résoudre le BOMCTP en combinant cette méthode mono-objectif avec l'approche ε -contrainte. La Section 6.5 montre les résultats expérimentaux des méthodes proposées pour le MCTP et le BOMCTP. Enfin, la Section 6.6 résume les conclusions de ce chapitre.

Contributions

Ce chapitre vise à résoudre d'une part le MCTP comme étudié dans les papiers de Hachicha et al. (2000), Lopes et al. (2013), Jozefowicz (2014), Ha et al. (2013), Murakami (2014) et Kammoun et al. (2017) et d'autre part le BOMCTP introduit par Artigues et al. (2018). Une nouvelle méthode est proposée et une comparaison avec les méthodes de l'état de l'art prouve son efficacité sur le problème mono-objectif. Cette méthode est ensuite utilisée avec succès pour résoudre la variante bi-objectif du problème.

Publication

Les méthodes de résolution du MCTP et ε -contrainte de référence pour le BOMCTP sont présentées dans un article de la revue *European Journal of Operations Research* (EJOR) intitulé '*Exact Methods for Mono-Objective and Bi-Objective Multi-Vehicle Covering Tour Problems*'. Cette publication a été faite en collaboration avec Roberto Roberti suite à une mobilité à l'Université Libre d'Amsterdam du département *Supply Chain Ana-*

lytics.

6.2 État de l'art

Current et Schilling (1994) considèrent que le CTP permet de concevoir des réseaux de transport à deux niveaux. Un véhicule principal effectue un tour de distance minimale pour servir des points subsidiaires tel que tous les points n'appartenant pas à ce tour peuvent atteindre un point du tour en un temps raisonnable.

Les problèmes de tournées couvrantes ont des applications réelles dans les domaines de l'humanitaire, du développement urbain et de la santé. Pour la logistique humanitaire, Naji-Azimi et al. (2012) et Nolz et al. (2010) ont étudié la distribution des biens de première nécessité dans des zones sinistrées. L'équipe de secours visite quelques centres de distribution et la population se sert dans ces centres s'ils sont assez proches. Hodgson et al. (1998) et Hachicha et al. (2000) se sont intéressés à la planification de routes pour une équipe médicale dans une zone rurale, visitant un nombre limité de villages telles que chaque habitant de la région est capable d'atteindre un centre de soins dans un temps acceptable. Dans le développement urbain, Current et Schilling (1994) et Labbé et Laporte (1986) déterminent les emplacements pour des services postaux nocturnes et des boîtes postales accessibles par l'ensemble de la population. Oliveira et al. (2015) planifient des itinéraires de patrouilles qui doivent visiter des sites critiques, comme les écoles et les hôpitaux, et passer à proximité d'autres endroits vulnérables, comme les parcs publics et les banques.

Pour des raisons de clarté, la littérature sur les problèmes de couverture est divisée en quatre parties : le CTP (Section 6.2.1), le MCTP (Section 6.2.2), le BOCTP (Section 6.2.3) et le BOMCTP (Section 6.2.4).

6.2.1 Problème de la tournée couvrante

Current et Schilling (1989) ont formulé le problème du voyageur de commerce avec couverture (*Covering Salesman Problem* = CSP) avec une formulation à deux indices. Le CSP diffère du CTP car les clients sont des points de service optionnels qui peuvent être visités et qu'il n'y a pas de points de service obligatoires. Ils ont conçu une heuristique qui résout le problème de l'ensemble couvrant (*Set Covering Problem* = SCP) sur le graphe initial et un problème du voyageur de commerce sur chaque graphe résultant de la solution optimale du SCP. Leur heuristique est testée sur des instances de 21 nœuds et 39 arêtes.

Gendreau et al. (1997) ont proposé des inégalités valides, ainsi qu'une méthode de séparations et coupes (*Branch-and-Cut* = B&C) pour résoudre la formulation à deux indices du CTP. Ils ont également introduit une heuristique pour trouver une borne supérieure à la racine de l'arbre de décision qui fournit généralement une solution à 3% de l'optimalité. La méthode exacte est testée sur des instances générées aléatoirement allant jusqu'à 600 nœuds et 100 points de service. Les instances ont été résolues en moins de deux heures.

Baldacci et al. (2005) ont formulé le CTP avec une formulation de flux de deux marchandises (*two-commodity flow* formulation). Ils ont comparé leur nouvelle formulation avec celle de Gendreau et al. et ont montré que la borne inférieure trouvée est plus resserrée. Une

métaheuristique évolutionnaire est développée et combinée à des contraintes d'élimination de sous-tours généralisés. Pour tester leur méthode, ils ont créé des instances aléatoires avec les mêmes paramètres que Gendreau et al. et elles ont été résolues en moins de 20 minutes.

6.2.2 Problème de tournées couvrantes

Le MCTP a été introduit par Hachicha et al. (2000) et formulé avec trois indices. Ils ont appliqué trois heuristiques différentes pour le MCTP : l'heuristique *savings* de Clarke et Wright (1964), l'algorithme *sweep* de Gillett et Miller (1974) et l'heuristique route-premier/cluster-second de Beasley (1983). Les performances de chaque algorithme ont été établies sur des instances générées aléatoirement ayant jusqu'à 200 points de service et 400 clients. Elles ont été résolues en moins de 30 minutes pour l'heuristique *sweep* et en moins de quatre minutes par les deux autres. Des instances réelles ont également été traitées en quelques secondes par l'algorithme de Beasley et confirme que cette méthode est le meilleur compromis entre le temps de résolution et la qualité de la solution.

Lopes et al. (2013) ont présenté des résultats préliminaires d'une méthode de séparations et génération de colonnes (*Branch-and-Price* = B&P) pour le MCTP basé sur une formulation de partitionnement. La résolution des sous-problèmes est accélérée avec une meta-heuristique GRASP. Ils ont testé leur méthode sur des instances aléatoires générées comme celles de Hachicha et al. où la distance maximale \bar{d} est égale à 200 et la cardinalité maximum d'une route \bar{n} vaut 4. Seulement 5 instances sur 15 sont résolues en moins de 4 heures sur des instances de 400 nœuds et de 200 points de service.

Jozefowicz (2014) a présenté une formulation de partitionnement pour le MCTP avec moins de contraintes que celle de Lopes et al. (2013). Il a conçu un algorithme de B&P dans lequel les sous-problèmes sont des *ring star* problèmes résolus par une méthode de B&C. Les instances sont générées aléatoirement et contiennent jusqu'à 60 points de service et 150 clients. La distance maximale \bar{d} est paramétrée à $+\infty$ et la cardinalité maximale \bar{n} varie entre 5 et 8. Malgré les bornes serrées au nœud racine, les temps de résolution sont assez longs.

Ha et al. (2013) ont proposé une nouvelle formulation de flux de deux marchandises du MCTP, basée sur celle de Baldacci et al. pour le CTP, ainsi que des inégalités valides. Leur formulation est résolue avec un algorithme de séparations et coupes. Ils ont également conçu une métaheuristique en deux phases qui crée des sous-ensembles de points de service aléatoires qui peuvent couvrir tous les clients du graphe et qui résout ensuite des problèmes de tournées de véhicules avec des demandes unitaires sur chaque sous-ensemble. Cette métaheuristique fournit des bornes supérieures initiales pour la méthode exacte. Ils ont comparé leur méthode avec celle de Jozefowicz sur des instances similaires et elle surpasse les performances des autres méthodes exactes pour le MCTP. Ainsi, cette méthode est la meilleure méthode exacte disponible dans la littérature pour résoudre le MCTP.

Murakami (2014) a introduit une heuristique basée sur la génération de colonnes pour le MCTP. Cette heuristique propose de meilleurs résultats que l'heuristique de Hachicha et al. en terme de coût, mais pas en terme de temps de résolution. Une autre étude (Murakami (2018)) utilise cette méthode approchée pour résoudre le problème de tournées de véhicules

généralisé où les points de services ont des demandes supérieures à un. Les résultats sur les instances de Jozefowicz (2014) montre son efficacité lorsque la méthode est comparée aux heuristiques de Hachicha et al. (2000). De nouvelles instances allant jusqu'à 400 points de service ont également été résolues en des temps raisonnables.

Kammoun et al. (2017) ont appliqué une heuristique de recherche locale à voisinages variables (*variable neighborhood search heuristic*) pour le MCTP. Ils ont comparé leur métaheuristique avec celle de Ha et al. (2013) sur les mêmes instances. La méthode de Kammoun et al. obtient de meilleurs résultats en des temps plus rapides. Les meilleures bornes supérieures connues de 4 instances sur les 96 ont été améliorées. Les instances de TSPLIB ont été modifiées pour obtenir des instances du MCTP avec 700 nœuds dont 362 points de service. La métaheuristique a obtenu de bonnes solutions sur ces instances en quelques secondes.

Récemment, Pham et al. (2017) ont introduit le problème de tournées de véhicules avec multi-couverture (*Multi-vehicle Multi-Covering Tour Problem* = MMCTP) où les clients doivent être couverts un certain nombre de fois. Pour le résoudre, il ont adapté la métaheuristique de Ha et al. (2013). Des instances aléatoires ont été générées et elles contiennent jusqu'à 200 nœuds et 50 points de service.

Enfin, Flores-Garza et al. (2017) ont également introduit une variation du MCTP : le problème de tournées couvrantes cumulatif (*Multi-vehicle Cumulative Covering Tour Problem*) qui minimise la somme des temps d'arrivée à chaque point de service. Ils ont utilisé une formulation à trois indices et une métaheuristique GRASP pour résoudre ce problème. Les instances de Jozefowicz (2014) ont été utilisées.

6.2.3 Problème de la tournée couvrante bi-objectif

Current et Schilling (1994) ont présenté deux problèmes appelés le problème de tour médian (*Median Tour Problem*) et le problème de la tournée couvrante maximale (*Maximal Covering Tour Problem*) qui sont des variantes bi-objectif du CSP présenté dans le papier de Current et Schilling (1989). Les deux problèmes minimisent la taille totale du tour. Cependant, le problème de tour médian minimise également la distance totale entre chaque nœud non visité et le nœud visité le plus proche pondérée par la demande de ces nœuds. D'autre part, le problème de la tournée couvrante maximale minimise la demande des nœuds non visités par le tour. Chaque problème a été formulé avec une formulation à deux indices et des heuristiques ont été conçues pour les résoudre. Elles sont initialisées avec une première solution qui minimise un seul des objectifs. L'ensemble des points non dominés $\mathcal{Y}_{\mathcal{N}}$ est complété en se déplaçant sur une solution adjacente le long du front dans le sens d'optimisation de l'autre objectif. Des instances réelles de 681 nœuds ont été utilisées pour les tester. Environ 26 points ont été trouvés dans $\mathcal{Y}_{\mathcal{N}}$, même si ce nombre dépend de la qualité de la solution initiale.

Jozefowicz et al. (2007) ont décrit une nouvelle variante bi-objectif du CTP où le premier objectif est toujours de minimiser le coût total du tour et l'autre est la minimisation de la distance de couverture maximale η . Ils ont conçu une métaheuristique en deux phases qui utilise un algorithme évolutionnaire multi-objectif pour générer des solutions réalisables en

entrée de la méthode de séparations et coupes de [Gendreau et al. \(1997\)](#). Des résultats expérimentaux ont été obtenus sur des instances générées aléatoirement de 120 points de service et 360 clients, ainsi que sur l'instance réelle de [Hodgson et al. \(1998\)](#). Une méthode exacte a également été construite en utilisant la méthode de Gendreau et al. intégrée dans une approche ε -contrainte. La métaheuristique donne des solutions de bonne qualité et est plus rapide sur les grandes instances.

[Nolz et al. \(2010\)](#) ont proposé une heuristique basée sur un algorithme génétique et la recherche locale pour résoudre le BOCTP qui minimise la distance totale entre les clients non couverts et le point de service visité le plus proche et qui minimise soit la distance totale du tour, soit le temps d'arrivée au dernier point de service visité par le tour. Ils ont utilisé des instances réelles de 41 points de service pour tester leur heuristique.

6.2.4 Problème de tournées couvrantes bi-objectif

[Tricoire et al. \(2012\)](#) ont introduit un problème bi-objectif de tournées couvrantes stochastique qui minimise la distance totale parcourue avec le coût d'ouverture de centres de distribution et les demandes non couvertes attendues. Dans ce problème, les demandes des clients sont incertaines et sont approximées par une répartition empirique. Chaque point de service a une capacité maximale de clients qu'il peut couvrir. Ils ont utilisé un algorithme de séparations et coupes intégré dans une méthode ε -contrainte. Ils ont également développé une heuristique qui permet de ne pas résoudre chaque problème à l'optimalité mais de s'arrêter avec un certain gap de tolérance fixe. Des instances réelles de 30 nœuds sont utilisées pour tester leurs méthodes dans un temps limite de trois jours. Le nombre de solutions dans $\mathcal{Y}_{\mathcal{N}}$ varie entre 6 et 112. L'heuristique semble donner une bonne approximation de $\mathcal{Y}_{\mathcal{N}}$ en des temps raisonnables.

[Oliveira et al. \(2015\)](#) ont planifié des itinéraires de patrouille qui visent à minimiser la distance totale parcourue et maximiser l'équité entre les routes, i.e. minimiser les différences entre les routes en ce qui concerne le nombre de points de service visités. Ils ont transformé le second objectif en contrainte et ont appliqué des heuristiques sur les instances de [Ha et al. \(2013\)](#) et sur des instances réelles.

Enfin, [Artigues et al. \(2018\)](#) ont construit la borne inférieure du BOMCTP où les objectifs sont de minimiser la distance totale parcourue et la distance maximale de couverture η . Ils ont utilisé la technique de génération de colonnes combinée à la méthode ε -contrainte. À notre connaissance, il n'existe pas de méthode exacte pour résoudre le BOMCTP tel qu'il est formulé dans le papier de [Artigues et al. \(2018\)](#).

Le tableau 6.1 récapitule les différentes publications de la littérature sur le BOCTP et le BOMCTP. La seconde colonne $|K|$ indique le nombre de véhicules disponibles au dépôt. Les colonnes restantes représentent les différents objectifs utilisés dans la littérature, soit :

- *Coût total* : minimisation du coût total des routes ;
- *Trajet des clients* : minimisation de la distance totale entre les nœuds non visités et leur plus proche nœud visité ;
- *Demande non satisfaite* : minimisation des demandes non satisfaites par les routes ;

- *Distance de couverture maximale* : minimisation de la distance de couverture maximale η ;
- *Dernière arrivée* : minimisation du temps d'arrivée au dernier point de service visité ;
- *Équité* : maximisation de l'équité entre les routes, i.e. minimisation de la différence de points de service visités entre la route de plus grande cardinalité et celle de cardinalité minimum.

Notre étude du BOMCTP se concentre sur les deux fonctions objectives introduites par Artigues et al. (2018).

TABLE 6.1: Tableau sur la littérature du BOCTP.

Auteurs	$ K $	Coût total	Trajet des clients	Demande non satisfaite	Distance de couverture maximale	Dernière arrivée	Équité
Current et Schilling (1994)	1	x	x				
	1	x		x			
Jozefowicz et al. (2007)	1	x			x		
Nolz et al. (2010)	1	x	x				
	1		x			x	
Tricoire et al. (2012)	m	x		x			
Oliveira et al. (2015)	m	x					x
Artigues et al. (2018)	m	x			x		

6.3 Méthode exacte pour le problème de tournées couvrantes

Dans cette section, nous présentons une formulation mathématique pour le MCTP et nous décrivons les spécificités de la méthode exacte mono-objectif présentée en Section 3.3 lorsqu'elle est appliquée au MCTP.

6.3.1 Formulation mathématique

Pour résoudre le MCTP, une formulation de partitionnement a été introduite par Lopes et al. (2013). Soit R l'ensemble des routes réalisables. Chaque route $r \in R$ part du dépôt, visite un ensemble de points de service S_r et retourne au dépôt. La cardinalité d'une route r est notée q_r . Une route est dite réalisable si elle ne visite pas plus de \bar{n} points de service, i.e., $q_r \leq \bar{n}$, et que la somme des distances des arêtes traversées, notée d_r , n'excède pas la distance maximale \bar{d} .

Soit a_{ir} un coefficient binaire indiquant si le point de service $i \in O \cup M$ est visité par la route $r \in R$. Soit $C_r \subseteq C$ le sous-ensemble de clients couverts par r (i.e., par les points de services visités S_r) et b_{ir} un coefficient binaire indiquant si le client $i \in C$ est dans C_r . Les variables binaires x_r de la formulation indiquent si la route $r \in R$ est sélectionnée dans la solution ($x_r = 1$) ou non ($x_r = 0$). Le MCTP peut être formulé comme suit :

$$\left\{ \begin{array}{l}
 \min z = \sum_{r \in R} d_r x_r \quad (6.1.1) \\
 t.q. \sum_{r \in R} a_{ir} x_r = 1 \quad i \in M \quad (6.1.2) \\
 \sum_{r \in R} a_{ir} x_r \leq 1 \quad i \in O \quad (6.1.3) \\
 \sum_{r \in R} b_{ir} x_r \geq 1 \quad i \in C \quad (6.1.4) \\
 \sum_{r \in R} x_r \leq |K| \quad (6.1.5) \\
 x_r \in \{0, 1\} \quad r \in R \quad (6.1.6)
 \end{array} \right. \quad (6.1)$$

La fonction objectif (6.1.1) minimise la distance totale des routes sélectionnées. Les contraintes (6.1.2) assurent que tous les points de service obligatoires sont visités; alors que les contraintes (6.1.3) autorisent une visite aux points de service optionnels, si nécessaire. Chaque client doit être couvert au moins une fois grâce aux contraintes (6.1.4). Les contraintes (6.1.5) garantissent que moins de $|K|$ routes sont sélectionnées dans la solution. Enfin, les contraintes (6.1.6) définissent l'espace de décision des variables.

Nous pouvons noter que les contraintes (6.1.3) sont redondantes si la matrice de distance \mathbf{d} satisfait l'inégalité triangulaire. Autrement, ces contraintes sont nécessaires à moins qu'une route r puisse passer sur un point de service i sans le visiter (i.e., augmenter le coefficient a_{ir}). Quelques formulations dans la littérature ont gardé ces contraintes (Hachicha et al. (2000), Lopes et al. (2013), Ha et al. (2013)), alors que d'autres non (Jozefowicz (2014), Artigues et al. (2018)).

La méthode exacte utilisée pour résoudre le MCTP est basée sur la formulation (6.1) qui est appelée le problème maître (PM) dans cette section. La relaxation linéaire du PM est le problème maître linéaire (PML), les deux problèmes PM et PML restreints sur un ensemble de routes \bar{R} sont notés $\text{PMR}(\bar{R})$ et $\text{PMLR}(\bar{R})$. Comme décrit en Section 3.3, cette méthode exacte repose également sur la méthode de Baldacci et al. (2008). Nous rappelons que cette dernière est composée de quatre étapes principales :

1. Calculer une borne inférieure LB , solution optimale du PML et la solution duale associée λ par génération de colonnes.
2. Calculer une borne supérieure UB du problème maître PM.
3. Générer l'ensemble \bar{R} des routes réalisables ayant un coût réduit, calculé par rapport à λ , inférieur ou égal au gap, $UB - LB$.
4. Résoudre $\text{PMR}(\bar{R})$ à l'optimalité avec un solveur de programmes linéaires en nombres entiers.

Dans la suite de cette section, les notations suivantes sont utilisées :

- R est l'ensemble des routes réalisables du problème maître;
- \bar{R} est l'ensemble des routes dans le gap et générées à l'étape 3;

- R_{ng} est l'ensemble des ng-routes (*cf.* Section 2.3.2.2) réalisables du problème maître ;
- \tilde{R}_{ng} est l'ensemble des ng-routes générées pour la borne inférieure à l'étape 1 ;
- \tilde{R}_{ng}^i est l'ensemble des ng-routes générées, à l'itération i , pour la borne inférieure à l'étape 1, avec l'ajout d'une technique de stabilisation, présentée en Section 6.3.2.

Les spécificités de chaque étape lors de leurs applications sur le MCTP sont expliquées dans les prochaines Sections 6.3.2 à 6.3.6. Les variables duales associées aux contraintes (6.1.2), (6.1.3), (6.1.4) et (6.1.5) sont respectivement $\lambda_i \in \mathbb{R}$ ($i \in M$), $\gamma_j \leq 0$ ($j \in O$), $\sigma_k \geq 0$ ($k \in C$) et $\gamma_0 \leq 0$.

