Sandra Ulrich NGUEVEU
Doctorat : Optimisation et Sûreté des Systèmes
Année 2009

## Problèmes de tournées de véhicules avec contraintes particulières pour la maîtrise des risques

Cette thèse porte sur la résolution de deux problèmes de transport qui intègrent des contraintes ou des objectifs de maîtrise des risques: le problème de tournées de véhicules m-péripatétiques (m-PTVP) et le problème de tournées de véhicules cumulatives avec contraintes de capacité (PTVCC). Les applications ciblées sont du domaine de la logistique sécurisée, tel le transport de fonds, et de la logistique humanitaire, tel le transport de 1ers secours.

Le m-PTVP a fait l'objet de la majeure partie de nos travaux. Il consiste à définir des tournées de véhicules de coût total minimal et sans arêtes communes sur m périodes telles que chaque client soit visité une fois par période. Nous proposons différents modèles mathématiques, des bornes inférieures polynomiales, deux métaheuristiques dont un algorithme de recherche taboue guidée par la solution d'un b-couplage parfait, une approche de type génération de colonnes qui combine des heuristiques duales avec la génération de q-tournées, et enfin deux algorithmes de branchements et de coupes.

Le PTVCC identifie les tournées de véhicules qui minimisent la somme des dates d'arrivées chez les clients tout en respectant les contraintes de capacité des véhicules. Nous proposons un algorithme mémétique (MA) dont l'efficacité repose sur le découpage optimal de chromosomes en solutions et l'utilisation de pré-calculs déduits des propriétés du PTVCC que nous avons identifiées, lesquelles produisent également des formules de calcul de bornes inférieures.

Mots clés : recherche opérationnelle - optimisation combinatoire – logistique – transport - tournées de véhicules – sécurité – humanitaire – métaheuristiques - génération de colonnes - branchements et coupes

## Vehicle Routing Problems with Special Constraints for the Risk Management

This research concerns solving two transportation problems that integrate constraints or objective-fonctions related to the risk management: the m-Peripatetic Vehicle Routing Problem (m-PVRP) and the Cumulative Capacitated Vehicle Routing Problem (CCVRP). Applications can be found in the security logistics field, such as money transportation, and the humanitarian logistics field, such as vital goods supply after a natural disaster.

Most of our work is about the m-PVRP. It consists in finding a set of routes of minimal total cost over m periods such that each customer is served once per period and two customers are never sequenced consecutively during two different periods. Different mathematical models are presented and the problem complexity is discussed. Then, we propose polynomial lower bounds, two metaheuristics including a tabu search with diversification and guided by a b-matching solution, a column-generation-like approach combining dual heuristics with q-routes generation, and finally two branch-and-cut algorithms.

The CCVRP consists in minimizing the sum of arrival times at customers, instead of the classical route cost or length, subject to the vehicle capacity. We propose a memetic algorithm (MA). Its efficiency lies upon the optimal splitting procedure of chromosomes without trip delimiters into feasible routes to obtain a feasible solution. Pre-computations to speed up MA and valid formulas to compute lower bounds are deduced from some CCVRP properties we identified.

Keywords: operations research - combinatorial optimization – logistics – transportation - vehicle routing – security – humanitarian – metaheuristics - column generation - branch-and-cut

**Ecole Doctorale** Sciences des Systèmes Technologiques et Organisationnels

Année 2009

Sandra Ulrich NGUEVEU – Tournées de véhicules avec maîtrise des risques

2009TROY0000

Thèse de doctorat de l'UTT

Sandra Ulrich NGUEVEU

# Problèmes de tournées de véhicules avec contraintes particulières pour la maîtrise des risques

Spécialité :
Optimisation et Sûreté des Systèmes

utt
université de technologie
Troyes

# THESE

*pour l'obtention du grade de*

## DOCTEUR de l'UNIVERSITE DE TECHNOLOGIE DE TROYES

**Spécialité : OPTIMISATION ET SURETE DES SYSTEMES**

*présentée et soutenue par*

**Sandra Ulrich NGUEVEU**

*le 27 octobre 2009*

# Problèmes de tournées de véhicules avec contraintes particulières pour la maîtrise des risques

## JURY

| | | |
|---|---|---|
| M. B. GRABOT | PROFESSEUR DES UNIVERSITES | Président |
| M. D. FEILLET | PROFESSEUR | Rapporteur |
| M. C. PRINS | PROFESSEUR DES UNIVERSITES | Directeur de thèse |
| M. F. SEMET | PROFESSEUR | Rapporteur |
| M. K. SORENSEN | PROFESSEUR ASSOCIE | Examinateur |
| M. R. WOLFLER CALVO | PROFESSEUR DES UNIVERSITES | Directeur de thèse |

# Résumé

Cette thèse porte sur la résolution de deux problèmes de transport qui intègrent des contraintes ou des objectifs de maîtrise des risques : le problème de tournées de véhicules $m$-péripatétiques ($m$-PTVP) et le problème de tournées de véhicules cumulatives avec contraintes de capacité (PTVCC). Les applications ciblées sont du domaine de la logistique sécurisée, tel le transport de fonds, et de la logistique humanitaire, tel le transport de 1ers secours.

Le $m$-PTVP a fait l'objet de la majeure partie de nos travaux. Il consiste à définir des tournées de véhicules de coût total minimal et sans arêtes communes sur $m$ périodes telles que chaque client soit visité exactement une fois par période. Nous proposons différents modèles mathématiques, des bornes inférieures polynomiales, deux métaheuristiques dont un algorithme de recherche taboue diversifiée et guidée avec la solution d'un $b$-couplage parfait, une approche de type génération de colonnes qui combine des heuristiques duales avec la génération de $\mathfrak{q}$-tournées, et enfin deux algorithmes de branchements et coupes.

Le PTVCC identifie les tournées de véhicules qui minimisent la somme des dates d'arrivées chez les clients tout en respectant les contraintes de capacité des véhicules. Nous proposons un algorithme mémétique (MA) dont l'efficacité repose sur le découpage optimal de chromosomes en solutions et l'utilisation de pré-calculs déduits des propriétés du PTVCC que nous avons identifiées, lesquelles produisent également des formules de calcul de bornes inférieures. Notre MA est à ce jour la meilleure métaheuristique publiée pour le cas particulier du problème du réparateur itinérant (= PTVCC mono-véhicule = TRP).

**Mots clés :** Recherche opérationnelle, Optimisation combinatoire, Logistique, Transport, Tournées de véhicules, Sécurité, Humanitaire, Métaheuristiques, Génération de colonnes, Branchements et coupes.

# Abstract

This research concerns solving two transportation problems that integrate constraints or objective-fonctions related to the risk management : the $m$-Peripatetic Vehicle Routing Problem ($m$-PVRP) and the Cumulative Capacitated Vehicle Routing Problem (CCVRP). Applications can be found in the security logistics field, such as money transportation, and the humanitarian logistics field, such as vital goods supply after a natural disaster.

Most of our work is about the $m$-PVRP. It consists in finding a set of routes of minimal total cost over $m$ periods such that each customer is served once per period and two customers are never sequenced consecutively during two different periods. Different mathematical models are presented and the problem complexity is discussed. Then, we propose polynomial lower bounds, two metaheuristics including a tabu search with diversification and guided by a $b$-matching solution, a column-generation-like approach combining dual heuristics with $\mathfrak{q}$-routes generation, and finally two branch-and-cut algorithms.

The CCVRP consists in minimizing the sum of arrival times at customers, instead of the classical route cost or length, subject to the vehicle capacity. We propose a memetic algorithm (MA). Its efficiency lies upon the optimal splitting procedure of chromosomes without trip delimiters into feasible routes to obtain a feasible solution. Pre-computations to speed up MA and valid formulas to compute lower bounds are deduced from some CCVRP properties we identified. Up to now, this MA is the best metaheuristic published on the special case of the Traveling Repairman Problem ( = single-vehicle PTVCC = TRP).

**Keywords :** Operations Research, Combinatorial Optimization, Logistics, Transportation, Vehicle Routing, Security, Humanitarian, Metaheuristics, Column generation, Branch-and-cut.

# Remerciements

Tout d'abord, je tiens à remercier mes directeurs de thèse, MM. les professeurs Christian PRINS et Roberto WOLFLER CALVO pour leur encadrement, leur disponibilité, leur soutien et leurs conseils tout au long de ces trois années de recherche. Ils m'ont apporté une compréhension plus approfondie des différents aspects du sujet et de la recherche.

Merci à MM. les professeurs Dominique FEILLET et Frédéric SEMET pour avoir accepté d'être rapporteurs de ce manuscrit, leurs commentaires et leurs remarques m'ont permis d'améliorer la qualité de ce dernier. Merci également au professeur Bernard GRABOT d'avoir présidé le jury de cette thèse, ainsi qu'au professeur Kenneth SÖRENSEN pour avoir accepté de participer au jury.

A présent, je voudrais adresser ma gratitude aux membres et personnels (passé et présent) du LOSI [1] et du pôle ROSAS [2] pour m'avoir accueilli dans un cadre aussi agréable et chaleureux. Je remercie également tous ceux dont j'ai pu faire la connaissance au cours de conférences, séminaires et autres occasions, qui m'ont offert des rencontres enrichissantes et aidé d'une manière ou d'une autre.

J'ai une pensée pour les personnes qui ont contribué à mon parcours à l'UTT [3] depuis mon entrée en tronc commun (Margret DECORS, Olivier BERGOSSI, et tous ceux que, faute d'espace, je ne peux citer). J'ai également une pensée pour les personnes qui ont rendu ces trois dernières années particulièrement agréables : ELLIDOC [4] (notamment Christophe, Léa, Mazen, ...), les équipes de basket (auxquelles j'associe Franck et les deux Stéphane) ... ainsi que tous ceux et celles qui m'ont fait parvenir des mots d'encouragements avant ma soutenance.

Enfin, cette thèse de doctorat est évidemment dédiée à ma famille, mes grand-parents, parents, frères et soeur pour leur présence, même à distance, et leur soutien inconditionnel.

1. LOSI : Laboratoire d'Optimisation des Systèmes Industriels de l'UTT
2. ROSAS : Pôle de Recherche Opérationnelle, Statistiques Appliquées et Simulation de l'UTT
3. UTT : Université de Technologie de Troyes
4. ELLIDOC : Association des doctorants et des docteurs de l'UTT

# Table des matières

**Liste des tableaux** **13**

**Table des figures** **15**

**Glossaire** **17**

**1 Introduction générale** **19**

1.1 Maîtrise des risques . . . . . 19
1.2 Logistique de sécurité . . . . . 19
1.3 Logistique humanitaire . . . . . 20
1.4 Recherche opérationnelle . . . . . 20
1.5 But de la thèse . . . . . 21

**2 État de l'art** **23**

2.1 Introduction . . . . . 23
2.2 Concepts de base . . . . . 24
2.2.1 Notions de graphes . . . . . 24
2.2.2 Notions de programmation linéaire . . . . . 25
2.3 Méthodes (classiques) de résolution . . . . . 27
2.3.1 Méthodes exactes . . . . . 27
2.3.2 Méthodes heuristiques . . . . . 33
2.4 Problèmes de tournées de Véhicules . . . . . 38
2.4.1 Les problèmes classiques . . . . . 39
2.4.2 Le problème du vendeur $m$-péripatétique . . . . . 43
2.4.3 Le problème du réparateur itinérant . . . . . 45
2.4.4 Les arbres et les problèmes de tournées de véhicules . . . . . 47
2.5 Logistique humanitaire et logistique de sécurité . . . . . 48

2.5.1 Tournées de véhicules pour la logistique humanitaire . . . . . 48
2.5.2 Tournées de véhicules pour la logistique de sécurité . . . . . 50
2.6 Conclusion . . . . . 52

**3 Le m-PTVP : formulations mathématiques et bornes inférieures 53**
3.1 Introduction . . . . . 53
3.2 Description, modélisations et complexité . . . . . 54
3.2.1 Données et contraintes . . . . . 54
3.2.2 Modèle basé sur les arêtes . . . . . 55
3.2.3 Modèle de partionnement . . . . . 57
3.2.4 Modèle basé sur le VRP . . . . . 58
3.2.5 Résultats de complexité . . . . . 59
3.3 Bornes inférieures simples . . . . . 60
3.3.1 LB1 : $m$ solutions VRP . . . . . 60
3.3.2 LB2 : $m$-1-arbres recouvrants disjoints . . . . . 60
3.3.3 LB3 : $b$-couplage parfait relaxé . . . . . 63
3.4 Résultats numériques . . . . . 65
3.4.1 Instances et implémentation . . . . . 65
3.4.2 Cas particuliers du VRP et $m$-PVP . . . . . 65
3.4.3 $m$-PTVP . . . . . 67
3.5 Conclusion . . . . . 69

**4 Heuristique et métaheuristiques pour le m-PTVP 71**
4.1 Introduction . . . . . 71
4.2 Heuristique adaptée de Clarke et Wright . . . . . 72
4.3 Recherche locale itérative . . . . . 72
4.3.1 Pénalisation de solutions non réalisables . . . . . 72
4.3.2 Perturbation . . . . . 73
4.3.3 Recherche locale et mouvements . . . . . 74
4.3.4 Mise à jour de la meilleure solution et critère d'arrêt . . . . . 76
4.3.5 Récapitulatif de l'ILS pour le $m$-PTVP . . . . . 76
4.4 Recherche taboue guidée et diversifiée . . . . . 77
4.4.1 Pénalisation des solutions non réalisables . . . . . 77
4.4.2 Recherche du meilleur voisin et gestion de la liste taboue . . . . . 77
4.4.3 Guidage par la solution d'un $b$-couplage parfait . . . . . 78

4.4.4 Diversification . . . . . 79
4.4.5 Critère d'aspiration et critère d'arrêt . . . . . 80
4.4.6 Récapitulatif du GTSD pour le $m$-PTVP . . . . . 80
4.5 Résultats numériques . . . . . 80
4.5.1 Instances et implémentation . . . . . 80
4.5.2 Tests sur le 1-PTVP (VRP) . . . . . 82
4.5.3 Tests sur le $m$-PVP . . . . . 83
4.5.4 Tests sur le $m$-PTVP . . . . . 84
4.6 Conclusion . . . . . 89

**5 Approche de type génération de colonnes pour le $m$-PTVP 91**
5.1 Introduction . . . . . 91
5.2 Formulations mathématiques . . . . . 92
5.3 Principes de l'heuristique duale H1 . . . . . 96
5.3.1 Approximer $\mathscr{R}$ avec l'ensemble $\overline{\mathscr{R}}$ des $\mathfrak{q}$-tournées . . . . . 96
5.3.2 Résoudre ($\overline{\text{RPP}}$) par génération de $\mathfrak{q}$-tournées . . . . . 97
5.3.3 Résoudre chaque problème-maitre sans utiliser de solveur . . . . . 97
5.4 Génération des meilleures $\mathfrak{q}$-tournées . . . . . 100
5.4.1 Génération de $\mathfrak{q}$-chemins . . . . . 101
5.4.2 Fusion de paires de $\mathfrak{q}$-chemins . . . . . 102
5.4.3 Ajout des meilleures $\mathfrak{q}$-tournées dans $\overline{\mathscr{R}}^{(t)}$ . . . . . 104
5.5 Mise à jour lagrangienne des pénalités $\lambda_i$ et $\mu_e$ . . . . . 104
5.5.1 Evaluation de $deg_i$ et $slk_e$ . . . . . 105
5.5.2 Mise à jour de $\lambda_i$ et $\mu_e$ en fonction de $deg_i$ et $slk_e$ . . . . . 105
5.6 Récapitulatif de l'heuristique duale H1 . . . . . 106
5.7 Génération de tournées élémentaires pour H2 . . . . . 106
5.7.1 Génération de chemins élémentaires . . . . . 109
5.7.2 Fusions de paires chemins élémentaires en tournées . . . . . 109
5.8 Résultats Numériques . . . . . 110
5.8.1 Instances et implémentation . . . . . 110
5.8.2 Résultats sur le VRP . . . . . 111
5.8.3 Résultats sur le $m$-PTVP . . . . . 111
5.9 Conclusion . . . . . 113

**6 Approche polyédrale pour le $m$-PTVP 115**

6.1 Introduction . . . 115
6.2 Présentation de la méthode . . . 116
6.2.1 Principe . . . 116
6.2.2 Formulations mathématiques utilisées . . . 117
6.3 Coupes pour la procédure $BC_{PA}$ . . . 120
6.3.1 Contraintes de capacité . . . 120
6.3.2 Contraintes de peigne . . . 121
6.3.3 Contraintes de multi-étoiles . . . 121
6.4 Coupes pour la procédure $BC_{RPA}$ . . . 123
6.4.1 Contraintes agrégées de capacité . . . 123
6.4.2 Contraintes agrégées de peigne . . . 123
6.4.3 Contraintes agrégées multi-étoiles . . . 124
6.5 Séparation des inégalités valides du modèle (PA) . . . 124
6.5.1 Séparation des contraintes de capacité . . . 125
6.5.2 Séparation des contraintes de peigne . . . 127
6.5.3 Séparation des contraintes de multi-étoiles . . . 128
6.6 Séparation des inégalités valides du modèle (RPA) . . . 129
6.7 Parcours et branchement sur les arbres . . . 130
6.8 Résultats numériques . . . 131
6.8.1 Instances et implémentation . . . 131
6.8.2 Résultats de $BC_{PA}$ . . . 132
6.8.3 Résultats de $BC_{RPA}$ . . . 132
6.9 Conclusion . . . 136

**7 Le PTVCC : formulation, métaheuristique et bornes inférieures 141**
7.1 Introduction . . . 141
7.2 Formulation mathématique et bornes inférieures . . . 143
7.2.1 Modèle mathématique . . . 143
7.2.2 Bornes inférieures . . . 147
7.3 Algorithme mémétique . . . 148
7.3.1 Codage et évaluation des chromosomes . . . 149
7.3.2 Initialisation de la population *Pop* et critère de distance . . . 151
7.3.3 Sélection et croisement de deux parents $P_1$ et $P_2$ . . . 151
7.3.4 Recherche locale : mouvements et pré-calculs . . . 153
7.3.5 Remplacement . . . 158

7.3.6 Récapitulatif de notre algorithme mémétique pour le PTVCC . . . 159
7.4 Résultats numériques . . . 159
7.4.1 Instances et implémentation . . . 160
7.4.2 Cas particulier du TRP . . . 160
7.4.3 PTVCC . . . 162
7.5 Conclusion . . . 164

**8 Conclusion générale** **167**

**Annexes** **171**

**Bibliographie** **191**

# Liste des tableaux


2.1 Correspondance primal - dual : règles et exemples . . . . . 26

3.1 Bornes inférieures simples appliquées au VRP = 1-PTVP . . . . . 66
3.2 Bornes inférieures simples appliquées au $m$-PVP ($\lambda = 1$) . . . . . 66
3.3 Bornes inférieures simples appliquées au $m$-PTVP : $m \in \{2, 3, 4, 5, 6, 7\}$ . . 68

4.1 Réglages de paramètres . . . . . 82
4.2 Résultats de $m$CW, ILS et GTSD appliqués au VRP = 1-PTVP . . . . . 83
4.3 Influence des différents composants du (G)TS(D) pour le VRP = 1-PTVP 83
4.4 Résultats de $m$CW, ILS et GTSD appliqués au $m$-PVP ($\lambda = 1$) . . . . . 84
4.5 Influence des différents composants du (G)TS(D) pour le $m$-PVP ($\lambda = 1$) . 85
4.6 Résultats de $m$CW, ILS et GTSD appliqués au $m$-PTVP : $m > 1$ . . . . . 86
4.7 Influence des différents composants du (G)TS(D) pour le $m$-PTVP : $m > 1$ 87
4.8 Robustesse de l'ILS testée sur les instances VRPNC pour le 2-PTVP . . . 87

5.1 Résultats de H1 et H1+H2 sur le cas particulier du VRP = 1-PTVP . . . . 111
5.2 Résultats des heuristiques duales H1 et H1+H2 appliquées au $m$-PTVP : $m \in \{2, 3, 4, 5, 6, 7\}$ . . . . . 112

6.1 Résultats de $BC_{PA}$ sur instances de type A pour le 2-PTVP . . . . . 132
6.2 Résultats de $BC_{PA}$ sur instances de type A pour le 3-PTVP . . . . . 133
6.3 Résultats de $BC_{PA}$ sur instances de type B pour le 2-PTVP . . . . . 133
6.4 Résultats de $BC_{PA}$ sur instances de type B pour le 3-PTVP . . . . . 134
6.5 Résultats de $BC_{PA}$ sur instances de type P pour le 2-PTVP . . . . . 134
6.6 Résultats de $BC_{PA}$ sur instances de type P pour le 3-PTVP . . . . . 135
6.7 Résultats de $BC_{PA}$ sur instances de type VRPNC pour le 2-PTVP . . . . 135
6.8 Résultats de $BC_{PA}$ sur instances de type VRPNC pour le 3-PTVP . . . . 135
6.9 Résultats de $BC_{RPA}$ sur instances de type A pour le 2-PTVP . . . . . 136

6.10 Résultats de $BC_{RPA}$ sur instances de type A pour le 3-PTVP . . . . . 137
6.11 Résultats de $BC_{RPA}$ sur instances de type B pour le 2-PTVP . . . . . 137
6.12 Résultats de $BC_{RPA}$ sur instances de type B pour le 3-PTVP . . . . . 138
6.13 Résultats de $BC_{RPA}$ sur instances de type P pour le 2-PTVP . . . . . 138
6.14 Résultats de $BC_{RPA}$ sur instances de type P pour le 3-PTVP . . . . . 139
6.15 Résultats de $BC_{RPA}$ sur instances de type VRPNC pour le 2-PTVP . . . . 139
6.16 Résultats de $BC_{RPA}$ sur instances de type VRPNC pour le 3-PTVP . . . . 139

7.1 Résultats des versions *normales* des algorithmes pour le $TRP$ . . . . . 161
7.2 Résultats des versions *rapides* des algorithmes pour le $TRP$ . . . . . 161
7.3 Résultats du PTVCC sur les instances VRPNC . . . . . 162
7.4 Robustesse des algorithmes mémétiques . . . . . 164

# Table des figures


3.1 Exemple de solution d'un 2-PTVP ($m = 2$) . . . . . 54
3.2 Détection de sous-tours par les contraintes (3.7) avec $Q = 50$ . . . . . 56
3.3 Respect de la capacité $Q = 50$ avec les contraintes (3.7) . . . . . 57
3.4 Solution optimale (coût = 1148) du 2-PTVP appliqué à l'instance P-n45-k5 59
3.5 Procédure d'*augmentation* pour insérer l'arête $(1, 5)$ sans cycles avec $m = 2$ 62
3.6 Evolution des bornes inférieures en fonction de $m$ . . . . . 67

4.1 Heuristique $m$CW appliquée à l'instance P-n45-k5 pour le 2-PTVP . . . . 73
4.2 Voisinage : les 8 manières d'insérer l'arête (A,B) au cours d'une période . . 75
4.3 Evolution de $\delta$ en fonction du temps pour les instances VRPNC . . . . . 88

5.1 Exemples de $\mathfrak{q}$-tournées . . . . . 96
5.2 Fusion d'une paire de $\mathfrak{q}$-chemins pour obtenir une $\mathfrak{q}$-tournée . . . . . 103

6.1 Comparaison de solutions optimales de (PA) et (RPA) pour $m = 2$ . . . . 119
6.2 Détection de coupes de peigne avec demandes unitaires et $Q = 7$ . . . . . 122
6.3 Détection de coupes de multi-étoiles partielles avec $Q = 3$ . . . . . 122
6.4 Perte de coupes potentielles lors de la contraction du graphe $G(X^*)$ . . . . 130

7.1 Comparaison des solutions optimales de CVRP et OVRP adaptées au PTVCC avec la meilleure solution obtenue par notre métaheuristique sur le PTVCC pour A-n32-k5 . . . . . 142
7.2 Détection de sous-tours par les contraintes (7.7) avec $n = 3$ . . . . . 144
7.3 PTVCC vs VRP . . . . . 145
7.4 Coût d'une tournée $k$ et de son inverse $\bar{k}$ . . . . . 146
7.5 Solution équilibrée/déséquilibrée et *coefficients* associés (ici $n = 7$, $R = 2$) . 147
7.6 Découpage optimal du chromosome *abcde* avec $Q = 10$ et $R = 3$ . . . . . 150
7.7 Exemple de croisement OX . . . . . 153

7.8 Calcul en $O(1)$ de l'impact de l'échange des clients $i$ et $j$ . . . . . 154
7.9 Mouvements de recherche locale . . . . . 155
7.10 Evolution du coût des solutions en fonction de $R$ . . . . . 163

8.1 Fusion de deux tournées . . . . . 172
8.2 Fusion de deux tournées de la période $k^1 = 1$ d'une solution de 2-PTVP . . 173

# Glossaire

BC : algorithme de Branchement et Coupes

CCVRP (*Cumulative Capacitated Vehicle Routing Problem*) : PTVCC

GTSD (*Guided Tabu Search with Diversification*) : algorithme de recherche taboue guidée et diversifiée

ILS (*Iterated Local Search*) : recherche locale itérative

$m$-ARCM : ensemble de $m$ Arbres Recouvrants sans arêtes communes et de Coût total Minimal

$m$-1-ARCM : ensemble de $m$ 1-Arbres Recouvrants sans arêtes communes et de Coût total Minimal

$m$-PSP ($m$-*Peripatetic Salesman Problem*) : $m$-PVP

$m$-PVP : Problème du Vendeur $m$-Péripatétique

$m$-PTVP : Problème de Tournées de Véhicules $m$-Péripatétiques

$m$-PVRP ($m$-*Peripatetic Vehicle Routing Problem*) : $m$-PTVP

MA (*Memetic Algorithm*) : algorithme mémétique = algorithme génétique hybridé avec une recherche locale

PLM : Programme Linéaire Mixte

PLNE : Programme Linéaire en Nombres Entiers

PTVCC : Problème de Tournées de Véhicules Cumulatives avec contraintes de Capacité = VRP dont la fonction-objectif est égale à la somme des dates d'arrivées chez les clients

TRP (*Traveling Repairman Problem*) : problème du réparateur itinérant = TSP dont la fonction-objectif est égale à la somme des dates d'arrivées chez les clients

TSP (*Traveling Salesman Problem*) : problème de voyageur de commerce

VRP (*Vehicle Routing Problem*) : problème de tournées de véhicules

VRPTW (*Vehicle Routing Problem with Time Windows*) : VRP + contraintes de fenêtres de temps

# Chapitre 1

# Introduction générale

Cette thèse s'intéresse à la résolution de deux problèmes de transport qui intègrent des contraintes ou des objectifs de maîtrise des risques : le problème de tournées de véhicules $m$-péripatétiques ($m$-PTVP) et le problème de tournées de véhicules cumulatives avec contraintes de capacité (PTVCC). Les applications ciblées sont du domaine de la logistique de sécurité et de la logistique humanitaire. Ce chapitre d'introduction présente l'intérêt des problèmes étudiés avant d'énoncer le plan de la thèse.

## 1.1 Maîtrise des risques

La maîtrise des risques est désormais un enjeu essentiel sur les plans technologiques, économiques et même sociétaux. Les principales stratégies utilisées sont de l'ordre de la prévention, des actions correctives ou des palliatifs. La prévention consiste à prendre des mesures pour diminuer la probabilité d'apparition de l'événement redouté, comme par exemple varier les itinéraires lors de transports réguliers de fonds pour limiter les risques de braquage. Les actions correctives visent à diminuer les effets et les conséquences du risque lorsque celui-ci intervient, par exemple optimiser l'approvisionnement des populations en denrées vitales après une catastrophe naturelle. Enfin, le palliatif cherche à profiter de l'occurrence du risque sans pour autant en diminuer la probabilité ou les conséquences. C'est le cas typique de l'assurance, qui n'empêche pas les accidents mais propose un "dédommagement" pour le préjudice subi. Le $m$-PTVP et le PTVCC étudiés dans cette thèse relèvent des deux premières stratégies.

## 1.2 Logistique de sécurité

La logistique sécurisée ou logistique de sécurité regroupe les métiers en relation avec la sécurité des systèmes bancaires et celle des réseaux de distribution, comprenant notamment le stockage en chambres sécurisées, les transferts de fonds, de valeurs, de diamants

ou de pierres précieuses, la gestion et le réapprovisionnement des automates de banques, la surveillance de bâtiments, ... Le domaine du transport de fonds en particulier est en pleine expansion non seulement à cause de la croissance régulière de la masse monétaire en circulation, notamment des pièces et billets de banque, mais aussi parce que les commerces et les établissements financiers transfèrent de plus en plus la manipulation et le transport des espèces qu'ils génèrent à des sociétés spécialisées pour la sécurité de leur personnel. Une des conséquences de ces évolutions est le recours régulier aux véhicules banalisés plus discrets que les véhicules blindés sur les territoires nationaux, le développement de systèmes de suivis à l'aide de puces électroniques et la multiplication d'équipements adaptés tels que les valises intelligentes blindées intégrant des systèmes de maculation des billets activés en cas d'effraction.

Le $m$-PTVP concerne justement le transport régulier de fonds, dont l'objectif est de minimiser les longueurs des trajets effectués sur un horizon de $m$ jours, semaines ou mois, tout en variant régulièrement l'ordre de visite d'un ensemble de clients, représentant des commerces ou établissements financiers, à partir d'un dépôt représentant la banque ou la chambre sécurisée.

## 1.3 Logistique humanitaire

Contrairement à la logistique commerciale classique qui a pour but d'assurer le profit et la rentabilité des entités impliquées, la logistique humanitaire a pour finalité de minimiser les pertes en vies humaines, les dommages et les préjudices subis suite à une catastrophe naturelle par exemple, en portant secours au plus vite aux populations sinistrées. L'urgence est donc à la satisfaction des demandes au plus tôt. Les enjeux sont très élevés puisque des vies sont en danger. De plus, les notions d'éthique et d'équité sont omniprésentes car au vu des conséquences potentielles trop importantes, il n'est pas acceptable qu'une population soit négligée par rapport à une autre au nom de critères tels que l'éloignement au dépôt choisi pour stocker les vivres à acheminer.

Ce domaine prend de l'importance car de nombreuses régions dans le monde ont été récemment victimes de catastrophes naturelles de grande ampleur telles que des ouragans, des tremblements de terre, ... Ces dernières ont mis en évidence un besoin important en méthodologies et outils adaptés pour optimiser l'organisation des secours tout en prenant en compte les spécificités de la logistique humanitaire. Le PTVCC s'inscrit dans ce cadre et a pour objectif de minimiser la somme des dates d'arrivées chez les clients au lieu de l'habituel coût total ou longueur du trajet.

## 1.4 Recherche opérationnelle

La recherche opérationnelle a pour but de résoudre des problèmes de décision ou d'optimisation à l'aide d'outils mathématiques et informatiques. Développée en Grande-Bretagne

au cours de la seconde guerre mondiale, lorsqu'il fut décidé d'employer des méthodes scientifiques pour étudier divers aspects des opérations militaires, elle s'est ensuite étendue à d'autres domaines tels que l'industrie, le transport ou encore la finance. En ce qui concerne l'optimisation de transport, qui correspond bien souvent à l'organisation et l'optimisation des tournées d'une ou plusieurs flottes de véhicules, les méthodes de résolution développées sont en général soit des méthodes exactes, dont le but est de trouver une solution optimale du problème posé, soit des méthodes approchées, dont le but est de trouver une solution de bonne qualité mais pas forcément optimale en des temps de calculs raisonnables.

## 1.5 But de la thèse

Deux problèmes de transport sont étudiés dans cette thèse : le problème de tournées de véhicules $m$-péripatétiques ($m$-PTVP) et le problème de tournées de véhicules cumulatives avec contraintes de capacité (PTVCC). Ils font partie de la classe des problèmes de tournées de véhicules sur nœuds, car ils consistent à satisfaire la demande d'un ensemble de clients donnés qui peuvent être représentés par des nœuds sur un réseau.

Le $m$-PTVP fait l'objet de la majeure partie de nos travaux. Il s'agit d'identifier des tournées à coût total minimal qui seront effectuées sur un horizon de $m$ périodes, par exemple des véhicules de transport de fonds devant collecter au cours de chaque période les fonds générés par des clients représentant des commerces ou des établissements financiers, pour les transférer au dépôt où se trouve une chambre sécurisé. L'ordre de visite des clients doit régulièrement varier au cours des $m$ périodes afin de réduire les risques de braquage ; ainsi deux clients quelconques ne doivent pas se succéder plus d'une fois au cours des $m$ périodes. Le $m$-PTVP cherche donc des tournées de véhicules à coût total minimal et sans arêtes communes sur un horizon de $m$ périodes.

Le PTVCC quant à lui consiste à identifier les tournées de véhicules qui minimisent la somme des dates d'arrivées chez des clients dont la demande doit être satisfaite, en partant d'un dépôt où sont stockés tous les produits et en respectant la capacité des véhicules. Cette fonction-objectif reflète la nécessité de satisfaire au plus tôt les besoins des populations sinistrées après une catastrophe naturelle alors que la flotte de véhicules est limitée.

La suite de la thèse est organisée telle que suit : l'état de l'art du chapitre 2 rappelle les concepts de base nécessaires à la compréhension de la suite ainsi que les principes des méthodes classiques d'optimisation de tournées de véhicules sur nœuds, avant de présenter quelques problèmes traités dans la littérature qui peuvent présenter des similitudes avec ceux traités dans cette thèse. Avant d'être résolu, tout problème d'optimisation doit être formalisé à l'aide d'un modèle mathématique. Le chapitre 3 est donc consacré aux formulations mathématiques du $m$-PTVP sur lesquelles seront basées les méthodes de résolution développées par la suite, ainsi qu'aux bornes inférieures polynomiales déduites de ces modèles. Le chapitre 4 décrit les heuristiques et métaheuristiques utilisées pour obtenir des solutions du $m$-PTVP avec des temps de calcul raisonnables. Il s'agit d'un algorithme de recherche taboue guidée et d'un algorithme de recherche locale itérative, tous deux initia-

lisés à partir d'une adaptation de l'heuristique de Clarke et Wright. Le chapitre 5 décrit une procédure de génération de bornes inférieures basée sur des appels successifs à des heuristiques duales et la génération progressive des colonnes du modèle mathématique résolu. Enfin, le chapitre 6 présente une méthode exacte basée sur une approche polyédrale.

Le chapitre 7 est entièrement consacré au PTVCC. Ce dernier possède des propriétés qui le différencient significativement des problèmes classiques de tournées de véhicules. Ces propriétés sont donc identifiées et sont ensuite exploitées au sein de formules de calculs de bornes inférieures, mais également lors de la conception de l'algorithme mémétique adapté au problème. Les solutions de cette métaheuristique sont également comparées à celle obtenues par l'heuristique du plus proche voisin. Enfin, un modèle mathématique est proposé ainsi qu'une étude de l'impact du nombre de véhicules disponibles sur la fonction-objectif.

Pour terminer, une conclusion générale sur les travaux de cette thèse ainsi que sur les perspectives de recherche dans le domaine sont données au chapitre 8.

# Chapitre 2

# État de l'art

## 2.1 Introduction

Bien que la logistique humanitaire et la logistique de sécurité soient plus que jamais d'actualité, comme expliqué dans l'introduction, les deux problèmes étudiés dans cette thèse sont nouveaux et n'ont fait l'objet d'aucune étude auparavant, à notre connaissance. Le problème de tournées de véhicules $m$-péripatétiques ($m$-PTVP) consiste à définir des tournées de véhicules de coût total minimal sur $m$ périodes telles que chaque client soit visité exactement une fois par période et que deux clients ne soient jamais servis consécutivement au cours de deux périodes différentes. Le problème de tournées de véhicules cumulatives avec contraintes de capacité (PTVCC), quant à lui, consiste à identifier les tournées de véhicules qui minimisent la somme des dates d'arrivées chez des clients dont la demande doit être satisfaite, en partant d'un dépôt où sont stockés tous les produits et en respectant la capacité des véhicules. Ces deux problèmes font partie de la classe des problèmes de tournées de véhicules sur nœuds, car ils consistent à satisfaire la demande d'un ensemble de clients donnés qui peuvent être représentés par des nœuds sur un réseau.

Ainsi, avant de développer nos modèles et nos méthodes de résolution, il est intéressant de rappeler dans ce chapitre les concepts de base utiles pour la compréhension des problèmes de tournées de véhicules sur nœuds, des approches classiques de résolution de ce type de problème. Nous présentons également les problèmes existants qui peuvent être considérés comme des cas particuliers du $m$-PTVP ou du PTVCC, ainsi que des travaux en rapport avec les thématiques abordées.

La suite de ce chapitre est donc organisée en cinq sections. Les deux premières rappellent les concepts de base et les méthodes classiques de résolution de problèmes de tournées de véhicules sur nœuds. Les deux suivantes présentent les travaux existants sur l'application des tournées de véhicules dans le cadre du transport de secours (logistique humanitaire) ou du transport régulier de sécurité (logistique de sécurité). Enfin, la conclusion resitue les deux problèmes étudiés dans cette thèse par rapport à la littérature scientifique.

## 2.2 Concepts de base

Un problème de tournées de véhicules consiste à déterminer les tournées qu'une flotte de véhicules doit effectuer afin de livrer un ensemble de clients, souvent en minimisant le coût ou tout autre objectif fixé, tout en respectant un certain nombre de contraintes, par exemple une capacité limitée pour chaque véhicule.

En optimisation combinatoire, deux types de représentations peuvent être utilisées pour modéliser le problème avant de le résoudre : un graphe ou un programme linéaire. Les paragraphes suivants en rappellent les définitions et quelques notions utiles pour la suite.

### 2.2.1 Notions de graphes

Un graphe orienté $G$ est défini par deux ensembles finis $(V, A)$, un ensemble $V$ non vide de sommets (ou nœuds) et un ensemble $A$ de couples de nœuds $(i, j)$ alors appelés arcs. Si quels que soient les nœuds $i$ et $j$ de $V$ les arcs $(i, j)$ et $(j, i)$ appartiennent à $A$, alors on dit que $G$ est un graphe complet. Si on ignore l'orientation, on a un graphe non-orienté $G = (V, E)$ avec des paires notées $[i, j]$ ou $[j, i]$ et appelées arêtes. Il est habituel et pratique de dessiner les graphes sur un plan en représentant les nœuds par des points, les arêtes par des lignes simples qui relient les points extrémités concernés, et les arcs par des flèches partant de leurs origines respectives. Enfin, il est possible d'associer des valeurs, entières ou réelles, positives ou non, aux nœuds et aux arcs ou aux arêtes. Dans ce cas on parle de graphe valué, généralement noté $G = (V, A, C)$ ou $G = (V, E, C)$.

Pour un problème de tournées de véhicules, un nœud correspond souvent à un dépôt ou un client, tandis qu'une arête représente le trajet entre les deux nœuds. Une arête non-orientée veut dire que le trajet entre les deux nœuds peut être effectué dans les deux sens, comme sur une rue à deux sens. Les valeurs associées aux clients peuvent représenter la demande à satisfaire, la durée de service de ce client, l'heure d'arrivée chez ce client, ... tandis que les valeurs des arêtes peuvent correspondre aux durées de trajets entre les deux extrémités, le coût du trajet, la probabilité qu'un incident survienne sur ce trajet, ... On parle de tournées sur nœuds lorsque le service à effectuer se situe sur les différents nœuds (par exemple, lors de la livraison d'œufs dans les boulangeries de la ville), en revanche il s'agit de tournées sur arcs lorsque le service est à effectuer le long d'un arc ou d'une arête du graphe (par exemple, lors du salage ou du nettoyage de routes).

Dans un graphe non-orienté, une chaîne est une suite d'arêtes consécutives tandis qu'un cycle est un cas particulier équivalent à une chaîne fermée. Le cycle est élémentaire s'il ne passe pas deux fois par un même nœud. Dans un graphe orienté, on parle de chemin et de circuit selon Berge [15]. Un graphe est connexe si quel que soit les sommets $i \in V$ et $j \in V$, il existe une chaîne de $i$ à $j$. Une tournée effectuée par un véhicule correspond souvent à un cycle.

### 2.2.2 Notions de programmation linéaire

Un programme mathématique est un problème d'optimisation sous contraintes, c'est à dire qu'il consiste à minimiser (ou maximiser) une fonction sous contraintes. Il s'écrit sous la forme suivante :

$$(\mathrm{P}) \quad \min f(x) \tag{2.1}$$

sous les contraintes (s. c.)

$$g_i(x) \leq 0, \quad i = 1, ..., m \tag{2.2}$$
$$x \in \mathbb{X} \tag{2.3}$$

Ici, le vecteur $x \in \mathbb{X}$ a pour composantes $x_1, x_2, \ldots, x_n$ qui sont les inconnues du problème. La fonction $f$ est appelée fonction-objectif ou fonction-économique. Les conditions (2.2) et (2.3) sont les contraintes du problème (P).

Une solution du problème (P) est un vecteur qui vérifie toutes les contraintes, c'est à dire tel que $g_i(x) \leq 0, (i = 1, \ldots, m)$ et $x \in \mathbb{X}$. On parle aussi de solution réalisable ou solution admissible. Le coût d'une solution est la valeur que prend alors la fonction-objectif $f$. Une solution optimale de (P) est une solution qui minimise $f(x)$ parmi l'ensemble de toutes les solutions réalisables. On parle aussi d'optimum (global).

Un programme mathématique est un programme linéaire (PL) si la fonction-objectif et les conditions à respecter sont linéaires ou affines. On parle de programme linéaire en nombres entiers (PLNE) lorsque les valeurs des variables du programme linéaire doivent être des nombres entiers alors qu'un programme linéaire mixte (PLM) est obtenu lorsque certaines variables doivent être entières et d'autres pas.

Modéliser un problème de tournées de véhicules par un PL ou PLNE revient donc à définir :

- les variables, représentant les décisions qui doivent être prises, par exemple le choix du véhicule qui visite chaque client, l'ordre de visite des clients, ...
- la fonction-objectif exprimée en fonction des variables, représentant le ou les objectifs à remplir, les termes à maximiser ou minimiser
- les contraintes exprimées en fonction des variables, représentant les conditions ou limites (physiques, logiques, temporelles, ...) à prendre en compte lors de la construction de solutions réalisables.

#### Relaxation

Relaxer un PLNE ou un PLM consiste en général à supprimer certaines de ses contraintes, ou alors les remplacer par des contraintes plus faibles. Le problème résultant est souvent plus simple à résoudre que l'original, mais si la relaxation est bien faite alors sa solution optimale peut être réalisable pour le problème original, et donc optimale pour ce dernier, ou alors ne pas être réalisable mais avoir un coût proche de l'optimum du problème original.

En revanche, une relaxation mal choisie produit des solutions dont le coût est très éloigné de l'optimum du problème original. La relaxation linéaire consiste à supprimer les contraintes d'intégrité des variables du PLNE ou du PLM, de manière à obtenir un PL, ce qui revient à autoriser des solutions non-entières, appelées solutions fractionnaires. D'autres de types de relaxations telles que la relaxation lagrangienne (voir paragraphe 2.3.1) modifient la fonction-objectif.

### Dualité

La notion de dualité est un concept fondamental en programmation linéaire. Soit (P) un problème linéaire quelconque dont $A$ est la matrice des contraintes, $x$ le vecteur des variables, $b$ le vecteur des seconds membres et $c$ le vecteur des coûts. On peut associer à (P) son dual (D), défini comme un programme linéaire de matrice de contraintes $A^T$, de vecteur de variables $u$, de vecteur de second membres $c$ et de vecteur de coût $b$, selon le tableau de conversion 2.1. On dit alors que (P) est le problème primal et (D) est son problème dual.

| (P) | **Primal** | **Dual** | (D) |
|---|---|---|---|
| $\min z = c.x$ | Fonction-objectif (min) | Second membre | $\max w = u.b$ |
| | Second membre | Fonction-objectif (max) | |
| s. c. | $A$ matrice des contraintes | $A^T$ matrice des contraintes | s. c. |
| | Contrainte $i : \geq$ | Variable $u_i \geq 0$ | |
| $A.x \leq b$ | Contrainte $i : =$ | Variable $u_i \in \mathbb{R}$ | $u.A^T \leq c$ |
| | Variable $x_j \geq 0$ | Contrainte $j : \leq$ | |
| $x \geq 0$ | Variable $x_j \in \mathbb{R}$ | Contrainte $j : =$ | $u \leq 0$ |

TABLE 2.1 – Correspondance primal - dual : règles et exemples

Ces deux problèmes entretiennent des relations spéciales, dont certaines sont rappelées ci-après :

- Le dual de (D) est (P).
- Si $\bar{x}$ est une solution de (P) et $\bar{u}$ une solution de (D), alors : $\bar{z} = c.\bar{x} \geq \bar{w} = \bar{u}.b$.
- Les solutions optimales $x^*$ et $u^*$ de (P) et (D) ont le même coût $z^* = w^*$.
- Si des solutions de (P) et de (D) ont le même coût, alors ce coût est optimal.
- Le coût réduit $\widetilde{c}_j$ de la variable $\bar{x}_j$ est :

$$\widetilde{c}_j = c_j - \bar{u}A^j \tag{2.4}$$

Ainsi donc, en minimisation, le coût d'une solution de (D) est un minorant pour l'optimum alors que le coût d'une solution (P) est un majorant pour l'optimum. Le coût réduit d'une variable correspond à l'impact sur la fonction-objectif d'une incrémentation (unitaire) de sa valeur courante. A l'optimum le coût réduit de toutes les variables est supérieur ou égal à zéro, car s'il existe des variables à coût réduit négatif, cela signifie que la valeur de ces variables devrait être augmentée pour réduire le coût de la fonction-objectif.

## 2.3 Méthodes (classiques) de résolution

Les méthodes de résolution de problèmes d'optimisation NP-difficiles tels que les problèmes de tournées de véhicules sont classées en deux catégories : les méthodes exactes et les méthodes heuristiques. Les méthodes exactes sont conçues pour trouver un optimum du problème. Toutefois, leurs durées de calcul tendent à augmenter exponentiellement avec la taille des instances des problèmes NP-difficiles qu'elles essaient de résoudre. Elles doivent donc être arrêtées prématurément lors de la résolution d'instances de grandes tailles, et ne produisent alors que des minorants appelés *bornes inférieures* ou des majorants appelés *bornes supérieures* du coût optimal. A l'inverse, les méthodes heuristiques sont des algorithmes conçus pour produire une solution réalisable du problème traité, de bonne qualité mais pas forcément optimale, sans nécessiter des temps de calcul importants. Elles comprennent les heuristiques constructives, qui construisent une solution, les recherches locales, qui génèrent une suite de solution de coûts décroissants jusqu'à l'optimum local, et les métaheuristiques, qui contiennent divers mécanismes pour se sortir des optima locaux. Les méthodes exactes et les méthodes heuristiques sont complémentaires, car leurs résultats permettent de s'évaluer mutuellement : un faible écart entre les meilleures bornes inférieures et bornes supérieures signifie que l'optimum est proche, un écart nul prouve que les valeurs sont déjà optimales. Enfin, il est courant d'utiliser les résultats d'une de ces approches pour améliorer la performance de l'autre.

### 2.3.1 Méthodes exactes

La plupart des problèmes de tournées de véhicules peuvent être formulés sous forme d'un PLNE dont le but est de minimiser une fonction linéaire à variables entières en respectant un ensemble de contraintes linéaires. Le PL résultant de la relaxation continue d'un PLNE peut être résolu à l'aide de l'algorithme du simplexe par exemple. Toutefois, arrondir une solution continue pour obtenir une solution entière n'est pas suffisant et conduit souvent à de mauvais résultats. Il est d'ailleurs facile de construire des exemples où l'optimum continu est très éloigné de l'optimum entier. Résoudre un PLNE nécessite donc l'application de méthodes dédiées, telles que les algorithmes de séparation et évaluation, méthodes de coupes, génération de colonnes, programmation dynamique, relaxation lagrangienne, ... Chacune des méthodes, résumées ci-après, requiert moins de temps de calcul et d'espace mémoire si une borne supérieure de bonne qualité lui est préalablement fournie, en général à l'aide de métaheuristiques efficaces.

#### Méthodes de recherche arborescente par séparation et évaluation

Les méthodes de recherche arborescente telles que l'algorithme de séparation et évaluation introduit par Land et Doig [75] sont basées sur l'idée d'énumérer implicitement les solutions possibles d'un problème PLNE pour y retrouver la meilleure. Il est évident qu'une énumération explicite demanderait beaucoup trop de temps : comme le montre

Minoux [89], un ordinateur capable d'évaluer une solution en $10^{-9}$ secondes (ce qui serait déjà une belle performance) aurait besoin de 30 ans pour énumérer l'ensemble $S$ des $2^n$ solutions potentielles d'un programme linéaire en nombres entiers à $n = 60$ variables, puis 30000 ans pour $n = 70$. Il s'agit donc de rechercher un principe algorithmique permettant de déterminer une solution optimale sans avoir à énumérer explicitement l'ensemble des solutions possibles.

Pour cela, la procédure de séparation divise le problème en un certain nombre de sous-problèmes qui ont chacun leur ensemble de solutions réalisables de telle sorte que tous ces ensembles forment un recouvrement (idéalement une partition) de l'ensemble $S$. Ainsi, en résolvant tous les sous-problèmes et en prenant la meilleure solution trouvée, on est assuré d'avoir résolu le problème initial. Ce principe de séparation peut être appliqué de manière récursive à chacun des sous-ensembles de solutions obtenus, tant qu'il y a des ensembles contenant plusieurs solutions. Les ensembles de solutions (et leurs sous-problèmes associés) ainsi construits peuvent être assimilés à des nœuds dont la hiérarchie naturelle en arbre est appelée arbre de recherche ou arbre de décision.

La procédure d'évaluation a pour but de réduire la taille de l'arbre de recherche en bloquant l'expansion des nœuds qui ne contiennent pas l'optimum. Pour cela, à chaque itération de la procédure de séparation, une borne inférieure (dans le cas d'une minimisation) du coût de la meilleure solution en chaque nœud est calculée. Cette borne inférieure est généralement obtenue en résolvant une relaxation (linéaire, lagrangienne, agrégée...) du problème correspondant au nœud. Si une de ces bornes inférieures excède le coût de la meilleure solution connue, alors on a l'assurance que le nœud correspondant ne contient pas l'optimum. Il est donc inutile de continuer d'explorer un tel nœud, on dit qu'il est "tué". Si le nœud reste vivant alors deux situations peuvent se présenter :

- la solution du problème relaxé est réalisable pour le problème non relaxé en ce nœud. Alors le nœud ne doit plus être séparé et peut être considéré comme une feuille. La solution trouvée est réalisable pour le PLNE initial. Elle remplace la meilleure connue, à condition que son coût soit meilleur.
- la solution du problème relaxé n'est pas réalisable pour le problème non relaxé en ce nœud. Alors la procédure de séparation doit être appliquée en ce nœud.

Avoir de bonnes procédures de calcul de bornes inférieures et une bonne borne supérieure ou solution connue au départ permet donc de tuer plus de nœuds plus vite, ce qui accélère la recherche de l'optimum.

L'exploration de l'arbre de recherche peut être notamment effectuée en largeur, en profondeur ou en donnant la priorité au meilleur nœud. Lors d'une exploration en profondeur (ou *depth-first search*) les nœuds sont traités, c'est à dire évalués, puis tués ou séparés, selon une règle *LIFO - Last In First Out*, c'est à dire en donnant la priorité aux nœuds les plus récents, ce qui permet de limiter les besoins en mémoire pour stocker les nœuds intermédiaires. En revanche, si la priorité est donnée au meilleur nœud (*best-first search*), tous les nœuds existants sont évalués et le nœud séparé en priorité est celui qui produit la borne inférieure la plus faible (dans le cas d'une minimisation). Cette approche consomme plus de mémoire que la précédente car de nombreux nœuds sont stockés en parallèle, mais

elle améliore plus rapidement les valeurs des solutions, d'où des temps d'exécution plus faibles.

Des exemples d'algorithmes de séparation et évaluation pour des problèmes de tournées de véhicules sur nœuds sont disponibles dans les articles de Toth et Vigo [125] ainsi que de Fischetti *et al.* [51].

### Méthodes de coupes

Les méthodes de coupes ou plans sécants introduites par Gomory [57] sont basées sur l'idée de simplifier la résolution d'un problème en relaxant certaines de ses contraintes, généralement les contraintes d'intégrité. Si la solution optimale du problème relaxé est une solution valide pour le problème original, alors elle est l'optimum recherché. Sinon, il existe certainement des contraintes du problème initial qui sont violées par la solution courante. Il faut donc trouver un sous-ensemble de ces contraintes alors appelées coupes, puis les ajouter à la relaxation avant de la résoudre à nouveau. La procédure se poursuit jusqu'à ce que la solution obtenue soit réalisable pour le problème original, ou alors que les procédures d'identification de coupes n'arrivent plus à trouver de contraintes violées. Dans ce dernier cas, le coût de la dernière solution trouvée est une borne inférieure du coût optimal s'il s'agit d'un problème de minimisation, ou alors une borne supérieure dans le cas contraire.

Cette approche est dite polyédrale car une solution d'un programme linéaire correspond à un sommet du polyèdre délimité par les contraintes et qui constitue l'enveloppe convexe des solutions possibles. Lorsque des contraintes sont relaxées, alors le polyèdre résultant est plus large que le précédent et sa solution optimale peut se situer hors du polyèdre original. L'objectif de l'ajout de coupes est donc de refaçonner le polyèdre relaxé en supprimant des parties de l'espace convexe supplémentaire, de manière à ce que la solution optimale du problème relaxé corresponde à une solution réalisable, et donc optimale, du problème original. Une coupe valide doit donc éliminer des solutions non valides pour le problème original sans toucher aux solutions valides de ce dernier. La performance d'une méthode de coupes dépend alors fortement de l'efficacité de sa procédure de génération de coupes.

L'algorithme de branchement et coupes (*branch-and-cut*) résulte de l'utilisation de méthodes de coupes pour calculer de bonnes bornes inférieures au cours de la procédure d'évaluation de la méthode de séparation et évaluation précédemment présentée, afin d'en améliorer la performance. Cet algorithme a par exemple été utilisé par Padberg et Rinaldi [109] pour un problème de voyageur de commerce ou encore Letchford *et al.* [80] pour un problème de tournées de véhicules sans retour au dépôt.

### Méthodes de génération de colonnes

La génération de colonnes est une méthode efficace pour résoudre des programmes linéaires pour lesquels le nombre de variables (ou colonnes) est tellement important qu'on ne peut pas les représenter de manière explicite. L'idée est d'essayer de résoudre un tel problème en tenant compte d'un nombre réduit de variables à la fois, puisque de toute

façon à l'optimum la plupart des variables sont hors base et nulles, c'est-à-dire que seul un (petit) sous-ensemble de variables compte pour résoudre le problème, au pire égal au nombre de contraintes. Une méthode utilisant la génération de colonnes initialise donc le programme linéaire avec un sous-ensemble de colonnes de petite taille, appelé problème-maitre, qui est résolu pour obtenir une solution. Deux situations peuvent alors se présenter :

- soit toutes les colonnes omises ont un coût réduit positif ou nul : cela signifie que la solution obtenue est réalisable et son coût est l'optimum de la relaxation linéaire du problème original, car même si les colonnes délaissées étaient prises en compte, l'impact sur la fonction-objectif serait nul et la solution serait la même.
- soit il existe des colonnes à coût réduit négatif parmi celles qui n'ont pas été prises en compte : cela signifie que si elles avaient été prises en compte, la solution obtenue serait différente et de coût inférieur à celui de la solution courante, qui n'est donc pas optimale pour le problème original. Des colonnes à coût réduit négatif doivent donc être insérées dans le problème-maitre et ce dernier doit être résolu à nouveau.

L'efficacité d'une méthode de génération de colonnes repose donc sur l'efficacité de deux procédures : (1) sa procédure de résolution de chaque problème-maitre, qui doit être rapide puisqu'elle sera répétée à chaque itération (2) sa procédure de détection ou génération des colonnes à coût réduit négatif, qui doit fonctionner sans expliciter toutes les colonnes négligées.

Les coûts réduits sont calculés suivant l'équation (2.4) à l'aide des variables duales $\bar{u}$ dérivant de la solution courante du problème-maitre. En pratique, ces valeurs duales oscillent de manière irrégulière au cours des itérations de l'algorithme. Afin d'en améliorer la convergence, il existe donc des méthodes de stabilisation de la procédure de résolution basées sur l'introduction de perturbations dans le problème-maitre, couplées avec des bornes sur les variables duales, développées par exemple par Du Merle *et al.* [39] pour des problèmes de transport aérien et de localisation. Lorsque la génération de colonnes est combinée avec une procédure de séparation et évaluation, on obtient un algorithme de *branch-and-price*, lequel combiné à une méthode de coupe permet d'obtenir un algorithme de *branch-and-cut-and-price*, comme celui de Fukasawa *et al.* [52] pour un problème de tournées de véhicules avec contraintes de capacité.