6.3.2 Étape 1 : Calcul d'une borne inférieure LB

Comme expliqué en Section 3.3.1, la borne inférieure LB est calculée par génération de colonnes et travaille sur l'ensemble de routes \tilde{R} . Cet ensemble \tilde{R} est initialisé avec les routes réalisables de cardinalité un et deux et avec une route artificielle qui visite l'ensemble des clients et qui a une distance \tilde{d} généralement suffisamment grande pour ne faire partie d'aucune solution optimale.

Le sous-problème de *pricing* du MCTP est un ESPPRC. La ng-relaxation sur les points de service, expliquée en Section 2.3.2.2, est appliquée telle quelle pour accélérer sa résolution. L'ensemble R de routes réalisables de la formulation considérée est maintenant remplacé par l'ensemble R_{ng} de ng-routes réalisables. Pour résoudre le sous-problème, nous proposons des nouvelles règles de dominance et des bornes de complétions adaptées, ainsi que l'adaptation d'une technique de stabilisation de la génération de colonnes et une heuristique pour générer rapidement des colonnes.

Algorithme d'étiquetage. Pour cet algorithme d'étiquetage, une étiquette E est un ng-chemin réalisable qui commence au dépôt, visite un ensemble de points de service S_E , couvre un ensemble de clients C_E et se termine au point de service v_E avec un ng-ensemble NG_E , une cardinalité q_E , une distance de trajet τ_E et un coût réduit \bar{c}_E . Soit le coût d'_{ij} d'une arête $(i, j) \in E$:

$$d'_{ij} = d_{ij} - (e_i + e_j)/2 \quad \forall i, j \in O \cup \{0\} \cup M \quad (6.2)$$

avec

$$e_i = \begin{cases} \lambda_i & i \in M \\ \gamma_i & i \in O \\ \gamma_0 & i = 0 \end{cases}$$

L'extension d'une étiquette E de son nœud terminal v_E à un point de service $w \in \{O \cup M\} \setminus NG_E$ produit une nouvelle étiquette E' via les règles (6.3) à (6.8) :

$$v_{E'} = w \quad (6.3)$$

$$q_{E'} = q_E + 1 \quad (6.4)$$

$$\tau_{E'} = \tau_E + d_{v_E w} \quad (6.5)$$

$$C_{E'} = C_E \cup C_w \quad (6.6)$$

$$\overline{c}_{E'} = \overline{c}_E + d'_{v_E w} - \sum_{i \in C_w \setminus C_E} \sigma_i \quad (6.7)$$

$$NG_{E'} = (NG_E \cap N_w) \cup \{w\} \quad (6.8)$$

où C_w , $w \in O \cup M$, est l'ensemble des clients de C qui peuvent être couverts par w et N_w , $w \in O \cup M$, est le ng-ensemble pré-défini pour le point de service w .

Les règles de propagation (6.3), (6.4), (6.5), (6.8) sont utilisées dans beaucoup de problèmes de tournées de véhicules. Au contraire, les règles (6.6) et (6.7) sont spécifiques au MCTP puisqu'elles dépendent de l'ensemble C des clients à couvrir. La règle de propagation (6.6) met à jour les clients couverts par E en ajoutant tous les clients de C qui sont couverts par le point de service w . La règle de propagation (6.7) met à jour le coût réduit en ajoutant au coût réduit de E , le coût réduit de l'arête $\{v_E, w\}$ et en enlevant le coût des variables duales des clients couverts par w qui ne sont pas couverts par E .

Règles de dominance. Les règles de dominance (6.9)-(6.13) sont utilisées. Une étiquette E^1 domine une étiquette E^2 différente de E^1 si les conditions suivantes sont satisfaites :

$$v_{E^1} = v_{E^2} \quad (6.9)$$

$$q_{E^1} \leq q_{E^2} \quad (6.10)$$

$$\tau_{E^1} \leq \tau_{E^2} \quad (6.11)$$

$$NG_{E^1} \subseteq NG_{E^2} \quad (6.12)$$

$$\overline{c}_{E^1} \leq \overline{c}_{E^2} - \sum_{i \in C_{E^1} \setminus C_{E^2}} \sigma_i \quad (6.13)$$

Les quatre premières conditions assurent que tous les points de service $i \in O \cup M$ qui peuvent être atteints par propagation de l'étiquette E^2 peuvent également être atteints par propagation de l'étiquette E^1 . La condition (6.13) s'assure que jamais le coût réduit d'une propagation de l'étiquette E^2 ne sera inférieur au coût réduit de cette même propagation de l'étiquette E^1 . La Proposition 1 suivante permet de vérifier la véracité de ces règles de dominance.

Proposition 1. *Soit deux étiquettes E^1 et E^2 satisfaisant les règles (6.9)-(6.13), toute extension de E^2 est dominée par une extension de E^1 .*

Démonstration. Soit i un point de service accessible par E^1 et E^2 et i couvre le client j .

Par la règle d'extension (6.7), les situations suivantes peuvent arriver :

- $j \in C_{E^1}$ et $j \in C_{E^2}$. La valeur de σ_j est soustraite de $\overline{c_{E^1}}$ et de $\overline{c_{E^2}}$, donc la règle (6.13) est toujours respectée.
- $j \in C_{E^1}$ et $j \notin C_{E^2}$. La valeur de σ_j est uniquement soustraite de $\overline{c_{E^2}}$. Cependant, la règle (6.13) assure déjà que le coût réduit $\overline{c_{E^1}}$ est inférieur ou égal au coût réduit $\overline{c_{E^2}}$ moins la valeur de $\sigma_j \geq 0$.
- $j \notin C_{E^1}$ et $j \in C_{E^2}$. La valeur de σ_j est uniquement soustraite de $\overline{c_{E^1}}$. Ainsi, seule la valeur de $\overline{c_{E^1}}$ diminue.
- $j \notin C_{E^1}$ et $j \notin C_{E^2}$. Les coûts duaux $\overline{c_{E^1}}$ et $\overline{c_{E^2}}$ ne changent pas.

Comme la règle (6.13) établit que les variables duales $\sigma_i \geq 0$ de tous les clients i couverts par E^1 mais pas par E^2 sont enlevées de $\overline{c_{E^2}}$, chaque extension de E^2 est dominé par la même extension de E^1 . \square

Bornes de complétion. Durant l'algorithme d'étiquetage, les bornes de complétion sont utilisées pour éliminer des étiquettes ne pouvant pas aboutir sur une ng-route de coût réduit négatif. Les bornes de complétion présentées ci-dessous sont la manière la plus directe d'appliquer les bornes de complétion de l'état de l'art pour résoudre des problèmes de tournées de véhicules. Pour définir ces bornes, nous utilisons la distinction entre un ng-chemin avant (i.e. qui commence au dépôt et termine à un point de service i dans $M \cup O$) et un ng-chemin arrière (i.e. commençant à un point de service i dans $M \cup O$ et finissant au dépôt). Ainsi, pour trouver les bornes de complétion d'un ng-chemin avant, il est nécessaire de borner le coût de tous les ng-chemins arrières qui peuvent le compléter pour former une route.

Un ng-chemin arrière (NG, q, i) est un chemin qui peut être non élémentaire $ch = (i_{|S_{ch}|}, \dots, i_1, 0)$ qui commence au point de service $i_{|S_{ch}|} = i$, visite un ensemble de points de service S_{ch} tel que $NG_{ch} = NG$, a une cardinalité $q_{ch} = q$ et qui finit au dépôt. Soit le coût c'_{ij} d'une arête $(i, j) \in E$, défini comme suit, grâce à la définition (6.2) de la distance d' :

$$c'_{ij} = d'_{ij} - \sum_{k \in C_i \setminus C_j} \sigma_k \quad \forall (i, j) \in O \cup \{0\} \cup M \quad (6.14)$$

où C_i , $i \in O \cup M \cup \{0\}$ est l'ensemble des clients de C couvert par i .

Chaque ng-chemin arrière ch a un coût c'_{ch} qui est la somme des distances c' des arcs traversés par ch . Soit $f(NG, q, i)$ une borne inférieure du ng-chemin arrière de coût c' minimum (NG, q, i) . Le calcul de ces bornes est expliqué dans le papier de [Baldacci et al. \(2011\)](#) et décrit dans la Section 2.3.2.3. Pour tous les ng-chemins arrières ch commençant au point de service v_{ch} , de cardinalité q_{ch} et visitant les points de service S_{ch} , le coût c'_{ch} de ch est borné comme suit :

$$c'_{ch} \geq \min_{NG \subseteq S_{ch} \cap N_{v_{ch}}} f(NG, q_{ch}, v_{ch})$$

Soit $F(C_{ch'}, q_{ch'}, v_{ch'})$ la borne inférieure du coût de toutes les ng-routes qui peuvent être construites à partir de l'extension du ng-chemin avant ch' .

$$F(S_{ch'}, q_{ch'}, v_{ch'}) = \min_{\substack{q_{v_{ch'}} \leq q \leq Q - q_{ch'} + q_{v_{ch'}} \\ NG \cap S_{ch'} = v_{ch'}}} f(NG, q, v_{ch'})$$

Il est important de noter que le coût c' de tout ng-chemin est inférieur ou égal a son coût réduit \bar{c} puisque le coût c' déduit plus de variables duales positives. Ainsi, toute extension du ng-chemin avant ch' a un coût réduit supérieur à une valeur δ si :

$$\bar{c}_{ch'} + F(C_{ch'}, q_{ch'}, v_{ch'}) > \delta$$

En fixant la valeur de δ à 0 et en calculant le coût c' des ng-chemins grâce aux variables duales associées à la solution optimale du PMLR(\tilde{R}_{ng}), les bornes de complétion empêchent l'expansion de certains ng-chemins qui aboutiront à des ng-routes de coût réduit positif.

Cependant, une difficulté du MCTP vient du fait que le coût réduit d'un chemin ch dépend des clients couverts par ch . Cette information n'est pas stockée durant l'algorithme de programmation dynamique qui calcule les bornes car cela rendrait la résolution trop coûteuse : passage d'une complexité linéaire ($O(Q * (|M| + |O|) * 2^{\Delta-1})$) à exponentielle ($O(Q * (|M| + |O|) * 2^{\Delta-1} * 2^{|C|-1})$) en fonction de la taille de V . Par conséquent, les coûts utilisés pour les bornes de complétion sont moins précis que le coût réduit.

Bornes de complétion adaptées. Comme indiqué précédemment, les bornes de complétion f ne sont pas calculées avec le coût réduit \bar{c} mais avec un autre coût c' . La différence entre ces deux coûts pour un chemin ch est le nombre de fois où les variables duales σ associées aux clients visités par ch sont déduites. Pour $i \in C_{ch}$, σ_i est déduit une fois dans le coût réduit \bar{c} et au moins une fois dans le coût c' . Par conséquent, pour chaque chemin ch , $c'_{ch} < \bar{c}_{ch}$ et les bornes f peuvent être améliorées en réduisant les nombres de fois où les variables duales σ sont déduites.

Soit le coût c''_{ij} d'une arête $(i, j) \in E$, défini comme suit, grâce à la définition (6.2) de la distance d' :

$$c''_{ij} = d'_{ij} - \sum_{k \in C_i \setminus \{C_j \cup \omega\}} \sigma_k \quad \forall (i, j) \in E \quad (6.15)$$

où C_i , $i \in O \cup M \cup \{0\}$ est l'ensemble des clients de C couvert par i .

Soit Ψ le sous-ensemble de clients de C avec les variables duales σ les plus élevées tel que $|\Psi| \leq MAX_SELECTED$. Soit Ω l'ensemble de toutes les combinaisons des clients dans Ψ . Nous pouvons alors définir $g(NG, q, i, \omega)$, avec $\omega \in \Omega$, en tant que borne inférieure du ng-chemin arrière (NG, q, i) de coût c'' minimum.

Pour chaque ng-chemin arrière ch qui commence au point de service v_{ch} , de cardinalité q_{ch} , visitant les points de service S_{ch} et couvrant les clients C_{ch} , le coût c'' est borné comme suit :

$$c''_{ch} \geq \min_{NG \subseteq \tilde{S}_{ch} \cap N_{v_{ch}}} g(NG, q_{ch}, v_{ch}, C_{ch} \cap \Psi)$$

Soit $G(C_{ch'}, q_{ch'}, v_{ch'}, C_{ch'})$ la borne inférieure du coût de toutes les ng-routes qui peuvent être construites à partir de l'extension du ng-chemin avant ch' .

$$G(S_{ch'}, q_{ch'}, v_{ch'}, C_{ch'}) = \min_{\substack{q_{v_{ch'}} \leq q \leq Q - q_{ch'} + q_{v_{ch'}} \\ NG \cap S_{ch'} = v_{ch'}}} g(NG, q, v_{ch'}, C_{ch'} \cap \Psi)$$

Avec ces nouvelles définitions, un ng-chemin avant ch' peut être éliminé pendant la génération de colonnes si :

$$\bar{c}_{ch'} + G(NG_{ch'}, q_{ch'}, v_{ch'}, C_{ch'}) > 0$$

En théorie, les bornes de complétion G sont plus efficaces que les bornes de complétion F comme elles permettent d'éliminer plus de chemins pendant la génération de colonnes.

Heuristique. Une heuristique est utilisée pour accélérer les premières itérations de la génération de colonnes. Ainsi, pour chaque cardinalité $0 \leq q \leq \bar{n}$ et chaque point de service $i \in O \cup M$, le nombre de labels de cardinalité q et de point de service terminal i est limité à MAX_LABEL. Lorsqu'aucune route de coût réduit négatif n'est trouvée alors que cette limite est atteinte, la valeur de MAX_LABEL peut être augmentée. À la fin de la génération de colonnes, le paramètre MAX_LABEL doit être fixé à $+\infty$ pour garantir qu'il n'existe plus de colonnes de coût réduit négatif.

Technique de stabilisation. Généralement, le problème de couverture par ensembles (*Set Covering Problem*) est fortement dégénéré. Ainsi, les variables duales peuvent varier au cours des itérations de la génération de colonnes et causer des problèmes de convergence. Pour éviter cette instabilité, des méthodes de stabilisation sont utilisées sur le problème maître (cf. Section 2.3.3). Du Merle et al. (1999) ont proposé la méthode de *Box Stabilization* qui impose une limite flexible et variable sur les variables duales, ainsi qu'une pénalisation de l'objectif si la limite est dépassée. Pigatti et al. (2005) ont alors proposé une version simplifiée de la méthode de *Box Stabilization*.

Nous appliquons la technique de Pigatti et al. (2005) au problème maître comme suit. Soit m^- , m^+ , o et w quatre vecteurs de variables artificielles. De telles variables ont un coût $\bar{\pi}$ et doivent être inférieures ou égales à une valeur α .

$$\left\{ \begin{array}{ll}
\min z(\bar{\pi}, \alpha) = \sum_{r \in R} d_r x_r + \sum_{i \in M} \bar{\pi}_i (m_i^+ - m_i^-) & \\
\quad - \sum_{i \in O} \bar{\pi}_i o_i + \sum_{i \in C} \bar{\pi}_i w_i & (6.16.1) \\
t.q. \sum_{r \in R} a_{ir} x_r - m_i^- + m_i^+ = 1 & i \in M \rightarrow \lambda_i \text{ non restreints} \quad (6.16.2) \\
\sum_{r \in R} a_{ir} x_r - o_i \leq 1 & i \in O \rightarrow \gamma_i \leq 0 \quad (6.16.3) \\
\sum_{r \in R} b_{ir} x_r + w_i \geq 1 & i \in C \rightarrow \sigma_i \geq 0 \quad (6.16.4) \quad (6.16) \\
\sum_{r \in R} x_r \leq |K| & \rightarrow \gamma_0 \leq 0 \quad (6.16.5) \\
0 \leq m_i^- \leq \alpha & i \in M \rightarrow \mu^- \geq 0 \quad (6.16.6) \\
0 \leq m_i^+ \leq \alpha & i \in M \rightarrow \mu^+ \geq 0 \quad (6.16.7) \\
0 \leq o_i \leq \alpha & i \in O \rightarrow \theta \geq 0 \quad (6.16.8) \\
0 \leq w_i \leq \alpha & i \in C \rightarrow \omega \geq 0 \quad (6.16.9) \\
x_r \geq 0 & r \in R \quad (6.16.10)
\end{array} \right.$$

Nous pouvons noter que $\forall \alpha \geq 0$ et $\forall \bar{\pi}$, $z(\bar{\pi}, \alpha) \leq z$. De plus, $\forall \bar{\pi}$, $z(\bar{\pi}, 0) = z$. Donc, résoudre cette formulation donne une borne inférieure de la formulation (6.1).

Soit $\mu^- \geq 0$, $\mu^+ \geq 0$, $\theta \geq 0$ et $\omega \geq 0$ les variables duales respectives des contraintes (6.16.6), (6.16.7), (6.16.8) et (6.16.9). Le dual de la formulation (6.16) présente les nouvelles contraintes suivantes : $\bar{\pi}_i - \mu_i^- \leq \lambda_i \leq \bar{\pi}_i + \mu_i^+$ ($i \in M$), $\bar{\pi}_i - \theta_i \leq \gamma_i \leq 0$ ($i \in O$) and $0 \leq \sigma_i \leq \bar{\pi}_i + \omega_i$ ($i \in C$). Ces contraintes impliquent qu'une déviation des variables duales λ_i ($i \in M$) de la valeur $\bar{\pi}_i$ ($i \in M$) est pénalisée par $\alpha|\bar{\pi}_i - \lambda_i|$. Le même processus de pénalisation a lieu pour les variables γ et σ .

Le réglage des paramètres $\bar{\pi}$ et α pour la génération de colonnes est fait de la manière suivante. Le coefficient $\bar{\pi}$ change durant l'exécution selon les valeurs des variables duales associées aux contraintes (6.16.2) à (6.16.4) et α est fixé à une certaine valeur. Soit $\text{PMR}'(\tilde{R})$ la formulation (6.1) sur un ensemble réduit de colonnes \tilde{R} et \tilde{R}_{ng}^0 l'ensemble de colonnes initial. En premier lieu, $\bar{\pi}$ est mis à 0 et α à 0.001. Au début de l'itération k de la génération de colonnes, la valeur de $\bar{\pi}$ est la valeur de la solution optimale duale du $\text{PMR}'(\tilde{R}'_{ng}{}^{k-1})$. Lorsque $\text{PMR}'(\tilde{R}'_{ng}{}^n)$ est résolu à l'optimalité pour $\alpha = 0.001$, le processus entier est relancé pour $\alpha = 0$; mais l'ensemble initial de colonnes est $\tilde{R}'_{ng}{}^n$ et $\bar{\pi}$ est la solution optimale duale du $\text{PMR}'(\tilde{R}'_{ng}{}^n)$.

6.3.3 Étape 2 : Calcul d'une borne supérieure UB

La même heuristique que celle présentée en Section 3.3.2 est utilisée. À la fin de la génération de LB , un ensemble $\tilde{R}'_{ng}{}^n \subseteq R_{ng}$ de ng-routes est disponible. Un sous-ensemble $\tilde{R}'_{ng}{}^{elem}$ de routes réalisables élémentaires peut être facilement obtenu en enlevant toutes les routes

non élémentaires de \widetilde{R}_{ng}^n . Une borne supérieure UB du PM est calculée en résolvant $\text{PMR}(\widetilde{R}_{elem})$.

6.3.4 Étape 3 : Énumération des colonnes dans le gap

Lors de la troisième étape, nous utilisons un algorithme d'étiquetage mono-directionnel. Une étiquette E est un chemin réalisable qui commence au dépôt, visite un ensemble de points de service S_E , couvre les clients C_E et termine au point de service v_E avec une cardinalité q_E , une distance de trajet τ_E et un coût réduit \bar{c}_E . Les règles d'extension de l'étiquette E pour obtenir l'étiquette E' sont définies par (6.3)-(6.7) plus la mise à jour des points de service visités (6.17).

$$S_{E'} = S_E \cup \{v_{E'}\} \quad (6.17)$$

La règle de dominance éliminant des étiquettes non prometteuses est la suivante. Une étiquette E^1 domine une étiquette E^2 si les conditions (6.9)-(6.11), (6.18) et (6.19) sont respectées.

$$S_{E^1} = S_{E^2} \quad (6.18)$$

$$\bar{c}_{E^1} \leq \bar{c}_{E^2} \quad (6.19)$$

La règle (6.18) assure que les mêmes points de service sont visités. La dernière règle (6.19) assure que le coût réduit de E^1 est inférieur à celui de E^2 . Par conséquent, ces inégalités n'affectent pas les performances de l'algorithme. En effet, la règle (6.18) implique qu'il n'existe pas de client $i \in C_{E^1} \setminus C_{E^2}$. Ainsi, toute extension de l'étiquette E^2 ne peut pas obtenir un meilleur coût réduit que la même extension de l'étiquette E^1 . Les mêmes bornes de complétion que celles pour le calcul de la borne inférieure sont utilisées pour une valeur de $\delta = UB - LB$. À la fin de l'énumération des colonnes dans le gap, l'ensemble \bar{R} contient les routes de la solution optimale entière de la formulation considérée.

6.3.5 Étape 4 : Calcul de la solution optimale entière

Après l'étape 3, nous résolvons la formulation (6.1) sur l'ensemble de colonnes \bar{R} avec un solveur de programmes linéaires en nombres entiers.

6.3.6 Détails d'implémentation

Soit LB la solution optimale du PML et UB_1 une borne supérieure trouvée lors de l'étape 2. Pour des raisons d'efficacité, si UB_1 forme un gap avec LB qui est supérieur à 4%, alors la borne supérieure utilisée pour l'étape 3 n'est pas une solution réalisable mais est calculée comme suit : $UB_2 = (1 + \frac{4}{100})LB$. Si aucune solution optimale n'est trouvée avec UB_2 en tant que borne supérieure à l'étape 4, ou si la solution optimale a un coût

supérieur à UB_2 , les étapes 3 et 4 sont relancées du début avec UB_1 en tant que borne supérieure.

6.4 Extension au problème de tournées couvrantes bi-objectif

Dans cette section, la méthode exacte présentée dans la section précédente est appliquée au BOMCTP où les objectifs sont la minimisation de la distance totale parcourue et la minimisation de la distance maximale de couverture η . Premièrement, nous introduisons une formulation de partitionnement pour le BOMCTP. Ensuite, la méthode exacte pour le MCTP est intégrée dans la méthode ε -contrainte par étapes (SEM). Certaines étapes particulières sont décrites comme le choix des valeurs de ε lors des itérations ou le calcul des points extrêmes. La dernière partie discute de l'application de la méthode ε -contrainte par étapes améliorée (ISEM).

6.4.1 Formulations mathématiques

Formulation bi-objectif. La formulation suivante du BOMCTP est une extension de la formulation (6.1) et a été présentée dans [Artigues et al. \(2018\)](#). Par contre, les contraintes (6.1.3) et (6.1.5), qui limitent respectivement le nombre de visites à un point de service optionnel à un et le nombre de véhicules, ne sont pas dans la formulation proposée par Artigues et al. Pour une ng-route $r \in R'$, on note κ_r la distance maximale entre chaque client de C_r et son plus proche point de service dans S_r (i.e. visité par r).

$$\left\{ \begin{array}{ll} \min z = \sum_{r \in R} d_r x_r & (6.20.1) \\ \min w = \eta & (6.20.2) \\ t.q. (6.1.2) - (6.1.6) & \\ \kappa_r x_r \leq \eta & r \in R \quad (6.20.3) \end{array} \right. \quad (6.20)$$

Le nouvel objectif de la formulation (6.20.2) minimise la distance maximale de couverture de toutes les routes η . Les contraintes (6.20.3) imposent à η d'excéder la distance maximale de couverture de chaque route κ_r . La résolution de cette formulation permet l'obtention d'un ensemble de points non dominés.

Formulation pour la méthode ε -contrainte. Une constante ε est introduite dans la formulation bi-objectif pour contraindre un des objectifs à être inférieur à une certaine constante.

$$\left\{ \begin{array}{ll}
 \min z = \sum_{r \in R} d_r x_r & (6.21.1) \\
 t.q. \sum_{r \in R} a_{ir} x_r = 1 & i \in M \quad (6.21.2) \\
 \sum_{r \in R} a_{ir} x_r \leq 1 & i \in O \quad (6.21.3) \\
 \sum_{r \in R} b_{ir} x_r \geq 1 & i \in C \quad (6.21.4) \\
 \sum_{r \in R} x_r \leq |K| & (6.21.5) \\
 \kappa_r x_r \leq \varepsilon & r \in R \quad (6.21.6) \\
 x_r \in \{0, 1\} & r \in R \quad (6.21.7)
 \end{array} \right. \quad (6.21)$$

Comme expliqué dans le papier de [Artigues et al. \(2018\)](#), la contrainte ε (6.21.6) peut être enlevée du problème maître et remplacée par une condition sur la définition d'une route réalisable. Il suffit d'imposer que les points de service optionnels d'une route réalisable ne couvrent que les clients situés à moins d'une distance ε . Une telle reformulation est possible pour le BOMCTP car l'objectif borné par ε concerne la définition de chaque route séparément. Cette reformulation n'est pas possible pour les problèmes des Chapitres 4 et 5 car l'objectif borné concerne la combinaison de plusieurs routes.

Pour la suite, nous utilisons donc la formulation (6.21) sans la contrainte (6.21.6) avec une distance maximale de couverture η fixée à la valeur de ε . Avec cette dernière remarque, la méthode exacte du MCTP est applicable à la formulation finale sans effectuer de modifications.

6.4.2 Adaptation de SEM

Pour pouvoir obtenir l'ensemble des points non dominés, la valeur ε doit varier entre les deux points extrêmes du problème. Le point extrême minimisant la distance totale parcourue par les véhicules est trouvé au début de la méthode pour $\varepsilon = +\infty$. La valeur ε décroît jusqu'à ce qu'il existe un client qui ne puisse plus être servi par les points de service optionnels.

Quand un point non dominé OPT est trouvé, la valeur ε peut être fixée à $w(OPT) - 1$ car η est un coefficient entier. Cependant, ε peut décroître plus rapidement comme expliqué par [Jozefowicz et al. \(2007\)](#). En effet, la prochaine valeur de ε peut être calculée facilement puisque c'est la distance de couverture maximale minimale pour laquelle OPT est toujours valide, moins un. Cette distance de couverture maximale minimale d'une solution peut être calculée comme suit. Pour chaque route r appartenant à la solution ($x_r > 0$), affecter chaque client couvert au point de service visité le plus proche et récupérer κ_r , le maximum de toutes les distances d'affectation. Enfin, ε est fixé à $\max_{x_r > 0} \kappa_r - 1$.

L'algorithme 2 de la Section 3.4.3 reste valide en intégrant les deux remarques précédentes, ainsi SEM s'applique sur le problème de tournées couvrantes bi-objectif.

6.4.3 Adaptation de ISEM

Lors du travail sur la formulation (6.21) sans la contrainte (6.21.6), appelée contrainte ε , les techniques du Chapitre 4 ne sont pas utiles ou applicables. En effet, les techniques T_{lb} , T_{valid} et T_{save} peuvent être appliquées mais les conditions d'utilisation deviennent nulles avec la suppression de la contrainte ε . De plus, la technique T_{mult_dual} , qui utilise plusieurs directions de recherche simultanées pour l'énumération des colonnes, ne peut plus être appliquée car seulement un objectif est présent. Seule la technique T_{gap} de la Section 4.3.1 qui réduit le gap d'une unité peut être utilisée.

Pour utiliser l'ensemble des techniques, il faudrait se baser sur la formulation (6.21) sans distance maximale de couverture η et en gardant la contrainte (6.21.6). Dans ce cas-là, l'ensemble de routes réalisables est beaucoup plus grand que celui sans la contrainte ε . Soit un graphe du BOMCTP où le dépôt est le nœud 0 et avec un point de service optionnel 1 couvrant les clients $\{a, b, c\}$ avec une distance inférieure à 100 unités. Lorsque ε vaut 100, l'ensemble des routes réalisables de la formulation (6.21) sans la contrainte ε contient deux routes 0 et $0 \rightarrow 1 \rightarrow 0$ couvrant $\{a, b, c\}$. Au contraire, l'ensemble des routes réalisables de la formulation (6.21) avec la contrainte ε contient 9 routes réalisables différentes dont 8 passant par le client 1, mais couvrant les 7 combinaisons des clients $\{a, b, c\}$. Ainsi, l'application d'ISEM sur la formulation (6.21) serait moins bonne que la méthode SEM sur la formulation (6.21) sans la contrainte ε . C'est pourquoi la méthode ISEM ne sera présentée dans les résultats expérimentaux.

6.5 Résultats expérimentaux

Dans cette section, les résultats des méthodes exactes des Sections 6.3 et 6.4 pour le MCTP et le BOMCTP sont présentés. Les expérimentations ont été conduites sur un processeur Xeon E5-2695 avec un CPU de 2.30GHz de fréquence d'horloge, sur un unique cœur. Le code a été implémenté en C++ et les problèmes linéaires et entiers sont résolus avec Gurobi 7.1. Les paramètres suivants ont été utilisés : $\Delta = 8$ (cf. Section 2.3.2.2), $MAX_SELECTED = 3$ (cf. Section 6.3.2) et $MAX_LABEL = 4$ et $+\infty$ (cf. Section 6.3.2).

Le temps limite est de deux heures pour la méthode mono-objectif, trois heures pour obtenir la borne inférieure du BOMCTP et six heures pour la méthode exacte bi-objectif. Si une méthode ne converge pas dans le temps limite, la ligne correspondante est notée avec un tiret (-). Les meilleures méthodes en termes de temps CPU et de gap sont notées en gras.

6.5.1 Problème de tournées couvrantes

Les instances du MCTP sont celles utilisées par Jozefowicz (2014) et Ha et al. (2013). Elles sont nommées $X-T-n-W-p$ où X est le nom de l'instance de Krolak qui est utilisée, T est la taille de l'ensemble des points de service obligatoires avec le dépôt, n est la taille de l'ensemble des points de service plus le dépôt, W est la taille de l'ensemble des clients à couvrir et p est le nombre maximal de points de service visités par une route. Dans toutes

TABLE 6.2: Synthèse des résultats de notre méthode et celle de Hà et al. par classes.

Instances	Nombre d'instances	Ha et al. (2013)			Notre méthode	
		Gap moyen	Temps_o moyen	Temps_c moyen	Gap moyen	Temps moyen
$ V = 100$ et $ M = 0$	32	17.0	37.6	19.4	0.5	6.5
$ V = 100$ et $ M > 0$	32	8.5	271.0	139.7	2.4	5.5
$ V = 200$	32	13.9	1455.4	750.2	2.0	393.1

TABLE 6.3: Nombre d'instances fermées par notre méthode et celle de Hà et al. par classes.

Instances	Ha et al. (2013)	Notre méthode
$ V = 100$ et $ M = 0$	32/32	32/32
$ V = 100$ et $ M > 0$	31/32	31/32
$ V = 200$	20/32	27/32
Total	83/96	90/96

les instances, la taille maximale d'une route \bar{d} est fixée à $+\infty$. Au total, 96 instances sont testées.

La méthode la plus performante dans la littérature pour le MCTP est présentée dans le papier de Ha et al. (2013) ; c'est pourquoi nous comparons nos méthodes avec elle dans les tableaux de résultats suivants. Comme le CPU utilisé par Hà et al. est différent du nôtre, nous utilisons le *passmark CPU score*¹ avec un seul cœur pour avoir une comparaison équitable. Ils utilisent un CPU Intel Core2 Duo avec une fréquence d'horloge de 2.40GHz. Ils ne spécifient pas précisément le modèle de leur micro-processeur et la note de leur CPU varie entre 842 et 991. Nous comparons nos résultats avec le processeur disponible le moins efficace. Ainsi, la note du CPU utilisé par Hà et al. est 842 et le nôtre est 1628. Dans le tableau 6.2 les temps du papier de Hà et al. sont divisés par 1.94 dans la colonne *Temps_c* et leurs temps originaux sont notés dans la colonne *Temps_o*.

Dans ce tableau, les en-têtes suivants sont utilisés : le temps CPU en secondes (*Temps*) et le gap entre la borne inférieure *LB* et l'optimum *UB* en pourcentage (*Gap*). Si aucune solution optimale n'est trouvée, le gap est calculé avec la meilleure borne supérieure valide trouvée. Le gap est calculé comme suit :

$$Gap = 100 - \frac{z(LB) * 100}{z(UB)} (\%);$$

De plus, le tableau 6.3 montre le nombre d'instances fermées par notre méthode et celle de Hà et al. Le détail de toutes les instances est donné en Annexe C.1.

Les résultats montrent que notre méthode est strictement plus efficace que celle de l'état de l'art sur 82 instances. Notre méthode est strictement moins efficace sur 7 instances. Parmi ces 7 instances, la différence de temps est supérieure à 5 secondes sur une seule instance, *D1-1-50-50-8*. Cela peut s'expliquer car la génération de colonnes est une technique

1. Données disponibles sur le site https://www.cpubenchmark.net/cpu_list.php

TABLE 6.4: Moyennes des résultats de notre méthode avec les bornes de complétion f et g sur des classes d'instances.

Instances	Borne de complétion de l'état-de-l'art f		Nouvelle borne de complétion g	
	Chemins	Temps	Chemins	Temps
$ V = 100$ et $ M = 0$	100723	20.6	27550	6.5
$ V = 100$ et $ M > 0$	4826	4.6	2560	5.1
$ V = 200$	467976	436.1	315583	407.5

moins efficace sur les problèmes de tournées de véhicules avec de longues routes en terme de points de service visités. Par exemple, pour $V = 200$, notre méthode surpasse celle de Hà et al. Notre méthode a permis de fermer 7 nouvelles instances. Sur les 6 instances qui restent ouvertes, la borne inférieure de notre méthode est meilleure.

Nous pouvons remarquer que la difficulté de résolution d'une instance augmente avec la cardinalité maximale autorisée. Le nombre de points de service (optionnels ou obligatoires) joue également un rôle important dans l'accroissement de la difficulté. Cependant, le nombre de clients ne semble pas être un paramètre critique. Ces conclusions confirment celles énoncées par Hà et al.

Pour évaluer l'efficacité de la nouvelle borne de complétion g , introduite en Section 6.3.2, nous comparons les résultats de la méthode avec la borne g et avec la borne de l'état de l'art f . La moyenne des résultats est présentée dans le tableau 6.4 avec les en-têtes suivants : le temps CPU en secondes (*Temps*) et le nombre d'étiquettes acceptées par les bornes de complétion lors des dernières itérations de la génération de colonne pour calculer la borne inférieure du MCTP (*Chemins*). Les dernières itérations correspondent au sous-problème de *pricing* résolu avec $\text{MAX_LABEL} = +\infty$ (cf. Section 6.3.2). Plus le nombre de chemins autorisés est petit, plus la borne de complétion est efficace. Le détail de toutes les instances est donné en Annexe C.2.

Les résultats montrent que la borne g est strictement plus efficace en ce qui concerne le temps de résolution sur 43 instances. Au contraire, la borne f est strictement plus efficace sur 31 instances. Le temps de résolution moyen des instances est de 139.7 secondes et de 153.8 secondes pour les bornes g et f respectivement.

Nous pouvons aussi observer que le nombre de chemins acceptés dans les sous-problèmes après la borne de complétion est toujours plus élevé pour la borne f . En effet, la borne g élimine plus d'étiquettes en théorie. Par exemple, l'instance $A1-1-50-50-8$ montre que le nombre de chemins acceptés est environ quatre fois plus grand pour la borne f et l'impact sur le temps de résolution est proportionnel. Cependant, sur l'instance $C1-10-50-50-5$, peu d'étiquettes sont acceptées par les deux bornes et donc, le temps de calcul de la borne g a un impact négatif sur le temps de résolution global.

TABLE 6.5: Résultats de notre méthode bi-objectif et de SOGA pour trouver la borne inférieure du BOMCTP.

Instance	SOGA			Notre méthode			Instance	SOGA			Notre méthode		
	Pt	Hv	Temps	Pt	Hv	Temps		Pt	Hv	Temps	Pt	Hv	Temps
A1-1-25-75-4	34	83.0	5.4	48	82.8	5.7	B1-1-25-75-4	40	84.6	10.6	60	84.5	9.0
A1-1-25-75-5	38	83.2	7.0	41	83.2	8.4	B1-1-25-75-5	39	84.9	16.6	42	84.9	9.0
A1-1-25-75-6	33	83.3	10.6	46	83.3	15.4	B1-1-25-75-6	33	85.2	14.9	44	85.1	16.2
A1-1-25-75-8	31	83.6	7.6	34	83.5	36.6	B1-1-25-75-8	33	85.5	15.6	41	85.4	52.8
A1-1-50-50-4	48	87.7	72.2	74	87.5	34.9	B1-1-50-50-4	57	86.6	174.0	75	86.5	31.7
A1-1-50-50-5	50	88.2	148.5	67	88.2	45.5	B1-1-50-50-5	58	87.0	238.4	72	87.0	56.6
A1-1-50-50-6	45	88.6	421.0	58	88.5	82.1	B1-1-50-50-6	53	87.3	946.2	68	87.3	97.0
A1-1-50-50-8	48	88.9	569.9	58	88.8	351.1	B1-1-50-50-8	47	87.7	858.6	60	87.6	1749.7
A2-1-50-150-4	71	84.0	260.7	137	83.9	181.5	B2-1-50-150-4	48	85.6	147.5	127	84.9	134.2
A2-1-50-150-5	63	84.8	338.1	77	84.7	199.6	B2-1-50-150-5	51	85.7	1103.4	91	85.5	191.9
A2-1-50-150-6	55	85.2	603.2	96	85.1	504.2	B2-1-50-150-6	55	85.9	1793.5	84	85.8	370.6
A2-1-50-150-8	54	85.5	835.6	72	85.5	1655.1	B2-1-50-150-8	60	86.2	4642.5	80	86.2	7600.1
A2-1-100-100-4	71	84.1	1812.8	131	83.8	971.0	B2-1-100-100-4	84	86.2	10522.5	169	85.8	901.8
A2-1-100-100-5	63	85.1	1037.2	113	84.5	2022.5	B2-1-100-100-5	-	-	-	162	86.5	1768.9
A2-1-100-100-6	56	85.3	831.5	91	84.7	5322.5	B2-1-100-100-6	-	-	-	149	87.0	2785.0
A2-1-100-100-8	69	85.2	6089.4	-	-	-	B2-1-100-100-8	-	-	-	-	-	-
Moyenne	52	85.4	815.7	76	85.2	762.4	Moyenne	51	86.0	1575.7	88	86.0	1051.6
C1-1-25-75-4	48	75.1	14.9	69	74.9	11.7	D1-1-25-75-4	23	84.2	4.7	59	82.5	8.3
C1-1-25-75-5	43	76.3	20.2	75	76.1	20.1	D1-1-25-75-5	28	83.7	9.3	47	83.3	8.3
C1-1-25-75-6	38	77.0	27.8	66	76.8	26.4	D1-1-25-75-6	31	83.8	17.0	43	83.8	15.5
C1-1-25-75-8	32	78.3	20.4	43	78.0	39.3	D1-1-25-75-8	30	84.5	14.9	33	84.4	46.9
C1-1-50-50-4	54	84.3	72.4	81	84.0	38.5	D1-1-50-50-4	57	83.9	166.5	94	83.6	40.4
C1-1-50-50-5	49	85.2	155.7	81	84.9	65.7	D1-1-50-50-5	53	84.6	425.6	93	84.4	81.7
C1-1-50-50-6	57	85.6	363.0	88	85.4	144.6	D1-1-50-50-6	59	85.2	862.6	85	85.0	191.8
C1-1-50-50-8	58	86.1	1883.5	63	86.1	649.5	D1-1-50-50-8	55	85.8	2104.0	80	85.6	1546.9
Moyenne	47	81	319.7	71	80.8	124.5	Moyenne	42	84.5	450.6	67	84.1	242.5

6.5.2 Problème de tournées couvrantes bi-objectif

La méthode de Artigues et al. (2018) permet la construction de la borne inférieure du BOMCTP. Le tableau 6.5 compare la précision et l'efficacité de la borne inférieure obtenue par notre méthode et celle de la littérature, appelée SOGA et présentée dans l'article d'Artigues et al. Les deux algorithmes sont lancés sur le même CPU et codés en C++, mais la méthode SOGA résout les problèmes linéaires et entiers avec CPLEX 12.9. Les instances utilisées pour le BOMCTP sont les mêmes que le MCTP avec une distance maximale de couverture η variable.