### Programmation dynamique

Inventée par Bellman [14], la programmation dynamique est une méthode d'optimisation opérant par phases (ou séquences) dont l'efficacité repose sur le principe d'optimalité de Bellman, selon lequel une séquence optimale (de décisions ou de choix) est composée de sous-séquences optimales. Ce principe permet donc une méthode de résolution ascendante, qui détermine une solution optimale d'un problème à partir des solutions de tous les sous-problèmes : on commence par les sous-problèmes les plus petits et on remonte vers les sous-problèmes de plus en plus difficiles en tirant profit des résultats des problèmes déjà obtenus pour que les calculs à effectuer restent simples et rapides, en général grâce à une formule de récurrence.

La difficulté essentielle de la technique de programmation dynamique réside dans la formulation de la solution du problème sous la forme d'une séquence de décisions ayant des relations de récurrence, permettant de calculer la solution du problème en fonction des solutions de ses sous-problèmes. Son efficacité dépend de l'identification de la propriété récursive et l'efficacité des procédures de résolution des problèmes intermédiaires.

Un exemple simple, mais classique, est celui du calcul du plus court chemin entre deux nœuds $i^*$ et $j^*$ dans un graphe. Considérons $k^* \notin \{i^*, j^*\}$ un sommet intermédiaire sur ce chemin, les sous-chemins de $i^*$ à $k^*$ et de $k^*$ à $j^*$ sont aussi de coût minimal, car sinon, en remplaçant un de ces sous-chemins par un chemin de moindre coût et de mêmes extrémités, on diminuerait le coût du chemin optimal de $i^*$ à $j^*$, ce qui est impossible. Ce problème satisfait donc le principe d'optimalité. L'algorithme de Floyd est une méthode de programmation dynamique qui le résout en calculant successivement, pour k croissant de 1 à $n$, pour tous les couples de sommets $(i, j)$, les chemins de coûts minimaux de $i$ à $j$ parmi ceux dont les sommets intermédiaires sont dans $\{1, \ldots, k\}$, que l'on appelle alors $k$-chemins. Notons $d_{ij}^k$ le coût du $k$-chemin de $i$ à $j$ de coût minimal. Le coût du plus court chemin entre $i$ et $j$ est donc $d_{ij}^n$, calculé de la manière suivante. Considérant un $k$-chemin de coût minimum, l'une des deux situations décrites ci-après se produit forcément :

- ce $k$-chemin ne contient pas $k$, c'est en fait un $(k-1)$-chemin de coût minimum
- ce $k$-chemin contient $k$, on peut le diviser en deux sous-chemins de $i$ à $k$ et de $k$ à $j$, et par principe d'optimalité, chacun de ces sous-chemins est un $(k-1)$-chemin de coût minimal

Par conséquent, l'équation (2.5) est toujours vérifiée.

$$d_{ij}^k = \min(d_{ij}^{k-1}, d_{ik}^{k-1} + d_{kj}^{k-1}) \tag{2.5}$$

Puisque $d_{ij}^0 = c_{ij}$, alors la valeur $d_{ij}^n$ peut être obtenue par récurrence en faisant varier $k$ de 1 à $n$ au sein de l'équation (2.5). Le coût $d_{i^*j^*}^n$ du plus court chemin entre deux nœuds $i^*$ et $j^*$ est ainsi calculé par programmation dynamique en $O(n^3)$, puisqu'une itération en $O(1)$ est effectuée par trinôme $(k, i, j)$ et que le résultat $d_{i^*j^*}^n$ n'est disponible qu'à la fin de l'algorithme.

### Relaxation lagrangienne et méthode de sous-gradient

La relaxation lagrangienne (voir par exemple le livre de Minoux [89]) consiste à transférer certaines des contraintes d'un programme linéaire vers la fonction-objectif de ce dernier et ce sous forme de pénalités, pour obtenir un problème relaxé (RP) qui soit résoluble rapidement. On dit qu'on dualise ces contraintes. Un aperçu de la méthode est donné ci-après.

Supposons que les contraintes $A.x = b$ rendent le problème (P) du tableau 2.1 particulièrement difficile. La relaxation lagrangienne permet de les transférer sous forme de pénalités dans la fonction-objectif de (P). Pour cela, on associe à chaque contrainte $i$ un multiplicateur de Lagrange ou variable duale $\pi_i$, réel positif. En notant $\pi$ le vecteur des pénalités $\pi_i$, la fonction de Lagrange $L(x, \pi)$ est définie tel que suit : $L(x, \pi) = cx + \pi.(b - A.x) =$

$f(x) + \pi.g(x)$. Il est possible de ne dualiser que les contraintes "gênantes", et laisser les autres dans la formulation relaxée (RP).

La relaxation lagrangienne est intéressante à condition de disposer d'un algorithme rapide pour minimiser la fonction de Lagrange pour $\pi$ fixé, c'est-à-dire pour calculer un point de la fonction duale $L(\pi) = \min_{x \in S} L(x, \pi)$. $L(\pi)$ est une borne inférieure de $L(x, \pi)$ pour tout $x$ de $S$ et tout vecteur $\pi$, et donc en particulier pour le coût optimal du primal (P) noté $z^*$.

Pour tout vecteur $\pi$ positif et toute solution réalisable $x$ de (P), $L(\pi) \leq f(x)$. En effet par définition $L(\pi) \leq L(x, \pi) = f(x) + \pi.g(x)$, or $f(x) + \pi.g(x) \leq f(x)$ car $\pi \geq 0$ et $g(x) \leq 0$. D'où $L(\pi) \leq f(x)$ et en particulier $L(\pi) \leq f(x^*)$. D'où la question : quel est le vecteur $\pi^*$ donnant la meilleure borne inférieure ? Ceci revient à résoudre le problème dual de (RP) : (DRP) $\max_{\pi \in \mathbb{R}^{m+}} \{\min_{x \in S} \{L(x, \pi)\}\}$, ou alors la fonction duale : $\max_{\pi \in \mathbb{R}^{m+}} L(\pi)$.

En général, le saut de dualité $z^*$ - $L(\pi^*)$ n'est pas nul car il y a une perte d'information en sommant dans la fonction-objectif des contraintes qui étaient indépendantes. En pratique, ce saut est d'autant plus grand que le problème est difficile, mais les bornes obtenues restent très bonnes.

La relaxation lagrangienne n'a aucun intérêt pour les PL continus, puisque le dual d'un PL continu peut être directement résolu avec l'algorithme du simplexe. Elle est utilisée surtout pour des problèmes d'optimisation combinatoire formulés sous forme de PLNE (et pour des programmes non linéaires). A cause du saut de dualité, elle fournit rarement une solution optimale. Toutefois, les bonnes bornes inférieures obtenues peuvent être utilisées au sein de méthodes arborescentes ou pour évaluer l'écart à l'optimum d'une heuristique.

***Méthode de sous-gradient***

La méthode de sous-gradient est une méthode d'augmentations successives qui permet de trouver le vecteur $\pi$ qui maximise $L(\pi)$, en améliorant les solutions successives trouvées au cours des itérations. Pour cela, notons $i^{(h)}$ la valeur du terme $i$ lors de l'itération $h$. Soit donc $y^{(h)}$ la solution trouvée lors de l'itération $h$, c'est à dire qui vérifie : $L(\pi^{(h)}) = f(y^{(h)}) + \pi^{(h)} g(y^{(h)}) = \min_{x \in S} \{f(x) + \pi^{(h)} g(h)\}$. On peut montrer que l'équation (2.6) est toujours vraie.

$$L(\pi^{(h+1)}) \leq L(\pi^{(h)}) + g(y^{(h)})(\pi^{(h+1)} - \pi^{(h)}) \tag{2.6}$$

Le terme $g(y^{(h)})$ est alors appelé sous-gradient de la fonction duale $L$ en $\pi^{(h)}$ lors de l'itération $h$. La valeur de la fonction peut donc être améliorée au cours de l'itération $h+1$ en utilisant des pénalités $\pi^{(h+1)}$ calculées suivant l'équation (2.7).

$$\pi^{(h+1)} = \pi^{(h)} + \lambda \frac{g(y^{(h)})}{||g(y^{(h)})||} = \pi^{(h)} + \lambda \frac{\gamma(\pi^{(h)})}{||\gamma(\pi^{(h)})||} \tag{2.7}$$

où $\gamma(\pi^{(h)}) = g(y^{(h)})$ tandis que $\lambda$ est un paramètre dont la valeur est régulièrement diminuée (par exemple multipliée par 0,90) après un nombre prédéfini d'itérations.

Au cours des itérations $h \in [1, h_{\max}]$, une variable $LB$ stocke les meilleures valeurs de $L$. Donc, si $LB < L(\pi^{(h)})$ lors de l'itération $h$, alors $LB$ est mis à jour $LB = L(\pi^{(h)})$. La méthode de sous-gradient s'arrête lorsque $||\gamma(\pi)|| = 0$, ou $h = h_{\max}$, ou lorsque $LB$ égale le coût d'une solution de (P) déjà connue, prouvant ainsi qu'elle est optimale.

### 2.3.2 Méthodes heuristiques

Les méthodes heuristiques ont pour but de produire des solutions réalisables de bonne qualité pour des problèmes NP-difficiles. Leurs durées de calcul sont souvent très inférieures à celles des méthodes exactes puisqu'elles ne sont souvent que de complexité polynômiale, ce qui leur permet de traiter des instances de grande taille, les rendant attractives pour la résolution de problèmes réels. L'efficacité de tels algorithmes est jugée selon deux types de critères : (1) de manière empirique en comparant le temps de calcul ainsi que la valeur de la solution obtenue avec la meilleure solution connue ou une borne inférieure (2) de manière mathématique en calculant le ratio dans le pire cas entre la valeur de la solution obtenue et l'optimum. Les paragraphes qui suivent récapitulent les concepts les plus courants pour des problèmes de tournées de véhicules.

#### Heuristiques constructives

Une heuristique constructive est un algorithme dont le but est de fournir rapidement, généralement en temps polynomial, une bonne solution à un problème d'optimisation, mais sans garantie d'optimalité. Les plus connues pour les problèmes de tournées de véhicules sur nœuds construisent une solution avec des principes simples.

L'heuristique du plus proche voisin construit une tournée à la fois ; la tournée courante débute au dépôt, visite à chaque fois le client disponible le plus proche et ne retourne au dépôt que lorsqu'aucun client supplémentaire ne peut y être ajouté, puis une nouvelle tournée débute. Cette heuristique est adaptée aux instances dont les clients sont regroupés par paquets (ou *clusters*). Elle est beaucoup moins efficace sur des instances dont les clients sont éparpillés tout autour du dépôt.

L'heuristique de Clarke et Wright [26] construit une marguerite résultant de la création de tournées à un seul client pour tous les clients, puis fusionne des tournées deux par deux, tant que cela réduit le coût de la solution courante et que chaque nouvelle tournée respecte les contraintes du problème, en particulier la capacité des véhicules. Les fusions sont appliquées en commençant par celles qui produisent les réductions de coût les plus importantes. Cette heuristique a le double avantage de réduire à la fois le nombre de véhicules utilisés et le coût total, ce qui la rend efficace sur la plupart des instances.

Enfin il existe des heuristiques constructives à deux phases de type *route-first cluster-second* ou de type *cluster-first route-second*. Les premières construisent un tour géant contenant tous les clients, puis découpent ce tour en tournées réalisables c'est à dire des tournées qui respectent les contraintes du problème (capacité des véhicules, etc.). C'est le cas par

exemple de l'heuristique d'Ulusoy [129]. Les secondes quant-à elles regroupent d'abord les clients en sous-groupes pouvant être servis par un seul véhicule, avant de construire la meilleure tournée pour chacun de ces sous-groupes. Le regroupage de clients peut être effectué en triant les clients par ordre croissant d'angle polaire par rapport au dépôt, pour ensuite en déduire des secteurs compatibles avec la capacité des véhicules. C'est le cas de l'heuristique Sweep de Gillet et Miller [54].

**Recherche locale**

Pour améliorer la solution produite par une heuristique constructive, il est possible (1) d'exécuter plusieurs heuristiques et garder la meilleure solution, ou (2) rendre aléatoire (randomisation) une composante de l'heuristique, appliquer cette dernière plusieurs fois et garder la meilleure solution obtenue, ou encore (3) appliquer une recherche locale à la solution obtenue. Comme exemple de randomisation, lors de l'application de l'heuristique du plus proche voisin, au lieu de choisir systématiquement le client le plus proche, il est possible de choisir aléatoirement le premier, deuxième ou troisième client le plus proche.

La recherche locale a pour but de vérifier s'il existe de meilleures solutions dans le voisinage d'une solution donnée. Le voisinage est défini comme l'ensemble des solutions qui peuvent être obtenues en appliquant une transformation simple à la solution courante. Pour éviter l'énumération complète de l'espace des solutions, la recherche locale applique donc les transformations ou mouvements qui réduisent le coût total et elle s'arrête dès qu'il n'existe plus de mouvement améliorant. Les principales politiques de recherche locale sont la recherche du premier améliorant ou du meilleur voisin. Dans le premier cas, le premier mouvement améliorant trouvé (c'est à dire, qui réduit le coût) est immédiatement appliqué, alors que dans le second cas, l'ensemble des mouvements possibles à chaque itération sont évalués avant que le meilleur d'entre eux ne soit choisi et appliqué. Cette dernière politique est souvent la plus efficace. La solution obtenue à la fin de la recherche locale est appelée optimum local ou minimum local (dans le cas d'une minimisation), puisqu'il s'agit de la solution de coût minimal dans son voisinage. De nombreux mouvements d'exploration ont été proposés dans la littérature, les plus classiques sont les mouvements *k-opt* [82], *Or-opt* [107], les échanges de chaînes [46] et les chaînes d'éjection [113].

Un mouvement *k-opt* ou *k-échange* consiste à enlever $k$ arêtes non consécutives d'une tournée et les remplacer par $k$ autres arêtes de manière à constituer une nouvelle solution réalisable pour le problème traité. En général $k = 2$ ou $k = 3$, car l'exploration du voisinage $k$-opt (recherche du meilleur mouvement et application de ce mouvement) s'effectue en $O(n^k)$. Un mouvement Or-opt déplace des suites consécutives de 1 à 3 arcs vers d'autres tournées tant que ces dernières respectent les contraintes du problème. Les chaînes d'éjection sont basées sur les mouvements Or-opt, mais si la tournée qui accueille les nouveaux clients devient infaisable alors l'algorithme tente d'éjecter un ou plusieurs des clients de cette dernière vers une autre tournée. Le processus d'éjection se poursuit jusqu'à ce qu'une solution admissible soit trouvée ou que la chaîne des clients éjectés-intégrés soit plus longue qu'un paramètre $h$. Seuls les mouvements qui conduisent à des solutions admissibles sont

finalement appliqués.

### Métaheuristiques

Les métaheuristiques sont des algorithmes capables de s'extraire de minima locaux, ce qui les rend plus performants que de simples heuristiques même combinées avec une recherche locale. Le mécanisme permettant à la métaheuristique de trouver l'optimum local de la solution courante est appelé *intensification* et se ramène souvent à une recherche locale. Par contre le mécanisme permettant de sortir de la région de ce minima local est appelé *diversification* et varie d'un algorithme à l'autre. Les meilleures métaheuristiques permettent d'obtenir d'excellents résultats sur des problèmes de grandes tailles, et ce avec des temps de calculs raisonnables, contrairement aux méthodes exactes présentées dans la section 2.3.1 qui peuvent nécessiter des temps de calculs prohibitifs sur de telles instances. Il existe deux grandes familles de métaheuristiques : les mono-solutions et les multi-solutions. Les premières manipulent une solution à la fois alors que les secondes manipulent une population de solutions et sont pour cela appelées métaheuristiques à population.

Les métaheuristiques mono-solutions les plus connues sont le GRASP (ou *Greedy Randomized Adaptive Search Procedure*), la recherche locale itérative (ou *Iterated Local Search*), la recherche à voisinage variable (ou *Variable Neighborhood Search*), le recuit simulé (ou *Simulated Annealing*) et la recherche taboue (ou *Tabu Search*). Leurs principes sont résumés ci-après.

#### *- Le GRASP, la recherche locale itérative et la recherche à voisinage variable*

Introduit par Feo *et al.* [49], le GRASP crée une nouvelle solution à chaque itération à l'aide d'un algorithme glouton randomisé et l'améliore avec une recherche locale. La métaheuristique s'arrête après un nombre prédéfini d'itérations et la meilleure solution trouvée lors des différentes itérations est restituée comme résultat. Il s'agit donc d'un échantillonnage aléatoire de l'espace des optima locaux (ou *random sampling*).

En ce qui concerne la recherche locale itérative, décrite par exemple dans l'article de Cowling et Keuthen [30], le principe est similaire à celui du GRASP, sauf que la solution initiale de chaque itération est obtenue en appliquant une perturbation aléatoire à la meilleure solution trouvée jusque-là. L'idée est de garder certaines caractéristiques de la meilleure solution au cours d'une marche aléatoire dans l'espace des solutions, de manière à obtenir d'autres solutions encore meilleures.

La recherche à voisinage variable présentée notamment par Mladenović et Hansen [90], est également inspirée du GRASP, mais utilise une série de $K$ voisinages de cardinalité croissante. Le tirage au sort de la nouvelle solution ainsi que la recherche locale s'effectuent alternativement sur les différents voisinages. Sachant que plusieurs variables d'une solution localement optimale gardent la même valeur dans une solution globalement optimale, explorer ainsi différents voisinages permet de remettre en cause un sous-ensemble de variables à la fois, dans le but de retrouver l'optimum global.

***- Le recuit simulé***

Introduit en optimisation combinatoire par Kirkpatrick *et al.* [71], le recuit simulé est inspiré du processus de recuit thermique utilisé en métallurgie dans le but de réduire l'énergie $E$ d'un matériau en alternant les phases de refroidissement lent et les phases de réchauffage (recuit). Dans un problème combinatoire, cette énergie correspond à la fonction-objectif dépendante d'un paramètre fictif $T$, nombre réel positif assimilé à la température du matériau. Partant d'une solution de départ produite par une heuristique, à chaque itération une transformation à effectuer sur la solution courante est tirée au sort. Si cette transformation entraîne une variation de coût $\Delta_E < 0$, alors elle est appliquée à la solution courante. Sinon, elle est acceptée avec une probabilité $e^{-\frac{\Delta_E}{T}}$ et la valeur de $T$ est légèrement diminuée, par exemple en la multipliant par 0,90. L'acceptation d'une « mauvaise » solution permet alors d'explorer une plus grande partie de l'espace des solutions et tend à éviter de s'enfermer trop vite dans la recherche d'un optimum local. Le processus se poursuit jusqu'à ce que $T$ devienne proche de zéro.

***- La recherche taboue***

Introduite par Glover et Laguna [55], la recherche taboue est une métaheuristique qui consiste à balayer complètement le voisinage de la solution courante (initialement fournie par une heuristique), puis à lui appliquer la meilleure transformation possible, même si cette dernière détériore la fonction-objectif. Il s'agit d'une métaheuristique déterministe, contrairement à celles présentées jusqu'ici, puisqu'aucun choix ou paramètre n'est aléatoire. Pour éviter de retourner à des solutions déjà visitées, une file (FIFO) de taille limitée et appelée liste taboue stocke l'historique des dernières solutions visitées. En pratique, stocker des solutions entières consommerait trop de mémoire, c'est pourquoi seules certaines des caractéristiques des dernières solutions visitées sont stockées, par exemple, les mouvements ayant permis de passer d'une solution à l'autre. Une transformation ne peut alors être effectuée que si elle conduit à une solution qui ne possède aucune de ces caractéristiques. L'algorithme s'arrête après un nombre prédéfini d'itérations et la meilleure solution rencontrée au cours de son exécution est restituée comme résultat.

Les métaheuristiques à population les plus connues sont : la recherche dispersée (ou *scatter search*), l'optimisation par essaims de particules (ou *particle swarm optimization*), les colonies de fourmis (ou *ant colony*) et les algorithmes génétiques (ou *genetic algorithms*). Leurs principes sont rappelés dans les lignes qui suivent.

***- La recherche dispersée***

La recherche dispersée est une méthode à population qui utilise des opérateurs de combinaison de solutions et de recherche locale. A chaque itération de cette méthode dont fait l'objet l'article de Martí *et al.* [85], un sous-ensemble de solutions de la population, dont les éléments répondent à des critères de qualité et diversité, est sélectionné et appelé ensemble de référence. Les solutions de l'ensemble de référence sont alors combinées entre elles pour

produire de nouvelles solutions, qui sont ensuite améliorées par la recherche locale. Ces nouvelles solutions remplacent des éléments moins bons de l'ensemble de référence si elles satisfont des critères de qualité et de diversité. Si aucune nouvelle solution n'est admise dans l'ensemble de référence, alors une re-génération partielle de cet ensemble est effectuée et une nouvelle itération commence. L'algorithme s'arrête après un nombre maximum d'itérations et la meilleure solution ayant fait partie de l'ensemble de référence est alors restituée comme résultat.

***- L'optimisation par essaims de particules***

Les algorithmes d'optimisation par essaim de particules ont été introduits par Eberhart et Kennedy [43] et sont inspirés des essaims d'insectes (ou des bancs de poissons ou des nuées d'oiseaux) et de leurs mouvements coordonnés. De même que ces animaux se déplacent en groupe pour trouver de la nourriture ou éviter les prédateurs, les algorithmes à essaim de particules recherchent des solutions pour un problème d'optimisation. Les individus de l'algorithme représentent des solutions et sont appelés particules, tandis que la population entière est appelée essaim. Dans cet algorithme, une particule décide de son prochain mouvement en fonction de sa propre expérience, qui est dans ce cas la mémoire de la meilleure position qu'elle a occupée, et en fonction de son meilleur voisin. Ce voisinage peut être défini en prenant par exemple la distance euclidienne entre les deux particules ou encore la position dans l'essaim de l'individu. Les vitesses et directions de la particule qui définissent le prochain mouvement seront calculées en fonction de trois tendances correspondant à autant de paramètres : la propension à suivre son propre chemin $(1-\rho)$, sa tendance à revenir vers sa meilleure position atteinte $(\rho_1)$ et sa tendance à aller vers son meilleur voisin $(\rho_2)$. Par exemple, une particule $i$ dont le vecteur-position est $\overrightarrow{x_i^{(h)}}$ lors de l'itération $h$ pourra être déplacée vers la position $\overrightarrow{x_i^{(h+1)}}$ qui vérifie : $\overrightarrow{x_i^{(h+1)}} = \overrightarrow{x_i^{(h)}} + \overrightarrow{v_i^{(h)}}$ avec $\overrightarrow{v_i^{(h)}} = \overrightarrow{v_i^{(h)}} + \rho_1(\overrightarrow{x_i^{\text{best}}} - \overrightarrow{x_i^{(h)}})$ où $\overrightarrow{v_i^{(h)}}$ est le vecteur vitesse de $i$ à l'itération $h$ et $\overrightarrow{x_i^{\text{best}}}$ est sa meilleure position connue jusque-là. Les algorithmes à essaim de particules peuvent s'appliquer aussi bien à des données discrètes qu'à des données continues. L'idée est qu'à partir d'optimums locaux et empiriques, l'ensemble des particules va normalement converger vers la solution optimale globale du problème traité. L'algorithme s'arrête dès qu'une convergence est avérée, bien que l'optimum ne soit pas garanti. Un exemple d'application au problème de tournées de véhicules avec fenêtres de temps est donné par Zhu *et al.* [136].

***- Algorithmes de colonies de fourmis***

Les algorithmes de colonies de fourmis proposés par Dorigo *et al.* [38] sont inspirés du comportement des fourmis qui recherchent de la nourriture depuis leur colonie. Quand elles en trouvent, elles marquent le chemin y conduisant en laissant des quantités de phéromones dépendantes de la qualité de la source trouvée (éloignement et quantité). Ainsi, les autres fourmis sont averties et attirées dans les directions les plus prometteuses. Avec le temps, les parcours conduisant aux meilleures sources de nourriture sont de plus en plus fréquentés,

les taux d'hormones augmentant par le passage de fourmis. Par contre, les chemins peu visités voient leur taux de phéromones s'amoindrir par évaporation. Une nouvelle classe de métaheuristiques a donc été proposée, assimilant les trajets réalisés par les fourmis à des parcours de graphe, et la qualité de la source à la valeur de la fonction-objectif. L'initialisation du graphe se fait en attribuant des taux de phéromones nuls sur les arcs. Les fourmis sont représentées par des agents construisant la solution. A chaque itération, ils avancent dans le graphe en effectuant des choix soumis à des probabilités sur les arcs à traverser. La construction de leur chemin est donc biaisée en favorisant les arcs fortement marqués de phéromones. Ensuite, selon la valeur de la fonction-objectif obtenue, les taux de phéromones sont mis à jour. L'algorithme s'arrête après un nombre prédéfini d'itérations. Des exemples d'applications sont proposés par Mullen *et al.* [94].

***- Algorithmes génétiques***

Introduits par Holland [63], les algorithmes génétiques sont des algorithmes d'optimisation inspirés de la génétique et l'évolution naturelle des espèces, qui utilisent des croisements, sélections et mutations pour faire évoluer vers l'optimum une population de solutions représentées sous forme de chromosomes. Ainsi donc, à chaque itération deux chromosomes-parents de la population sont sélectionnés en favorisant les plus prometteurs selon une mesure appelée fitness, puis croisés pour obtenir un ou deux chromosomes-enfants combinant les caractéristiques des parents. Des mutations peuvent être appliquées aux enfants à titre de diversification pour éviter une convergence prématurée de la population. Deux modes de gestion de la population peuvent être utilisés : (1) la gestion incrémentale, où les enfants remplacent certains chromosomes de la population courante, en général choisis parmi les plus mauvais selon le critère de fitness (2) la gestion générationnelle, où la population courante est entièrement remplacée par la population-enfant à chaque itération. L'algorithme s'arrête après un nombre prédéfini d'itérations.

En général, ce type de métaheuristique est moins efficace que les méthodes taboues (par exemple) sur les problèmes de tournées : il doit alors être hybridé avec une recherche locale pour intensifier la recherche. L'algorithme hybride ainsi obtenu est appelé algorithme mémétique. Introduit par Moscato [92], il a été adapté avec succès à de nombreux problèmes de tournées de véhicules, voir par exemple les articles de Prins [112] et de Lima *et al.* [81].

## 2.4 Problèmes de tournées de Véhicules

Cette section est constituée de 5 parties. Les deux premières présentent les deux problèmes classiques dont dérivent tous les autres problèmes de type tournées de véhicules, ainsi que les arbres et leurs liens avec ces problèmes. Les deux suivantes sont consacrées respectivement au problème du vendeur $m$-péripatétique et au problème du réparateur itinérant, deux problèmes dont nous étudions la généralisation dans cette thèse. Enfin, la dernière section présente différents travaux existants sur les problématiques du transport de secours, notamment après une catastrophe naturelle, et du transport de sécurité,

notamment pour le transport régulier de fonds.

### 2.4.1 Les problèmes classiques

Les paragraphes qui suivent présentent les deux problèmes les plus classiques en optimisation de tournées de véhicules. En effet, la plupart des autres problème de cette classe résultent tout simplement de modifications d'un ou plusieurs des éléments suivants :

- la demande, qui peut être déterministe ou stochastique, connue au départ ou seulement à l'arrivée chez le client, nécessiter un temps de livraison ou pas, ...
- la flotte de véhicules, qui peut être homogène ou hétérogène, avoir un coût fixe (coût d'activation) ou pas, ...
- le critère d'optimisation, qui peut être le coût total et/ou le nombre de véhicules utilisés et/ou le nombre de livraisons en retard (hors de la fenêtre horaire), ...
- ...