L'efficacité est évaluée avec le temps CPU utilisé pour trouver l'ensemble de la borne inférieure dans la colonne *Temps*. La précision est déterminée par deux valeurs : *Pt* représente le nombre de points dans la borne inférieure et *Hv* représente la mesure de l'hypervolume (cf. Section 1.2.4).

Les résultats montrent que notre méthode a des bornes inférieures plus précises pour toutes les instances. En effet, notre méthode génère plus de points dans la borne inférieure et cela peut expliquer la précision de nos bornes, ainsi que la différence de temps entre les deux méthodes. Par exemple, pour l'instance A2-1-100-100-5, notre méthode trouve 113

TABLE 6.6: Résultats de notre méthode bi-objectif pour résoudre le BOMCTP.

Instance	$ \mathcal{Y} $	Temps	Instance	$ \mathcal{Y} $	Temps	Instance	$ \mathcal{Y} $	Temps
A1-1-25-75-4	30	5.0	D1-1-25-75-6	29	11.1	C1-10-50-50-5	46	1313.3
A1-1-25-75-5	32	6.7	D1-1-25-75-8	29	103.4	C1-10-50-50-6	42	17398.6
A1-1-25-75-6	32	12.3	D1-1-50-50-4	61	39.9	D1-5-25-75-4	27	12.6
A1-1-25-75-8	29	54.7	D1-1-50-50-5	64	74.2	D1-5-25-75-5	25	538.5
A1-1-50-50-4	50	25.0	D1-1-50-50-6	62	353.8	D1-10-50-50-4	45	1824.9
A1-1-50-50-5	49	101.9	A1-5-25-75-4	24	7.1	D1-10-50-50-5	42	6561.1
A1-1-50-50-6	49	277.2	A1-5-25-75-5	21	9.1	A2-1-50-150-4	58	126.5
A1-1-50-50-8	50	1621.7	A1-5-25-75-6	18	11.3	A2-1-50-150-5	52	145.1
B1-1-25-75-4	34	5.5	A1-5-25-75-8	19	13.2	A2-1-50-150-6	53	1108.6
B1-1-25-75-5	37	7.6	A1-10-50-50-4	39	207.7	A2-1-100-100-4	90	7225.1
B1-1-25-75-6	37	13.9	A1-10-50-50-5	47	4955.1	A2-1-100-100-5	79	1714.6
B1-1-25-75-8	36	78.9	B1-5-25-75-4	21	4.5	A2-1-100-100-6	78	21354.7
B1-1-50-50-4	58	306.7	B1-5-25-75-5	21	6.4	A2-10-50-150-4	47	247.4
B1-1-50-50-4	58	306.7	B1-5-25-75-6	21	15.3	A2-10-50-150-5	37	1426.6
B1-1-50-50-6	55	477.7	B1-5-25-75-8	19	203.6	A2-10-50-150-6	29	3330.9
C1-1-25-75-4	39	9.7	B1-10-50-50-4	33	20.7	A2-20-100-100-4	60	4263.5
C1-1-25-75-5	38	9.8	B1-10-50-50-5	35	46.0	B2-1-50-150-4	57	192.4
C1-1-25-75-6	38	29.9	B1-10-50-50-6	29	337.5	B2-1-50-150-5	53	160.2
C1-1-25-75-8	36	659.1	C1-5-25-75-4	14	2.5	B2-1-50-150-6	51	328.2
C1-1-50-50-4	60	48.4	C1-5-25-75-5	15	5.0	B2-1-100-100-4	100	2092.4
C1-1-50-50-5	63	370.5	C1-5-25-75-6	14	13.8	B2-10-50-150-4	37	100.8
C1-1-50-50-6	60	5167.2	C1-5-25-75-8	12	11.0	B2-10-50-150-5	42	136.1
D1-1-25-75-4	29	4.1	C1-10-50-50-4	57	156.9	B2-10-50-150-6	43	324.7
D1-1-25-75-5	29	6.1						

points en 2022.5 secondes pour un hypervolume de 84.5%. La méthode SOGA, quant à elle, trouve 63 points en 1037.2 secondes pour un hypervolume de 85.1%. Il existe tout de même quelques instances pour lesquelles notre méthode est plus rapide. Par exemple, dans l'instance $D1-1-50-50-8$, notre méthode trouve 80 points en 1546.9 secondes et SOGA calcule 55 points en 2104.0 secondes.

Comme noté dans la Section 6.2.4, il n'y a pas de méthodes exactes pour le BOMCTP traité dans ce chapitre. Nous allons donc comparer SEM sur les instances utilisées pour le MCTP avec une distance maximale de couverture η variable. Les résultats sont reportés dans le tableau 6.6 qui présente une colonne $|\mathcal{Y}|$ pour le nombre de points non dominés et une colonne *Temps* pour le temps CPU en secondes pour obtenir l'ensemble complet minimal.

Nous pouvons observer que 70 instances sur 96 sont résolues à l'optimalité en 6 heures. Le nombre de points non dominés est assez élevé : entre 12 et 100, avec une moyenne de 40. Cela signifie que la variante du BOMCTP étudiée dans ce chapitre offre un compromis entre les deux objectifs. Par exemple, l'instance $D1-1-25-75-6$ contient 29 points non dominés et les deux points extrêmes sont les couples (*coût*; *distance de couverture*) suivants : (730; 2934) et (11047; 658). La solution optimale du MCTP trouvée pour l'instance $D1-1-25-75-6$ correspond au point non dominé (6651; 931) et est située au milieu de \mathcal{Y}_N .

Nous pouvons également noter que la valeur de la cardinalité maximale \bar{n} n'influence pas le nombre de points non dominés. Au contraire, plus le nombre de points de service

obligatoires est élevé et moins il y a de points non dominés; comme le montre les instances $B1-1-25-75-8$ et $B1-5-25-75-8$ qui ont respectivement 36 et 19 points non dominés. Enfin, plus il y a de points de service optionnels et plus \mathcal{V}_N contient de points non dominés; comme le montre les instances $A1-1-25-75-4$ et $A1-1-50-50-4$ qui ont respectivement 30 et 50 points non dominés.

6.5.3 Conclusion

Dans un premier temps, les résultats expérimentaux montrent que la méthode exacte pour le MCTP surpasse les méthodes courantes de l'état de l'art. Les nouvelles bornes de complétion g montrent également leur pertinence puisque le nombre de routes éliminées est plus important. Cependant, le nombre d'opérations supplémentaires à effectuer pour les obtenir peut avoir un impact négatif sur le temps de résolution comparé aux bornes f , plus rapides à calculer.

Ensuite, l'ensemble des bornes inférieures pour le BOMCTP sont plus précises que celles proposées dans l'état de l'art. Ces bornes précises permettent de mettre en place les méthodes bi-objectif basées sur l'approche ε -contrainte. L'application de l' ε -contrainte sur ce problème est particulièrement adaptée grâce à l'objectif de type *min-max*. Cependant, cela rend caduque la majorité des techniques de ISEM, présentées dans le Chapitre 4.

6.6 Conclusion du chapitre

Dans ce chapitre, nous avons proposé une méthode exacte pour résoudre le problème de tournées couvrantes. Elle est basée sur la formulation de partitionnement, utilise la relaxation ng-route pour le problème de *pricing* et inclut une formulation stabilisée. Nous décrivons toutes les spécificités de la génération de colonnes pour le MCTP comme l'extension d'une étiquette, les règles de dominance et particulièrement les bornes de complétion. L'efficacité de notre méthode a été prouvée lors de la comparaison avec la méthode de l'état-de-l'art pour le MCTP. Quelques instances ont été fermées grâce à notre algorithme. De plus, la méthode mono-objectif a été intégrée dans une méthode ε -contrainte par étapes pour résoudre le problème de tournées couvrantes bi-objectif qui vise à minimiser la distance parcourue et la distance maximale de couverture autorisée pour les véhicules. La taille de l'ensemble des points non dominés de chaque instance montre que le problème bi-objectif propose un compromis intéressant entre ses deux objectifs. La résolution de problèmes de type min-max est efficace avec la méthode ε -contrainte par étapes. Cependant, l'utilisation de la méthode ε -contrainte par étapes améliorée n'est pas possible pour les problèmes où la contrainte ε n'apparaît pas dans le problème maître.

Conclusions et perspectives

Conclusions

Cette thèse propose une méthode exacte pour les problèmes de tournées de véhicules bi-objectif. Les contributions se divisent en deux axes principaux : l'application de la génération de colonnes dans une méthode générique pour l'ensemble des problèmes de tournées de véhicules bi-objectif et l'accélération de cette méthode grâce à de multiples techniques basées sur la méthode de l' ε -contrainte. Plusieurs problèmes sont considérés dans cette thèse et montrent la généralité de la méthode.

Le Chapitre 1 présente les notions d'optimisation combinatoire multi-objectif nécessaires à la compréhension de la suite de la thèse, ainsi que les principales approches de résolution exacte. Ce chapitre met en exergue la variété d'approches pouvant être utilisées pour résoudre des problèmes multi-objectif. Les méthodes peuvent être notamment scindées en deux groupes : celles basées sur la scalarisation des objectifs et celles basées sur l'espace des solutions.

Le Chapitre 2 définit les problèmes de tournées de véhicules mono-objectif et les méthodes exactes classiques de résolution. La technique de génération de colonnes est utilisée dans les méthodes de l'état-de-l'art, ce qui démontre son efficacité pour ce type de problèmes. L'état de l'art des méthodes exactes pour les problèmes de tournées de véhicule multi-objectif est également donné dans ce chapitre. En conclusion, la plupart des méthodes de la littérature utilisent des méthodes basées sur la scalarisation des objectifs, notamment l'approche ε -contrainte. Cependant, très peu de méthodes utilisent la génération de colonnes.

À l'issue de ces deux chapitres, un axe de recherche se dessine pour l'élaboration d'une méthode exacte pour les tournées de véhicules bi-objectif qui est l'utilisation d'une méthode de scalarisation combinée avec de la génération et de l'énumération de colonnes. Deux questions majeures se posent alors :

- Quelle est la méthode de scalarisation la plus efficace pour ces problèmes ?
- Comment la combiner avec les techniques de génération et d'énumération de colonnes ?

Ainsi, le Chapitre 3 répond à ces questions en utilisant les deux méthodes de scalarisation les plus utilisées, soit la méthode ε -contrainte et la méthode à deux phases, avec la génération de colonnes. Pour chacune des approches de scalarisation, deux méthodes sont construites. Des expérimentations sur un problème de tournées de véhicules bi-objectif avec fenêtres de temps et deux coûts entiers sur les arêtes valident l'approche ε -contrainte. Cependant, l'aspect le plus critique de la méthode est d'utiliser l'énumération de colonnes sur de petites zones de l'espace des objectifs.

La méthode du Chapitre 3, nommée méthode ε -contraintes par étapes (*Stepwise ε -constraint Method* = SEM), est analysée en détail dans le Chapitre 4. Plus l'écart entre la borne inférieure et la borne supérieure est grand, plus il y a de colonnes générées par

l'énumération des colonnes et plus le temps de résolution d'une itération est longue. Ainsi, plusieurs axes d'amélioration sont envisagés :

Réduction du gap d'optimalité. Tout d'abord, l'ajout de coupes classiques du problème de tournées de véhicules mono-objectif a été envisagée, mais cela ne permet pas de renforcer la borne inférieure efficacement. Lors du travail sur des coûts objectif entiers, ce gap peut être réduit d'une unité.

Réduction du nombre de colonnes générées. Une nouvelle technique de construction de la borne inférieure réduit le nombre de colonnes générées pour le calcul des bornes. De plus, la validation prématurée des points non dominés et l'utilisation de plusieurs directions de recherche réduisent le nombre de colonnes lors de l'énumération des colonnes pour trouver les points non dominés.

Traitement des colonnes. Une gestion adaptée des colonnes durant l'énumération permet de réduire les traitements multiples des colonnes.

La plupart de ces techniques sont basées sur les propriétés structurelles des formulations contenant une contrainte bornant l'un des objectifs. Les techniques les plus efficaces ajoutées à SEM forment la méthode ε -contrainte par étapes améliorée (*Improved Stepwise ε -constraint Method=ISEM*) et se montrent très performantes pour résoudre le problème de tournées de véhicules bi-objectif avec deux coûts entiers sur les arêtes.

Dans le Chapitre 5, SEM et ISEM sont appliquées sur un autre problème : le problème d'orientation par équipes bi-objectif minimisant un coût sur les arcs et maximisant un profit collecté entier. L'efficacité des techniques est démontrée, ainsi que la généralité de SEM et ISEM.

Enfin, le Chapitre 6 traite le problème de tournées couvrantes mono-objectif et bi-objectif. Tout d'abord, la génération et l'énumération de colonnes du problème dans une version mono-objective sont présentées et démontrent leur efficacité par rapport à l'état de l'art. Le problème bi-objectif peut se réduire à une succession de problèmes mono-objectifs où la contrainte ε est éliminée astucieusement. La méthode SEM peut être facilement appliquée. Cependant, la disparition de la contrainte bornant l'un des objectifs rend caduque l'utilisation des techniques de ISEM et montre qu'elle n'est pas adaptée au problème bi-objectif de type min-max.

En conclusion, une méthode efficace et générique a été proposée pour résoudre les problèmes de tournées de véhicules bi-objectif. Des techniques originales permettent d'accélérer l'utilisation de la méthode ε -contrainte lorsque la contrainte bornant l'un des objectifs est présente dans la formulation.

Publications

Le contenu du Chapitre 3 a été présenté dans des conférences internationales (ISCO 2018, Odysseus 2018, VEROLOG 2017) et nationales (ROADEF 2018, MOPGP 2017). La conférence ISCO a donné lieu à une publication dans les *Lecture Notes in Computer Science*. De plus, ISEM et les résultats expérimentaux des Chapitres 4 et 5 sont présentés dans un article soumis à *Computers & Operations Research (C&OR)* et intitulé '*An ε -constraint column generation-and-enumeration algorithm for bi-objective vehicle*

routing problems'. Une partie de ce travail a également été exposée à la conférence internationale VEROLOG 2019. Enfin, les méthodes de résolution du MCTP et du BOMCTP sont présentées dans un article de la revue *European Journal of Operations Research* (EJOR) intitulé '*Exact Methods for Mono-Objective and Bi-Objective Multi-Vehicle Covering Tour Problems*'.

Perspectives

Améliorations de la méthode. Tout d'abord, d'autres techniques peuvent être ajoutées à SEM pour en améliorer encore plus l'efficacité. Le fait que les coupes classiques mono-objectif ne soient pas efficaces n'empêchent pas de trouver d'autres coupes inhérentes à la contrainte bornant l'un des objectifs. Par exemple, une coupe possible est de limiter le nombre de clients dans un sous-ensemble de nœuds par rapport au coût objectif borné entrant. Ensuite, l'utilisation d'une métaheuristique permettrait de générer des bornes supérieures de qualité en un temps limité pour SEM et ISEM. Une telle méthode peut être développée pour les formulations mono-objectif et appliquée à chaque nouvelle valeur de ε . Au contraire, la métaheuristique peut être conçue pour générer un ensemble de solutions approchées du problème bi-objectif au début ou au cours de la résolution de SEM et ISEM.

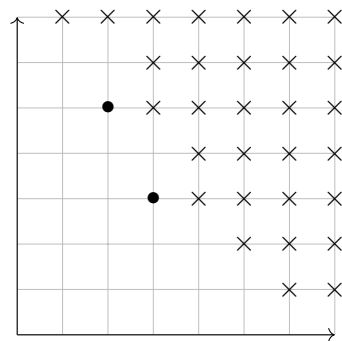
Preuve de la généralité. L'application à d'autres familles de problèmes comme l'ordonnement ou le *bin packing* pourrait démontrer la généralité de ISEM et la pertinence des techniques proposées. Avec l'énumération des colonnes, des problèmes présentant de faibles gap d'optimalité semblent plus indiqués à cette méthode.

Conception d'une matheuristique. La résolution des problèmes bi-objectif à l'optimalité peut être une tâche longue et fastidieuse. Concevoir une matheuristique basée sur ISEM pour obtenir une approximation de l'ensemble des points non dominés est une perspective intéressante. Pour que cette approximation soit de qualité, deux caractéristiques, représentées dans la Figure 6.2, doivent être prises en compte :

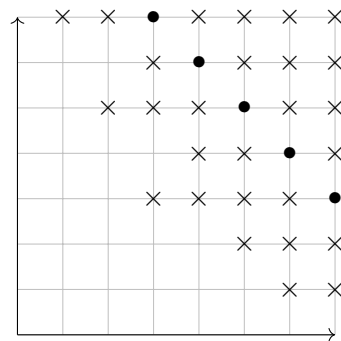
- l'intensification qui est la propriété d'une méthode à converger vers le front de Pareto ;
- la diversification qui est la capacité d'une méthode à estimer l'ensemble du front.

Stopper la résolution de la méthode au bout d'un certain temps de calcul n'est pas une solution viable. En effet, il n'y a aucune garantie sur la qualité de la solution courante et les points trouvés risquent d'être restreints dans une zone de l'espace des objectifs ne permettant pas une bonne diversité des solutions. Ne générer qu'une partie du front est également une approche limitant la diversité des solutions. D'autres pistes sont envisageables ; par exemple en autorisant un certain gap d'optimalité pour valider un point non dominé, ou en limitant le nombre de colonnes à générer dans le gap. Cependant, elles peuvent affecter l'intensification du front de Pareto obtenu pour limiter le temps de résolution.

Extension du nombre d'objectifs. Une dernière perspective est d'étendre la méthode ISEM sur des problèmes combinatoires à plus de deux objectifs. En effet, des méthodes ont déjà été proposées pour étendre l'approche ε -contrainte. Cependant, l'application sur des problèmes à variante mono-objectif \mathcal{NP} -complet risque d'être coûteuse en temps de résolution. La méthode *quadrant shrinking* de Boland et al. (2017) est une méthode efficace



(a) Ensemble estimé avec une bonne intensification.



(b) Ensemble estimé avec une bonne diversification.

FIGURE 6.2: Représentation des qualités d'une estimation d'un front de Pareto où les points sont des points de l'ensemble estimé et les croix sont des points associés à des solutions réalisables.

pour les problèmes combinatoires tri-objectif. Elle est basée sur une exploration astucieuse de l'espace des objectifs en projetant les points sur un espace bidimensionnel pour éliminer efficacement les zones à explorer. Combiner la génération de colonnes et les techniques de ISEM à la méthode *quadrant shrinking* semble être une piste possible.

Liste de publications

Les travaux de cette thèse ont abouti à des communications scientifiques et des articles.

Revues

1. Estèle Glize, Roberto Roberti, Nicolas Jozefowicz, Sandra Ulrich Ngueveu. *Exact Methods for Mono-Objective and Bi-Objective Multi-Vehicle Covering Tour Problems*. In *European Journal of Operations Research* (EJOR).
2. Estèle Glize, Nicolas Jozefowicz, Sandra Ulrich Ngueveu. *An ε -constraint column generation-and-enumeration algorithm for bi-objective vehicle routing problems*. In *Computers & Operations Research* (C&OR). **Soumis**.

Conférence internationale avec actes

1. Estèle Glize, Nicolas Jozefowicz, Sandra Ulrich Ngueveu. *An exact column generation-based algorithm for Bi-Objective Vehicle Routing Problems*. In International Symposium on Combinatorial Optimization (ISCO 2018), Marrakech, Maroc, April 2018.

Conférences internationales avec actes à diffusion limitée

1. Estèle Glize, Nicolas Jozefowicz, Sandra Ulrich Ngueveu. *Branch-and-price algorithms for bi-objective vehicle routing problems*. In ODYSSEUS, the Seventh International Workshop on Freight Transportation and Logistics, Sardaigne, Italie, June 2018.
2. Estèle Glize, Nicolas Jozefowicz, Sandra Ulrich Ngueveu. *Branch-and-price algorithms for the Bi-Objective Vehicle Routing Problem with Time Windows*. In Multiple Objective Programming and Goal Programming (MOPGP 2017), Metz, France, October 2017.

Conférences internationales sans acte

1. Estèle Glize, Nicolas Jozefowicz, Sandra Ulrich Ngueveu. *Exact Method for Bi-objective Vehicle Routing Problems*. In the 7th meeting of EURO Working group on Vehicle Routing and Logistics optimization (VeRoLog 2019), Seville, Espagne, June 2019.
2. Estèle Glize, Nicolas Jozefowicz, Sandra Ulrich Ngueveu. *Exact methods for the Bi-Objective Vehicle Routing Problem*. In the 6th meeting of EURO Working group on Vehicle Routing and Logistics optimization (VeRoLog 2017), Amsterdam, Pays-Bas, July 2017.

Conférence nationale sans acte

1. Estèle Glize, Nicolas Jozefowicz, Sandra Ulrich Ngueveu. *Algorithme de branch-and-price pour les problèmes de tournées de véhicules bi-objectifs*. In ROADEF, 19^{ème} congrès annuel de la société Française de Recherche Opérationnelle et d'Aide à la Décision, Lorient, France, Février 2018.

Résultats expérimentaux du problème de tournées de véhicules bi-objectif avec fenêtres de temps

Cette annexe présente les résultats expérimentaux du Chapitre 3 et 4 sur les instances du problème de tournées de véhicules bi-objectif avec fenêtres de temps et deux coûts entiers sur les arêtes. Chaque instance est le résultat de la concaténation de deux instances de Solomon¹. La première instance de Solomon nommée A fournit le coût c^1 sur les arcs qui sera également le temps de parcours des véhicules, les fenêtres de temps, les temps de service et les demandes de chaque client. La seconde instance nommée B ne fournit que les seconds coûts c^2 sur les arcs. Chaque nouvelle instance est marquée A_B_X , avec X le nombre de nœuds dans le graphe.

A.1 Comparaison des méthodes d'exploration de l'espace des objectifs

Les tableaux A.1, A.2, A.3 et A.4 présentent les temps CPU en secondes des quatre méthodes du Chapitre 3 pour les instances qui ont été résolues en quatre heures. La dernière ligne fait la moyenne du temps des instances qui ont été résolues par toutes les méthodes.

A.2 Comparaison de SEM et ISEM

Les tableaux A.5, A.6 et A.7 présentent les temps CPU en secondes des méthodes SEM et ISEM pour les instances qui ont été résolues en quatre heures. La dernière ligne fait la moyenne du temps des instances qui ont été résolues par les deux méthodes. Dans ces tableaux, T_{valid} , T_{save} et T_{mult_dual} indiquent respectivement le pourcentage d'utilisation de chaque technique sur l'ensemble des points non dominés.

1. Instances de Solomon disponibles sur <https://www.sintef.no/nearp>

TABLE A.1: Résultats expérimentaux des méthodes sur les instances de type R_C et R_RC pour $|V| = 25$.

Instance	Temps M1	Temps M2	Temps M3	Temps M4
R101_C1_25	2.1	8.8	2.4	7.0
R101_C2_25	2.1	8.4	2.6	5.5
R101_RC1_25	2.7	13.1	3.7	8.6
R102_C1_25	729.6	252.4	1575.7	66.1
R102_C2_25	658.4	41.9	1631.8	29.3
R102_RC1_25	712.4	40.7	1417.3	24.5
R103_C1_25	10846.0	39.0	3848.1	41.5
R103_C2_25	20375.5	54.9	12491.1	56.1
R103_RC1_25	20021.4	66.2	13274.3	70.0
R104_C1_25	-	68.7	7841.0	102.9
R104_RC1_25	-	966.7	-	643.4
R104_C2_25	-	179.3	-	516.4
R105_C1_25	24.0	36.6	31.1	25.5
R105_C2_25	19.0	28.1	29.6	21.9
R105_RC1_25	18.5	25.7	30.5	18.4
R106_C1_25	4518.4	91.3	4653.8	567.7
R106_C2_25	4717.3	119.6	4459.5	233.0
R106_RC1_25	6235.5	241.4	4276.1	1044.5
R107_C1_25	-	1827.1	-	1295.8
R107_C2_25	-	262.9	-	485.3
R107_RC1_25	-	9619.0	-	540.1
R108_C1_25	-	76.1	5449.1	1865.7
R108_C2_25	-	-	-	487.1
R108_RC1_25	-	-	-	4085.4
R109_C1_25	172.9	31.0	104.3	34.7
R109_C2_25	156.5	45.2	426.5	23.8
R109_RC1_25	215.7	61.1	460.5	46.6
R110_C1_25	1599.1	330.7	2514.5	477.0
R110_C2_25	1234.5	159.6	2112.3	529.8
R110_RC1_25	1747.0	109.3	2308.1	104.7
R111_C1_25	13808.6	448.9	4974.9	2344.4
R111_C2_25	-	406.9	12882.3	2302.8
R111_RC1_25	-	280.6	11105.6	469.6
R112_C1_25	-	407.2	17462.9	2466.3
R112_C2_25	-	964.0	-	2830.8
R112_RC1_25	-	3484.0	-	1100.7
R201_C2_25	-	4963.1	-	-
R201_RC1_25	-	1500.0	-	-
Moyenne	3991.7	102.5	2755.9	262.8
Instances Fermées	22	36	27	36

TABLE A.2: Résultats expérimentaux des méthodes sur les instances de type RC_C et RC_R pour $|V| = 25$.

Instance	Temps M1	Temps M2	Temps M3	Temps M4
RC101_C1_25	7.6	7.5	5.9	10.5
RC101_C2_25	9.1	9.2	7.8	12.3
RC101_R1_25	16.0	21.4	15.2	25.5
RC102_C1_25	59.4	20.3	8.9	16.6
RC102_C2_25	269.8	32.1	133.0	36.1
RC102_R1_25	330.9	58.0	217.0	112.4
RC103_C1_25	-	39.7	526.6	39.4
RC103_C2_25	4290.7	99.7	2553.0	74.9
RC103_R1_25	4016.9	44.5	1173.7	74.8
RC104_C1_25	-	183.0	130.3	139.8
RC104_C2_25	-	28.5	7318.2	21.3
RC104_R1_25	-	102.3	14928.9	360.8
RC105_C1_25	63.2	21.6	18.5	22.8
RC105_C2_25	72.7	14.1	5.9	10.0
RC105_R1_25	53.5	19.2	33.7	39.7
RC106_C1_25	81.4	13.2	7.4	11.6
RC106_C2_25	120.5	32.7	88.6	39.4
RC106_R1_25	115.0	24.5	66.5	29.7
RC107_C1_25	33.9	19.2	8.4	20.3
RC107_C2_25	1648.0	28.5	1693.5	18.9
RC107_R1_25	1976.9	257.2	1410.1	1133.5
RC108_C1_25	175.4	39.8	8.6	23.2
RC108_C2_25	297.6	43.2	25.2	29.6
RC108_R1_25	19983.5	90.6	10701.0	1359.1
RC201_C1_25	-	1491.3	404.6	-
RC201_C2_25	-	252.9	-	-
RC205_C2_25	-	724.4	-	-
RC205_R1_25	-	1067.7	-	-
Moyenne	1681.1	44.8	909.1	155.0
Instances Fermées	20	28	25	24

TABLE A.3: Résultats expérimentaux des méthodes sur les instances de type C_C , C_R et C_{RC} pour $|V| = 25$.

Instance	Temps M1	Temps M2	Temps M3	Temps M4
C101_C2_25	6217.3	19.6	68.4	66.8
C101_RC1_25	3220.8	11.3	45.6	53.3
C101_R1_25	9612.0	68.5	1760.8	945.0
C102_C2_25	-	72.1	10187.6	103.6
C102_RC1_25	-	403.2	2063.7	17532.3
C102_R1_25	-	1428.6	-	-
C103_C2_25	-	5984.2	-	1418.4
C103_RC1_25	-	412.8	9274.5	14745.9
C103_R1_25	-	7095.5	-	-
C105_C2_25	620.4	22.1	109.0	168.3
C105_RC1_25	-	608.6	1939.5	5124.0
C105_R1_25	-	320.3	21587.9	-
C106_C2_25	7689.3	24.9	68.0	29.7
C106_RC1_25	3331.6	14.2	58.6	60.5
C106_R1_25	14198.0	67.5	2380.5	339.8
C107_C2_25	659.6	26.9	656.5	374.6
C107_RC1_25	-	10625.7	4423.4	-
C107_R1_25	-	808.8	-	4739.4
C108_C2_25	-	162.1	303.9	12112.1
C108_RC1_25	-	437.0	1261.0	8365.2
C109_C2_25	-	762.2	4387.1	-
C109_RC1_25	-	13488.4	3761.3	-
C109_C1_25	-	762.2	4387.1	-
C201_RC1_25	-	6447.8	-	-
Moyenne	5693.6	31.9	634.4	254.8
Instances Fermées	8	24	19	16

TABLE A.4: Résultats expérimentaux des méthodes sur les instances avec $|V| = 50$.

Instance	Temps M1	Temps M2	Temps M3	Temps M4
R101_C1_50	122.4	255.1	124.9	137.3
R101_C2_50	135.1	264.2	141.9	143.9
R102_C1_50	-	1874.0	-	7845.7
R102_C2_50	-	704.0	-	-
R103_C1_50	-	12118.9	-	-
R103_C2_50	-	3858.7	-	-
R105_C1_50	3291.7	961.2	3212.1	2432.2
R105_C2_50	5435.0	864.1	3624.4	3534.0
R106_C1_50	-	2609.3	-	-
R109_C1_50	-	2084.9	-	-
R109_C2_50	-	1637.9	-	-
C101_C2_50	-	68.5	288.9	296.0
C101_R1_50	-	2187.4	-	-
C102_C2_50	-	756.4	-	-
C107_C2_50	-	295.8	3254.8	8152.2
C108_C2_50	-	2759.4	-	-
Moyenne	2246.1	586.2	1775.8	1561.9
Instances Fermées	4	16	6	7

TABLE A.5: Résultats expérimentaux détaillés de SEM et de ISEM sur les instances de type R_C et R_RC pour $|V| = 25$.

Instance	SEM		ISEM		
	Temps	Temps	T_{valid}	T_{save}	T_{mult_dual}
R101_C1_25	8.7	7.5	17	42	0
R101_C2_25	8.4	8.8	39	22	4
R101_RC1_25	12.8	11.3	49	17	3
R102_C1_25	129.4	67.0	9	67	0
R102_C2_25	42.9	45.7	31	35	5
R102_RC1_25	40.1	31.5	49	25	0
R103_C1_25	37.6	34.3	49	19	0
R103_C2_25	58.5	49.7	26	26	0
R103_RC1_25	66.6	62.9	27	40	0
R104_C1_25	64.2	58.7	17	28	0
R104_RC1_25	576.8	324.6	23	39	4
R104_C2_25	103.0	86.6	24	31	0
R105_C1_25	34.8	32.5	21	45	0
R105_C2_25	26.7	24.0	19	41	0
R105_RC1_25	24.9	22.2	20	43	0
R106_C1_25	86.5	84.6	22	40	0
R106_C2_25	103.6	100.5	22	22	0
R106_RC1_25	231.5	169.6	28	32	0
R107_C1_25	975.3	436.7	31	36	0
R107_C2_25	189.1	171.2	14	28	0
R107_RC1_25	10031.4	688.8	31	38	5
R108_C1_25	70.8	73.2	9	34	0
R108_C2_25	13452.6	476.7	31	24	0
R108_RC1_25	-	839.2	14	35	4
R109_C1_25	24.3	23.9	27	27	0
R109_C2_25	34.6	28.9	31	24	0
R109_RC1_25	48.1	41.4	26	37	0
R110_C1_25	220.8	210.8	9	35	0
R110_C2_25	103.8	96.2	19	22	0
R110_RC1_25	85.0	88.8	39	37	0
R111_C1_25	162.0	295.3	4	63	0
R111_C2_25	305.3	260.4	17	28	0
R111_RC1_25	208.3	152.3	35	31	0
R112_C1_25	149.1	136.6	14	41	0
R112_C2_25	464.9	344.6	9	24	0
R112_RC1_25	193.4	135.8	18	34	3
R201_C1_25	-	2829.5	57	30	57
R201_C2_25	402.5	404.7	45	23	3
R201_RC1_25	131.0	118.7	35	33	0
R202_RC1_25	4423.8	1787.0	49	13	41
R203_RC1_25	-	9283.4	28	5	30
R205_RC1_25	766.3	1332.4	72	18	81
R206_RC1_25	-	4800.3	58	27	55
R207_RC1_25	-	8840.0	67	15	62
R209_RC1_25	-	2748.6	64	20	69
R210_RC1_25	-	8556.1	23	9	28
Moyenne	874.3	218.6	27	32	4
Instances fermées	39	46			

TABLE A.6: Résultats expérimentaux détaillés de SEM et de ISEM sur les instances de type RC_C et RC_R pour $|V| = 25$.

Instance	SEM		ISEM		
	Temps	Temps	T_{valid}	T_{save}	T_{mult_dual}
RC101_C1_25	6.9	6.8	11	22	0
RC101_C2_25	8.8	7.8	40	7	0
RC101_R1_25	18.0	17.1	22	33	0
RC102_C1_25	18.9	22.0	39	22	6
RC102_C2_25	25.8	32.3	8	33	13
RC102_R1_25	49.0	45.5	37	33	0
RC103_C1_25	37.9	45.5	47	13	10
RC103_C2_25	90.9	90.4	16	34	16
RC103_R1_25	42.0	54.9	41	19	11
RC104_C1_25	88.3	101.5	55	14	5
RC104_C2_25	26.7	34.2	20	33	7
RC104_R1_25	81.6	95.6	54	10	12
RC105_C1_25	20.6	28.8	33	29	19
RC105_C2_25	13.5	18.3	48	9	13
RC105_R1_25	15.7	17.2	35	20	5
RC106_C1_25	12.4	11.3	33	13	0
RC106_C2_25	30.0	38.6	21	32	14
RC106_R1_25	21.4	25.4	21	25	8
RC107_C1_25	19.0	45.3	13	7	40
RC107_C2_25	31.1	46.3	35	19	15
RC107_R1_25	156.2	119.9	24	24	2
RC108_C1_25	38.4	52.8	44	6	11
RC108_C2_25	41.4	54.2	30	20	10
RC108_R1_25	69.6	58.9	28	24	3
RC201_C1_25	187.1	164.1	66	8	5
RC201_C2_25	93.9	63.0	59	4	0
RC201_R1_25	-	303.7	47	24	3
RC202_C2_25	3726.0	1252.6	57	20	55
RC203_C2_25	-	4560.2	68	14	71
RC205_C1_25	11232.8	2960.9	70	16	72
RC205_C2_25	288.9	280.5	46	18	7
RC205_R1_25	370.6	256.0	21	29	2
RC206_C1_25	-	2896.9	43	49	55
RC206_C2_25	-	1466.1	54	19	30
RC206_R1_25	2719.1	844.2	49	38	57
Moyenne	631.7	222.3	36	20	13
Instances fermées	31	35			

TABLE A.7: Résultats expérimentaux détaillés de SEM et de ISEM sur les instances de type C_C , C_{RC} et R_{RC} pour $|V| = 25$.

Instance	SEM	ISEM			
	Temps	Temps	T_{valid}	T_{save}	T_{mult_dual}
C101_C2_25	16.9	14.0	17	17	0
C101_RC1_25	10.9	9.8	43	14	0
C101_R1_25	52.4	54.5	57	13	4
C102_C2_25	65.4	94.0	6	18	6
C102_RC1_25	132.2	240.6	63	6	56
C102_R1_25	841.1	1017.0	35	20	4
C103_C2_25	1793.8	1991.0	14	23	27
C103_RC1_25	182.5	369.4	58	17	50
C103_R1_25	2686.3	1004.8	40	15	33
C104_RC1_25	11070.9	-	-	-	-
C104_R1_25	1825.5	2273.3	33	26	21
C105_C2_25	15.8	14.7	0	25	0
C105_RC1_25	133.8	302.8	69	13	75
C105_R1_25	241.3	222.5	31	19	4
C106_C2_25	20.4	18.6	17	17	0
C106_RC1_25	10.7	8.6	43	14	0
C106_R1_25	61.5	63.7	41	18	5
C107_C2_25	22.7	27.5	20	20	0
C107_RC1_25	306.5	1011.4	63	21	63
C107_R1_25	465.7	359.6	26	30	2
C108_C2_25	63.4	53.6	53	24	0
C108_RC1_25	153.3	263.8	48	28	52
C108_R1_25	-	2260.4	47	16	25
C109_C2_25	189.1	134.2	42	21	0
C109_RC1_25	308.2	355.4	17	19	13
C109_R1_25	4756.8	1717.3	51	22	42
C201_RC1_25	2218.6	453.3	47	33	80
C201_R1_25	1458.4	1413.6	85	9	94
C201_C1_25	-	1173.6	21	36	57
C202_RC1_25	5471.9	2092.9	69	14	72
C202_R1_25	-	11508.3	76	12	86
C202_C1_25	-	2644.0	62	17	72
C203_RC1_25	-	5295.0	68	13	71
C203_C1_25	-	8798.0	50	25	67
C204_RC1_25	10430.5	12097.2	58	25	67
C205_RC1_25	-	1145.1	74	13	78
C205_R1_25	-	5970.1	66	16	59
C205_C1_25	-	1130.6	59	15	47
C206_RC1_25	-	773.7	50	26	41
C206_C1_25	-	553.9	55	14	32
C207_C1_25	-	2485.9	65	12	62
C208_RC1_25	3228.0	3254.6	77	14	86
C208_R1_25	-	6816.3	55	12	41
C208_C1_25	-	807.8	61	9	43
Moyenne	1281.5	1066.7	42	19	30
Instances fermées	30	43			

Résultats expérimentaux du problème de la course d'orientation par équipes bi-objectif avec fenêtre de temps

Cette annexe présente les résultats expérimentaux du Chapitre 5 sur les instances du problème de la course d'orientation par équipes bi-objectif avec fenêtre de temps. Les tableaux B.1 et B.2 présentent les caractéristiques des différentes instances avec le nombre de points non dominés $|ND|$, le nombre de points non supportés $|NS|$ et le gap moyen entre la borne inférieure et la solution optimale à chaque étape de SEM en pourcentage, pour chaque valeur ε . La formule du gap G pour une valeur ε fixée est :

$$G = 100 - \frac{c(LB)*100}{c(O)} (\%);$$

Les tableaux indiquent également les temps CPU en secondes pour la méthode de Parragh et Tricoire [Parragh et Tricoire \(2019\)](#) (*B&B*), SEM, SEM avec T_{valid} et ISEM. Pour SEM avec T_{valid} et ISEM, les colonnes T_{lb} , T_{valid} , T_{save} et T_{mult_dual} montrent le pourcentage respectif d'utilisation de chaque technique par rapport au nombre de points non dominés. Enfin, la ligne *Moyenne* fait la moyenne des caractéristiques des instances qui ont été résolues par toutes les méthodes et la ligne *Fermées* représente le nombre d'instances fermées par chaque méthode si elles ne le sont pas toutes.

TABLE B.1: Résultats expérimentaux des méthodes sur les instances de type *c101* et *pr01*.