#### Le problème de voyageur de commerce

Le problème du voyageur de commerce (ou *Travelling Salesman Problem* = TSP) consiste à trouver un cycle de coût total minimal qui traverse chacun des nœuds d'un graphe exactement une fois. On parle alors de recherche de cycle hamiltonien de coût minimal. Il équivaut à minimiser le parcours effectué par un véhicule de capacité égale à la demande totale des clients, qui quitte le dépôt plein et dessert l'ensemble des clients avant de retourner vide au dépôt. Parmi les nombreuses formulations proposées dans la litérature, l'une des plus utilisées est celle de Dantzig *et al.* [31], définie sur un graphe $G = (V, E, C)$ où $V$ est l'ensemble des nœuds et $E$ l'ensemble des arêtes. La variable de décision $x_e$ est égale à 1 si l'arête $e$ de coût $c_e \in C$ est utilisée, et 0 sinon. Soit $\delta(S)$ l'ensemble des arêtes dont une seule des extrémités se trouve dans $S \subset V$ et soit $E(S)$ l'ensemble des arêtes dont les deux extrémités se trouvent dans l'ensemble $S \subset V$, le TSP est défini par les équations (2.8) à (2.11).

$$(\text{TSP}) \min \sum_{e \in E} c_e x_e \tag{2.8}$$

s. c.

$$\sum_{e \in \delta(\{i\})} x_e = 2, \qquad i \in V \tag{2.9}$$

$$\sum_{e \in E(S)} x_e \leq |S| - 1, \qquad S \subset V, 2 \leq |S| \leq n-2 \tag{2.10}$$

$$x_e \in \{0, 1\}, \qquad e \in E \tag{2.11}$$

La fonction-objectif (2.8) cherche à minimiser le coût total du cycle, égal à la somme des coûts des arêtes utilisées. Les contraintes (2.9) précisent que tous les nœuds doivent

être de degré 2, c'est à dire avoir une arête entrante et une arête sortante. Les contraintes (2.10) éliminent les sous-tours, c'est à dire empêchent la création de multiples cycles sans aucuns nœuds en commun. Enfin, les contraintes (2.11) sont des contraintes d'intégrité des variables. Dans cette formulation, les contraintes sont respectivement en nombre $n$, $2^{n-1}-2$ et $\frac{n(n-1)}{2}$. D'autres formulations pour les contraintes de sous-tours ont été proposées, notamment celle de Miller *et al* [87] en nombre $n^2$. Le choix de la formulation à utiliser dépend de la méthode de résolution utilisée. Celle de Miller *et al* [87] par exemple n'est pas adaptée aux méthodes arborescentes parce que ses relaxations conduisent à des bornes inférieures de très faibles valeurs, ce qui ne permet pas de limiter la taille de l'arbre de recherche.

Ce problème est déjà NP-difficile, comme l'ont rappelé Garey et Johnson [53]. De nombreux travaux ont été publiés sur le sujet, notamment le livre de Lawler *et al.* [78], ou plus récemment celui de Applegate *et al.* [2], qui donnent un aperçu des méthodes modernes de résolution de TSP. La TSPLIB [115] est une bibliothèque qui recense les instances connues de la littérature à utiliser lors des tests numériques et comparatifs de nouveaux algorithmes. Il existe également des logiciels Open Source tels que CONCORDE, disponible à l'adresse [27] ou encore LKH, disponible à l'adresse [62] et capables de résoudre la plupart des instances de la TSPLIB. Parmi les développements les plus récents de la littérature, nous pouvons mentionner les méthodes de séparation et évaluation de Naddef et Thienel [95] [96], l'approche colonies de fourmies de Tsai *et al.* [128] ou encore l'approche lagrangienne de Zamani et Lau [135]. A l'heure actuelle, l'instance de TSP euclidien de plus grande taille résolue optimalement comporte 85900 nœuds.

De nombreuses variantes de ce problème existent, entre-autres le problème de collecte et livraisons (Renaud *et al.* [116]), le problème d'orientation (Keller [69]), le TSP avec gain (Feillet *et al.* [48]) ou encore le TSP avec fenêtres horaires (Pesant *et al.* [110]).

### Le problème de tournées de véhicules

Le problème de tournées de véhicules (ou *Vehicle Routing Problem* = VRP) est une généralisation du TSP obtenue lorsque $K$ véhicules de capacité $Q$ sont disponibles au nœud-dépôt pour satisfaire les demandes des nœuds-clients. Chaque véhicule doit donc effectuer une tournée réalisable, c'est à dire quitter le dépôt, visiter une fois des clients dont la somme des demandes ne dépasse pas la capacité $Q$, avant de retourner au dépôt. Chaque client doit être servi par un seul véhicule qui satisfait totalement sa demande.

Une des formulations de VRP les plus utilisées est proposée par Laporte et Norbert [76], qui généralise celle de Dantzig *et al.* [31] pour le TSP. Elle définit donc un graphe $G = \{V, E, C\}$ où $V$ est l'ensemble des nœuds comprenant le dépôt 0 et l'ensemble des clients $V' = V \setminus \{0\} = \{1, \dots n\}$, tandis que $E$ est l'ensemble des arêtes. Chaque client $i \in V'$ a une demande $q_i$ et chacun des $K$ véhicules disponibles a une capacité $Q$ tels que : $\sum_{i \in V'} q_i \leq KQ$. La variable $x_e$ compte le nombre d'utilisations de l'arête $e$ de coût $c_e$. Soit

$\delta(S)$ l'ensemble des arêtes dont une seule des extrémités est dans $S$.

$$(\text{VRP}) \min \sum_{e \in E} c_e x_e \tag{2.12}$$

s. c.

$$\sum_{e \in \delta(\{i\})} x_e = 2, \qquad i \in V \setminus \{0\} \tag{2.13}$$

$$\sum_{e \in \delta(\{0\})} x_e \leq 2K \tag{2.14}$$

$$\sum_{e \in \delta(S)} x_e \geq 2r(S), \qquad \forall S \subseteq V \setminus \{0\}, S \neq \emptyset \tag{2.15}$$

$$x_e \in \{0,1\}, \qquad \forall e \in E \setminus \delta(\{0\}) \tag{2.16}$$

$$x_e \in \{0,1,2\}, \qquad \forall e \in \delta(\{0\}) \tag{2.17}$$

La fonction-objectif (2.12) minimise le coût total de la solution, égal à la somme des coûts des arêtes utilisées. Les contraintes (2.13) précisent que deux arêtes doivent être incidentes à chaque client pour qu'un véhicule puisse y arriver et repartir. Les contraintes (2.14) assurent qu'au maximum $2K$ arêtes sont incidentes au dépôt de manière à ce qu'au plus $K$ véhicules puissent quitter puis retourner au dépôt. Les contraintes (2.15) de sous-tours, où $r(S) = \left\lceil \frac{\sum_{i \in V'} q_i}{Q} \right\rceil$ est une borne inférieure du nombre de véhicules nécessaires pour satisfaire la demande des clients contenus dans $S$, empêchent à la fois la création de cycles non-connectés au dépôt et la création de cycles dont la demande dépasse la capacité $Q$. Enfin, les contraintes (2.16) et (2.17) sont des contraintes d'intégrité. Notons qu'une arête sortant du dépôt $e \in \delta(\{0\})$ peut être utilisée plus d'une fois car une tournée peut visiter un seul client. Les contraintes sont respectivement en nombre $n$, 1, $2^n - 2$, $\frac{n(n-1)}{2}$ et $n$.

Le VRP est évidemment NP-difficile puisqu'il généralise le TSP déjà NP-difficile. Il a également été largement traité dans la littérature et un récapitulatif des travaux est disponible dans le livre de Toth et Vigo [126]. En ce qui concerne les développements les plus récents, une méthode efficace est l'algorithme mémétique de Prins [112]. Cet algorithme résulte de l'hybridation d'une recherche locale au sein d'un algorithme génétique en version incrémentale (décrit dans la section 2.3.2). Il utilise un critère de distance pour accepter ou rejeter les chromosomes-enfants créés après chaque croisement, rendant inutile le recours à la mutation. Il a également la particularité de transformer le problème de calcul du fitness d'un chromosome en problème recherche de plus court chemin dans un graphe auxiliaire à l'aide de l'algorithme de Bellman, ce qui a permis de l'adapter avec succès à d'autres problèmes de tournées de véhicules. Des algorithmes de recherche tabous ont également été utilisés avec succès (Toth et Vigo [127]) et plus récemment des algorithmes de colonies de fourmis (Reimann *et al.* [114]).

En ce qui concerne les méthodes exactes récentes, Letchford *et al.* [80] ont proposé une méthode de branchement et de coupes pour laquelle Lysgaard a développé une biblio-

thèque Open Source CVRPSEP [84] de fonctions de détection de coupes de VRP. Cette bibliothèque a notamment été utilisée par Fukasawa *et al.* [52] dans leur algorithme de *branch-and-cut-and-price* basé sur une formulation qui combine un modèle de partitionnement ayant un nombre exponentiel de colonnes avec le modèle de Laporte et Norbert [76] présenté plus haut. L'algorithme résultant fut jusque très récemment la méthode produisant les meilleurs résultats, permettant par exemple de résoudre optimalement la plupart des instances VRP proposées par Augerat [8]. Mais à ce jour la méthode exacte la plus performante est celle de Baldacci *et al.* [10] basée sur une formulation de partitionnement du VRP où $\mathscr{R}$ est l'ensemble des tournées réalisables et la variable de décision $x_r^k$ est égale à 1 si la tournée $k \in \mathscr{R}$ est utilisée, et 0 sinon. Cette formulation est difficile à résoudre car elle contient un nombre exponentiel de variables correspondant au nombre exponentiel de tournées admissibles. De telles formulations sont normalement adaptées à une approche de type génération de colonnes, mais dans le cas présent Garey et Johnson [53] rappellent que la génération de tournées à coût réduit négatif est un problème NP-complet, ce qui rend difficile une telle approche. En effet, générer un cycle élémentaire de coût inférieur ou égal à une valeur $\gamma$ donnée est un problème polynomial si tous les coûts du graphe sont positifs ou nuls (d'après Itai et Rodeh [66]), mais NP-complet sinon.

L'algorithme de Baldacci *et al.* [10] est appliqué sur une formulation de partitionnement du VRP à laquelle ont été ajoutées des inégalités supplémentaires de capacité et cliques afin que ses relaxations soient plus fortes. Il fait appel à une procédure de calcul de solutions duales proches de l'optimum du dual de la relaxation linéaire de la formulation résultante. Cette procédure est composée de trois heuristiques duales ascendantes (*dual ascent*) chacune étant exécutée à partir des résultats de la précédente, pour améliorer les résultats. La première heuristique est basée sur l'approximation de tournées par des $\mathfrak{q}$-tournées, qui sont des tournées pas forcément élémentaires mais de capacité fixée. La seconde combine la relaxation lagrangienne, la génération de colonnes et la génération de coupes. Enfin, la troisième tente de comber l'écart dual laissé par les deux premières heuristiques en utilisant des techniques classiques de génération de colonnes et de génération de coupes. La solution duale finale est utilisée pour générer un problème de taille réduite ne contenant que les tournées dont le coût réduit est plus faible que l'écart entre la borne inférieure et la borne supérieure. Si les procédures précédentes ont été suffisamment efficaces, le problème résultant peut être résolu par un solveur pour obtenir l'optimum du VRP. Les résultats numériques ont montré que l'algorithme de Baldacci *et al.* domine toutes les autres méthodes exactes à ce jour sur le VRP, il produit de très bonnes bornes inférieures et souvent des solutions optimales, le tout pour un temps de calcul très inférieur à celui des autres méthodes exactes du VRP. Sa généralisation à des variantes du VRP n'est pas triviale (comme pour toute méthode exacte), mais lorsqu'elle est réussie, les résultats produits sont toujours d'excellente qualité, comme le montre l'article de Baldacci *et al.*[11] sur le VRP hétérogène (ou *Heterogenous Vehicle Routing Problem*), une des généralisations possibles du VRP.

De nombreuses variantes du VRP existent et ont été étudiées dans la littérature. Nous pouvons mentionner par exemple le VRP avec collectes et livraisons (Savelsbergh et Sol [122]), le VRP avec fenêtre horaires (Desrochers *et al.* [36]), le VRP avec livraisons divi-

sibles (Archetti *et al.* [6]), ou encore le VRP multi-dépôt (Renaud *et al.* [117]). Enfin, une catégorie de problèmes émergents combinent le VRP avec d'autres problèmes combinatoires de manière à prendre en compte plusieurs niveaux de décision en même temps ou de mieux représenter le problème industriel traité. Nous pouvons citer par exemple le problème de localisation-routage (Von Boventer [131]), ou le problème de tournées de véhicules combinées à la gestion de stock (Federgruen et Zipkin [47]), ou encore le problème le tournées de véhicules avec contraintes de chargement (Ladany et Mehrez [74]).

### 2.4.2 Le problème du vendeur $m$-péripatétique

Le problème du vendeur $m$-péripatétique ($m$-PVP) consiste à trouver $m$ cycles hamiltoniens disjoints à coût total minimal sur un graphe normalement complet $G = (V, E, C)$ où $V$ est l'ensemble des nœuds et $E$ est l'ensemble des arêtes. Chaque arête $(i, j), \forall i \in V, j \in V$ a un coût $c_{ij}$. La formulation (2-PVP) ci-après a été proposée par De Kort [35]. La variable de décision est $x_{ij}^k$ égale à 1 si l'arête $(i, j)$ est utilisée au cours de la période $k$ et 0 sinon.

$$(2\text{−PVP}) \quad \min \sum_{k=1}^{2} \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}^k \tag{2.18}$$

s. c.

$$\sum_{i=1}^{n} x_{ij}^k = 1, \qquad \forall k \in \{1, 2\}, \forall j \in V \tag{2.19}$$

$$\sum_{j=1}^{n} x_{ij}^k = 1, \qquad \forall k \in \{1, 2\}, \forall i \in V \tag{2.20}$$

$$\sum_{i,j \in S} x_{ij}^k \leq |S| - 1, \qquad \forall k \in \{1, 2\}, \forall S \subset V, S \neq V, S \neq \emptyset \tag{2.21}$$

$$x_{ij}^1 + x_{ij}^2 \leq 1, \qquad \forall i, j \in V \tag{2.22}$$

$$x_{ij}^k \in \{0, 1\}, \qquad \forall k = \{1, 2\}, \forall i, j \in V \tag{2.23}$$

La fonction-objectif (2.18) minimise le coût total des cycles hamiltoniens. Les contraintes de degré (2.19) et (2.20) imposent que chaque nœud soit traversé exactement une fois par cycle. Les inégalités (2.21) empêchent la création de sous-tours tandis que les contraintes (2.22) s'assurent que les cycles construits n'aient aucune arête en commun.

Le $m$-PVP est un problème NP-difficile comme l'a montré De Kort [34]. Il est relativement peu étudié dans la littérature, moins d'une dizaine de publications sont disponibles. Krarup [72] a introduit le problème pour la première fois en 1975 et proposé une heuristique simple pour trouver une solution réalisable mais rarement optimale : résoudre un TSP sur le graphe initial, retirer de ce graphe les arêtes utilisées par le cycle trouvé, résoudre un second TSP sur le graphe restant et fusionner les deux cycles trouvés pour obtenir une solution de 2-PVP.

De Kort est l'auteur de 3 articles sur le sujet. Son premier article [33] propose une borne inférieure basée sur le calcul de $2m$ chemins sans arêtes communes allant d'un nœud 1 à un nœud 2 et visitant tous les autres nœuds exactement $m$ fois au total, tout en minimisant le coût total. Il propose aussi d'utiliser comme borne inférieure la valeur de "$m \times c_{TSP}*$" où $c_{TSP}*$ est le coût de la solution optimale du TSP résolu sur le graphe $G$. Dans son second article [34] consacré tout particulièrement au 2-PVP, il propose en plus de $2 \times c_{TSP}*$ d'utiliser une borne inférieure basée sur le calcul de 2 arbres recouvrants sans arêtes communes et de coût total minimal. Puis, des heuristiques réparent les arbres pour les transformer en cycles hamiltoniens, de manière à obtenir une solution réalisable de 2-PVP. Ses résultats obtenus sur des instances euclidiennes de la littérature de 48 à 60 nœuds et des instances euclidiennes générées aléatoirement de 60 à 140 nœuds ont montré un écart entre ses bornes supérieures et inférieures variant de 0,77% à 2,43%, toutefois De Kort montrera par la suite que ses heuristiques donnent de très mauvais résultats sur des instances non-euclidiennes. Enfin, son dernier article [35] présente un algorithme de séparation et évaluation pour le 2-PVP qui utilise comme borne inférieure deux 1-arbres (voir la section 2.4.4) sans arêtes communes et de coût total minimal, qu'il améliore en pénalisant les violations des contraintes de degré des nœuds par une relaxation lagrangienne. La borne inférieure résultante est utilisée au sein de sa procédure de séparation et évaluation qui utilise également toutes les autres bornes (inférieures et supérieures) développées dans les articles précédents pour résoudre optimalement 103 des 105 instances traitées, jusqu'à 60 nœuds pour des instances de la littérature et jusqu'à 130 nœuds pour des instances générées aléatoirement.

Duchenne *et al.* [40] ont proposé des algorithmes de branchement et coupes pour résoudre des instances de la littérature jusqu'à 60 nœuds. Les violations de contraintes sont identifiées en détectant les coupes valides obtenues en généralisant des contraintes de TSP. Des bornes inférieures sont également obtenues en résolvant le problème relaxé résultant de l'agrégation sur $m$ des différentes contraintes. Ce problème agrégé admet comme variables de décision $x_e$ égale à 1 si l'arête est utilisée et 0 sinon, sans distinction de périodes. Ces algorithmes ont été améliorés dans un deuxième article [41] pour proposer de nouvelles inégalités, stratégies de branchement et de vérification de la validité des solutions relaxées, pour pouvoir résoudre des instances de la littérature jusqu'à 280 nœuds. Enfin, des heuristiques [42] ont été développées et testées sur des instances de 100 à 318 nœuds, produisant en moyenne des bornes supérieures à 0.52% des meilleures bornes inférieures connues.

Une étude de cas pratique a été publiée par Cordone et Wolfler [132]. Ils y présentent une approche heuristique ayant permis de résoudre un problème réel d'optimisation de tournées des gardiens d'une société de sécurité, lesquelles incluent des contraintes péripatétiques, mais également d'autres types de contraintes spécifiques à la société de sécurité, telles que des horaires de visites variant d'une période à l'autre pour les tournées ne soient pas prévisibles pour d'éventuels cambrioleurs, ou encore le respect permanent d'une distance minimale du gardien par rapport aux sites dont il est responsable, de manière à mettre un seuil au temps maximum qu'il mettra à intervenir en cas de cambriolage sur n'importe lequel de ses sites. Enfin, l'article De Brey et Volgenant [32] s'intéresse au 2-PVP sur des

graphes non-complets de structures particulières.

Les applications du $m$-PVP, outre les tournées de gardiennage et patrouilles de sécurité, incluent par exemple le design des parcours de véhicules auto-guidés (AGV) au sein d'usines automatisées. Plusieurs parcours sans pistes communes doivent alors être créés pour limiter les encombrements et les accidents qui peuvent se produire lorsque deux ou plusieurs AGV partagent le même espace (Blazewicz *et al.* [18]). Le $m$-PVP correspond alors au cas où tous les parcours passent par tous les nœuds. Enfin, les problèmes de transport de matières dangereuses demandent souvent de créer $m$ cycles hamiltoniens entre plusieurs emplacements. Il est préférable dans ce cas que les cycles créés aient un nombre minimum d'arêtes en commun pour répartir le risque (Lindner-Dutton *et al.* [83]). Enfin, dans le domaine des télécommunications, la fiabilité du réseau augmente lorsqu'au lieu d'un seul cycle, plusieurs cycles sont créés et n'ont aucune arête en commun.

### 2.4.3 Le problème du réparateur itinérant

Le problème du réparateur itinérant (ou *traveling repairman problem* = TRP) consiste à trouver un cycle hamiltonien qui minimise la somme des latences des nœuds-clients, sachant que la latence d'un client est définie comme la distance ou le temps de parcours le long du cycle depuis le nœud-dépôt pour pouvoir atteindre ce client. Il s'agit donc de minimiser la somme des dates d'arrivées chez les clients, ou encore la somme des temps d'attente avant la satisfaction de la demande de chaque client si l'on considère que toutes les demandes arrivent simultanément à l'instant 0. Le TRP est défini sur un graphe $G = (V, E, W)$ où $V$ est l'ensemble des nœuds comprenant le dépôt 0 et l'ensemble des clients $V' = V \setminus \{0\} = \{1, \ldots, n\}$, tandis que $E$ est l'ensemble des arêtes. Chaque arête $e \in E$ a un poids ou une durée de parcours $w_e$. Notons $\mathcal{C}^H$ l'ensemble des cycles hamiltoniens du graphe $G$ et $x = (x_1, x_2, \ldots, x_n) \in \mathcal{C}^H$ un cycle hamiltonien où $x_i$ est le client servi en $i$ème position après le dépôt dans $x$, alors la formulation (TRP) formalise celle de Salehipour *et al.* [120] pour le TRP.

$$\text{(TRP)} \quad \min \sum_{i=1}^{n} l(x_i) \tag{2.24}$$

s. c.

$$l(x_i) = \sum_{j=1}^{i} w_{e=(x_{j-1},x_j)} \tag{2.25}$$

$$x_0 = 0 \tag{2.26}$$

$$x \in \mathcal{C}^H \tag{2.27}$$

Sahni et Gonzalez [119] ont montré que ce problème est NP-difficile. Blum *et al.* [19] ont montré que le TRP ne peut pas être considéré comme une simple variante du classique TSP car il possède des propriétés spécifiques qui le rendent beaucoup plus difficile à

résoudre et à approximer. Ils ont notamment montré que, contrairement à la plupart des problèmes d'optimisation sur graphe, le TRP n'est pas divisible en sous-problèmes à optimiser localement puis regrouper pour obtenir une solution du problème global, car tout changement sur un des composants du TRP a de forte chances de se répercuter sur les autres composants.

Blum *et al.* [19] ont également proposé le premier algorithme d'approximation sur le TRP, de ratio $8\beta$, basé sur une $\beta$-approximation du $k$-arbre de cout minimal recouvrant au moins $k$ nœuds y compris un nœud-racine $r$ prédéfini (ou *rooted k-Minimum Spanning Tree* = $k$-MST). L'idée est que le coût du $k$-MST optimal de racine $r$ est une borne inférieure de la latence du $k$ème nœud visité par le cycle TRP optimal. Cette idée a été améliorée par Geomans et Kleinberg [56] qui ont proposé un ratio de $3{,}59\beta$ avant que Chaudhari *et al.* [22] ne réduisent $\beta$ à $1 + \epsilon$ avec $\epsilon$ une valeur très faible, de manière à obtenir par un algorithme en $O(n^4)$ le meilleur ratio d'approximation connu à ce jour, à savoir 3,59. A titre de comparaison, le meilleur ratio d'approximation de TSP est de 1,5. Notons également que Archer et Williamson [5] ont proposé des algorithmes d'approximation un peu plus rapides en $O(n^3 \log n)$, mais leur ratio n'est que de 9,28 (=3,59 × 2,585).

Concernant les méthodes exactes, nous pouvons citer la programmation dynamique proposée par Wu *et al.* [133] pour résoudre des instances jusqu'à 23 nœuds. Fischetti *et al.* [50], ont résolu le problème de livraison de pizzas à l'aide d'un algorithme de séparation et évaluation en utilisant des bornes inférieures déduites de la formulation du problème sous forme de matroïde, pour résoudre des instances jusqu'à 60 nœuds. Les algorithmes de Bianco *et al.* [17] incorporent des bornes inférieures fournies par une relaxation lagrangienne du TRP et une heuristique basée sur la programmation dynamique, et ce afin de résoudre jusqu'à l'optimum des instances de moins de 30 nœuds et jusqu'à 3% de l'optimum des instances de moins de 60 nœuds. La publication la plus récente est celle de Sarubbi *et al.* [121] qui résouds des instances jusqu'à 80 nœuds après avoir formulé le TRP comme un problème de recherche de plus court chemin avec des contraintes additionnelles sur un graphe multi-niveaux dont les nœuds représentent les positions potentielles de chaque client dans le cycle.

A notre connaissance, la seule métaheuristique sur le TRP a été publiée par Salehipour *et al.* [120]. Il s'agit d'un GRASP combiné à une descente à voisinage variable (ou *GRASP + Variable Neighborhood Descent = GRASP+VND*). Testé sur des instances de 10 à 1000 nœuds, cet algorithme a permis d'améliorer les résultats produits par l'heuristique du plus proche voisin de 2% à 12%. Ils présentent également une borne inférieure simple déduite d'un arbre recouvrant grâce à l'affection de coefficients aux coûts de ses arêtes. L'écart entre cette borne inférieure simple et les bornes supérieures produites par les métaheuristiques varie entre 33% et 55%.

Le TRP est connu sous plusieurs noms : le TSP avec coûts cumulatifs (ou *TSP with cumulative costs*) le problème de minimisation des latences (ou *minimum latency problem*), le problème du livreur (ou *delivery man problem*), ou encore le problème du conducteur de bus scolaire (ou *school-bus driver problem*). Contrairement au TSP, il considère souvent un dépôt ou nœud-origine, et certains articles prennent parfois en compte le retour à ce dépôt,

ce qui ne change pas le principe des méthodes appliquées, mais modifie le coût optimal obtenu sur une instance. De nombreuses applications, outre la distribution, ont été décrites dans la littérature, de l'ordonnancement sur machines industrielles, aux tournées de vehicules automatisés (ou *Automated Guided Vehicle* = AGV) dans des usines automatisées, la recherche d'une information stockée sur un réseau ou encore l'optimisation d'énergie pour les systèmes de télécommunications sans fils, comme l'illustrent les articles de Simchi-Levi et Berman [123] ou encore de Bertacco *et al.* [16].

### 2.4.4 Les arbres et les problèmes de tournées de véhicules

Les arbres ont souvent été utilisés pour produire des bornes inférieures pour des problèmes de tournées de véhicules, en particulier le TSP et ses variantes. Un arbre $A$ est défini comme un graphe connexe acyclique, c'est à dire un ensemble de nœuds et d'arêtes sans cycle et tel que quels que soient deux de ses nœuds $u$ et $v$, il existe un chemin de $u$ à $v$ composé uniquement d'arêtes de $A$. Un arbre recouvrant défini dans un graphe est un arbre qui utilise tous les nœuds de ce graphe. Une forêt est définie comme un graphe acyclique pas forcément connexe. Elle correspond donc à une union d'arbres et/ou de nœuds (d'où le nom).

Dans un graphe où tous les coûts sont positifs ou nuls, l'arbre recouvrant de coût minimal (ARCM) fournit une borne inférieure au TSP. En effet, soit $s$ une solution d'un TSP, soit $c_s$ son coût et soit $c_{\text{ARCM}}$ le coût du ARCM défini sur le même graphe. Si l'on retire une arête $e$ de $s$ alors on obtient $s' = s \setminus \{e\}$ qui vérifie $c_{s'} \leq c_s$. De plus, $s'$ est un arbre recouvrant dont tous les nœuds sont de degré 2 à l'exception des deux extremités qui sont de degré 1. Il est donc évident que $c_{s'} \geq c_{\text{ARCM}}$ puisque par définition aucun arbre recouvrant ne peut avoir un coût inférieur à $c_{\text{ARCM}}$. D'où $c_{\text{ARCM}} \leq c_{s'} \leq c_s$. Le problème de recherche d'un ARCM peut être résolu polynomialement avec l'algorithme de Kruskal.

Le TSP fournissant une borne inférieure au VRP, le coût d'un ARCM est donc également une borne inférieure du VRP, mais selon les instances traitées, elle peut être de très faible qualité, puisque la capacité des véhicules n'est pas prise en compte. Chaque tournée d'un problème de tournées de véhicules peut être approximée par un arbre, en supprimant une arête par tournée et en calculant le coût d'arbres disjoints à coût total minimal. Notons toutefois que rajouter des contraintes au ARCM permet d'augmenter son coût et donc la borne inférieure, mais le problème résultant devient souvent NP-difficile, comme l'ont rappelé Garey et Johnson [53]. Par exemple, des contraintes de degré sur les nœuds peuvent être ajoutées, mais le problème résultant est NP-difficile. Une alternative consiste à pénaliser les violations de contraintes avec la relaxation lagrangienne.

Proposé par Held et Karp [61] au cours des années 70, un 1-arbre (ou 1-ARCM) résulte de l'union de deux arbres : d'une part un arbre de coût minimal recouvrant $n-1$ nœuds du graphe (par exemple $V \setminus \{i_0\}$), et d'autre part l'arbre formé par les deux arêtes de coût minimal incidentes à ce $i_0$ et les nœuds associés. L'objet résultant contient donc un seul cycle, et comme l'ont montré Held et Karp, son coût est une borne inférieure du cycle

hamiltonien de coût minimum, et donc du TSP.

## 2.5 Logistique humanitaire et logistique de sécurité

Cette section présente un récapitulatif des travaux disponibles sur les thèmes du transport de secours et du transport régulier de sécurité.

### 2.5.1 Tournées de véhicules pour la logistique humanitaire

Après une catastrophe naturelle, le délai d'arrivée des secours et des approvisionnements auprès des populations affectées a un impact évident sur le taux de survie des habitants et les préjudices occasionnés. Pourtant, au sein de tournées de véhicules créées pour un problème classique, certaines populations pourraient être servies bien plus tard que d'autres pour minimiser la distance parcourue ou le coût total. Il est donc évident que la modélisation du transport de secours sous la forme d'un problème de tournées de véhicules nécessite la définition d'une nouvelle fonction-objectif qui reflète au mieux les priorités du premier cité.