Description instances			B&B	SEM	SEM + T_{valid}			ISEM				
Instance	$ ND $	$ NS $	MGap	Temps	Temps	Temps	T_{valid}	Temps	T_{lb}	T_{valid}	T_{save}	T_{mult_dual}
c101_1_15	18	10	24.9	9.8	11.9	9.7	61.1	7.8	61.1	61.1	5.6	0.0
c101_1_20	20	10	24.5	44.9	23.3	13.6	70.0	9.7	60.0	70.0	0.0	0.0
c101_1_25	21	9	21.7	94.9	30.8	14.9	71.4	10.8	66.7	66.7	4.8	0.0
c101_1_30	21	9	21.7	197.2	37.4	22.7	66.7	15.1	66.7	66.7	4.8	0.0
c101_1_35	23	9	20.5	263.9	47.7	24.3	69.6	16.0	69.6	69.6	4.3	0.0
c101_2_15	24	13	20.6	44.9	28.9	19.0	58.3	16.6	58.3	58.3	4.2	0.0
c101_2_20	32	16	17.8	755.3	56.8	41.1	62.5	38.0	56.3	62.5	0.0	0.0
c101_2_25	37	20	14.9	5931.0	154.7	103.1	56.8	96.3	51.4	54.1	2.7	2.7
c101_2_30	37	20	14.9	-	135.2	94.5	59.5	95.0	54.1	56.8	2.7	2.7
c101_2_35	41	18	13.4	-	78.2	58.7	61.0	49.9	58.5	61.0	7.3	0.0
c101_3_15	24	13	20.6	40.7	25.7	20.9	58.3	18.7	58.3	58.3	4.2	0.0
c101_3_20	33	16	17.4	844.4	356.1	184.8	63.6	442.1	57.6	54.5	0.0	27.3
c101_3_25	42	21	13.7	-	2712.7	1195.7	59.5	2193.3	57.1	52.4	2.4	26.2
c101_3_30	-	-	-	-	-	-	-	-	-	-	-	-
c101_3_35	-	-	-	-	-	-	-	-	-	-	-	-
c101_4_15	24	13	20.6	44.3	27.7	22.7	58.3	18.5	58.3	58.3	4.2	0.0
c101_4_20	33	16	17.4	838.7	372.8	139.5	60.6	477.6	57.6	51.5	3.0	27.3
c101_4_25	42	21	13.7	-	4110.8	3318.6	59.5	4331.5	57.1	52.4	2.4	26.2
c101_4_30	-	-	-	-	-	-	-	-	-	-	-	-
c101_4_35	-	-	-	-	-	-	-	-	-	-	-	-
Moyenne	26	13	20.2	759.2	97.8	51.3	63.1	97.2	60.1	61.0	3.1	4.8
Fermées				12	16	16		16				
pr01_1_15	26	7	20.3	4.0	22.3	11.0	65.4	9.4	73.1	61.5	11.5	0.0
pr01_1_20	39	5	15.5	19.6	68.8	40.7	84.6	12.4	84.6	84.6	5.1	0.0
pr01_1_25	72	7	9.6	135.6	128.2	70.5	86.1	21.5	90.3	84.7	5.6	0.0
pr01_1_30	93	7	8.2	1222.6	1468.8	405.7	87.1	56.2	91.4	84.9	7.5	0.0
pr01_1_35	85	9	0.0	3345.3	1678.2	664.9	85.9	173.3	88.2	82.4	10.6	0.0
pr01_2_15	36	8	15.5	3.4	25.0	17.0	69.4	12.6	77.8	69.4	11.1	0.0
pr01_2_20	72	13	11.8	36.9	126.9	86.9	66.7	78.5	79.2	69.4	9.7	1.4
pr01_2_25	121	13	7.1	341.4	2905.7	1796.1	81.8	134.1	84.3	81.0	5.0	0.0
pr01_2_30	153	10	6.5	1800.9	757.8	553.9	87.6	476.7	89.5	83.7	7.8	2.0
pr01_2_35	-	-	-	-	-	-	-	-	-	-	-	-
pr01_3_15	37	8	15.2	2.7	29.8	17.1	70.3	12.2	75.7	70.3	10.8	0.0
pr01_3_20	67	10	12.6	46.6	68.0	48.0	67.2	32.7	79.1	68.7	10.4	0.0
pr01_3_25	134	11	6.5	201.6	202.5	138.0	88.1	56.8	91.8	86.6	4.5	0.0
pr01_3_30	169	10	5.9	1297.0	551.6	380.7	89.3	178.5	93.5	89.3	4.7	0.0
pr01_3_35	195	15	4.3	-	2865.6	2562.4	79.5	1657.0	88.7	81.5	8.7	1.5
pr01_4_15	37	8	15.2	2.7	25.1	16.8	70.3	11.8	75.7	70.3	10.8	0.0
pr01_4_20	67	11	12.6	40.2	67.2	47.4	70.1	32.3	79.1	68.7	11.9	0.0
pr01_4_25	135	11	6.4	166.6	236.6	133.2	90.4	45.5	91.9	88.9	3.0	0.0
pr01_4_30	169	10	5.9	1175.0	472.8	304.1	87.0	121.3	93.5	89.9	4.7	0.0
pr01_4_35	195	15	4.3	-	2530.0	2167.8	81.5	2341.5	88.2	82.1	9.7	1.5
Moyenne	78	9	10.9	491.3	520.7	269.8	78.1	77.7	83.4	77.0	8.3	0.2
Fermées				17	19	19		19				

TABLE B.2: Résultats expérimentaux des méthodes sur les instances de type *r101* et *rc101*.

Description instances				B&B	SEM	SEM + T_{valid}		ISEM				
Instance	$ ND $	$ NS $	MGap	Temps	Temps	Temps	T_{valid}	Temps	T_{lb}	T_{valid}	T_{save}	T_{mult_dual}
r101_1_15	9	3	29.3	0.3	9.0	5.2	55.6	5.0	55.6	55.6	22.2	0.0
r101_1_20	10	5	28.2	0.2	12.9	6.5	80.0	4.4	70.0	80.0	10.0	0.0
r101_1_25	10	5	28.2	0.5	16.3	8.1	80.0	5.5	70.0	80.0	10.0	0.0
r101_1_30	16	3	24.2	1.6	20.3	12.7	87.5	6.9	81.3	81.3	6.3	0.0
r101_1_35	23	4	17.7	3.0	38.4	22.7	82.6	15.8	82.6	73.9	8.7	4.3
r101_2_15	25	6	14.8	0.6	15.7	11.7	48.0	11.4	76.0	44.0	32.0	0.0
r101_2_20	27	8	15.1	1.3	23.1	15.6	66.7	12.5	77.8	66.7	18.5	0.0
r101_2_25	32	6	14.8	2.0	42.0	21.2	71.9	14.1	84.4	71.9	15.6	0.0
r101_2_30	40	7	12.5	6.3	52.9	33.1	77.5	21.1	80.0	75.0	7.5	0.0
r101_2_35	47	7	10.8	10.1	74.8	48.5	61.7	38.8	85.1	61.7	23.4	0.0
r101_3_15	45	8	9.9	1.6	28.3	19.1	62.2	15.3	82.2	60.0	22.2	0.0
r101_3_20	44	9	10.7	2.5	39.1	25.4	63.6	18.9	79.5	63.6	18.2	0.0
r101_3_25	54	10	10.2	4.5	58.5	37.4	72.2	25.3	83.3	70.4	14.8	0.0
r101_3_30	62	10	9.1	12.5	78.4	55.5	64.5	40.1	83.9	62.9	22.6	0.0
r101_3_35	73	11	8.0	30.3	111.8	78.4	64.4	59.3	83.6	63.0	21.9	0.0
r101_4_15	51	9	9.1	1.9	32.9	22.0	62.7	18.4	82.4	60.8	21.6	0.0
r101_4_20	56	10	9.0	3.6	47.3	31.9	62.5	25.2	83.9	64.3	21.4	0.0
r101_4_25	65	11	9.2	7.0	70.3	44.7	72.3	32.7	83.1	72.3	13.8	0.0
r101_4_30	77	12	8.2	18.2	120.5	65.6	77.9	48.0	81.8	74.0	11.7	0.0
r101_4_35	102	11	6.7	37.9	157.6	108.0	67.6	68.2	87.3	67.6	18.6	0.0
Moyenne	43	8	14.3	7.3	52.5	33.6	69.1	24.3	79.7	67.4	17.0	0.2
rc101_1_15	11	6	25.3	1.1	10.0	4.7	81.8	2.8	72.7	81.8	0.0	0.0
rc101_1_20	11	6	25.5	2.4	10.5	6.3	81.8	3.7	72.7	81.8	0.0	0.0
rc101_1_25	12	6	24.4	4.8	18.1	8.6	83.3	4.2	75.0	83.3	0.0	0.0
rc101_1_30	12	6	24.4	5.4	21.5	10.4	83.3	5.2	75.0	83.3	0.0	0.0
rc101_1_35	12	6	24.4	7.0	24.9	12.2	83.3	6.5	75.0	83.3	0.0	0.0
rc101_2_15	25	10	15.0	6.2	19.9	10.9	64.0	9.1	72.0	64.0	12.0	0.0
rc101_2_20	27	10	14.5	22.4	29.2	15.2	66.7	12.3	74.1	66.7	11.1	0.0
rc101_2_25	29	9	13.9	48.2	31.7	20.1	69.0	11.7	79.3	72.4	10.3	0.0
rc101_2_30	29	9	13.9	57.6	40.1	30.5	69.0	13.9	79.3	72.4	10.3	0.0
rc101_2_35	29	9	13.9	81.6	44.1	30.9	69.0	16.4	79.3	72.4	10.3	0.0
rc101_3_15	25	10	15.1	8.5	18.5	11.0	68.0	9.0	72.0	64.0	12.0	0.0
rc101_3_20	34	11	12.5	35.1	30.2	20.9	64.7	16.1	73.5	64.7	14.7	0.0
rc101_3_25	46	14	10.0	180.2	50.8	34.2	65.2	25.7	76.1	65.2	13.0	0.0
rc101_3_30	46	14	10.0	220.4	59.8	47.9	65.2	26.3	76.1	65.2	13.0	0.0
rc101_3_35	46	14	10.0	332.1	70.0	50.0	65.2	31.3	76.1	65.2	13.0	0.0
rc101_4_15	25	10	15.1	6.9	15.6	12.5	68.0	9.9	72.0	64.0	12.0	0.0
rc101_4_20	36	13	12.2	46.1	33.0	26.0	63.9	18.0	75.0	63.9	16.7	0.0
rc101_4_25	49	16	9.9	262.9	54.2	44.0	63.3	27.0	75.5	61.2	18.4	0.0
rc101_4_30	53	16	9.6	478.6	71.0	51.4	62.3	36.5	73.6	64.2	15.1	0.0
rc101_4_35	59	17	8.4	946.2	93.3	66.0	62.7	52.0	76.3	62.7	15.3	0.0
Moyenne	31	11	15.4	137.7	37.3	25.6	70.0	16.9	75.0	70.1	9.9	0.0

Résultats expérimentaux du problème de tournées couvrantes

Dans cette annexe, nous présentons le détail des résultats expérimentaux du Chapitre 6 pour les problèmes de tournées couvrantes. Les instances du MCTP sont celles utilisées par Jozefowicz (2014) et Ha et al. (2013). Elles sont nommées $X-T-n-W-p$ où X est le nom de l'instance de Krolak qui est utilisée, T est la taille de l'ensemble des points de service obligatoires avec le dépôt, n est la taille de l'ensemble des points de service plus le dépôt, W est la taille de l'ensemble des clients et p est le nombre maximal de points de service visités par une route. Dans toutes les instances, la taille maximale d'une route \bar{d} est fixée à $+\infty$. Au total, 96 instances sont testées.

C.1 Comparaison des méthodes

Dans les tableaux C.1, C.2 et C.3, les temps du papier de Ha et al. sont divisés par 1.94 dans la colonne *Temps_c* pour pouvoir être comparés avec nos temps d'après le *passmark CPU score*¹ et leurs temps originaux sont notés dans la colonne *Temps_o*. Dans ces trois tableaux, les en-têtes suivants sont utilisés : le temps CPU en secondes (*Temps*), la solution optimale de chaque instance (*UB*), la borne inférieure du MCTP (*LB*) et le gap entre la borne inférieure *LB* et la solution optimale *UB* en pourcentage (*Gap*). Si aucune solution optimale n'est trouvée, le gap est calculé avec la meilleure borne supérieure valide trouvée. Le gap est calculé comme suit :

$$Gap = 100 - \frac{z(LB)*100}{z(UB)} (\%);$$

C.2 Comparaison des bornes de complétion

Les tableaux C.4 et C.5 comparent l'utilisation de notre méthode avec les bornes de complétion f et g , introduites en Section 6.3.2. Dans ces tableaux, les en-têtes suivants sont utilisés : le temps CPU en secondes (*Temps*) et le nombre d'étiquettes acceptées par les bornes de complétion lors des dernières itérations de la génération de colonne pour calculer la borne inférieure du MCTP (*Chemins*). Les dernières itérations correspondent au sous-problème de *pricing* résolu avec $MAX_LABEL = +\infty$ (cf. Section 6.3.2). Plus le nombre de chemins autorisés est petit, plus la borne de complétion est efficace.

1. Données disponibles sur le site https://www.cpubenchmark.net/cpu_list.php

150 Annexe C. Résultats expérimentaux du problème de tournées couvrantes

TABLE C.1: Résultats de notre méthode et celle de Hà et al. sur des instances avec $|V| = 100$ et $|M| = 0$.

Instance	UB	Ha et al. (2013)				Notre méthode		
		LB	Gap	Temps_o	Temps_c	LB	Gap	Temps
A1-1-25-75-4	8479.0	8218.0	3.1	1.1	0.6	8479.0	0.0	0.1
A1-1-25-75-5	8479.0	7521.1	11.3	3.3	1.7	8479.0	0.0	0.2
A1-1-25-75-6	8479.0	7411.3	12.6	3.3	1.7	8402.3	0.9	0.6
A1-1-25-75-8	7985.0	6340.3	20.6	20.1	10.4	7985.0	0.0	0.6
A1-1-50-50-4	10271.0	9346.0	9.0	9.9	5.1	10271.0	0.0	0.6
A1-1-50-50-5	9220.0	7925.4	14.0	12.4	6.4	9220.0	0.0	0.6
A1-1-50-50-6	9130.0	7186.3	21.3	24.8	12.8	9130.0	0.0	3.1
A1-1-50-50-8	9130.0	6325.9	30.7	203.9	105.1	9031.0	1.1	24.8
B1-1-25-75-4	7146.0	6563.8	8.1	1.8	0.9	7146.0	0.0	0.2
B1-1-25-75-5	6901.0	5821.7	15.6	3.2	1.6	6782.0	1.7	0.3
B1-1-25-75-6	6450.0	5009.6	22.3	4.3	2.2	6450.0	0.0	0.5
B1-1-25-75-8	6450.0	4464.5	30.8	10.9	5.6	6450.0	0.0	2.4
B1-1-50-50-4	10107.0	8706.1	13.9	16.6	8.6	10107.0	0.0	0.6
B1-1-50-50-5	9723.0	7750.3	20.3	84.1	43.4	9723.0	0.0	1.0
B1-1-50-50-6	9382.0	7193.3	23.3	162.2	83.6	9273.0	1.2	3.4
B1-1-50-50-8	8348.0	5744.5	31.2	76.1	39.2	8348.0	0.0	10.1
C1-1-25-75-4	6161.0	5265.9	14.5	2.8	1.4	6161.0	0.0	0.2
C1-1-25-75-5	6161.0	4946.5	19.7	5.8	3.0	6161.0	0.0	0.3
C1-1-25-75-6	6161.0	4678.2	24.1	7.7	4.0	6161.0	0.0	0.5
C1-1-25-75-8	6161.0	4279.7	30.5	9.4	4.8	6161.0	0.0	1.6
C1-1-50-50-4	11372.0	10462.3	8.0	8.1	4.2	11042.0	2.9	0.9
C1-1-50-50-5	9900.0	8897.0	10.1	13.3	6.9	9900.0	0.0	0.7
C1-1-50-50-6	9895.0	8210.6	17.0	56.9	29.3	9719.0	1.8	3.1
C1-1-50-50-8	8699.0	7479.8	14.0	8.5	4.4	8684.0	0.2	21.7
D1-1-25-75-4	7671.0	7471.4	2.6	1.0	0.5	7671.0	0.0	0.2
D1-1-25-75-5	7465.0	6630.6	11.2	5.4	2.8	7363.3	1.4	0.6
D1-1-25-75-6	6651.0	5764.1	13.3	3.8	2.0	6651.0	0.0	0.3
D1-1-25-75-8	6651.0	5187.7	22.0	12.9	6.6	6651.0	0.0	1.3
D1-1-50-50-4	11606.0	10704.4	7.8	9.3	4.8	11606.0	0.0	0.7
D1-1-50-50-5	10770.0	9139.6	15.1	29.3	15.1	10770.0	0.0	0.9
D1-1-50-50-6	10525.0	8264.9	21.5	281.3	145.0	10197.0	3.1	4.0
D1-1-50-50-8	9361.0	6954.8	25.7	110.6	57.0	9341.4	0.2	121.2
Moyenne			17.0	37.6	19.4		0.5	6.5
Instances fermées				32/32				32/32

TABLE C.2: Résultats de notre méthode et celle de Hà et al. sur des instances avec $|V| = 100$ et $|M| > 0$.

Instance	Ha et al. (2013)					Notre méthode		
	UB	LB	Gap	Temps_o	Temps_c	LB	Gap	Temps
A1-5-25-75-4	10827.0	9466.4	12.6	9.5	4.9	9894.8	8.6	0.3
A1-5-25-75-5	8659.0	8659.0	0.0	0.1	0.1	8659.0	0.0	0.2
A1-5-25-75-6	8659.0	8576.6	1.0	6.9	3.6	8659.0	0.0	0.3
A1-5-25-75-8	8265.0	7398.2	10.5	4.2	2.2	8265.0	0.0	0.5
A1-10-50-50-4	17953.0	16223.0	9.6	4828.6	2489.0	16981.0	5.4	11.1
A1-10-50-50-5	15440.0	14317.2	7.3	173.6	89.5	15051.9	2.5	2.0
A1-10-50-50-6	14064.0	12769.9	9.2	1586.2	817.6	13831.5	1.7	12.3
A1-10-50-50-8	-	11280.3	17.7	-	-	12050.0	12.1	-
B1-5-25-75-4	9465.0	9465.0	0.0	0.2	0.1	9465.0	0.0	0.2
B1-5-25-75-5	9460.0	8713.2	7.9	5.7	2.9	9293.3	1.8	0.3
B1-5-25-75-6	9148.0	8291.1	9.4	18.1	9.3	9011.0	1.5	0.5
B1-5-25-75-8	8306.0	7193.7	13.4	8.9	4.6	8306.0	0.0	1.1
B1-10-50-50-4	15209.0	14106.9	7.2	127.6	65.8	14878.3	2.2	0.6
B1-10-50-50-5	13535.0	12125.2	10.4	149.2	76.9	13089.5	3.3	2.3
B1-10-50-50-6	12067.0	10815.9	10.4	104.7	54.0	11791.0	2.3	1.7
B1-10-50-50-8	10344.0	9343.0	9.7	32.3	16.6	10344.0	0.0	1.9
C1-5-25-75-4	9898.0	9795.5	1.0	0.4	0.2	9898.0	0.0	0.2
C1-5-25-75-5	9707.0	9205.3	5.2	3.0	1.5	9314.4	4.0	0.4
C1-5-25-75-6	9321.0	8688.0	6.8	4.2	2.2	8534.7	8.4	6.1
C1-5-25-75-8	7474.0	7474.0	0.0	9.4	4.8	7474.0	0.0	0.9
C1-10-50-50-4	18212.0	17217.0	5.5	164.4	84.7	17898.0	1.7	0.9
C1-10-50-50-5	16362.0	15583.4	4.8	126.8	65.4	15697.8	4.1	17.2
C1-10-50-50-6	14749.0	13932.4	5.5	240.4	123.9	14224.2	3.6	11.2
C1-10-50-50-8	12394.0	12000.6	3.2	5.6	2.9	12394.0	0.0	2.7
D1-5-25-75-4	11820.0	11241.8	4.9	1.7	0.9	11783.0	0.3	0.4
D1-5-25-75-5	10982.0	9870.1	10.1	16.7	8.6	10465.3	4.7	13.1
D1-5-25-75-6	9669.0	8976.1	7.2	3.4	1.8	9517.8	1.6	0.6
D1-5-25-75-8	8200.0	7516.4	8.3	1.3	0.7	8200.0	0.0	1.2
D1-10-50-50-4	20982.0	20346.9	3.0	10.9	5.6	20982.0	0.0	0.8
D1-10-50-50-5	18576.0	9139.6	50.8	393.5	202.8	17984.2	3.2	58.2
D1-10-50-50-6	16330.0	15073.8	7.7	116.1	59.8	16156.0	1.1	3.2
D1-10-50-50-8	14204.0	12613.5	11.2	248.4	128.0	13909.0	2.1	6.9
Moyenne			8.5	271.0	139.7		2.4	5.5
Instances fermées				31/32				31/32

152 Annexe C. Résultats expérimentaux du problème de tournées couvrantes

TABLE C.3: Résultats de notre méthode et celle de Hà et al. sur des instances avec $|V| = 200$.

Instance	Ha et al. (2013)					Notre méthode		
	UB	LB	Gap	Temps_o	Temps_c	LB	Gap	Temps
A2-1-50-150-4	11550.0	10437.8	9.6	82.2	42.4	11508.2	0.4	1.8
A2-1-50-150-5	10407.0	9168.5	11.9	340.6	175.6	10407.0	0.0	2.8
A2-1-50-150-6	10068.0	8527.6	15.3	1075.8	554.5	10068.0	0.0	3.6
A2-1-50-150-8	8896.0	7169.3	19.4	153.4	79.1	8896.0	0.0	23.9
A2-1-100-100-4	11885.0	9675.9	18.6	4593.9	2368.0	11512.2	3.1	20.8
A2-1-100-100-5	10234.0	8212.1	19.8	1440.1	742.3	10234.0	0.0	13.9
A2-1-100-100-6	10020.0	7162.9	28.5	-	-	10020.0	0.0	29.5
A2-1-100-100-8	9093.0	5883.0	35.3	-	-	9093.0	0.0	826.8
A2-10-50-150-4	17083.0	16049.6	6.0	1257.0	647.9	16761.0	1.9	1.8
A2-10-50-150-5	14977.0	13905.0	7.2	494.7	255.0	14803.5	1.2	4.5
A2-10-50-150-6	13894.0	12959.8	6.7	978.9	504.6	13453.5	3.2	40.8
A2-10-50-150-8	11942.0	11207.4	6.2	280.2	144.4	11942.0	0.0	9.8
A2-20-100-100-4	26594.0	25042.6	5.8	-	-	26174.0	1.6	10.2
A2-20-100-100-5	23419.0	21331.4	8.9	-	-	22715.5	3.0	4559.9
A2-20-100-100-6	-	18775.1	11.4	-	-	20372.0	3.8	-
A2-20-100-100-8	-	15552.1	17.0	-	-	17511.1	6.6	-
B2-1-50-150-4	11175.0	9573.0	14.3	166.0	85.6	11175.0	0.0	1.5
B2-1-50-150-5	10502.0	8245.4	21.5	1114.7	574.6	10339.0	1.6	2.8
B2-1-50-150-6	9799.0	7604.1	22.4	1274.0	656.7	9799.0	0.0	3.5
B2-1-50-150-8	8846.0	6213.4	29.8	166.0	85.6	8846.0	0.0	8.7
B2-1-100-100-4	18370.0	16748.9	8.8	6615.0	3409.8	17864.2	2.8	314.4
B2-1-100-100-5	15876.0	14214.5	10.5	1472.0	758.8	15838.5	0.2	11.7
B2-1-100-100-6	14867.0	12394.7	16.6	-	-	14531.2	2.3	107.7
B2-1-100-100-8	-	10366.0	25.7	-	-	12831.0	8.1	-
B2-10-50-150-4	16667.0	15125.6	9.2	5972.5	3078.6	15675.3	6.0	9.7
B2-10-50-150-5	14188.0	13574.2	4.3	124.2	64.0	13967.5	1.6	4.5
B2-10-50-150-6	12954.0	11947.6	7.8	773.6	398.8	12561.5	3.0	4.4
B2-10-50-150-8	11495.0	10078.3	12.3	732.7	377.7	11277.2	1.9	13.2
B2-20-100-100-4	34062.0	32913.3	3.4	-	-	33790.0	0.8	9.2
B2-20-100-100-5	29405.0	27494.1	6.5	-	-	28769.8	2.2	4961.8
B2-20-100-100-6	-	23954.7	8.2	-	-	25262.7	2.7	-
B2-20-100-100-8	-	19488.6	14.8	-	-	21441.0	6.3	-
Moyenne			13.9	1455.4	750.2		2.0	393.1
Instances fermées				20/32				27/32

TABLE C.4: Comparaison des bornes de complétion f et g sur des instances avec $|V| = 100$ et $|M| \geq 0$.

Instance	Borne de complétion de l'état-de-l'art f		Nouvelle borne de complétion g		Instance	Borne de complétion de l'état-de-l'art f		Nouvelle borne de complétion g	
	Chemins	Temps	Chemins	Temps		Chemins	Temps	Chemins	Temps
A1-1-25-75-4	498	0.1	88	0.1	A1-5-25-75-4	166	0.3	80	0.3
A1-1-25-75-5	2198	0.2	1522	0.2	A1-5-25-75-5	274	0.2	50	0.2
A1-1-25-75-6	11060	1.0	5088	0.6	A1-5-25-75-6	2126	0.3	483	0.3
A1-1-25-75-8	65634	1.9	13872	0.6	A1-5-25-75-8	18655	0.7	4430	0.5
A1-1-50-50-4	270	0.4	130	0.6	A1-10-50-50-4	346	10.9	158	11.1
A1-1-50-50-5	8446	0.8	1130	0.6	A1-10-50-50-5	702	1.5	368	2.0
A1-1-50-50-6	74643	2.7	56142	3.1	A1-10-50-50-6	4218	12.7	1644	12.3
A1-1-50-50-8	228512	130.1	69878	24.8	A1-10-50-50-8	-	-	-	-
B1-1-25-75-4	608	0.1	250	0.2	B1-5-25-75-4	66	0.1	66	0.2
B1-1-25-75-5	4584	0.4	1262	0.3	B1-5-25-75-5	3632	0.3	857	0.3
B1-1-25-75-6	14200	0.6	3600	0.5	B1-5-25-75-6	13778	0.9	3800	0.5
B1-1-25-75-8	384250	11.3	52298	2.4	B1-5-25-75-8	33277	1.1	19655	1.1
B1-1-50-50-4	934	0.6	374	0.6	B1-10-50-50-4	340	0.7	290	0.6
B1-1-50-50-5	5405	0.8	4379	1.0	B1-10-50-50-5	710	2.1	604	2.3
B1-1-50-50-6	32358	7.6	13144	3.4	B1-10-50-50-6	866	2.0	686	1.7
B1-1-50-50-8	410434	10.9	183948	10.1	B1-10-50-50-8	8770	2.3	7926	1.9
C1-1-25-75-4	358	0.2	222	0.2	C1-5-25-75-4	214	0.2	96	0.2
C1-1-25-75-5	15273	0.5	1365	0.3	C1-5-25-75-5	580	0.4	492	0.4
C1-1-25-75-6	47721	1.6	8404	0.5	C1-5-25-75-6	308	5.2	162	6.1
C1-1-25-75-8	483856	13.3	48235	1.6	C1-5-25-75-8	2342	0.8	860	0.9
C1-1-50-50-4	1522	0.8	1294	0.9	C1-10-50-50-4	242	0.8	236	0.9
C1-1-50-50-5	6074	1.0	410	0.7	C1-10-50-50-5	1635	15.5	1552	17.2
C1-1-50-50-6	84814	14.5	17333	3.1	C1-10-50-50-6	4668	10.2	3392	11.2
C1-1-50-50-8	257838	32.9	142493	21.7	C1-10-50-50-8	7528	2.7	3684	2.7
D1-1-25-75-4	674	0.2	214	0.2	D1-5-25-75-4	182	0.4	104	0.4
D1-1-25-75-5	3620	0.5	2502	0.6	D1-5-25-75-5	410	11.3	204	13.1
D1-1-25-75-6	9244	0.6	1058	0.3	D1-5-25-75-6	1678	0.5	788	0.6
D1-1-25-75-8	447168	7.6	38186	1.3	D1-5-25-75-8	21128	1.3	18012	1.2
D1-1-50-50-4	604	0.6	418	0.7	D1-10-50-50-4	428	0.7	224	0.8
D1-1-50-50-5	14524	1.0	2726	0.9	D1-10-50-50-5	414	41.8	278	58.2
D1-1-50-50-6	9672	4.0	6132	4.0	D1-10-50-50-6	1646	3.0	1026	3.2
D1-1-50-50-8	596132	409.6	203502	121.2	D1-10-50-50-8	18281	10.8	7140	6.9
Moyenne	100723	20.6	27550	6.5	Moyenne	4826	4.6	2560	5.1

154 Annexe C. Résultats expérimentaux du problème de tournées couvrantes

TABLE C.5: Comparaison des bornes de complétion f et g sur des instances avec $|V| = 200$.

Instance	Borne de complétion de l'état-de-l'art f		Nouvelle borne de complétion g		Instance	Borne de complétion de l'état-de-l'art f		Nouvelle borne de complétion g	
	Chemins	Temps	Chemins	Temps		Chemins	Temps	Chemins	Temps
A2-1-50-150-4	1462	2.6	1000	1.8	B2-1-50-150-4	692	2.1	668	1.5
A2-1-50-150-5	27693	5.4	10452	2.8	B2-1-50-150-5	3016	3.6	1500	2.8
A2-1-50-150-6	104062	7.4	18681	3.6	B2-1-50-150-6	13672	3.5	6538	3.5
A2-1-50-150-8	849383	43.2	278680	23.9	B2-1-50-150-8	84521	8.5	36993	8.7
A2-1-100-100-4	4182	19.1	2984	20.8	B2-1-100-100-4	1450	269.3	1198	314.4
A2-1-100-100-5	88582	12.0	71965	13.9	B2-1-100-100-5	4710	8.4	4582	11.7
A2-1-100-100-6	704792	80.2	188834	29.5	B2-1-100-100-6	32613	149.6	31614	107.7
A2-1-100-100-8	10610182	624.2	7795891	826.8	B2-1-100-100-8	-	-	-	-
A2-10-50-150-4	216	2.4	186	1.8	B2-10-50-150-4	120	12.4	76	9.7
A2-10-50-150-5	680	2.8	666	4.5	B2-10-50-150-5	1433	5.9	1223	4.5
A2-10-50-150-6	3677	36.0	2848	40.8	B2-10-50-150-6	2074	3.3	1860	4.4
A2-10-50-150-8	74259	15.5	45510	9.8	B2-10-50-150-8	16382	9.1	12242	13.2
A2-20-100-100-4	520	9.2	410	10.2	B2-20-100-100-4	700	12.5	666	9.2
A2-20-100-100-5	2315	5472.0	1771	4559.9	B2-20-100-100-5	1970	4954.3	1702	4961.8
A2-20-100-100-6	-	-	-	-	B2-20-100-100-6	-	-	-	-
A2-20-100-100-8	-	-	-	-	B2-20-100-100-8	-	-	-	-
					Moyenne	467976	436.1	315583	407.5

Bibliographie

- M. Abdel-Basset, L. Abdel-Fatah, et A. K. Sangaiah. Metaheuristic algorithms : A comprehensive review. In *Computational Intelligence for Multimedia Big Data on the Cloud with Engineering Applications*, pages 185–231. Elsevier, 2018. [6]
- R. Abounacer, M. Rekik, et J. Renaud. An exact solution approach for multi-objective location–transportation problem for disaster response. *Computers & Operations Research*, 41 :83–93, 2014. [38]
- Y. P. Aneja et K. P. Nair. Bicriteria transportation problem. *Management Science*, 25(1) : 73–78, 1979. [12 et 13]
- L. H. Appelgren. A column generation algorithm for a ship scheduling problem. *Transportation Science*, 3(1) :53–68, 1969. [27]
- R. Arab, S. Ghaderi, et R. Tavakkoli-Moghaddam. Solving a new multi-objective inventory-routing problem by a non-dominated sorting genetic algorithm. *International Journal of Engineering-Transactions A : Basics*, 31(4) :588–596, 2018. [39]
- C. Archetti, D. Feillet, A. Hertz, et M. G. Speranza. The capacitated team orienteering and profitable tour problems. *Journal of the Operational Research Society*, 60(6) :831–842, 2009. [98]
- C. Artigues, N. Jozefowicz, et B. M. Sarpong. Column generation algorithms for bi-objective combinatorial optimization problems with a min–max objective. *EURO Journal on Computational Optimization*, pages 1–26, 2018. [109, 113, 114, 115, 123, 124 et 128]
- P. Augerat, J. M. Belenguer, E. Benavent, A. Corberán, D. Naddef, et G. Rinaldi. *Computational results with a branch and cut code for the capacitated vehicle routing problem*. IMAG, 1995. [34]
- R. Baldacci, M. A. Boschetti, V. Maniezzo, et M. Zamboni. Scatter search methods for the covering tour problem. In *Metaheuristic Optimization via Memory and Evolution*, pages 59–91. Springer, 2005. [110]
- R. Baldacci, N. Christofides, et A. Mingozzi. An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts. *Mathematical Programming*, 115(2) :351–385, 2008. [31, 32, 34, 36, 42, 46, 49, 67 et 115]
- R. Baldacci, A. Mingozzi, et R. Roberti. New route relaxation and pricing strategies for the vehicle routing problem. *Operations Research*, 59(5) :1269–1283, 2011. [30, 31, 36, 87 et 118]
- M. L. Balinski et R. E. Quandt. On an integer program for a delivery problem. *Operations Research*, 12(2) :300–304, 1964. [25 et 42]

- C. Bazgan, H. Hugot, et D. Vanderpooten. Solving efficiently the 0–1 multi-objective knapsack problem. *Computers & Operations Research*, 36(1) :260–279, 2009. [20]
- J. E. Beasley. Route first—cluster second methods for vehicle routing. *Omega*, 11(4) : 403–408, 1983. [111]
- T. Bektaş et G. Laporte. The pollution-routing problem. *Transportation Research Part B : Methodological*, 45(8) :1232–1250, 2011. [37]
- R. Bellman. Dynamic programming. *Science*, 153(3731) :34–37, 1966. [20]
- J.-F. Bérubé, M. Gendreau, et J.-Y. Potvin. An exact ε -constraint method for bi-objective combinatorial optimization problems : Application to the traveling salesman problem with profits. *European Journal of Operational Research*, 194(1) :39–50, 2009. [17]
- N. Boland, H. Charkhgard, et M. Savelsbergh. A criterion space search algorithm for biobjective integer programming : The balanced box method. *INFORMS Journal on Computing*, 27(4) :735–754, 2015. [17, 20 et 39]
- N. Boland, H. Charkhgard, et M. Savelsbergh. The quadrant shrinking method : A simple and efficient algorithm for solving tri-objective integer programs. *European Journal of Operational Research*, 260(3) :873–885, 2017. [133]
- H. Bouly, A. Moukrim, D. Chanteur, et L. Simon. Un algorithme de destruction/construction itératif pour la résolution d’un probleme de tournées de véhicules spécifique. In *MOSIM*, volume 8, pages 1593–1602, 2008. [96]
- S. Boussier, D. Feillet, et M. Gendreau. An exact algorithm for team orienteering problems. *4or*, 5(3) :211–230, 2007. [98]
- S. E. Butt et T. M. Cavalier. A heuristic for the multiple tour maximum collection problem. *Computers & Operations Research*, 21(1) :101–111, 1994. [95]
- S. E. Butt et D. M. Ryan. An optimal solution procedure for the multiple tour maximum collection problem using column generation. *Computers & Operations Research*, 26(4) : 427–441, 1999. [97]
- M. E. Captivo, J. Clímaco, J. Figueira, E. Martins, et J. L. Santos. Solving bicriteria 0–1 knapsack problems using a labeling algorithm. *Computers & Operations Research*, 30 (12) :1865–1886, 2003. [20]
- L. Chalmet, L. Lemonidis, et D. Elzinga. An algorithm for the bi-criterion integer programming problem. *European Journal of Operational Research*, 25(2) :292–300, 1986. [12 et 14]
- I.-M. Chao, B. L. Golden, et E. A. Wasil. The team orienteering problem. *European Journal of Operational Research*, 88(3) :464–474, 1996. [95, 97 et 98]

- V. Chvatal. Edmonds polytopes and a hierarchy of combinatorial problems. *Discrete Mathematics*, 4(4) :305–337, 1973. [74]
- G. Clarke et J. W. Wright. Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12(4) :568–581, 1964. [111]
- C. Contardo et R. Martinelli. A new exact algorithm for the multi-depot vehicle routing problem under capacity and route length constraints. *Discrete Optimization*, 12 :129–146, 2014. [31, 33 et 36]
- L. Costa, C. Contardo, et G. Desaulniers. Exact branch-price-and-cut algorithms for vehicle routing. *Transportation Science*, 2019. [34 et 36]
- J. R. Current et D. A. Schilling. The covering salesman problem. *Transportation Science*, 23(3) :208–213, 1989. [107, 110 et 112]
- J. R. Current et D. A. Schilling. The median tour and maximal covering tour problems : Formulations and heuristics. *European Journal of Operational Research*, 73(1) :114–126, 1994. [110, 112 et 114]
- K. Dächert, J. Gorski, et K. Klamroth. An augmented weighted tchebycheff method with adaptively chosen parameters for discrete bicriteria optimization problems. *Computers & Operations Research*, 39(12) :2929–2943, 2012. [15]
- D.-C. Dang, R. El-Hajj, et A. Moukrim. A branch-and-cut algorithm for solving the team orienteering problem. In *International Conference on AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, pages 332–339. Springer, 2013. [98]
- G. B. Dantzig et J. H. Ramser. The truck dispatching problem. *Management Science*, 6(1) :80–91, 1959. [23]
- M. P. de Aragao et E. Uchoa. Integer program reformulation for robust branch-and-cut-and-price algorithms. In *Mathematical Program in Rio : a Conference in Honour of Nelson Maculan*, pages 56–61. Citeseer, 2003. [36]
- E. Demir, T. Bektaş, et G. Laporte. The bi-objective pollution-routing problem. *European Journal of Operational Research*, 232(3) :464–478, 2014. [37]
- G. Desaulniers, F. Lessard, et A. Hadjar. Tabu search, partial elementarity, and generalized k-path inequalities for the vehicle routing problem with time windows. *Transportation Science*, 42(3) :387–404, 2008. [33]
- M. Desrochers, J. Desrosiers, et M. Solomon. A new optimization algorithm for the vehicle routing problem with time windows. *Operations Research*, 40(2) :342–354, 1992. [30]
- J. Desrosiers, F. Soumis, et M. Desrochers. Routing with time windows by column generation. *Networks*, 14(4) :545–565, 1984. [34 et 36]

- J. Desrosiers, J. B. Gauthier, et M. E. Lübbecke. Row-reduced column generation for degenerate master problems. *European Journal of Operational Research*, 236(2) :453–460, 2014. [33]
- C. Dhaenens, J. Lemesre, et E.-G. Talbi. K-ppm : A new exact method to solve multi-objective combinatorial optimization problems. *European Journal of Operational Research*, 200(1) :45–53, 2010. [17]
- M. Dror. Note on the complexity of the shortest path models for column generation in vrptw. *Operations Research*, 42(5) :977–978, 1994. [30 et 46]
- O. Du Merle, D. Villeneuve, J. Desrosiers, et P. Hansen. Stabilized column generation. *Discrete Mathematics*, 194(1-3) :229–237, 1999. [33 et 120]
- M. Ehrgott et X. Gandibleux. A survey and annotated bibliography of multiobjective combinatorial optimization. *OR-Spektrum*, 22(4) :425–460, 2000. [20]
- M. Ehrgott et X. Gandibleux. Bounds and bound sets for biobjective combinatorial optimization problems. In *Multiple Criteria Decision Making in the New Millennium*, pages 241–253. Springer, 2001. [10]
- M. Ehrgott et X. Gandibleux. Bound sets for biobjective combinatorial optimization problems. *Computers & Operations Research*, 34(9) :2674–2694, 2007. [10]
- M. Ehrgott et S. Ruzika. Improved ε -constraint method for multiobjective programming. *Journal of Optimization Theory and Applications*, 138(3) :375, 2008. [16]
- M. Ehrgott et D. M. Ryan. Constructing robust crew schedules with bicriteria optimization. *Journal of Multi-Criteria Decision Analysis*, 11(3) :139–150, 2002. [16 et 17]
- M. Ehrgott et D. Tenfelde-Podehl. Computation of ideal and nadir values and implications for their use in mcdm methods. *European Journal of Operational Research*, 151(1) : 119–139, 2003. [10]
- R. El-Hajj, D.-C. Dang, et A. Moukrim. Solving the team orienteering problem with cutting planes. *Computers & Operations Research*, 74 :21–30, 2016. [98]
- D. Feillet. A tutorial on column generation and branch-and-price for vehicle routing problems. *4or*, 8(4) :407–424, 2010. [27]
- D. Feillet, P. Dejax, M. Gendreau, et C. Gueguen. An exact algorithm for the elementary shortest path problem with resource constraints : Application to some vehicle routing problems. *Networks : An International Journal*, 44(3) :216–229, 2004. [29]
- J. R. Figueira, L. Paquete, M. Simões, et D. Vanderpooten. Algorithmic improvements on dynamic programming for the bi-objective $\{0, 1\}$ knapsack problem. *Computational Optimization and Applications*, 56(1) :97–111, 2013. [20]