Partant de ce constat Campbell *et al.* [20] se sont intéressés à deux fonctions-objectifs alternatives pour le transport de secours (ou *vehicle routing for relief problems*) : (1) la minimisation de la date d'arrivée maximale sur chaque site et, (2) la minimisation de la date moyenne d'arrivée sur les différents sites à servir. Elles peuvent être assimilées à (1) la minimisation de la distance maximale à parcourir depuis le dépôt pour atteindre chaque site et, (2) la minimisation de la distance moyenne à parcourir depuis le dépôt pour que chaque site soit servi. Campbell *et al.* ont ensuite mené une étude préliminaire afin de démontrer la pertinence de ces fonctions-objectifs, à travers quelques études de pire cas et l'application d'heuristiques sur des instances simples (demande unitaire, ...) de différents types (randomisées, clusterisées, ...). Ayant montré que le changement de fonction-objectif pouvait améliorer les solutions trouvées et donc le secours des victimes, Campbell *et al.* suggèrent donc que les prochaines études sur le transport de secours se penchent vers le développement de méthodes plus sophistiquées prenant par exemple en compte la différence de demandes entre les sites sinistrés.

Ces travaux de Campbell *et al.* reflètent un effort émergent pour utiliser les techniques de recherche opérationnelle dans le cadre de la logistique humanitaire, actuellement en demande croissante d'expertise et d'outils performants, comme le montre par exemple le rapport de l'institut Fritz [65]. Peu d'articles de la littérature modélisent et résolvent des problèmes de logistique humanitaire à l'aide de tournées de véhicules ; Campbell *et al.* sont les seuls à avoir proposé de modéliser le problème de transport de secours sous la forme d'une variante du VRP obtenue en modifiant sa fonction-objectif.

Hsueh *et al.* [64] ont proposé de modéliser le problème transport de vivres (nourriture, eau, ...) et non vivres (médicaments, kits médicaux, ...) sous la forme d'un problème de tournées de véhicules dynamique (ou *dynamic vehicle routing problem for relief logistics* =

DVRP-RL). Les demandes de collecte ou de livraison des différents produits peuvent varier en temps réel et sont considérées imprévisibles. Certaines variations des durées de trajets peuvent être prévisibles (par exemple le rétablissement de certains tronçons, l'évolution du trafic en fonction du temps, ...) et d'autres non, en cas d'incidents de trafic par exemple. A chaque site-demandeur, plusieurs types de vivres et non vivres sont réclamés, avec des niveaux d'urgence et des délais différents. De plus, il est possible que de la collecte et de la livraison soient requises simultanément, puisque les produits en excès sur un site peuvent être récupérés pour être réutilisés dans n'importe lesquels des sites qui suivent dans le même trajet. Afin d'avoir une réponse rapide aux demandes futures, un véhicule dispatché ne retourne au dépôt que s'il a besoin d'être rempli à nouveau. Dans le cas contraire, il attend chez son dernier site visité jusqu'à réception des prochaines instructions du centre de coordination logistique. Afin de restituer l'aspect dynamique du problème, Hsueh *et al.* ont adopté le concept de Chang *et al.* [21] et ont reformulé le problème comme une série de modèles PLM dans un horizon de temps déroulant. Chaque modèle représente un cas particulier de VRPTW à un instant donné lorsque les temps de parcours et les demandes ont changé. Une fois qu'une nouvelle demande est apparue ou que les temps de trajets ont changé, le domaine du problème restant doit être ajusté et résolu à nouveau.

Gong *et al.* [58] considèrent des modèles d'allocation et réallocation d'ambulances pour une opération de secours post-catastrophe. Ils formulent un modèle déterministe qui décrit comment croît un cluster de victimes après l'occurrence d'une catastrophe, et considèrent l'objectif de minimiser le makespan (= durée totale = différence entre début et fin du déploiement) pour déterminer l'allocation et réallocation des ambulances. Yi and Özdamar [134] décrivent un modèle de localisation-distribution pour coordonner les moyens logistiques et les opérations d'évacuation en réponse à une catastrophe naturelle. La planification logistique considérée inclue l'expédition de biens (matériels médicaux, équipes de secours, ...) à des centres de distribution au sein de régions affectées ainsi que l'évacuation et le transfert des blessés vers des unités d'urgence (ou *emergency unit*). Il s'agit donc d'un problème de multiflots modélisé par un PLM qui traite les véhicules comme des flux de commodités entiers plutôt que des variables binaires.

Özdamar *et al.* [108] proposent un modèle de planification à intégrer dans un système d'aide aux décisions de logistique de crise suite à une catastrophe naturelle. Ce modèle s'applique au problème de transport dynamique qui doit être résolu de manière répétitive après un certain intervalle de temps, et ce durant toute la durée de la distribution de l'aide. Le modèle re-génère de nouveaux plannings qui incorporent les nouvelles demandes en vivres, matériels et les moyens de transport devenu disponibles. Ce planning indique les emplois du temps optimaux de collecte et livraison pour les véhicules sur l'horizon de planification, ainsi que les quantités optimales et les types de chargements collectés et livrés tout au long des tournées.

Mohaymany *et al.* [91] ont proposé une méthode pour obtenir des "chemins de secours" ou "chemins d'urgence" (ou *emergency paths*), basée sur deux critères permettant de diminuer les pertes en vies humaines. Le premier est la minimisation de la distance entre les centres de secours et la population en détresse ; le deuxième maximise la capacité de

service des forces de secours. Les principaux éléments pris en compte dans le processus de sélection des chemins sont d'une part les régions les plus habitées, les plus vulnérables, ou les plus à risque et d'autre part les centres de secours. La procédure de sélection de chemins fonctionne avec un système d'information géographique (SIG).

Barbarosoglu *et al.* [13] ont présenté un modèle mathématique pour la planification de missions d'hélicoptères au cours d'une opération de secours suite à une catastrophe naturelle. Ils décomposent les décisions à prendre en deux sous-problèmes correspondant à deux niveaux de décision : (1) les décisions tactiques prises par la hiérarchie en amont concernant la flotte d'hélicoptères, l'affectation des pilotes aux hélicoptères, le nombre de tournées effectuées par chaque hélicoptère et, (2) les décisions opérationnelles prises en aval concernant les trajets et les chargements à effectuer. Chacun des sous-problèmes est NP-difficile et est résolu par un solveur commercial pour de petites instances. Une procédure de coordination itérative assure la cohérence entre les deux sous-problèmes, en transférant des prévisions en aval pour améliorer les décisions prises en amont. A cause du caractère conflictuel des objectifs des deux niveaux, une approche multi-critères est utilisée, puis une procédure itérative définie au niveau tactique sélectionne la solution préférentielle parmi les solutions non-dominées.

Altay et Green III [1] ont publié en 2006 un récapitulatif des publications concernant la recherche opérationnelle et les sciences du management appliquées à la logistique de crise (ou *disaster management*). Ils soulignent le fait que de nombreuses questions restent en suspend, comme par exemple le problème de ramassage de débris qui n'a été abordé que récemment (Lauritzen [77]).

### 2.5.2 Tournées de véhicules pour la logistique de sécurité

Les problèmes de transport régulier incluant des contraintes de sécurité sont peu étudiés dans la littérature. Mourgaya et Vanderbeck [93] ont proposé une heuristique pour un problème de tournées de véhicules multipériodiques qu'ils ont ensuite testé sur une instance fournie par un partenaire industriel pour un problème de transport de fonds. Toutefois, leur modèle n'inclut pas de contraintes spécifiques à la sécurité. Le problème de tournées de véhicules multipériodiques consiste à planifier des visites de clients sur un horizon de temps donné en les affectant à des tournées de véhicules. Les fréquences de visite et les demandes des clients dans certains cas peuvent varier en fonction des périodes.

Mourgaya et Vanderbeck [93] abordent le problème en considérant deux niveaux, un niveau tactique et un niveau opérationnel. Au niveau tactique sont décidées les périodes de visite pour chaque client et l'affectation de chaque client au véhicule qui le servira, alors que les tournées à réaliser par chacun des véhicules sont définies au niveau opérationnel. Leurs objectifs ne concernent que le niveau tactique, laissant le soin à un logiciel-maison ou dans certains cas aux chauffeurs eux-mêmes de définir les tournées. Deux objectifs sont alors considérés : (1) la régionalisation, qui consiste à concentrer chaque tournée autour d'une région, en particulier lorsque la connaissance géographique d'un secteur est un para-

mètre important, (2) l'équilibrage des tournées par période et par véhicule. En revanche, l'optimisation du coût des tournées est fait au niveau opérationnel. Contrairement à la plupart des méthodes qui reposent sur l'utilisation intensive des procédures d'amélioration, Mourgaya et Vanderbeck utilisent une heuristique constructive qui explore largement l'espace des solutions et une recherche locale qui se contente d'améliorer la régionalisation des tournées sans détruire l'équilibrage de la charge de la solution.

L'heuristique construit des solutions Pareto-optimales du point de vue des critères d'équilibrage de la charge et de la régionalisation des clients visités. En optimisation multi-critère, une solution est Pareto-optimale si et seulement si pour au moins un des critères d'optimisation, elle est meilleure que n'importe quelle autre solution. Dans ce cas, on dit qu'elle fait partie du front de Pareto, dans le cas contraire on dit qu'elle est dominée. L'heuristique est basée sur trois concepts : (1) ordonner les clients en donnant la priorité à ceux susceptibles de provoquer les déséquilibres les plus importants dans le planning, par exemple ceux ayant une forte demande, puis (2) affecter les clients de manière gloutonne à un scénario possible, c'est à dire un ensemble de tournées potentielles parmi les journées potentielles, créant ainsi des solutions intermédiaires qui sont complétées au fur et à mesure des itérations, et enfin (3) évaluer chaque solution selon les deux critères de répartition de charge et de régionalisation. Le nombre de solutions intermédiaires augmente de manière exponentielle, et pourtant il n'est pas possible de prédire le devenir de chacune des solutions partielles, même avec la dominance au sens de Pareto, car une solution intermédiaire dominée peut donner lieu à une solution-finale qui soit Pareto-optimale. Il faut donc appliquer des critères heuristiques pour limiter le nombre de solutions considérées, par exemple, imposer une borne supérieure $K$ du nombre de solutions intermédiaires. A la fin de l'algorithme nous disposons donc d'un panel de K solutions pareto-optimales. La recherche locale tente soit d'échanger complètement les clients qui ont les mêmes fréquences de visite, soit d'échanger seulement au sein d'une période les clients de mêmes ou différents véhicules.

Les tests numériques sur une instance de 6172 clients sur 20 jours montrent des résultats numériques similaires à ceux de l'industriel, pour un temps de calcul plus raisonnable (3 heures au lieu de 20 heures). Ce modèle a l'avantage d'être applicable à n'importe quel problème multipériodique. Toutefois, il n'inclut pas de contraintes de sécurité spécifiques au transport régulier de fonds. En effet, une des caractéristiques du transport régulier de fonds, est que les tournées doivent régulièrement varier. Si les clients visités varient systématiquement, alors l'imprévisibilité devient implicite, toutefois, s'il s'agit principalement de clients réguliers et bien installés (banques, commerces, ...) alors il devient important de varier les itinéraires, ce qui correspond aux contraintes de péripatéticité du $m$-PTVP.

Rappelons que le $m$-PVP, qui trouve $m$ cycles hamiltoniens disjoints à coût total minimal, interdit donc de dupliquer le même ordre de livraison / séquence de visite des clients sur un horizon de $m$ périodes. Il a notamment été utilisé par Wolfler-Calvo et Cordone pour la création de l'heuristique de définition des tournées de gardiens et patrouilles. Un découpage de la ville en neuf régions étant fait au préalable par l'entreprise, l'heuristique résout donc un $m$-PVP par région. Toutefois, si ce découpage était levé dans un second

temps, comme l'ont suggéré les auteurs, on retomberait alors sur une généralisation du problème avec contraintes de capacité (un seul gardien ne peut pas tout faire !) et des demandes unitaires.

D'autres problèmes tels que le transport de matières dangereuses (Lindner-Dutton *et al.* [83]) demandent également de varier la solution pour étaler les risques, ce qui se traduit par des contraintes de péripatéticité.

## 2.6 Conclusion

Le but de ce chapitre était de présenter une vue d'ensemble pour une bonne appréhension des problèmes de tournées de véhicules et de leurs méthodes classiques de résolution avant d'aborder les problématiques liées à la logistique humanitaire et à la logistique de sécurité. Il en ressort que, malgré des publications en nombre croissant sur ces thématiques, les problèmes que nous aborderons de la suite de la thèse ont été peu ou pas étudiés.

En ce qui concerne la logistique de sécurité dont fait l'objet une grande partie de nos travaux, nous étudierons le problème de tournées de véhicules $m$-péripatétiques qui consiste à servir un ensemble de clients au cours de $m$ périodes tout en respectant les contraintes de péripatéticité sur cette période qui interdisent, par sécurité, la réutilisation de séquences de clients identiques au cours des $m$ périodes. Les méthodes de résolution seront évaluées non seulement sur notre problème, mais également sur deux de ses cas particuliers, le VRP et le $m$-PVP.

En ce qui concerne le logistique humanitaire, nous étudierons le problème de tournées de véhicules cumulatives qui consiste à minimiser la somme des latences des populations à secourir. Bien qu'il ait émergé indépendamment, il peut être considéré équivalent au min-avg VRP qui minimise la date moyenne d'arrivée chez les clients. Nous proposerons des outils de résolution plus élaborés et plus performants sur ce problème, que nous testerons également sur un de ses cas particuliers connus, le TRP. Nous étudierons également les propriétés du problème qui le différencient des variantes classiques du VRP.

Dans les chapitres qui suivent, la description plus précise de chacun des deux problèmes abordés, ainsi que leur intérêt théorique et pratique seront présentés avant leurs modèles mathématiques et méthodes de résolution.

# Chapitre 3

# Le m-PTVP : formulations mathématiques et bornes inférieures

## 3.1 Introduction

En France, comme dans la plupart des pays d'Europe, les banques, commerces, sociétés de services et administrations sous-traitent le transport des fonds dont ils ont besoin ou qu'ils ont générés au cours de leur fonctionnement auprès de sociétés spécialisées telles que Loomis®, Brinks®, Ivri® et Mondialpol®. Ces dernières sont en général chargées de deux missions complémentaires : le transfert vers un coffre sécurisé des recettes collectées en fin de journée ou semaine ; ainsi que le réapprovisionnement, en début de journée ou semaine, en devises nécessaires à l'activité économique des clients jusqu'au prochain transfert. Les tournées de collecte sont effectuées par des véhicules banalisés ou blindés qui quittent leur dépôt vide, récupèrent les fonds mis à disposition par les clients qu'ils visitent avant de retourner au dépôt où se trouve un coffre blindé. Les tournées de distribution quant à elles sont effectuées par des véhicules dédiés qui quittent le dépôt avec suffisamment de fonds pour satisfaire les besoins des clients à visiter et retournent vides au dépôt. Par sécurité, le plafond du montant des fonds autorisés par véhicules doit être respecté et les séquences de clients doivent régulièrement varier car il est déconseillé de réutiliser souvent les itinéraires. Les établissements chargés de tels transports ont besoin de logiciels spécialisés pour définir et optimiser les tournées de leurs véhicules et les plannings associés. Leurs activités de collecte et de distribution peuvent être modélisées comme des problèmes de tournées de véhicules $m$-péripatétiques ($m$-PTVP).

Défini sur un graphe complet, le $m$-PTVP consiste à déterminer des tournées de véhicules sans arêtes communes, de coût total minimal sur $m$ périodes, telles que chaque client soit visité exactement une fois par période. La figure 3.1 illustre une solution de 2-PTVP où 3 tournées sont utilisées au cours de la période 1 et 2 tournées sont utilisées au cours de la période 2. Ses applications, outre le transport de fonds, incluent par exemple la définition de tournées de gardiennage dans lesquelles les trajets des gardiens doivent avoir

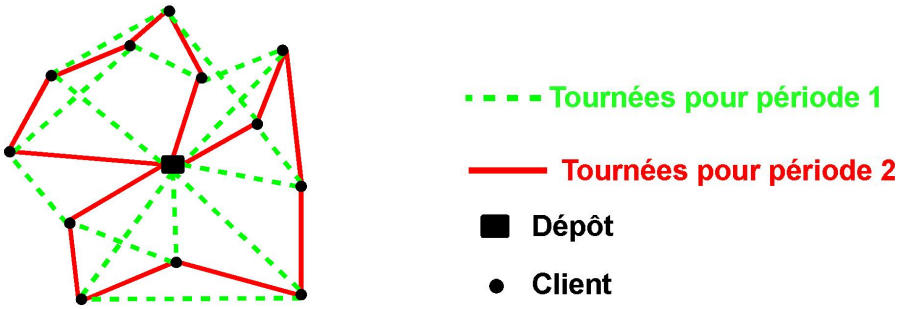



FIGURE 3.1 – Exemple de solution d'un 2-PTVP ($m = 2$)

une apparence irrégulière. Ce problème, nouveau à notre connaissance, peut être considéré comme une généralisation du VRP et du $m$-PVP, deux problèmes NP-difficiles connus en recherche opérationnelle et déjà présentés au chapitre 2. En effet, le VRP est équivalent au 1-PTVP puisqu'il consiste à trouver les meilleures tournées de véhicules permettant de visiter chaque client une et une seule fois, ce qui correspond à $m = 1$. De même, sachant que le problème du voyageur de commerce (TSP) est un cas particulier du VRP sans contrainte de capacité, nous pouvons en déduire, bien que cela ne soit pas trivial, que tout problème $m$-PVP équivaut à un problème $m$-PTVP sans contrainte de capacité (voir Annexe 8).

Ce chapitre est consacré aux formulations mathématiques du $m$-PTVP et à des bornes inférieures simples. L'évaluation numérique est effectuée sur les cas particuliers du VRP et du $m$-PVP en plus du cas général $m$-PTVP avec $m > 1$.

## 3.2 Description, modélisations et complexité

### 3.2.1 Données et contraintes

Le $m$-PTVP est défini sur un ensemble de périodes $\mathbb{M} = \{1 \dots m\}$ et un graphe complet et non-orienté $G = (V, E, C)$ où $V = \{0...n\}$ est l'ensemble des nœuds composé du dépôt ou nœud 0 et de l'ensemble des clients $V' = V \backslash \{0\}$. A chaque arête $e \in E$ est affecté un coût $c_e \in C$ et l'inégalité triangulaire est supposée vérifiée. Chaque client $i \in V'$ a une demande $q_i$ à satisfaire au cours de chaque période $k \in \mathbb{M}$. La flotte de véhicules a priori non limitée et basée au dépôt est composée de véhicules identiques de capacité $Q$. Le but est de définir les tournées de véhicules à réaliser au cours de chaque période $k \in \mathbb{M}$ de manière à minimiser le coût total des arêtes utilisées tout en respectant trois principaux types de contraintes :

1. *Livraison unique par période* : chaque client $i \in V'$ doit être servi par une seule tournée au cours de chaque période $k \in \mathbb{M}$.
2. *Respect de la capacité* $Q$ : la somme des demandes des clients de chacune des tournées réalisables, lesquelles débutent et se terminent au dépôt, ne doit pas excéder la capacité des véhicules $Q$.

3. *Péripatéticité* : chaque arête $e \in E$ peut être utilisée au maximum une fois au cours des $m$ périodes.

Afin de simplifier l'écriture et la lecture des modèles mathématiques du $m$-PTVP, nous définissons $\lambda$ et $r(S)$ qui désignent respectivement le nombre total minimal de tournées par période et le nombre minimal de tournées nécessaire pour satisfaire la demande cumulée d'un sous-ensemble de clients $S \subseteq V'$. Leurs valeurs respectives sont calculées à l'aide des équations (3.2) et (3.1).

$$r(S) = \left\lceil \sum_{i \in S} \frac{q_i}{Q} \right\rceil \tag{3.1}$$

$$\lambda = r(V') \tag{3.2}$$

Une condition nécessaire pour qu'une instance admette une solution réalisable de $m$-PTVP est que : $m \leq \frac{n}{2\lambda}$. En effet, il faut un minimum de $\lambda$ tournées par période pour respecter les contraintes de capacité, ce qui équivaut à $2\lambda$ arêtes connectées au dépôt. Sachant qu'il existe au total $n$ arêtes différentes connectées au dépôt dans le graphe complet, nous pouvons en déduire que le nombre maximum $m_{\max}$ de périodes possibles est donné par l'équation (3.3).

$$m_{\max} = \left\lfloor \frac{n}{2\lambda} \right\rfloor \tag{3.3}$$

### 3.2.2 Modèle basé sur les arêtes

Le $m$-PTVP peut être modélisé sous forme de programme linéaire en 0-1 où la variable de décision $x_e^k$ est égale à 1 si et seulement si l'arête $e \in E$ est utilisée par une tournée au cours de la période $k \in \mathbb{M}$. Dans cette formulation $\delta(S) \subset E$ est l'ensemble des arêtes ayant une et une seule extremité dans $S \subseteq V'$. Par souci de simplicité, $\delta(\{i\})$ avec $i \in V$ sera remplacé par $\delta(i)$.

$$\text{(PA)} \quad \min \sum_{k \in \mathbb{M}} \sum_{e \in E} c_e x_e^k \tag{3.4}$$

s. c.

$$\sum_{e \in \delta(i)} x_e^k = 2, \qquad \forall k \in \mathbb{M}, \forall i \in V' \tag{3.5}$$

$$\sum_{e \in \delta(0)} x_e^k \geq 2\lambda, \qquad \forall k \in \mathbb{M} \tag{3.6}$$

$$\sum_{e \in \delta(S)} x_e^k \geq 2r(S), \qquad \forall S \subseteq V', S \neq \varnothing, \forall k \in \mathbb{M} \tag{3.7}$$

$$\sum_{k \in \mathbb{M}} x_e^k \leq 1, \qquad \forall e \in E \tag{3.8}$$

$$x_e^k \in \{0, 1\}, \qquad \forall e \in E, \forall k \in \mathbb{M} \tag{3.9}$$

La fonction-objectif à minimiser correspond au coût total des arêtes qui sont utilisées au cours des $m$ périodes. Les contraintes (3.5) sont des contraintes de degré, c'est à dire qu'il doit y avoir exactement deux arêtes incidentes à chaque client dans chaque période, car chaque tournée de véhicule a besoin d'une arête pour atteindre un client et d'une autre pour en repartir. Les contraintes (3.6) stipulent que le nombre d'arêtes incidentes au dépôt doit être au moins égal au double du nombre minimum de tournées nécessaires au cours de chacune des $m$ périodes ($= 2\lambda$), car une tournée a besoin de deux arêtes incidentes au dépôt. Les contraintes (3.7) détaillées ci-après font respecter la capacité maximale des tournées et interdisent les sous-tours ou sous-ensembles de clients non connectés au dépôt ; pour cela, elles vérifient que n'importe quel sous-ensemble $S \subseteq V'$ de clients soit traversé par au moins $2r(S) \in \mathbb{N}$ arêtes, c'est à dire ayant une extrémité dans $S$ et une extrémité hors de $S$. Les contraintes péripatétiques (3.8) s'assurent qu'aucune arête n'est utilisée plus d'une fois au cours des $m$ périodes. Enfin, les contraintes d'intégrité (3.9) s'assurent que les variables restent binaires.

Les figures 3.2 et 3.3 illustrent les deux rôles des contraintes (3.7) : faire respecter la capacité maximale des tournées et interdire les sous-tours ou sous-ensembles de clients non connectés au dépôt. En effet, chaque sous-ensemble de clients $S \subseteq V'$ doit être desservi par au moins $r(S)$ tournées, ce qui correspond à $2r(S)$ arêtes incidentes.

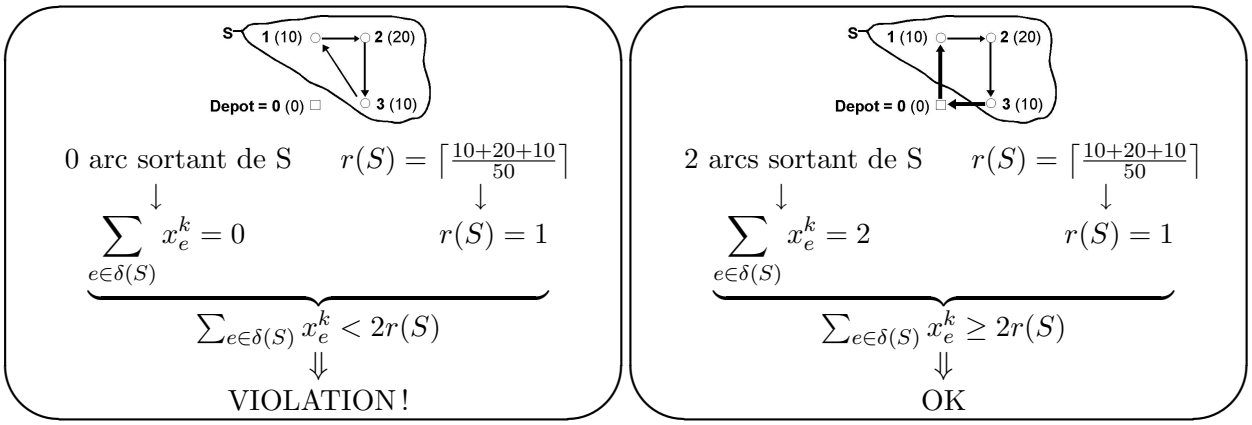



FIGURE 3.2 – Détection de sous-tours par les contraintes (3.7) avec $Q = 50$

Le modèle mathématique (PA) généralise la formulation à 3-indices du $m$-PVP de Duchenne *et al.* [40], ainsi que la formulation du VRP rappelée dans le livre de Toth et Vigo [126]. Il utilise un nombre de variables de l'ordre de $mn^2$. Les contraintes de sous-tours (3.7) sont en nombre exponentiel d'ordre $m2^n$ alors que tous les autres types sont d'ordre $n^2$. Cette formulation est donc hors de portée des solveurs commerciaux pour les instances de taille supérieure à une dizaine de nœuds. Toutefois, pour les instances de taille moyenne, nous pouvons envisager la relaxation des contraintes de sous-tours dans un premier temps, puis l'ajout progressif des contraintes non respectées par les solutions tour à tour obtenues, en espérant que l'ajout d'un nombre limité de contraintes permette à un solveur d'atteindre la solution optimale du $m$-PTVP dans un temps de calcul raisonnable. Ce principe inérant

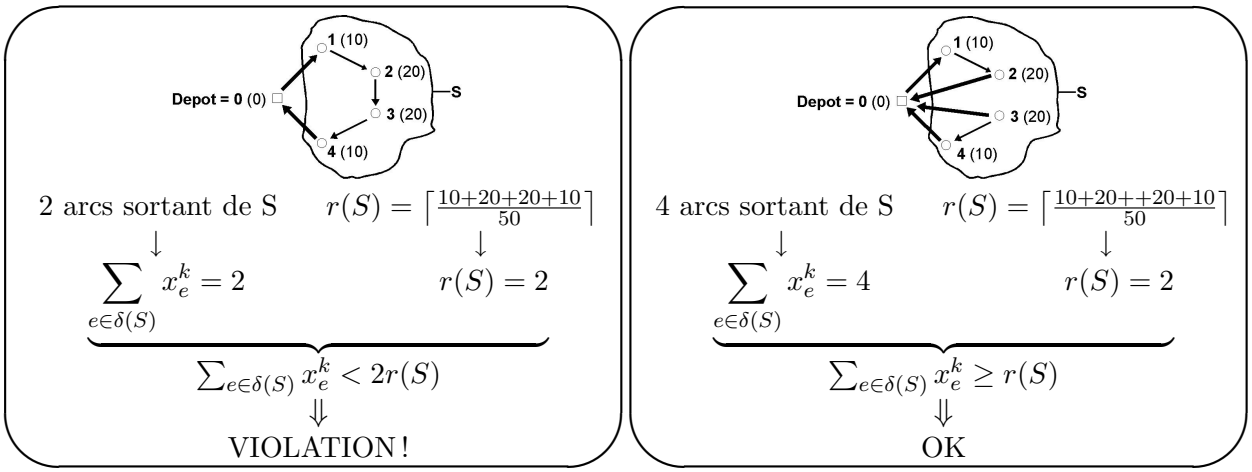



FIGURE 3.3 – Respect de la capacité $Q = 50$ avec les contraintes (3.7)

aux méthodes de coupes permettra de développer l'approche polyédrale décrites au sein du chapitre 6.