- M. Fischetti, J. J. S. Gonzalez, et P. Toth. Solving the orienteering problem through branch-and-cut. *INFORMS Journal on Computing*, 10(2) :133–148, 1998. [97]
- D. A. Flores-Garza, M. A. Salazar-Aguilar, S. U. Nguerveu, et G. Laporte. The multi-vehicle cumulative covering tour problem. *Annals of Operations Research*, 258(2) :761–780, 2017. [112]
- H. Fouchal, X. Gandibleux, et F. Lehu  d  . Preferred solutions computed with a label setting algorithm based on choquet integral for multi-objective shortest paths. In *2011 IEEE Symposium on Computational Intelligence in Multicriteria Decision-Making (MDCM)*, pages 143–150. IEEE, 2011. [11]
- R. Fukasawa, H. Longo, J. Lysgaard, M. P. de Arag  o, M. Reis, E. Uchoa, et R. F. Werneck. Robust branch-and-cut-and-price for the capacitated vehicle routing problem. *Mathematical Programming*, 106(3) :491–511, 2006. [33, 34 et 36]
- D. Gavalas, C. Konstantopoulos, K. Mastakas, et G. Pantziou. A survey on algorithmic approaches for solving tourist trip design problems. *Journal of Heuristics*, 20(3) :291–328, 2014. [97]
- B. Gavish. The delivery problem : New cutting planes procedure. In *TIMS XXVI conference, Copenhagen*, 1984. [25]
- M. Gendreau, G. Laporte, et F. Semet. The covering tour problem. *Operations Research*, 45(4) :568–576, 1997. [110 et 113]
- M. Gendreau, G. Laporte, et F. Semet. A branch-and-cut algorithm for the undirected selective traveling salesman problem. *Networks : An International Journal*, 32(4) :263–273, 1998. [97]
- A. M. Geoffrion. Proper efficiency and the theory of vector maximization. *Journal of Mathematical Analysis and Applications*, 22(3) :618–630, 1968. [12 et 44]
- B. E. Gillett et L. R. Miller. A heuristic algorithm for the vehicle-dispatch problem. *Operations Research*, 22(2) :340–349, 1974. [111]
- M. T. Godinho, L. Gouveia, et T. L. Magnanti. Combined route capacity and route length models for unit demand vehicle routing problems. *Discrete Optimization*, 5(2) :350–372, 2008. [108]
- R. E. Gomory et al. Outline of an algorithm for integer solutions to linear programs. *Bulletin of the American Mathematical society*, 64(5) :275–278, 1958. [35]
- A. Gunawan, H. C. Lau, et P. Vansteenwegen. Orienteering problem : A survey of recent variants, solution approaches and applications. *European Journal of Operational Research*, 255(2) :315–332, 2016. [97]

- W. J. Gutjahr et N. Dzubur. Bi-objective bilevel optimization of distribution center locations considering user equilibria. *Transportation Research Part E : Logistics and Transportation Review*, 85 :1–22, 2016. [38]
- M. H. Ha, N. Bostel, A. Langevin, et L.-M. Rousseau. An exact algorithm and a meta-heuristic for the multi-vehicle covering tour problem with a constraint on the number of vertices. *European Journal of Operational Research*, 226(2) :211–220, 2013. [109, 111, 112, 113, 115, 125, 126, 149, 150, 151 et 152]
- M. Hachicha, M. J. Hodgson, G. Laporte, et F. Semet. Heuristics for the multi-vehicle covering tour problem. *Computers & Operations Research*, 27(1) :29–42, 2000. [109, 110, 111, 112 et 115]
- Y. Haimès, L. Lasdon, et D. Wismer. On a bicriterion formulation of the problems of integrated system identification and system optimization. *IEEE transactions on systems, man, and cybernetics*, 1(3) :296–297, 1971. [15, 42, 52 et 53]
- E. E. Halvorsen-Weare et M. W. Savelsbergh. The bi-objective mixed capacitated general routing problem with different route balance criteria. *European Journal of Operational Research*, 251(2) :451–465, 2016. [37 et 38]
- H. W. Hamacher, C. R. Pedersen, et S. Ruzika. Finding representative systems for discrete bicriterion optimization problems. *Operations Research Letters*, 35(3) :336–344, 2007. [16]
- P. Hansen. Bicriterion path problems. In *Multiple criteria decision making theory and application*, pages 109–127. Springer, 1980. [8]
- M. J. Hodgson, G. Laporte, et F. Semet. A covering tour model for planning mobile health care facilities in suhumdistrict, ghama. *Journal of Regional Science*, 38(4) :621–638, 1998. [110 et 113]
- S. Irnich et G. Desaulniers. Shortest path problems with resource constraints. In *Column generation*, pages 33–65. Springer, 2005. [30]
- G. R. Jahanshahloo, F. Hosseinzadeh, N. Shoja, et G. Tohidi. A method for generating all efficient solutions of 0-1 multi-objective linear programming problem. *Applied Mathematics and Computation*, 169(2) :874–886, 2005. [14]
- M. Jepsen, B. Petersen, S. Spoorendonk, et D. Pisinger. Subset-row inequalities applied to the vehicle-routing problem with time windows. *Operations Research*, 56(2) :497–511, 2008. [74 et 75]
- J. Jorge. *Nouvelles propositions pour la résolution exacte du sac à dos multi-objectif uni-dimensionnel en variables binaires*. PhD thesis, 2010. [19]
- N. Jozefowicz. A branch-and-price algorithm for the multivehicle covering tour problem. *Networks*, 64(3) :160–168, 2014. [109, 111, 112, 115, 125 et 149]

- N. Jozefowicz, F. Semet, et E.-G. Talbi. The bi-objective covering tour problem. *Computers & Operations Research*, 34(7) :1929–1942, 2007. [112, 114 et 124]
- N. Jozefowicz, F. Semet, et E.-G. Talbi. Multi-objective vehicle routing problems. *European Journal of Operational Research*, 189(2) :293–309, 2008. [36]
- M. Kammoun, H. Derbel, M. Ratli, et B. Jarboui. An integration of mixed vnd and vns : the case of the multivehicle covering tour problem. *International Transactions in Operational Research*, 24(3) :663–679, 2017. [109 et 112]
- M. Keshtkaran, K. Ziarati, A. Bettinelli, et D. Vigo. Enhanced exact solution methods for the team orienteering problem. *International Journal of Production Research*, 54(2) : 591–601, 2016. doi : 10.1080/00207543.2015.1058982. [98]
- S. Kim, K.-N. Chang, et J.-Y. Lee. A descent method with linear programming subproblems for nondifferentiable convex optimization. *Mathematical Programming*, 71(1) :17–28, 1995. [33]
- G. Kirlik et S. Sayın. A new algorithm for generating all nondominated solutions of multiobjective discrete optimization problems. *European Journal of Operational Research*, 232(3) :479–488, 2014. [10, 15 et 17]
- K. Klamroth et M. M. Wiecek. Dynamic programming approaches to the multiple criteria knapsack problem. *Naval Research Logistics (NRL)*, 47(1) :57–76, 2000. [20]
- A. A. Kovacs, S. N. Parragh, et R. F. Hartl. The multi-objective generalized consistent vehicle routing problem. *European Journal of Operational Research*, 247(2) :441–458, 2015. [39]
- N. Labadie et C. Prodhon. A survey on multi-criteria analysis in logistics : Focus on vehicle routing problems. In *Applications of Multi-Criteria and Game Theory Approaches*, pages 3–29. Springer, 2014. [36]
- M. Labbé et G. Laporte. Maximizing user convenience and postal service efficiency in post box location. *Belgian Journal of Operations Research, Statistics and Computer Science*, 26 :21–35, 1986. [110]
- G. Laporte et S. Martello. The selective travelling salesman problem. *Discrete Applied Mathematics*, 26(2-3) :193–207, 1990. [97]
- G. Laporte et Y. Nobert. A branch and bound algorithm for the capacitated vehicle routing problem. *Operations-Research-Spektrum*, 5(2) :77–85, 1983. [24]
- G. Laporte, Y. Nobert, et M. Desrochers. Optimal routing under capacity and distance restrictions. *Operations Research*, 33(5) :1050–1073, 1985. [36]
- M. Laumanns, L. Thiele, et E. Zitzler. An efficient, adaptive parameter variation scheme for metaheuristics based on the epsilon-constraint method. *European Journal of Operational Research*, 169(3) :932–942, 2006. [15, 16, 17, 37 et 39]

- F. Lehuédé, R. Masson, S. N. Parragh, O. Péton, et F. Tricoire. A multi-criteria large neighbourhood search for the transportation of disabled people. *Journal of the Operational Research Society*, 65(7) :983–1000, 2014. [37]
- M. Leitner, I. Ljubić, et M. Sinnl. A computational study of exact approaches for the bi-objective prize-collecting steiner tree problem. *INFORMS Journal on Computing*, 27(1) :118–134, 2014. [17]
- J. Lemesre, C. Dhaenens, et E.-G. Talbi. Parallel partitioning method (ppm) : A new exact method to solve bi-objective problems. *Computers & Operations Research*, 34(8) :2450–2462, 2007. [17]
- R. Lopes, V. A. Souza, et A. S. da Cunha. A branch-and-price algorithm for the multi-vehicle covering tour problem. *Electronic Notes in Discrete Mathematics*, 44 :61–66, 2013. [109, 111, 114 et 115]
- J. Lysgaard, A. N. Letchford, et R. W. Eglese. A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Mathematical Programming*, 100(2) :423–445, 2004. [34 et 36]
- R. E. Marsten, W. Hogan, et J. W. Blankenship. The boxstep method for large-scale optimization. *Operations Research*, 23(3) :389–405, 1975. [33]
- E. Q. V. Martins. On a multicriteria shortest path problem. *European Journal of Operational Research*, 16(2) :236–245, 1984. [20]
- P. Matl, R. F. Hartl, et T. Vidal. Workload equity in vehicle routing : The impact of alternative workload resources. *Computers & Operations Research*, 110 :116–129, 2019. [38]
- G. Mavrotas. Effective implementation of the ε -constraint method in multi-objective mathematical programming problems. *Applied Mathematics and Computation*, 213(2) :455–465, 2009. [16]
- G. Mavrotas et K. Florios. An improved version of the augmented ε -constraint method (augmecon2) for finding the exact pareto set in multi-objective integer programming problems. *Applied Mathematics and Computation*, 219(18) :9652–9669, 2013. [16]
- J. C. Molina, I. Eguia, J. Racero, et F. Guerrero. Multi-objective vehicle routing problem with cost and emission functions. *Procedia-Social and Behavioral Sciences*, 160 :254–263, 2014. [37]
- K. Murakami. A column generation approach for the multi-vehicle covering tour problem. In *2014 IEEE International Conference on Automation Science and Engineering (CASE)*, pages 1063–1068. IEEE, 2014. [109 et 111]
- K. Murakami. Iterative column generation algorithm for generalized multi-vehicle covering tour problem. *Asia-Pacific Journal of Operational Research*, 35(04) :1850021, 2018. [111]

- Z. Naji-Azimi, J. Renaud, A. Ruiz, et M. Salari. A covering tour approach to the location of satellite distribution centers to supply humanitarian aid. *European Journal of Operational Research*, 222(3) :596–605, 2012. [110]
- P. Neumayer et D. Schweigert. Three algorithms for bicriteria integer linear programs. *Operations-Research-Spektrum*, 16(4) :267–276, 1994. [16]
- P. C. Nolz, K. F. Doerner, W. J. Gutjahr, et R. F. Hartl. A bi-objective metaheuristic for disaster relief operation planning. In *Advances in Multi-Objective Nature Inspired Computing*, pages 167–187. Springer, 2010. [110, 113 et 114]
- W. A. d. Oliveira, A. C. Moretti, et E. F. Reis. Multi-vehicle covering tour problem : building routes for urban patrolling. *Pesquisa Operacional*, 35(3) :617–644, 2015. [110, 113 et 114]
- M. Özlen et M. Azizoglu. Multi-objective integer programming : a general approach for generating all non-dominated solutions. *European Journal of Operational Research*, 199(1) :25–35, 2009. [15, 16 et 17]
- M. Özlen, B. A. Burton, et C. A. MacRae. Multi-objective integer programming : An improved recursive algorithm. *Journal of Optimization Theory and Applications*, 160(2) :470–482, 2014. [15]
- Ö. Özpeynirci et M. Köksalan. An exact algorithm for finding extreme supported nondominated points of multiobjective mixed integer programs. *Management Science*, 56(12) : 2302–2315, 2010. [13]
- V. Pareto. *Cours d'économie politique*, volume 1. 1896. [5]
- J. Park, J. Lee, S. Ahn, J. Bae, et H. Tae. Exact algorithm for the capacitated team orienteering problem with time windows. *Mathematical Problems in Engineering*, 2017, 2017. [98]
- S. N. Parragh et F. Tricoire. Branch-and-bound for bi-objective integer programming. *INFORMS Journal on Computing*, 31(4) :805–822, 2019. [19, 20, 39, 99, 104 et 145]
- D. Pecin, A. Pessoa, M. Poggi, et E. Uchoa. Improved branch-cut-and-price for capacitated vehicle routing. *Mathematical Programming Computation*, 9(1) :61–100, 2017. [31, 33, 34 et 36]
- A. Pessoa, R. Sadykov, et E. Uchoa. Enhanced branch-cut-and-price algorithm for heterogeneous fleet vehicle routing problems. *European Journal of Operational Research*, 270(2) :530–543, 2018. [34]
- T. A. Pham, M. H. Hà, et X. H. Nguyen. Solving the multi-vehicle multi-covering tour problem. *Computers & Operations Research*, 88 :258–278, 2017. [112]

- A. Pigatti, M. P. De Aragao, et E. Uchoa. Stabilized branch-and-cut-and-price for the generalized assignment problem. *Electronic Notes in Discrete Mathematics*, 19 :389–395, 2005. [33 et 120]
- M. Poggi, H. Viana, et E. Uchoa. The team orienteering problem : Formulations and branch-cut and price. *10th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS'10)*, 2010. [98]
- A. Przybylski et X. Gandibleux. Multi-objective branch and bound. *European Journal of Operational Research*, 260(3) :856–872, 2017. [18]
- A. Przybylski, X. Gandibleux, et M. Ehrgott. A recursive algorithm for finding all non-dominated extreme points in the outcome set of a multiobjective integer programme. *INFORMS Journal on Computing*, 22(3) :371–386, 2010. [13]
- R. Ramesh, Y.-S. Yoon, et M. H. Karwan. An optimal algorithm for the orienteering tour problem. *ORSA Journal on Computing*, 4(2) :155–165, 1992. [97]
- L. B. Reinhardt et D. Pisinger. Multi-objective and multi-constrained non-additive shortest path problems. *Computers & Operations Research*, 38(3) :605–616, 2011. [20]
- P. Reiter et W. J. Gutjahr. Exact hybrid algorithms for solving a bi-objective vehicle routing problem. *Central European Journal of Operations Research*, 20(1) :19–43, 2012. [37]
- G. Righini et M. Salani. Symmetry helps : Bounded bi-directional dynamic programming for the elementary shortest path problem with resource constraints. *Discrete Optimization*, 3(3) :255–273, 2006. [32]
- G. Righini et M. Salani. Decremental state space relaxation strategies and initialization heuristics for solving the orienteering problem with time windows with dynamic programming. *Computers & Operations Research*, 36(4) :1191–1203, 2009. [97]
- S. Røpke. Branching decisions in branch-and-cut-and-price algorithms for vehicle routing problems. *Presentation in Column Generation*, 2012, 2012. [36]
- L.-M. Rousseau, M. Gendreau, et D. Feillet. Interior point stabilization for column generation. *Operations Research Letters*, 35(5) :660–668, 2007. [33]
- J. Sáez-Aguado et P. C. Trandafir. Variants of the ε -constraint method for biobjective integer programming problems : application to p-median-cover problems. *Mathematical Methods of Operations Research*, 87(2) :251–283, 2018. [17]
- W. Souffriau, P. Vansteenwegen, J. Vertommen, G. V. Berghe, et D. V. Oudheusden. A personalized tourist trip design algorithm for mobile tourist guides. *Applied Artificial Intelligence*, 22(10) :964–985, 2008. [97]

- F. Sourd et O. Spanjaard. A multiobjective branch-and-bound framework : Application to the biobjective spanning tree problem. *INFORMS Journal on Computing*, 20(3) : 472–484, 2008. [10]
- R. E. Steuer et E.-U. Choo. An interactive weighted tchebycheff procedure for multiple objective programming. *Mathematical Programming*, 26(3) :326–344, 1983. [15]
- T. Stidsen, K. A. Andersen, et B. Dammann. A branch and bound algorithm for a class of biobjective mixed integer programs. *Management Science*, 60(4) :1009–1032, 2014. [19]
- J. Sylva et A. Crema. A method for finding the set of non-dominated vectors for multiple objective integer linear programs. *European Journal of Operational Research*, 158(1) : 46–55, 2004. [12, 14 et 43]
- H. Tang et E. Miller-Hooks. A tabu search heuristic for the team orienteering problem. *Computers & Operations Research*, 32(6) :1379–1407, 2005. [96]
- E. M. Toro, J. F. Franco, M. G. Echeverri, et F. G. Guimarães. A multi-objective model for the green capacitated location-routing problem considering environmental impact. *Computers & Industrial Engineering*, 110 :114–125, 2017. [37]
- P. Toth et A. Tramontani. An integer linear programming local search for capacitated vehicle routing problems. In *The vehicle routing problem : Latest advances and new challenges*, pages 275–295. Springer, 2008. [108]
- P. Toth et D. Vigo. *Vehicle routing : problems, methods, and applications*. SIAM, 2014. [24 et 27]
- F. Tricoire, A. Graf, et W. J. Gutjahr. The bi-objective stochastic covering tour problem. *Computers & Operations Research*, 39(7) :1582–1592, 2012. [38, 113 et 114]
- T. Tsiligirides. Heuristic methods applied to orienteering. *Journal of the Operational Research Society*, 35(9) :797–809, 1984. [96 et 97]
- E. L. Ulungu et J. Teghem. The two phases method : An efficient procedure to solve bi-objective combinatorial optimization problems. *Foundations of Computing and Decision Sciences*, 20(2) :149–165, 1995. [12 et 14]
- F. Vanderbeck. Implementing mixed integer column generation. In *Column generation*, pages 331–358. Springer, 2005. [33]
- P. Vansteenwegen, W. Souffriau, et D. Van Oudheusden. The orienteering problem : A survey. *European Journal of Operational Research*, 209(1) :1–10, 2011. [97]
- C. A. Vega-Mejía, J. R. Montoya-Torres, et S. M. Islam. Consideration of triple bottom line objectives for sustainability in the optimization of vehicle routing and loading operations : a systematic literature review. *Annals of Operations Research*, pages 1–65, 2017. [36]

- B. Villarreal et M. H. Karwan. An interactive dynamic programming approach to multi-criteria discrete programming. *journal of mathematical analysis and applications*, 81(2) : 524–544, 1981. [10]
- M. Visée, J. Teghem, M. Pirlot, et E. L. Ulungu. Two-phases method and branch and bound procedures to solve the bi-objective knapsack problem. *Journal of Global Optimization*, 12(2) :139–155, 1998. [19]
- K. Wicksell et al. *Vilfredo Pareto's Manuel d'économie politique*. 1913. [7]
- H. Yu et W. Solvang. An improved multi-objective programming with augmented ε -constraint method for hazardous waste location-routing problems. *International Journal of Environmental Research and Public Health*, 13(6) :548, 2016. [39]
- W. Zhang et M. Reimann. A simple augmented[U+220A]-constraint method for multi-objective mathematical integer programming problems. *European Journal of Operational Research*, 234(1) :15–24, 2014. [16]
- E. Zitzler et L. Thiele. Multiobjective optimization using evolutionary algorithms—a comparative case study. In *International Conference on Parallel Problem Solving from Nature*, pages 292–301. Springer, 1998. [10]
- E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, et V. Da Fonseca Grunert. Performance assessment of multiobjective optimizers : An analysis and review. *TIK-Report*, 139, 2002. [10]