### 3.2.3 Modèle de partionnement

Pour formuler le $m$-PTVP en terme de problème de partionnement (PP), nous avons besoin de définir $\mathscr{R}$ l'ensemble des tournées réalisables extractibles du graphe $G$ et $x_r^k$ la variable de décision égale à 1 si et seulement si la tournée $r \in \mathscr{R}$ de coût $c_r$ est utilisée au cours de la période $k \in \mathbb{M}$. Rappelons qu'une tournée de véhicule réalisable débute au dépôt, visite un sous-ensemble de clients une unique fois chacun et retourne au dépôt avec une demande totale inférieure ou égale à la capacité $Q$. Son coût est égal à la somme des coûts des arêtes qui la composent. Dans la formulation (PP), $\mathscr{R}_i$ désigne l'ensemble des tournées qui traversent le nœud $i \in V$ tandis que $\mathscr{R}(e)$ est l'ensemble des tournées qui traversent l'arête $e \in E$.

$$\text{(PP)} \quad \min \sum_{k \in \mathbb{M}} \sum_{r \in \mathscr{R}} c_r x_r^k \tag{3.10}$$

s. c.

$$\sum_{r \in \mathscr{R}_i} x_r^k = 1, \qquad \forall k \in \mathbb{M}, \forall i \in V' \tag{3.11}$$

$$\sum_{r \in \mathscr{R}} x_r^k \geq \lambda, \qquad \forall k \in \mathbb{M} \tag{3.12}$$

$$\sum_{r \in \mathscr{R}(e)} \sum_{k \in \mathbb{M}} x_r^k \leq 1, \qquad \forall e \in E \tag{3.13}$$

$$x_r^k \in \{0,1\}, \qquad \forall r \in \mathscr{R}, \forall k \in \mathbb{M} \tag{3.14}$$

La fonction-objectif minimise la somme des coûts des tournées utilisées. Les contraintes (3.11) imposent que chaque client soit servi par une et une seule tournée par période, tandis que les contraintes (3.12) imposent que les tournées choisies au cours de chaque période $k \in \mathbb{M}$ soient en nombre suffisant. Les contraintes péripatétiques (3.13) empêchent toute arête $e \in E$ d'être utilisée plus d'une fois au cours des $m$ périodes. Enfin, les contraintes d'intégrité (3.14) s'assurent de la binarité de chaque variable.

Le modèle mathématique (PP) généralise la formulation en partitionnement du VRP initialement proposée par Balinski et Quandt [12]. Malgré un nombre de contraintes de l'ordre de $n^2$, ce modèle mathématique est peu adapté aux solveurs commerciaux en raison du nombre exponentiel de ses variables, lié au nombre exponentiel de tournées extractibles du graphe $G$. Il est toutefois envisageable d'y appliquer des méthodes de type génération de colonnes, qui s'initialisent avec un nombre réduit de variables et ajoutent progressivement les variables les plus pertinentes, en espérant atteindre la solution optimale du problème après ajout d'un nombre limité de variables. Ce principe sera en partie utilisé pour la construction de bornes inférieures plus élaborées décrites au chapitre 5.

### 3.2.4 Modèle basé sur le VRP

La formulation du $m$-PTVP en terme de "*solutions-VRP*" est basée sur l'idée selon laquelle l'ensemble des arêtes utilisées au cours de chaque période $k \in \mathbb{M}$ du $m$-PTVP constitue une solution réalisable du VRP appliqué sur le graphe $G$, comme illustré par la figure 3.4. Notons $\mathscr{S}$ l'ensemble des solutions réalisables de VRP extractibles de $G$ et $E(s)$ l'ensemble des arêtes utilisées par $s \in \mathscr{S}$. Chaque *solution-VRP*, $s \in \mathscr{S}$, vérifie donc les contraintes (3.15) à (3.17) :

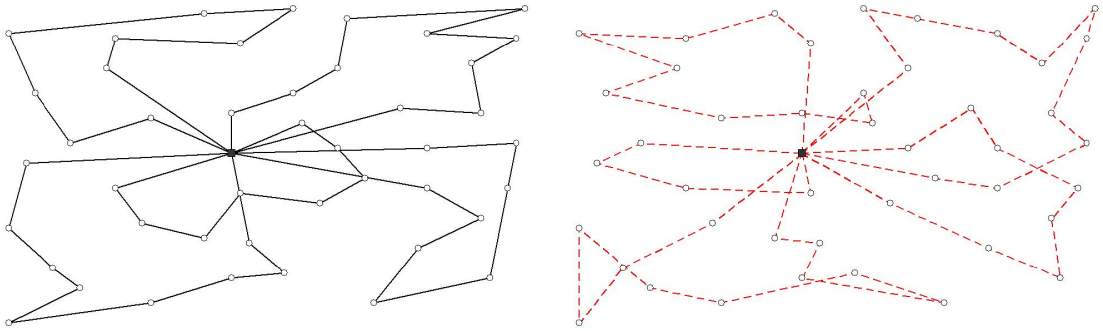
$$\sum_{e \in (\delta(i) \cap E(s))} x_e = 2, \qquad \forall i \in V' \tag{3.15}$$

$$\sum_{e \in (\delta(0) \cap E(s))} x_e \geq 2\lambda, \tag{3.16}$$

$$\sum_{e \in (\delta(S) \cap E(s))} x_e \geq 2r(S), \qquad \forall S \subseteq V', S \neq \varnothing \tag{3.17}$$

Soit $\mathscr{S}_e \subset \mathscr{S}$ l'ensemble des *solutions-VRP* qui utilisent l'arête $e \in E$. La formulation (PS) ci-après modélise donc le $m$-PTVP avec une variable de décision $y_s^k$ égale à 1 si et seulement si $s \in \mathscr{S}$ de coût $c_s = \sum_{e \in E(s)} c_e$ est choisie pour servir les clients au cours de la période $k \in \mathbb{M}$.

$$\text{(PS)} \min \sum_{k \in \mathbb{M}} \sum_{s \in \mathscr{S}} c_s y_s^k \tag{3.18}$$



(a) Tournées pour la période 1 : coût = 572 (b) Tournées pour la période 2 : coût = 576

FIGURE 3.4 – Solution optimale (coût = 1148) du 2-PTVP appliqué à l'instance P-n45-k5

s. c.

$$\sum_{s \in \mathscr{S}} y_s^k = 1, \qquad \forall k \in \mathbb{M} \tag{3.19}$$

$$\sum_{k \in \mathbb{M}} \sum_{s \in \mathscr{S}_e} y_s^k \leq 1, \qquad \forall e \in E \tag{3.20}$$

$$y_s^k \in \{0,1\}, \qquad \forall s \in \mathscr{S}, \forall k \in \mathbb{M} \tag{3.21}$$

La fonction-objectif (3.18) à minimiser est la somme des coûts des *solutions-VRP*, $s \in \mathscr{S}$, sélectionnées pour constituer la solution du $m$-PTVP. Les contraintes (3.19) imposent le choix d'une seule *solution-VRP* par période. Les contraintes péripatétiques (3.20) empêchent qu'une arête soit utilisée plus d'une fois au sein des $m$ *solutions VRP* sélectionnées. Enfin, les contraintes d'intégrité (3.21) s'assurent de la binarité de chaque variable. Le principal intérêt de cette formulation est la possibilité d'extraire des bornes inférieures simples du $m$-PTVP à partir de solutions ou bornes inférieures connues du VRP. Deux des trois bornes inférieures simples présentées dans la section 3.3 sont effectivement déduites de ce modèle.

### 3.2.5 Résultats de complexité

**Théorème 1** *Le m-PTVP est NP-difficile.*

***Preuve*** *S'il existe un algorithme polynomial pour résoudre le m-PTVP , cet algorithme pourrait résoudre également le 1-PTVP = VRP. Ceci est impossible puisque le VRP est NP-difficile.*

**Théorème 2** *Soit $P$ une solution réalisable de m-PTVP et soit $\mathscr{P}$ le graphe induit par $P$, composé de l'ensemble des nœuds du graphe original $G$ et de l'ensemble des arêtes utilisées*

*par $\mathscr{P}$. Aucun algorithme polynomial ne peut extraire de $\mathscr{P}$ une solution réalisable de $m$-PTVP.*

***Preuve*** *Le théorème de De Kort [33] stipule qu'un graphe régulier est un graphe dont tous les nœuds sont de degré 4, et extraire 2 cycles Hamiltoniens sans arêtes communes d'un tel graphe est NP-complet. Ce théorème peut être généralisé pour obtenir le lemme 1 puis le théorème 2 car s'il existait un algorithme polynômial pour extraire de $\mathscr{P}$ une solution réalisable de $m$-PTVP, le théorème de De Kort [33] serait faux.*

**Lemme 1** *Extraire $m$ cycles Hamiltoniens d'un graphe $2m$-régulier est NP-complet.*

En d'autres termes, une solution de $m$-PTVP répond à deux questions : (1) quel sous-ensemble d'arêtes sélectionner ? (2) comment répartir les arêtes sélectionnées entre les $m$ périodes pour obtenir une solution réalisable ? Même en supposant que la réponse à la 1ère question soit connue, répondre à la 2ème question reste NP-complet (à l'exception du cas trivial $m = 1$).

## 3.3 Bornes inférieures simples

Les trois bornes inférieures simples LB1, LB2 et LB3 présentées ci-après ont été obtenues en relaxant simplement les formulations mathématiques (PA) et (PS). Leurs performances sont évaluées et comparées dans la section 3.4.

### 3.3.1 LB1 : $m$ solutions VRP

La première borne inférieure LB1 résulte de la relaxation des contraintes péripatétiques (3.20) du modèle (PS), ce qui autorise la réutilisation de *solutions-VRP* identiques au cours des $m$ périodes. Soit $c_{vrp}$ le coût d'une solution optimale du VRP déduite du graphe $G$, l'équation (3.22) permet de calculer LB1, coût d'une solution optimale du problème relaxé et donc borne inférieure du $m$-PTVP.

$$LB1 = m \times c_{vrp} \tag{3.22}$$

### 3.3.2 LB2 : $m$-1-arbres recouvrants disjoints

Notre deuxième borne inférieure résulte de l'approximation de l'ensemble $\mathscr{S}$ des *solutions-VRP* du modèle mathématique (PS) par l'ensemble $\mathscr{T}$ des 1-arbres recouvrants et l'application de la relaxation lagrangienne sur les contraintes de degré des clients. Elle généralise donc celle définie par De Kort pour le 2-PVP [35].

Un 1-arbre (voir par exemple l'article de Held et Karp [61]) est un ensemble d'arêtes résultant de l'union d'un arbre recouvrant l'ensemble des clients $V'$ avec deux arêtes incidentes au dépôt. L'approximation de $\mathscr{S}$ par $\mathscr{T}$ se justifie, car une solution du voyageur

de commerce (ou TSP) est un 1-arbre particulier où tous les nœuds sont de degré 2. Par conséquent, un 1-arbre recouvrant à coût minimal donne une borne inférieure au TSP, et donc au VRP. Par conséquent, notre borne inférieure est basée sur le calcul du coût de $m$ 1-arbres recouvrants sans arêtes communes et de coût total minimal ($m$-1-ARCM).

Nous calculons les $m$-1-ARCM en appliquant l'algorithme de Roskind et Tarjan [118] sur $V'$ pour trouver $m$ arbres recouvrants disjoints sur $V'$, puis en ajoutant à la solution obtenue les $2m$ arêtes les moins chères de $\delta(0)$. L'algorithme [118] calcule $m$ arbres recouvrants disjoints à coût total minimal ($m$-ARCM) en généralisant l'algorithme de Kruskal [73] qui calcule l'arbre recouvrant de coût minimal. Ce dernier fonctionne comme suit : partant de la liste des arêtes triées par coûts croissants et d'une forêt réduite à $n$ nœuds isolés, il considère les arêtes une par une et ajoute l'arête courante à la forêt si aucun cycle n'est créé. Le principe de l'algorithme de Roskind et Tarjan pour le $m$-ARCM est assez voisin, mais fait croître $m$ forêts en parallèle. Lors de l'insertion d'une arête, en cas de création d'un cycle, une procédure dite d'*augmentation* illustrée par la figure 3.5 tente de déplacer les arêtes qui le constituent vers d'autres forêts, pour faire disparaître le cycle. En cas d'échec de cette procédure, l'arête initiale est rejetée. Cet algorithme polynomial peut être implémenté en $O(m^2n^2)$.

Sachant qu'un 1-arbre recouvrant de coût minimal ne prend en compte ni des contraintes de degré sur les clients, ni des contraintes de capacité sur les tournées, nous constatons donc que les $m$-1-ARCM nous fournissent une solution qui ne respecte pas forcément les contraintes (3.15), (3.16) et (3.17). Par conséquent, le modèle mathématique (PT) produit une borne inférieure du $m$-PTVP car il résulte de l'ajout des contraintes de degré des clients (3.26) à la formulation mathématique du problème des $m$-1-ARCM. Soit $d_{ti}$ le degré du client $i$ au sein du 1-arbre $t$, soit $E(t)$ l'ensemble des arêtes qui composent $t$ et soit $\mathscr{T}_e$ l'ensemble des 1-arbres qui utilisent l'arête $e$. Le modèle mathématique (PT) utilise une variable de décision binaire $y_t^k$ égale à 1 si et seulement si le 1-arbre $t$ de coût $c_t = \sum_{e\in E(t)} c_e$ est choisi pour la période $k$.

$$\text{(PT)} \quad \min \sum_{k\in\mathbb{M}} \sum_{t\in\mathscr{T}} c_t y_t^k \tag{3.23}$$

s. t.

$$\sum_{t\in\mathscr{T}} y_t^k = 1, \qquad \forall k \in \mathbb{M} \tag{3.24}$$

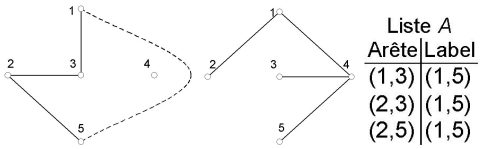
$$\sum_{t\in\mathscr{T}_e} \sum_{k\in\mathbb{M}} y_t^k \leq 1, \qquad \forall e \in E \tag{3.25}$$

$$\sum_{t\in\mathscr{T}} \sum_{k\in\mathbb{M}} y_t^k d_{ti} = 2m, \qquad \forall i \in V' \tag{3.26}$$

$$y_t^k \in \{0,1\}, \qquad \forall t \in \mathscr{T}, \forall k \in \mathbb{M} \tag{3.27}$$

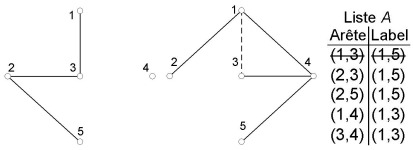
Les contraintes de degré (3.26) peuvent être dualisées en définissant le vecteur de pénalité $\pi = \{\pi_i : i \in V'\}$ avec $\pi_i$ une pénalité affectée au client $i$. La fonction duale résultante $L(y,\pi)$ peut être réécrite sous la forme (3.28) où $\hat{c}_t = \sum_{e\in E(t)} \hat{c}_e$ avec $\hat{c}_e = c_e + \pi_i + \pi_j$, $\forall e = (i,j) \in E$.

### 0 - Initialisation



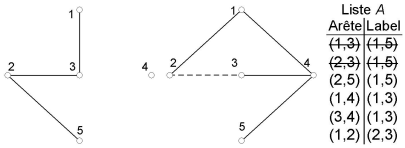

L'insertion de l'arête $(1,5)$ en pointillé dans la forêt 1 à gauche crée un cycle avec les arêtes $(1,3)$, $(2,3)$ et $(2,5)$, ce qui déclenche la procédure d'*augmentation* pour vérifier si des arêtes peuvent être déplacées pour permettre cette insertion. Les arêtes candidates au déplacement sont celles du cycle créé et sont stockées dans une liste $A$. On initialise $A$ avec les arêtes $(1,3)$, $(2,3)$ et $(2,5)$ avec le label $(1,5)$ indiquant l'arête qui a créé le cycle. A va être balayée du haut vers le bas.

### 1 - Test de déplacement de l'arête $(1,3)$



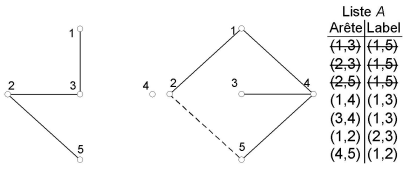

L'arête $(1,3)$ est retirée de la liste $A$. Le déplacement de cette arête vers la forêt qui suit sa forêt actuelle 1, c'est à dire la forêt 2 à gauche, crée un cycle avec les arêtes $(1,4)$ et $(3,4)$. On ajoute ces dernières en fin de liste $A$ avec le label $(1,3)$.

### 2 - Test de déplacement de l'arête $(2,3)$



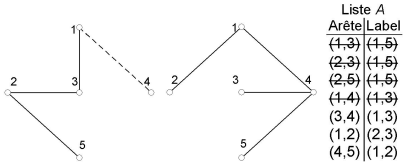

L'arête $(2,3)$ est retirée de la liste $A$. Le déplacement de cette arête vers la forêt qui suit sa forêt actuelle 1, c'est à dire la forêt 2, crée un cycle avec les arêtes $(1,2)$, $(1,4)$ et $(3,4)$. $(1,4)$ et $(3,4)$ font déjà partie de $A$. On ajoute donc l'arête $(1,2)$ en fin de liste $A$ avec le label $(2,3)$.

### 3 - Test de déplacement de l'arête $(2,5)$



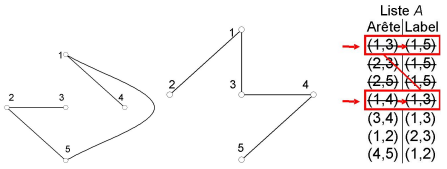

L'arête $(2,5)$ est retirée de la liste $A$. Le déplacement de cette arête vers la forêt qui suit sa forêt actuelle 1, c'est à dire la forêt 2, crée un cycle avec les arêtes $(1,2)$, $(1,4)$ et $(4,5)$. Les deux premières font déjà partie de $A$, on ajoute donc l'arête $(4,5)$ en fin de liste $A$ avec le label $(2,5)$.

### 4 - Test de déplacement de l'arête $(1,4)$




L'arête $(1,4)$ est retirée de la liste $A$. Le déplacement de cette arête vers la forêt qui suit sa forêt actuelle 2, c'est à dire la forêt 1, ne crée pas de cycle. Une chaîne d'*augmentation* a donc été trouvée et la procédure peut s'arrêter. Notons que si $A$ se vidait sans qu'une chaîne d'*augmentation* ne soit trouvée, alors l'*augmentation* serait impossible et l'arête $(1,5)$ devrait être rejetée. Ce n'est pas le cas dans cet exemple.

### 5 - Récupération de la chaîne d'augmentation




La chaîne d'*augmentation* est reconstituée en remontant les labels à partir de la dernière arête déplacée avec succès jusqu'à l'arête que l'on souhaitait insérer au départ. Dans cet exemple, on retrouve donc $(1,4)$, puis $(1,3)$ et enfin $(1,5)$. La chaîne d'*augmentation* est donc $(1,5)-(1,3)-(1,4)$ à interpréter comme suit : insérer l'arête $(1,5)$ dans la forêt 1, déplacer l'arête $(1,3)$ vers la forêt $2(=1 \mod m+1)$ et enfin déplacer l'arête $(1,4)$ vers la forêt $1(=2 \mod m+1)$.

FIGURE 3.5 – Procédure d'*augmentation* pour insérer l'arête $(1,5)$ sans cycles avec $m=2$

$$L(y,\pi) = \min \sum_{k\in\mathbb{M}} \sum_{t\in\mathscr{T}} c_t y_t^k + \sum_{i=1}^{n} \pi_i \Big(\sum_{k\in\mathbb{M}} \sum_{t\in\mathscr{T}} y_t^k d_{t,i} - 2m\Big) = -2m \sum_{i=1}^{n} \pi_i + \min \sum_{k\in\mathbb{M}} \sum_{t\in\mathscr{T}} \hat{c}_t y_t^k \quad (3.28)$$

Pour des valeurs fixées de $\pi_i$, il est possible de minimiser $L(x, \pi)$ simplement en calculant les $m$ 1-arbres recouvrants disjoints à coût total minimal avec un coût modifié $\hat{c}_e$ pour chaque arête $e = (i, j)$. Le vecteur $\pi$ qui maximise la borne peut être déterminé à l'aide d'une méthode de sous-gradients suivant l'équation (3.29) où chaque terme de type $x^{(t)}$ représente la valeur du terme $x$ lors de l'itération $t$, $UB$ est la meilleure borne supérieure fournie et $\epsilon$ initialement fixé à 1,5 est divisé par 2 toutes les 3 itérations sans amélioration. L'optimisation de $\pi$ par la méthode de sous-gradients s'arrête dès qu'il y a convergence, c'est à dire à l'itération $h'$ qui vérifie $L(y, \pi^{(h'-1)}) = UB$ ou $\| \Delta_i^{(h'-1)} \| = 0$.

$$\pi_i^{(h)} = \pi_i^{(h-1)} + \epsilon.step.\Delta_i^{(h-1)} \tag{3.29}$$

$$\text{où} \qquad \Delta_i^{(h-1)} = \sum_{k=1}^{m} \sum_{t \in \mathscr{T}} y_t^{k^{(h-1)}} d_{t,i} - 2m \qquad \text{et} \qquad step = \frac{UB - L(y, \pi^{(h-1)})}{\| \Delta_i^{(h-1)} \|}$$

### 3.3.3 LB3 : $b$-couplage parfait relaxé

Notre troisième borne inférieure LB3 est obtenue en résolvant un problème de $b$-couplage parfait relaxé, obtenu en relaxant des contraintes du $m$-PTVP. Décrit par les équations (3.44)-(3.47), le $b$-couplage parfait fait partie de la classe des problèmes linéaires résolus polynomialement par Edmonds et Johnson [45], et utilisés par Miller et Wright [88] pour le VRP. Il consiste à identifier un sous-ensemble d'arêtes $E'$ de coût minimal tel que chaque nœud $i$ soit incident à $b_i$ arêtes de $E'$, sans prendre en considération les autres types de contraintes. Notre $b$-couplage parfait relaxé, décrit par les équations (3.40)-(3.43), peut être déduit de la formulation mathématique du $m$-PTVP basée sur les arêtes (PA) en suivant les trois étapes de la procédure suivante :

1. Relaxer les contraintes de capacité (3.7) du modèle mathématique (PA) puis sommer les contraintes (3.5) et (3.6) sur $k \in \mathbb{M}$ pour obtenir le problème (3.30)-(3.34) :

$$\min \sum_{k \in \mathbb{M}} \sum_{e \in E} c_e x_e^k \tag{3.30}$$

s. c.

$$\sum_{k \in \mathbb{M}} \sum_{e \in \delta(i)} x_e^k = 2m, \qquad \forall i \in V' \tag{3.31}$$

$$\sum_{k \in \mathbb{M}} \sum_{e \in \delta(0)} x_e^k \geq 2m\lambda \tag{3.32}$$

$$\sum_{k \in \mathbb{M}} x_e^k \leq 1, \qquad \forall e \in E \tag{3.33}$$

$$x_e^k = \{0, 1\}, \qquad \forall e \in E, \quad \forall k \in \mathbb{M} \tag{3.34}$$

2. Effectuer le changement de variable $X_e = \sum_{k \in \mathbb{M}} x_e^k$ avec $X_e \in \{0,1\}$, $\forall e \in E$ pour obtenir le sous-problème (3.35)-(3.39) :

$$\min \sum_{e \in E} c_e X_e \tag{3.35}$$

s. c.

$$\sum_{e \in \delta(i)} X_e = 2m, \qquad \forall i \in V' \tag{3.36}$$

$$\sum_{e \in \delta(0)} X_e \geq 2m\lambda \tag{3.37}$$

$$X_e \leq 1, \qquad \forall e \in E \tag{3.38}$$

$$X_e \in \{0,1\}, \qquad \forall e \in E \tag{3.39}$$

3. Enfin, supprimer la contrainte (3.38) devenue redondante avec (3.39) pour obtenir un problème de $b$-couplage parfait relaxé, lequel fournit la borne inférieure LB3. Seule la contrainte de degré du dépôt, représentée par une inégalité (3.42) au lieu d'une égalité (3.46), distingue le $b$-couplage parfait classique de notre relaxation.

<u>$b$-couplage parfait relaxé</u>

$$\min \sum_{e \in E} c_e X_e \tag{3.40}$$

s. c.

$$\sum_{e \in \delta(i)} X_e = 2m, \quad \forall i \in \{1...n\} \tag{3.41}$$

$$\sum_{e \in \delta(0)} X_e \geq 2m\lambda \tag{3.42}$$

$$X_e \in \{0,1\}, \qquad \forall e \in E \tag{3.43}$$

<u>$b$-couplage parfait</u>

$$\min \sum_{e \in E} c_e X_e \tag{3.44}$$

s. c.

$$\sum_{e \in \delta(i)} X_e = 2m, \quad \forall i \in \{1...n\} \tag{3.45}$$

$$\sum_{e \in \delta(0)} X_e = 2m\lambda \tag{3.46}$$

$$X_e \in \{0,1\}, \qquad \forall e \in E \tag{3.47}$$

Le sous-problème obtenu étant polynomial, nous pouvons espérer obtenir des bornes inférieures du $m$-PTVP en un temps de calcul limité. Contrairement aux $m$-1-ARCM, ce $b$-couplage parfait relaxé prend partiellement en compte les contraintes de degré (3.5) et (3.6) de la formulation PA : la solution obtenue respecte le nombre total d'arêtes incidentes à chaque noeud, mais cette contrainte n'est pas forcément respectée au cours de chacune des $m$ périodes. En revanche, ce $b$-couplage parfait relaxé ne conserve aucune propriété de connectivité ou de capacité (3.7). De plus, à cause du théorème 2 (page 59), même si une solution de ce problème correspondait à une solution réalisable de $m$-PTVP, extraire cette dernière à partir de l'ensemble des arêtes fournies par le $b$-couplage serait NP-complet.

## 3.4 Résultats numériques

### 3.4.1 Instances et implémentation

En l'absence d'instances spécifiques au $m$-PTVP dans la littérature, nous avons testé nos bornes inférieures sur les cas particuliers du VRP et du $m$-PVP avant de les appliquer au $m$-PTVP avec $m > 1$. Quatre types d'instances VRP sont utilisées : A, B, P et VRPNC, disponibles sur le site internet [115]. Les trois premiers sont proposés par Augerat [8] et contiennent respectivement 27, 23 et 23 instances de 19 à 101 nœuds. Le type VRPNC proposé par Christofides *et al.* [23] contient 7 instances de 50 à 200 nœuds sans limitations supplémentaires sur la longueur des tournées. Nous avons également testé nos bornes inférieures sur 20 des plus grandes instances de la TSPLIB [115], allant de 70 à 280 nœuds, parmi celles déjà utilisées par Duchenne *et al.* [41] pour le $m$-PVP.

La première borne inférieure LB1 se déduit facilement du coût des solutions optimales du VRP à l'aide de l'équation (3.22). Notre implémentation de l'algorithme des $m$-1-ARCM de LB2 est faite en Delphi. Le $b$-couplage parfait relaxé de LB3 est résolu à l'aide du solveur Open Source Glpk pour la plupart des cas, et Cplex pour les cas les plus difficiles. Les calculs de bornes inférieures durent en moyenne moins d'une quinzaine de seconde sur un PC Pentium de 3,39 GHz et 1 Go de mémoire. Dans les pages qui suivent, les résultats obtenus sont présentés sous forme aggrégée par les tableaux 3.1, 3.2 et 3.3, afin d'en simplifier la lecture et l'analyse. Soit UB la meilleure borne supérieure trouvée par les métaheuristiques du chapitre 4, les colonnes $\theta_{\%}$ et $\sigma$ de chaque borne inférieure LBi, $\forall i \in \{1, 2, 3\}$ contiennent respectivement la moyenne et l'écart-type des ratios $\frac{100LBi}{UB}$ obtenus par type d'instance. Le nombre d'instances utilisées par type est précisé dans la première colonne de chaque tableau. Le détail des résultats est disponible en annexe.

### 3.4.2 Cas particuliers du VRP et $m$-PVP

Le tableau 3.1 illustre la performance des bornes inférieures simples LB2 et LB3 sur le cas particulier du VRP et les compare à celle basée sur les *q-tournées* de Baldacci *et al.* [10] notée ici $LB_{vrp}$. Cette dernière est à ce jour la meilleure borne inférieure connue pour le VRP et elle est calculable en quelques secondes. Toutefois, elle n'est pas disponible pour les instances VRPNC. Les bornes supérieures UB utilisées pour les calculs des ratios $\frac{100LBi}{UB}, \forall i \in \{1, 2, 3\}$ ont été prouvées optimales pour les instances de type A, B et P soit par Fukasawa *et al.* [52], soit par Baldacci *et al.* [10].

Les résultats montrent que LB2 et LB3 sont peu serrées sur le cas particulier du VRP. Selon les types d'instances, LB2 est en moyenne entre 54% et 66% de l'optimum, tandis que LB3 est entre 74% et 79%. $LB_{vrp}$ est bien meilleure puisqu'elle atteint 96% à 99%. Cela s'explique par le fait que les $m$-1-ARCM d'où proviennent LB2 ne tiennent pas compte des contraintes de capacité et très peu des contraintes de degré. Le $b$-couplage parfait relaxé qui produit LB3 donne de meilleurs résultats parce qu'il prend partiellement en compte les contraintes de degré, en s'assurant qu'il y a bien 2 arêtes incidentes à chaque client et

| | $LB_{vrp}$ | | | LB2 | | | LB3 | | |
|---|---|---|---|---|---|---|---|---|---|
| Type | $\theta_{\%}$ | $\sigma$ | min − max | $\theta_{\%}$ | $\sigma$ | min − max | $\theta_{\%}$ | $\sigma$ | min − max |
| A (27) | 96,48 | 1,24 | 92,50 - 98,00 | 53,77 | 9,45 | 38,61 - 76,69 | 74,14 | 6,13 | 62,26 - 87,67 |
| B (23) | 99,04 | 0,70 | 94,40 - 100,00 | 40,96 | 7,75 | 27,97 - 60,72 | 62,25 | 8,89 | 47,24 - 75,89 |
| P (23) | 98,13 | 0,97 | 95,6 - 99,80 | 67,61 | 16,13 | 34,40 - 92,07 | 88,72 | 4,31 | 83,07 - 96,18 |
| VRPNC (7) | | | | 66,01 | 10,52 | 50,51 - 80,79 | 78,61 | 16,07 | 45,34 - 92,31 |

Note : les instances VRPNC n'ont pas été traitées dans l'article de Baldacci *et al.* [10]

TABLE 3.1 – Bornes inférieures simples appliquées au VRP = 1-PTVP

(a) 2-PVP

| | LB1 | | | LB2 | | | LB3 | | |
|---|---|---|---|---|---|---|---|---|---|
| | $\theta_{\%}$ | $\sigma$ | min − max | $\theta_{\%}$ | $\sigma$ | min − max | $\theta_{\%}$ | $\sigma$ | min − max |
| 19 instances* | 83,67 | 3,16 | 79,09 - 90,06 | 89,25 | 2,78 | 82,50 - 100,00 | 95,71 | 8,28 | 69,88 - 100,00 |

* st70, eil-76 -101, pr-76 -107 -136 -226 -264, rat99, Kro-B100 -C100 -D100 -E100 -A150 -A200, lin105, ch130, d198, a280

(b) 3-PVP

| | LB1 | | | LB2 | | | LB3 | | |
|---|---|---|---|---|---|---|---|---|---|
| | $\theta_{\%}$ | $\sigma$ | min − max | $\theta_{\%}$ | $\sigma$ | min − max | $\theta_{\%}$ | $\sigma$ | min − max |
| 10 instances* | 71,39 | 4,59 | 66,60 - 78,51 | 88,85 | 2,52 | 83,39 - 90,77 | 97,41 | 6,84 | 78,12 - 100,00 |

* eil-76 -101, pr76, Kro-B100 -C100 -D100 -E100, pr-107 -124 -136

(c) 4-PVP

| | LB1 | | | LB2 | | | LB3 | | |
|---|---|---|---|---|---|---|---|---|---|
| | $\theta_{\%}$ | $\sigma$ | min − max | $\theta_{\%}$ | $\sigma$ | min − max | $\theta_{\%}$ | $\sigma$ | min − max |
| 6 instances* | 61,08 | 3,51 | 57,25 - 67,30 | 88,49 | 2,49 | 84,29 - 90,92 | 96,76 | 7,11 | 82,32 - 100,00 |

* st70, pr-76 -107 -124, Kro-B100 -D100

TABLE 3.2 – Bornes inférieures simples appliquées au $m$-PVP ($\lambda = 1$)

suffisamment d'arêtes incidentes au dépôt. Nos deux bornes inférieures sont globalement plus efficaces sur les instances de type P et VRPNC que sur les instances de type A et B. Enfin, leur performance varie significativement d'une instance à une autre, comme le montrent les valeurs élevées des écart-types.

Le tableau 3.2 présente les résultats obtenus par les bornes inférieures LB1, LB2 et LB3 sur le cas particulier du $m$-PVP avec les instances de la TSPLIB. Dans ce cas, la valeur de LB1 est en fait égale à $m \times c_{tsp}$ où $c_{tsp}$ est le coût de la solution optimale du TSP, ce qui la rend équivalente à la borne inférieure proposée dans l'article de De Kort [33]. Les bornes supérieures UB utilisées pour les calculs des ratios $\theta_{\%}$ correspondent aux valeurs des solutions optimales trouvées par Duchenne *et al.* [41], car nous n'avons gardé que les couples "$m$ - instance" qui y sont résolus optimalement. LB2 et LB3 sont assez serrées sur le $m$-PVP et surclassent LB1 puisqu'elles atteignent en moyenne environ 88% et 96% de l'optimum contre environ 70% pour LB1. Ces résultats sont cohérents avec les résultats obtenus sur le VRP puisque les contraintes de capacité n'existent pas pour un $m$-PVP.

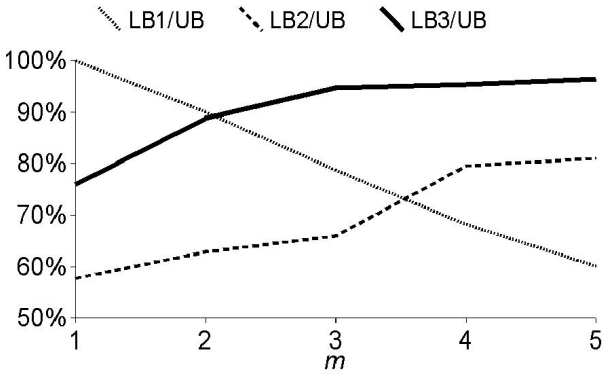



FIGURE 3.6 – Evolution des bornes inférieures en fonction de $m$

Toutefois, la restriction des couples "$m$ - instance" aux cas résolus par Duchenne *et al.* [41] et surtout la forme aggrégée du tableau 3.2 masquent légèrement l'impact du nombre de périodes $m$ sur l'efficacité des bornes inférieures : LB2 et LB3 produisent en général de meilleurs résultats lorsque $m$ augmente. La borne inférieure LB3 est particulièrement efficace, car elle atteint l'optimum pour 14 couples "$m$ - instance" sur les 35 testés.

### 3.4.3 $m$-PTVP

Le tableau 3.3 présente les résultats obtenus par les bornes inférieures LB1, LB2 et LB3 sur le $m$-PTVP, en utilisant les instances de type A, B, P et VRPNC avec $m > 1$. Pour chaque valeur de $m$, seules sont retenues les instances qui respectent la condition de faisabilité exprimée par l'équation (3.3), c'est pourquoi le nombre d'instances disponibles diminue lorsque $m$ augmente. Les ratios $\theta_{\%}$ ont été calculés par rapport aux meilleures bornes supérieures connues $UB$, dont l'optimalité n'a pas encore été démontré en ce qui concerne les instances les plus grandes (voir chapitre 6 sur l'approche polyédrale). Les résultats montrent combien nos bornes inférieures sont efficaces sur le $m$-PTVP, car globalement, le ratio $\frac{\max(LB1,LB2,LB3)}{UB}$ est environ égal à 94%. Comme l'illustre la figure 3.6, leur performance évolue lorsque le nombre de périodes $m$ augmente. En effet, LB1 est la plus efficace tant que $m$ reste faible, mais cette borne se dégrade très vite lorsque $m$ croît, contrairement à LB3. Cette dernière voit sa performance s'améliorer avec $m$ et devient rapidement la meilleure des trois bornes inférieures. Elle devrait être encore meilleure une fois que des contraintes de capacité ou connectivité auront été prises en compte. Enfin, LB2 surclasse LB1 dès que $m \geq 4$, mais reste inférieure à LB3.

Il est important de signaler que le $m$-PTVP ne devient pas plus facile lorsque $m$ augmente, malgré la bonne performance de LB3. En effet, même en supposant que LB3 corresponde à une solution optimale de $m$-PTVP, extraire cette solution en partitionnant les arêtes sélectionnées par LB3 en $m$ périodes reste NP-complet comme le montre le théorème 2 (page 59). Rappelons que ce théorème généralise celui de De Kort [33] selon lequel trouver deux cycles hamiltoniens disjoints, dans un graphe dont tous les nœuds sont de degré 4,

(a) 2-PTVP

| | LB1 | | | LB2 | | | LB3 | | |
|---|---|---|---|---|---|---|---|---|---|
| Type | $\theta_{\%}$ | $\sigma$ | $\min - \max$ | $\theta_{\%}$ | $\sigma$ | $\min - \max$ | $\theta_{\%}$ | $\sigma$ | $\min - \max$ |
| A (26) | 90,27 | 2,54 | 84,52 - 93,57 | 60,90 | 8,52 | 44,47 - 77,39 | 91,89 | 3,76 | 83,03 - 98,01 |
| B (23) | 93,85 | 1,98 | 88,57 - 96,90 | 45,88 | 7,88 | 30,20 - 63,46 | 79,22 | 7,92 | 57,11 - 93,53 |
| P (19) | 87,38 | 2,33 | 82,58 - 90,58 | 75,53 | 11,21 | 58,52 - 91,90 | 97,18 | 1,68 | 92,49 - 98,83 |
| VRPNC (7) | 88,60 | 1,54 | 86,37 - 90,49 | 69,01 | 9,92 | 53,47 - 82,47 | 86,82 | 15,51 | 53,79 - 97,29 |

(b) 3-PTVP

| | LB1 | | | LB2 | | | LB3 | | |
|---|---|---|---|---|---|---|---|---|---|
| Type | $\theta_{\%}$ | $\sigma$ | $\min - \max$ | $\theta_{\%}$ | $\sigma$ | $\min - \max$ | $\theta_{\%}$ | $\sigma$ | $\min - \max$ |
| A (25) | 77,56 | 3,92 | 69,83 - 84,03 | 61,97 | 7,95 | 46,11 - 76,50 | 97,24 | 1,84 | 91,93 - 99,53 |
| B (22) | 83,75 | 3,41 | 77,33 - 89,06 | 50,53 | 8,37 | 35,92 - 66,57 | 92,67 | 4,44 | 82,64 - 99,81 |
| P (14) | 74,70 | 2,48 | 70,49 - 78,54 | 80,82 | 8,80 | 65,72 - 91,34 | 98,61 | 1,05 | 95,79 - 100,00 |
| VRPNC (7) | 78,93 | 3,39 | 73,97 - 84,35 | 70,37 | 8,41 | 57,54 - 82,00 | 90,27 | 13,23 | 61,28 - 98,20 |

(c) 4-PTVP

| | LB1 | | | LB2 | | | LB3 | | |
|---|---|---|---|---|---|---|---|---|---|
| Type | $\theta_{\%}$ | $\sigma$ | $\min - \max$ | $\theta_{\%}$ | $\sigma$ | $\min - \max$ | $\theta_{\%}$ | $\sigma$ | $\min - \max$ |
| P (8) | 65,09 | 2,25 | 61,79 - 67,71 | 87,49 | 2,97 | 81,55 - 91,58 | 98,84 | 0,74 | 97,60 - 99,71 |
| VRPNC (6) | 71,33 | 4,63 | 65,27 - 79,01 | 71,51 | 8,38 | 60,09 - 81,87 | 92,09 | 10,13 | 71,96 - 98,18 |

(d) $5, 6, 7$-PTVP

| | LB1 | | | LB2 | | | LB3 | | |
|---|---|---|---|---|---|---|---|---|---|
| Type | $\theta_{\%}$ | $\sigma$ | $\min - \max$ | $\theta_{\%}$ | $\sigma$ | $\min - \max$ | $\theta_{\%}$ | $\sigma$ | $\min - \max$ |
| P (11) | 56,78 | 3,41 | 51,92 - 61,65 | 89,11 | 1,91 | 86,36 - 91,91 | 98,82 | 0,59 | 97,88 - 99,68 |
| VRPNC (10) | 63,27 | 4,56 | 58,34 - 73,65 | 71,62 | 8,32 | 60,28 - 82,22 | 93,51 | 7,37 | 79,05 - 98,85 |

TABLE 3.3 – Bornes inférieures simples appliquées au $m$-PTVP : $m \in \{2, 3, 4, 5, 6, 7\}$

est un problème NP-complet.

## 3.5 Conclusion

Ce chapitre a introduit le problème de tournées de véhicules $m$-péripatétiques ($m$-PTVP), qui consiste à extraire d'un graphe complet des tournées de véhicules de coût total minimal sans arêtes communes sur $m$ périodes telles que chaque client soit visité exactement une fois par période. Il s'agit d'un problème nouveau dont les applications comprennent par exemple le transport de fonds ou encore la définition de tournées de gardiennage. Il peut être considéré comme une généralisation de deux problèmes NP-difficiles connus que sont le VRP, correspondant au cas mono-période 1-PTVP et le $m$-PVP, correspondant au cas mono-véhicule.

Après avoir décrit les données et contraintes spécifiques au problème, nous avons présenté trois modèles mathématiques, basés respectivement sur les arêtes, les tournées réalisables et les solutions de VRP extractibles du graphe. Trois bornes inférieures simples ont été déduites de ces modèles et pour chacune d'elles plusieurs pistes d'amélioration ont été envisagées, telles que l'ajout de contraintes de connectivité sur le $b$-couplage parfait relaxé. Les tests numériques ont été effectués non seulement sur le $m$-PTVP en général, mais aussi sur les cas particuliers du VRP et du $m$-PVP. Il ressort principalement que notre $b$-couplage parfait relaxé produit la meilleure des trois bornes inférieures simples et que son efficacité croît avec $m$.

Ce travail a fait l'objet de présentations lors des conférences internationales *EURO XXII* [97] et *MOSIM* [99], ainsi que d'un article en cours de révision pour la revue *4OR* [106]. Les pistes d'amélioration envisagées dans ce chapitre ont permis de développer la méthode exacte du chapitre 6 basée sur l'approche polyédrale. Mais avant d'y arriver, le chapitre 4 va présenter les métaheuristiques conçues pour le $m$-PTVP et le chapitre 5 sera consacré à une borne inférieure plus élaborée, de type génération de colonnes.

# Chapitre 4

# Heuristique et métaheuristiques pour le m-PTVP

## 4.1 Introduction

Rappelons que le $m$-PTVP consiste à trouver des tournées réalisables de coût total minimal sur $m$ périodes, telles qu'aucune arête ne soit utilisée plus d'une fois. Dans chaque période, chaque client est servi une seule fois, chaque véhicule effectue au maximum une tournée et chacune de ces tournées débute puis se termine au dépôt avec une charge totale (somme des demandes des clients visités) inférieure ou égale à la capacité des véhicules. Le chapitre précédent était consacré aux modèles mathématiques et aux bornes inférieures polynomiales.

Dans ce chapitre nous proposons deux métaheuristiques pour ce problème, un algorithme de recherche taboue et une recherche locale itérative. Puisque le coût total de toute solution du $m$-PTVP dépend exclusivement de l'ensemble des arêtes utilisées, indépendamment de leur répartition par période, nos métaheuristiques essaient d'introduire dans des solutions des arêtes qui ne sont utilisées dans aucune de leurs $m$ périodes. Toutes les deux sont initialisées avec la solution obtenue par une version adaptée de l'algorithme de Clarke et Wright. Elles utilisent la même structure de voisinage, mais elles l'explorent avec des mécanismes différents d'intensification et de diversification. Pour le rendre plus efficace, nous guidons l'algorithme de recherche taboue avec la solution du problème de $b$-couplage parfait défini au chapitre précédent. Nous y avons également ajouté une procédure de diversification supplémentaire.

La suite de ce chapitre est organisée en quatre sections respectivement consacrées à l'heuristique, aux deux métaheuristiques et aux résultats numériques.

## 4.2 Heuristique adaptée de Clarke et Wright

Pour initialiser très rapidement les métaheuristiques du $m$-PTVP, nous proposons une heuristique inspirée des algorithmes de Krarup [72] pour le $m$-PVP et de Clarke et Wright [26] pour le VRP. Krarup propose d'obtenir une solution initiale du 2-PVP en combinant deux solutions de PVC, la première calculée sur le graphe d'origine et la deuxième calculée sur le graphe obtenu en retirant du graphe d'origine les arêtes utilisées par la première solution. Quant à l'algorithme de Clarke et Wright pour le VRP, il consiste à créer des tournées à un seul client pour servir chaque client du graphe, puis fusionner ces dernières deux par deux tant que le coût total décroît et que la capacité des véhicules est respectée. Cet algorithme a le double avantage de réduire à la fois le coût total et le nombre de véhicules utilisés.

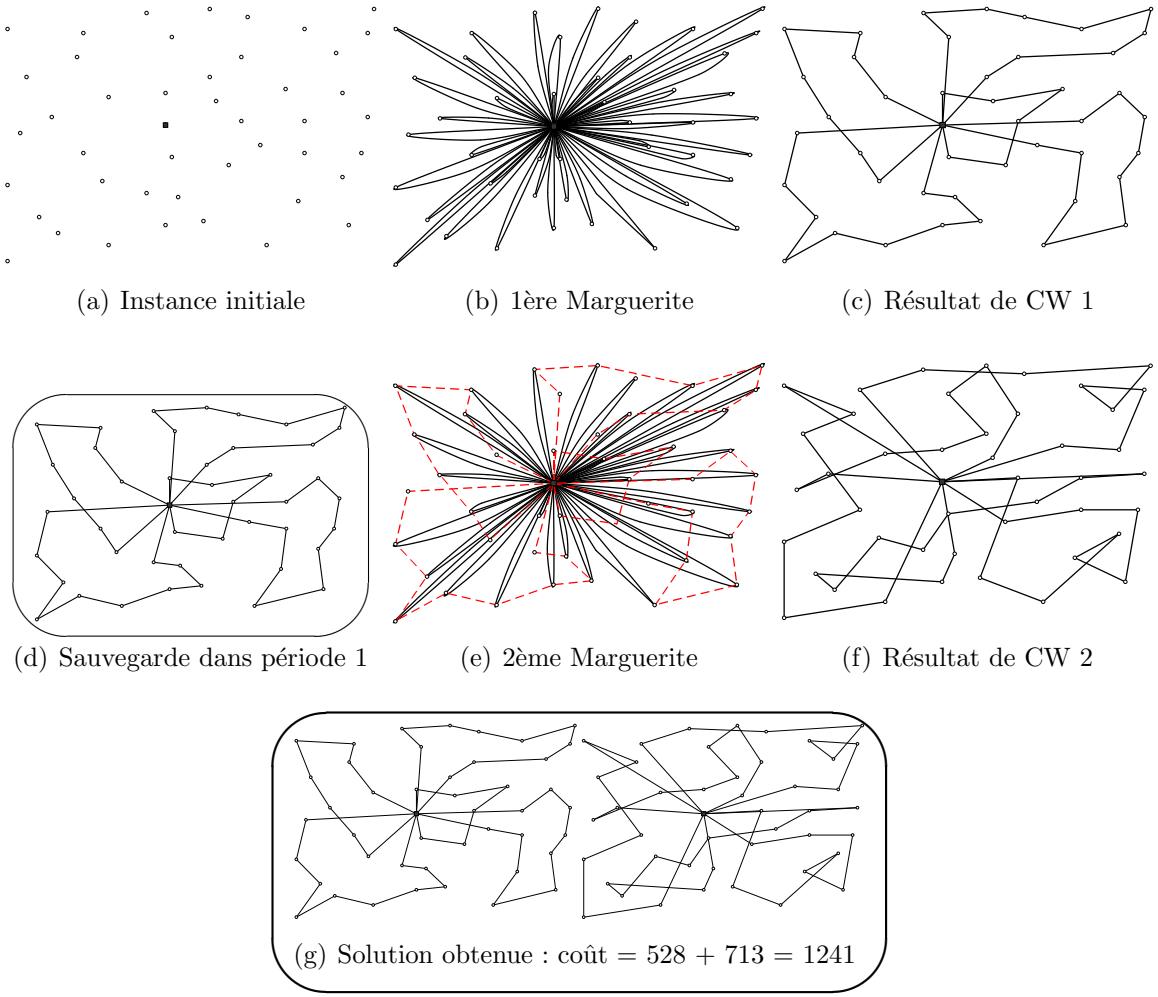
Notre heuristique $m$CW illustrée par la figure 4.1 applique $m$ fois l'algorithme de Clarke et Wright, en pénalisant les arêtes déjà utilisées après chaque itération, pour trouver une solution initiale du $m$-PTVP. La pénalité affectée aux coûts des arêtes déjà utilisées doit être suffisamment importante pour que l'heuristique privilégie les fusions qui produisent des tournées qui ne réutilisent pas des arêtes ayant déjà servis. Si la solution obtenue par l'heuristique $m$CW n'utilise aucune arête plus d'une fois, alors cette solution est réalisable pour le $m$-PTVP. Dans le cas contraire, la solution est non réalisable.

## 4.3 Recherche locale itérative

La recherche locale itérative ou ILS est une métaheuristique stochastique qui explore l'espace des solutions en appliquant une recherche locale à des copies aléatoirement modifiées de la meilleure solution connue. Soit $s$ la solution initiale d'une ILS. A chaque itération une perturbation est appliquée à $s$ pour obtenir $s'$ sur laquelle on applique la recherche locale. $s'$ remplace alors $s$ si elle est meilleure. La métaheuristique obtenue oscille entre les phases d'intensification liées à la recherche locale et les phases de diversification à travers les différentes perturbations. Sachant que le coût d'une solution de $m$-PTVP dépend uniquement du coût des arêtes utilisées, indépendamment de leur répartition parmi les $m$ périodes, toutes nos procédures sont basées sur l'insertion d'arêtes non utilisées dans la solution courante. Notre métaheuristique doit également pouvoir traiter les cas où la solution initiale fournie par l'heuristique constructive $m$CW est non réalisable.

### 4.3.1 Pénalisation de solutions non réalisables

Il est possible que la solution initiale fournie par l'heuristique $m$CW soit non réalisable pour le $m$-PTVP parce qu'au moins une arête y est utilisée plus d'une fois. Pour permettre à notre métaheuristique de s'initialiser à partir d'une telle solution, nous relaxons les contraintes péripatétiques (3.8) et nous ajoutons la pénalité $\sum_{e\in E} \alpha \times \max\{0, (\sum_{k\in\mathbb{M}} x_e^k - 1)\}$ à la fonction-objectif. Cela revient à autoriser mais en la pénalisant l'utilisation multiple



(a) Instance initiale (b) 1ère Marguerite (c) Résultat de CW 1

(d) Sauvegarde dans période 1 (e) 2ème Marguerite (f) Résultat de CW 2

(g) Solution obtenue : coût = 528 + 713 = 1241

FIGURE 4.1 – Heuristique $m$CW appliquée à l'instance P-n45-k5 pour le 2-PTVP

des arêtes, par exemple au cours de deux périodes différentes. La valeur de la constante $\alpha$ doit être suffisamment grande pour que l'algorithme privilégie toujours les solutions réalisables par rapport aux solutions non réalisables.

### 4.3.2 Perturbation

La procédure de perturbation qui consiste à modifier aléatoirement une copie de la meilleure solution connue doit être suffisamment forte pour empêcher le retour à l'optimum local courant, mais également suffisamment faible pour que des caractéristiques de l'actuel optimum local soient préservées. En effet, une perturbation trop faible conduirait systématiquement au même minimum local après application de la recherche locale, alors qu'une perturbation trop forte rendrait les solutions successives trop différentes les unes

des autres et l'algorithme se comporterait alors comme un GRASP.

Notre perturbation pour le $m$-PTVP consiste à insérer successivement $\beta$ arêtes choisies aléatoirement parmi l'ensemble des arêtes non utilisées. Chacune d'entre elles est insérée à l'aide du mouvement qui produit la meilleure solution parmi les $8m$ mouvements décrits au paragraphe 4.3.3 et illustrés par la figure 4.2. L'intensité $\beta$ de la perturbation varie dynamiquement entre deux seuils $\beta_{\min}$ et $\beta_{\max}$. Initialisé à $\beta_{\min}$, ce paramètre est incrémenté toutes les $k_{\max}^{ILS}$ itérations consécutives sans amélioration de la meilleure solution connue. Par contre, à chaque fois que cette dernière est améliorée, $\beta$ est réinitialisée à $\beta_{\min}$.

### 4.3.3 Recherche locale et mouvements

Une procédure de recherche locale consiste à améliorer une solution donnée en lui appliquant des transformations locales successives telles qu'un échange de deux clients, un déplacement d'un client, un mouvement $k$-opt ou une chaîne d'éjection. Rappelons qu'un mouvement $k$-opt consiste à supprimer $k$ arêtes non consécutives de la solution courante et les remplacer par $k$ autres arêtes de manière à obtenir une nouvelle solution réalisable.

Pour le $m$-PTVP, les transformations locales (ou mouvements) que nous avons choisies sont basées sur le principe selon lequel le coût total de toute solution dépend exclusivement de l'ensemble des arêtes utilisées, indépendamment de la répartition de ces arêtes par période. Il s'agit donc d'essayer d'introduire dans la solution courante, des arêtes actuellement inutilisées dans l'ensemble des $m$ périodes. La figure 4.2 illustre les 8 différentes manières d'insérer une arête (A, B) au sein d'une période d'une solution, inspirés du 2-opt. Il y a donc $8m$ mouvements potentiels par arête candidate à l'insertion.

Les mouvements impliquant deux tournées ne sont autorisés que si les nouvelles tournées obtenues respectent les contraintes de capacité : la somme des demandes des clients de chacune des tournées ne doit pas excéder la capacité $Q$ des véhicules. Ces tests de capacité sont effectués en $O(1)$ en pré-calculant et mettant à jour pour chaque tournée $k$ la valeur $charge_k$ égale au total des demandes de ses clients ; et pour chaque client $i$ de la tournée $k$ les valeurs $q_i^+$ et $q_i^-$ égales respectivement au total des demandes depuis le dépôt jusqu'au client $i$ ($q_i^+ = q_i + \sum_{j \in V' | j \text{ est servi avant } i} q_j$), et à la capacité disponible restante après $i$ dans la tournée $k$ (il est évident que $q_i^+ + q_i^- = Q$). En supposant que les tournées illustrées par la figure 4.2 sont parcourues dans le sens des aiguilles d'une montre, les tests de capacité à effectuer avant l'insertion de l'arête (A, B) à l'aide des mouvements de recherche locale de type 5 à 8 sont les suivants :

$$\text{Insertion de Type 5} : \max\{q_C^+ + q_D + q_D^-, q_A^- + q_A + q_B^+\} \leq Q \tag{4.1}$$

$$\text{Insertion de Type 6} : \max\{q_A^+ + q_B^- + q_B, q_E^- + q_E + q_F^+\} \leq Q \tag{4.2}$$

$$\text{Insertion de Type 7} : \max\{q_C^+ + q_F^+, q_A^- + q_A + q_B + q_B^-\} \leq Q \tag{4.3}$$

$$\text{Insertion de Type 8} : \max\{q_A^+ + q_B^+, q_E^- + q_E + q_B + q_B^+\} \leq Q \tag{4.4}$$

Contrairement aux implémentations classiques des mouvements 2-opt qui testent le remplacement de toutes les paires d'arêtes non-consécutives de la solution courante, nous

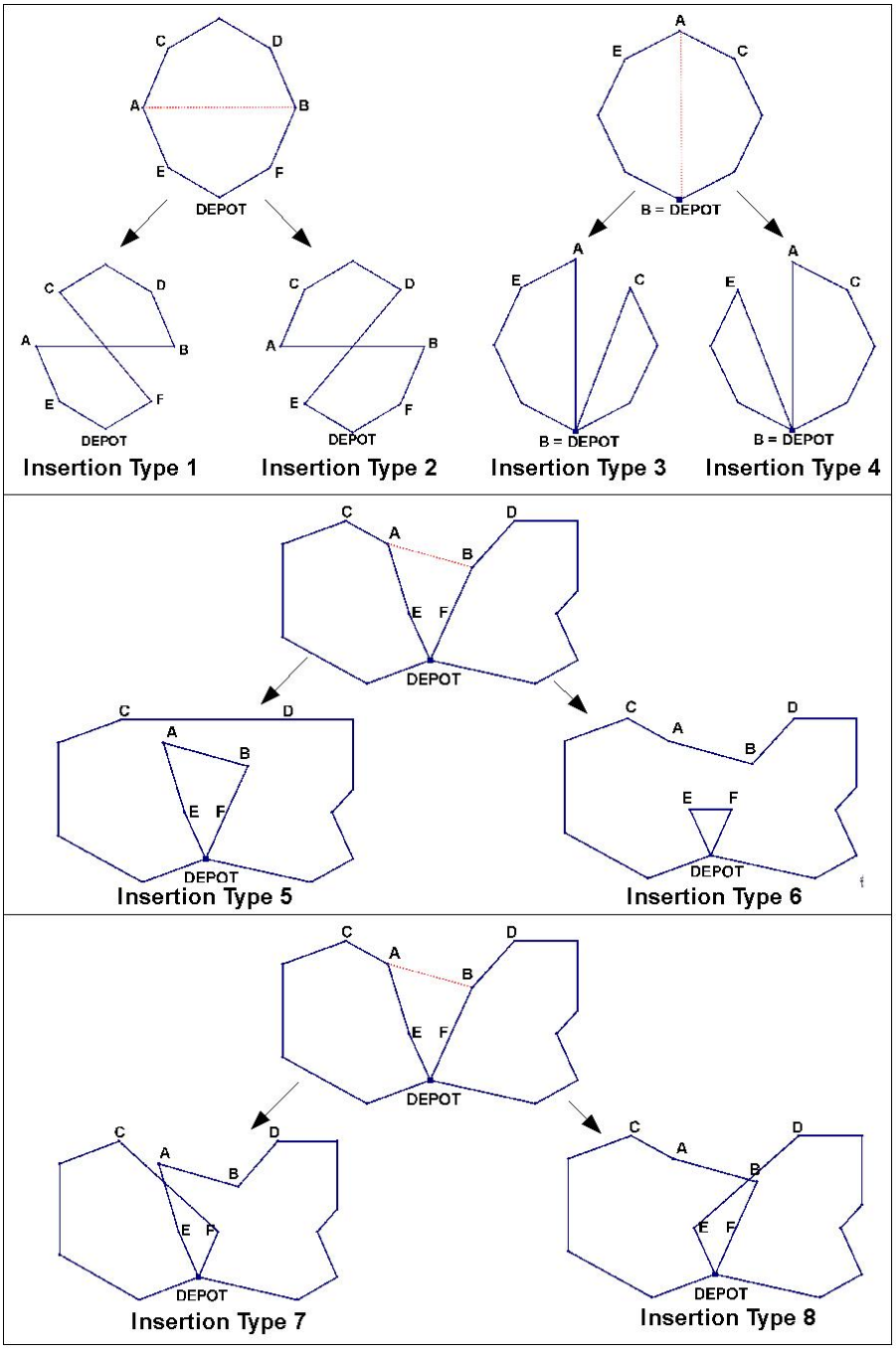



FIGURE 4.2 – Voisinage : les 8 manières d'insérer l'arête (A,B) au cours d'une période

cherchons à y insérer certaines arêtes déjà identifiées (notamment les arêtes non utilisées). De plus, nos mouvements autorisent la scission d'une tournée en deux (cas 3 et 4) ainsi que la fusion de deux tournées ensemble (si l'arête nouvellement insérée connecte les extrémités de deux tournées). Cela est d'autant plus pertinent que le nombre de véhicules n'est pas préalablement fixé, et donc le nombre de tournées de la solution optimale n'est pas connu a priori.

### 4.3.4 Mise à jour de la meilleure solution et critère d'arrêt

Nous utilisons un mécanisme classique selon lequel à la fin de chaque appel à la recherche locale, l'ILS remplace la meilleure solution connue par l'optimum local si et seulement si le coût total de ce dernier est plus faible. Dans le cas contraire l'optimum local trouvé est rejeté. Notre ILS s'arrête dès que $t \geq t_{\max}$, où $t$ est le nombre cumulé de mouvements effectués lors des appels successifs à la recherche locale et $t_{\max}$ est un nombre prédéfini correspondant au nombre maximal autorisé de mouvements. Le critère d'arrêt a été ainsi défini pour comparer l'ILS avec la méthode taboue dans la section 4.5.1, pour un même nombre d'explorations de voisinage.

### 4.3.5 Récapitulatif de l'ILS pour le $m$-PTVP

L'algorithme 1 récapitule l'ILS développé pour le $m$-PTVP. Les mécanismes de perturbations et de recherche locale ont été présentés dans les paragraphes précédents.

**Algorithm 1** Recherche Locale Itérative pour le $m$-PTVP

$\beta := \beta_{\min}$ ; $t := 1$ ; $k := 1$
$mCW(s)$
$RechercheLocale(s, t)$
**répéter**
  $s' := s$
  $Perturbation(s', \beta)$
  $RechercheLocale(s', t)$
  **si** $cout(s') < cout(s)$ **alors**
    $s := s'$
    $\beta := \beta_{\min}$ ; $k := 1$
  **fin si**
  $k := k + 1$
  **si** $(k > k^{ILS}_{\max})$ et $(\beta < \beta_{\max})$ **alors**
    $\beta := \beta + 1$
    $k := 1$
  **fin si**
**jusqu'à** $t \geq t_{\max}$
**renvoyer** $s$

## 4.4 Recherche taboue guidée et diversifiée

La recherche taboue ou TS est une métaheuristique déterministe qui explore l'espace des solutions d'un problème en se déplaçant à chaque itération de la solution courante $s'$ vers la meilleure solution non taboue du voisinage de $s'$, c'est à dire la meilleure solution qui peut être obtenue en appliquant une transformation locale non taboue à $s'$. En effet, pour éviter de boucler indéfiniment sur une séquence de solutions, une liste taboue stocke généralement des attributs des $\tau$ dernières solutions obtenues au cours de la recherche et une transformation locale n'est effectuée que si elle conduit à une solution qui ne possède aucun de ces attributs. D'autres mécanismes tels que la diversification et la granularité sont souvent implémentés. La granularité vise à accélérer le temps d'exécution de l'algorithme par l'utilisation d'un nombre réduit de mouvements, alors que la diversification s'assure que la recherche ne se limite pas à une portion restreinte de l'espace des solutions. Les paragraphes qui suivent décrivent les composants de l'algorithme de recherche taboue guidée et diversifiée (GTSD) que nous avons développé pour le $m$-PTVP.

### 4.4.1 Pénalisation des solutions non réalisables

Notre métaheuristique accepte les solutions non réalisables, mais les pénalise comme expliqué dans le paragraphe 4.3.1.

### 4.4.2 Recherche du meilleur voisin et gestion de la liste taboue

Rappelons que le coût d'une solution du $m$-PTVP dépend uniquement du sous-ensemble des arêtes qui la composent. C'est pourquoi la recherche du meilleur voisin de $s'$ consiste à essayer d'insérer dans $s'$ des arêtes non encore utilisées. D'autre part, il faut définir des critères tabous qui empêchent de retomber sur les mêmes solutions. Sachant cela, notre GTSD identifie à chaque itération $t$ l'ensemble $E'$ des arêtes non taboues et non utilisées par la solution $s'$. Il évalue ensuite le gain potentiel résultant de l'insertion de chacune d'elles dans $s'$ à l'aide des mouvements définis au paragraphe 4.3.3 et illustrés par la figure 4.2. Le meilleur mouvement ainsi trouvé est appliqué à $s'$ et les arêtes enlevées deviennent taboues jusqu'à l'itération $t + \tau$.

Dans une implémentation classique, le paramétrage de $\tau$ est très délicat. Si sa valeur est trop grande de nombreux mouvements sont interdits au risque de passer à côté de très bonnes solutions tandis que si sa valeur est trop petite l'algorithme retrouve des optima locaux déjà visités. C'est pourquoi on utilise souvent une taille dynamique variant entre $\tau_{\min}$ et $\tau_{\max}$, et avec des règles de gestion visant à diminuer $\tau$ pour trouver l'optimum local et augmenter $\tau$ pour sortir d'un bassin d'attraction.

Notre GTSD utilise une approche différente dont le but est de réduire la sensibilité de l'algorithme à la valeur de $\tau$ tout en laissant ce dernier jouer son rôle de mémoire à court terme. Pour cela nous définissons un "tabou partiel" selon lequel l'arête $e'$ à insérer dans $s'$

doit être choisie dans $E'$ mais la deuxième arête entrante $e''$ est libre. Cette dernière peut être déjà utilisée ou pas, mais surtout elle peut être taboue ou pas. A titre d'exemple si le mouvement de type 5 de la figure 4.2 est appliqué, alors $(A, B) \in E'$ alors que $(C, D)$ peut être taboue ou pas, déjà utilisée ou pas. Dans ce dernier cas, nous tiendrons compte de la pénalité définie au paragraphe 4.3.1. L'algorithme "partiellement tabou" ainsi obtenu est moins sensible à la valeur du paramètre $\tau$ tout en préservant le principal avantage de la liste taboue qui est d'empêcher les cycles. Le paramétrage du GTSD en est facilité, puisque la taille de la liste taboue peut rester fixe tout au long de la recherche.

Son implémentation est simple. Il suffit d'affecter à chaque arête $e$ un label $tabou_e$ qui contient le numéro de la dernière itération au cours de laquelle l'arête $e$ a été éjectée de la solution. Ce label est à mettre à jour après chaque mouvement de recherche locale effectué. Ainsi, à l'itération $t$, les arêtes non taboues $e$ sont celles qui vérifient la condition suivante : $tabou_e + \tau \leq t$. Seules ces arêtes peuvent être candidates à l'insertion dans $s'$, si elles ne sont pas déjà utilisées.

### 4.4.3 Guidage par la solution d'un $b$-couplage parfait

Le principe de la granularité introduit par Toth et Vigo [127] est d'accélérer l'exécution d'une métaheuristique en utilisant des voisinages réduits, c'est à dire en ne testant qu'un nombre limité de mouvements, ceux qui a priori ont les plus grandes probabilités de produire de bonnes solutions. Son implémentation classique consiste à retarder l'introduction des arêtes les plus chères dans la solution. Par conséquent à chaque itération les arêtes les plus chères sont supprimées de la liste des arêtes candidates à l'insertion dans la solution courante.

Les résultats préliminaires de Ngueveu *et al.* [99] montrent un écart moyen inférieur à 10% entre le coût optimal du $m$-PTVP et la borne du $b$-couplage parfait présenté au chapitre 3. Nous avons pensé à réparer des solutions de ce dernier pour obtenir de bonnes bornes supérieures pour notre problème. Néanmoins, le $b$-couplage ne produit qu'un sous-ensemble d'arêtes et en extraire une solution de $m$-PTVP est loin d'être évident, même si elle existe, car cela nécessite de partitionner les arêtes entre les $m$ périodes en plus des tournées réalisables, ce qui est NP-difficile comme l'a montré le théorème 2 (page 59). Pour pallier à cette difficulté, nous avons défini une granularité basée sur la solution du $b$-couplage parfait, et nous l'avons combinée avec la trame classique de l'algorithme de recherche taboue. Ainsi, le problème polynomial guide la métaheuristique dans l'espace des solutions. Nous obtenons un algorithme de recherche taboue guidée qui essaie d'insérer en priorité les arêtes non taboues et inutilisées qui composent la solution du $b$-couplage parfait relaxé, puisque celles-ci ont une plus grande probabilité que les autres de faire partie de la solution optimale du $m$-PTVP.

Certes, la résolution du $b$-couplage parfait relaxé produit un ensemble d'arêtes potentiellement bonnes pour le $m$-PTVP puisqu'il s'agit de l'ensemble le moins coûteux qui satisfait les contraintes aggrégées de degré. Toutefois, le faible nombre d'arêtes qui le composent

(par exemple 10% du graphe pour l'instance B-n45-k7 pour le 2-PTVP), entraîne une très courte liste d'arêtes candidates à l'insertion, ce qui induit un voisinage trop restreint. Nous avons trouvé deux manières d'élargir ce voisinage, pour que notre métaheuristique ait accès à plus de solutions, sans perdre l'avantage des données du $b$-couplage parfait.

1. Relaxer les contraintes d'intégrité de notre $b$-couplage pour augmenter le nombre d'arêtes pertinentes dans la liste des candidats, avec toutefois le risque de perdre de la pertinence si trop de fractions d'arêtes sont choisies dans la solution linéaire.
2. Compléter cette liste avec les arêtes les plus petites négligées par le précédent ensemble, suivant l'idée première de Toth et Vigo.

Ce dernier sous-ensemble est composé d'arêtes ayant un coût inférieur ou égal à $\mu\bar{c}$ où $\bar{c}$ est le coût moyen des arêtes utilisées dans la solution initiale et $\mu$ un paramètre. La pénalité appliquée aux solutions non réalisables (voir paragraphe 4.3.1) a été incluse dans le calcul de $\bar{c}$. L'idée derrière le fait de garder cette pénalité dans le calcul est la suivante : en l'absence de pénalités, la solution initiale non réalisable risque d'être moins chère que les solutions réalisables. Par conséquent, les arêtes inclues dans la liste des candidats doivent être légèrement plus chères pour que la métaheuristique trouve des solutions réalisables.

La granularité ($b$-couplage parfait relaxé + courtes arêtes) est appliquée à chaque fois que la meilleure solution connue est améliorée et retirée après $k_{\max}^{GTS}$ itérations sans améliorer la meilleure solution. Au cours de la recherche, l'algorithme oscille donc implicitement entre des phases d'intensification lorsque la granularité est activée et des phases de diversification lorsque la granularité est désactivée. Pour améliorer l'efficacité de notre GTSD, nous avons défini une procédure de diversification supplémentaire qui garantit l'exploration de régions différentes.

### 4.4.4 Diversification

La diversification s'assure que la procédure de recherche ne se limite pas à une portion réduite de l'espace des solutions. L'implémentation proposée par Taillard [124] pénalise les coûts des arêtes en fonction de leurs fréquences d'utilisation au cours de la recherche, puisque des arêtes utilisées trop souvent retiennent l'algorithme dans une région et l'empêchent de visiter l'ensemble de l'espace des solutions.

Nous n'avons pas voulu pénaliser les arêtes utilisées très souvent, car elles peuvent être requises pour atteindre la solution optimale. Par conséquent, après un nombre prédéfini d'itérations, notre procédure de diversification recherche la meilleure manière d'introduire dans la solution courante une des arêtes les moins chères qui n'ont jamais été utilisées. De cette manière, la recherche est diversifiée puisque l'algorithme atteint une région non encore visitée, et ce sans empêcher l'insertion d'une quelconque arête. Pour concilier ce composant avec la granularité déjà définie, la procédure de diversification est appliquée une fois que les deux conditions suivantes sont satisfaites :

1. Au moins $\gamma_{\max}$ itérations se sont écoulées depuis la dernière amélioration de la meilleure solution connue, et ce après désactivation de la granularité 4.4.3.

2. Le mouvement précédent n'était pas améliorant : $improved = false$.

De cette manière, la diversification ne perturbe pas la granularité mais renforce le GTSD si nécessaire. Son implémentation est simple. Il suffit d'affecter à chaque arête $e$ un label $servi_e$ qui contient le nombre de fois où l'arête $e$ a été insérée dans la solution courante $s'$ au cours de la recherche. Ce label est initialement nul pour toutes les arêtes qui ne sont pas dans la solution initiale $(E \setminus E(s'))$ et égal à 1 pour les autres. Ensuite à chaque itération, les labels des arêtes insérées dans $s'$ sont incrémentés. Ainsi, la diversification forcera l'insertion de l'arête la moins chère qui vérifie la condition : $servi_e = 0$.

### 4.4.5 Critère d'aspiration et critère d'arrêt

Nous utilisons un critère d'aspiration classique qui consiste à autoriser un mouvement tabou s'il améliore la meilleure solution trouvée jusqu'à présent. Le GTSD est stoppé lorsque $t = t_{\max}$, où $t$ est le nombre de mouvements appliqués, tabous ou pas, et $t_{\max}$ est le nombre maximal autorisé de mouvements.

### 4.4.6 Récapitulatif du GTSD pour le $m$-PTVP

L'algorithme 2 récapitule la structure générale de notre recherche taboue guidée et diversifiée. Les mécanismes de recherche de meilleur voisin, de diversification et de mise à jour ont été expliqués dans les paragraphes précédents.

## 4.5 Résultats numériques

### 4.5.1 Instances et implémentation

Nous avons testé nos algorithmes sur les cas particuliers du VRP et du $m$-PVP avant de les appliquer au $m$-PTVP avec $m > 1$. Quatre types d'instances VRP de 19 à 101 nœuds sont utilisées, A, B, P et VRPNC, ainsi que 20 des plus grandes instances de la TSPLIB [115], allant de 70 à 280 nœuds, parmi celles déjà utilisées par Duchenne *et al.* [41] pour le $m$-PVP. Pour plus de détails sur ces instances, se référer au chapitre 3.

L'heuristique et les métaheuristiques sont programmées en C sur un PC Pentium de 3,39 GHz et 1 Go de mémoire. Notons que TSD et GTS désignent respectivement les versions sans guidage et sans diversification de notre algorithme de recherche taboue GTSD. Dans les pages qui suivent, les résultats obtenus sont présentés sous forme aggrégée par les tableaux 4.2 à 4.7, afin d'en simplifier la lecture et l'analyse. Le détail des résultats est disponible en annexe. Le nombre d'instances utilisées par type est précisé dans la première colonne de chaque tableau. Soit $UB$ la meilleure borne supérieure obtenue avec différents réglages, les colonnes suivantes sont utilisées dans les différents tableaux :

**Algorithm 2** Recherche taboue taboue guidée et diversifiée pour le $m$-PTVP

1: $mCW(s)$
2: $s' := s$ ; $granular := true$ ; $improved := true$
3: $t := 1$ ; $k := 1$ ; $\gamma := 1$ ; $\forall e \in E$, $tabou_e := -\infty$ et $servi_e := 0$ ; $\forall e \in E \setminus E(s')$, $servi_e := 1$
4: **pour** $t = 1$ **à** $t_{\max}$ **faire**
5: $RechercheMeilleurVoisin(s', cout(s), granular, improved)$
6: **si** $cout(s') < cout(s)$ **alors**
7: $s := s'$
8: $k := 1$ ; $\gamma := 1$
9: $granular := true$
10: **sinon**
11: $\gamma := \gamma + 1$
12: **si** $granular = true$ **alors**
13: $k := k + 1$
14: **si** $k \geq k^{GTS}_{\max}$ **alors**
15: $granular := false$
16: $k := 1$ ; $\gamma := 1$
17: **fin si**
18: **fin si**
19: **si** $(\gamma > \gamma_{\max})$ et $(improved = false)$ **alors**
20: $Diversification(s', servi)$
21: **fin si**
22: **fin si**
23: $MiseàjourLabels(tabou, t)$
24: **fin pour**
25: **renvoyer** $s$

| Algo | Param | Description | Réglage 1 | Réglage 2 |
|---|---|---|---|---|
| GTSD, ILS | $\alpha$ | Pénalisation | $2\overline{c}_{max}$ | |
| GTSD, ILS | $t_{\max}$ | Nbre max d'itérations (it.) | 10000 | |
| GTSD | $\tau$ | Taille de liste taboue | $n$ | |
| GTSD | $\mu$ | Coefficient appliqué au coût moyen des arêtes | 1.30 | |
| GTSD | $k_{\max}^{GTS}$ | Nbre max d'it. avant retrait de la granularité | $2n/3$ | $2n$ |
| GTSD | $\gamma_{\max}$ | Nbre max d'it. avant diversification | $2n$ | |
| ILS | $\beta_{\max}$ | Nbre max d'arêtes insérées pour perturber l'ILS | 5 | |
| ILS | $\beta_{\min}$ | Nbre min d'arêtes insérées pour perturber l'ILS | 1 | |
| ILS | $k_{\max}^{ILS}$ | Nbre d'it. avant une mise-à-jour de $\beta$ | $1500/n$ | |

TABLE 4.1 – Réglages de paramètres

1. $\delta_{\%}$ : moyenne du ratio $\frac{100(UB_i - UB)}{UB}$ pour la (méta)heuristique $i = m$CW, ILS ou GTSD.
2. $\sigma$ : écart-type lié à $\delta_{\%}$
3. *Nbf* : nombre d'instances pour lesquelles $m$CW a trouvé une solution réalisable. Seules les solutions réalisables sont utilisées dans le calcul du $\delta_{\%}$ de $m$CW.
4. $T$ : temps moyen en secondes pour atteindre la meilleure solution
5. $Bp$ : ratio entre le nombre d'arêtes choisies par le $b$-couplage parfait relaxé et $|E|$

Pour que la comparaison entre ILS et GTSD soit équitable, les deux métaheuristiques sont arrêtées après le même nombre de mouvements réalisés $t_{\max}$. En effet, l'essentiel du temps de calcul est consommé par les explorations de voisinage des deux métaheuristiques. Une itération du GTSD correspond à un seul mouvement tandis que la recherche locale utilisée par l'ILS peut effectuer plusieurs mouvements successifs. Les résultats préliminaires pour paramétrer les métaheuristiques ont conduit à un réglage différent du GTSD par problème et par type d'instances. Afin de limiter le nombre de réglages utilisés, nous avons choisi de n'en garder que deux : réglage 1 pour la plupart des problèmes (VRP, $m$-PVP et $m$-PTVP avec $m \in \{2, 3, 5, 6, 7\}$) et réglage 2 pour le 4-PTVP. Les paramètres correspondants sont listés dans la table 4.1.

### 4.5.2 Tests sur le 1-PTVP (VRP)

Le tableau 4.2 illustre les résultats de nos méthodes sur le cas particulier du VRP, tandis que le tableau 4.3 mets en lumière les variantes avec ou sans guidage et avec ou sans diversification de l'algorithme de recherche taboue. Les bornes supérieures *UB* utilisées pour les calculs des ratios $\delta_{\%}$ sont les optimum connus des instances de type A, B et P.

Nous obtenons de bons résultats sur le VRP car l'écart par rapport à *UB* est en moyenne de 0,80%. Le GTSD en particulier domine l'ILS qui a un écart à l'optimum nettement supérieur à 1,00% pour les instances de type A, B, P et 1,26% pour le type VRPNC. Les deux algorithmes sont initialisés avec une heuristique $m$CW qui est déjà entre 4,42% et 7,01%

| | $m$CW | | | ILS | | | GTSD | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $\delta_{\%}$ | $\sigma$ | *Nbf* | $\delta_{\%}$ | $\sigma$ | $T$ | $\delta_{\%}$ | $\sigma$ | $T$ | *Bp* |
| A (27) | 4,42 | 1,92 | 24 | 1,82 | 1,36 | 0,86 | 0,54 | 0,57 | 3,43 | 0,05 |
| B (23) | 3,74 | 2,61 | 23 | 1,58 | 1,64 | 1,75 | 0,93 | 1,45 | 3,78 | 0,05 |
| P (23) | 8,04 | 3,05 | 17 | 3,81 | 2,47 | 1,38 | 0,41 | 0,53 | 3,45 | 0,06 |
| VRPNC (7) | 7,01 | 3,66 | 7 | 4,73 | 3,25 | 30,52 | 1,26 | 1,85 | 17,76 | 0,02 |

TABLE 4.2 – Résultats de $m$CW, ILS et GTSD appliqués au VRP = 1-PTVP

| | TS | | | TSD | | | GTS | | | GTSD | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $\delta_{\%}$ | $\sigma$ | $T$ | $\delta_{\%}$ | $\sigma$ | $T$ | $\delta_{\%}$ | $\sigma$ | $T$ | $\delta_{\%}$ | $\sigma$ | $T$ |
| A (27) | 0,56 | 0,76 | 3,28 | **0,53** | 0,73 | 2,80 | **0,54** | 0,50 | 2,88 | **0,54** | 0,57 | 3,43 |
| B (23) | **0,84** | 1,47 | 1,97 | 0,95 | 1,49 | 2,46 | 0,96 | 1,50 | 3,29 | 0,93 | 1,45 | 3,77 |
| P (23) | 0,50 | 0,56 | 3,63 | 0,56 | 0,61 | 2,91 | 0,47 | 0,54 | 3,04 | **0,41** | 0,53 | 3,45 |
| VRPNC (7) | 1,49 | 1,71 | 12,69 | **1,22** | 1,52 | 25,02 | **1,23** | 1,37 | 26,02 | 1,26 | 1,85 | 17,76 |
| Moyenne | 0,85 | 1,12 | 5,39 | 0,81 | 1,10 | 8,30 | 0,80 | 0,98 | 8,81 | **0,78** | 1,10 | 7,10 |

TABLE 4.3 – Influence des différents composants du (G)TS(D) pour le VRP = 1-PTVP

de l'optimum. Les neuf solutions initiales non réalisables sont dues aux tournées à un seul client obtenues par $m$CW, qui utilisent la même arête deux fois, mais les métaheuristiques arrivent toujours à trouver des solutions sans pénalités. Comme prévu, le $b$-couplage parfait relaxé fournit un faible nombre d'arêtes ($\approx 4\%$ de $|E|$).

En ce qui concerne l'influence du guidage par le $b$-couplage et de la diversification de la recherche taboue, nous constatons que GTSD produit de meilleurs résultats que TSD sur 3 types d'instances sur 4 et le guidage réduit l'écart-moyen à l'optimum. Les résultats de GTSD sur le VRP peuvent être encore améliorés si la procédure de diversification était activée un peu plus tard sur les instances de type A ou un peu plus tôt sur les instances de type B : pour A, $\delta_{\%}$ =0,48% si $\gamma_{\max} = 3n$ au lieu de $2n$, tandis que pour B, $\delta_{\%}$ =0,89% si $\gamma_{\max} = 3n/2$.

### 4.5.3 Tests sur le $m$-PVP

Le tableau 4.4 présente les résultats obtenus par nos algorithmes sur le cas particulier du $m$-PVP, tandis que le tableau 4.5 mets en lumière les quatre variantes de l'algorithme de recherche taboue. Les bornes supérieures *UB* utilisées pour les calculs des ratios $\delta_{\%}$ correspondent aux valeurs des solutions optimales proposées dans l'article de Duchenne *et al.* [41], car nous n'avons gardé que les couples "$m$ - instances" qui y sont résolus optimalement.

L'heuristique initiale $m$CW produit des solutions entre 12,28% et 14,44% de l'optimum, et les solutions initiales sont réalisables dans la plupart des cas, 33 sur 35. Les deux métaheuristiques donnent des résultats similaires ($\delta_{\%} \approx$2,00%), mais GTSD est presque deux fois plus rapide que ILS. Le $b$-couplage parfait relaxé sélectionne seulement 2% des arêtes

(a) 2-PVP

| | $m$CW | | | ILS | | | GTSD | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $\delta_{\%}$ | $\sigma$ | *Nbf* | $\delta_{\%}$ | $\sigma$ | $T$ | $\delta_{\%}$ | $\sigma$ | $T$ | *Bp* |
| 19 instances* | 14,44 | 3,90 | 19 | 2,09 | 0,91 | 222,65 | 2,26 | 0,67 | 61,42 | 0,02 |

* st70, eil-76 -101, pr-76 -107 -136 -226 -264, rat99, Kro-B100 -C100 -D100 -E100 -A150 -A200, lin105, ch130, d198, a280

(b) 3-PVP

| | $m$CW | | | ILS | | | GTSD | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $\delta_{\%}$ | $\sigma$ | *Nbf* | $\delta_{\%}$ | $\sigma$ | $T$ | $\delta_{\%}$ | $\sigma$ | $T$ | *Bp* |
| 10 instances* | 12,51 | 2,78 | 9 | 2,04 | 0,55 | 136,09 | 1,71 | 0,55 | 30,89 | 0,03 |

* eil-76 -101, pr76, Kro-B100 -C100 -D100 -E100, pr-107 -124 -136

(c) 4-PVP

| | $m$CW | | | ILS | | | GTSD | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $\delta_{\%}$ | $\sigma$ | *Nbf* | $\delta_{\%}$ | $\sigma$ | $T$ | $\delta_{\%}$ | $\sigma$ | $T$ | *Bp* |
| 5 instances* | 12,28 | 3,26 | 6 | 1,69 | 0,59 | 164,82 | 1,52 | 0,57 | 30,90 | 0,04 |

* st70, pr-76 -107 -124, Kro-B100 -D100

TABLE 4.4 – Résultats de $m$CW, ILS et GTSD appliqués au $m$-PVP ($\lambda = 1$)

du graphe, ce qui conduit à une granularité très petite, laquelle explique probablement pourquoi ILS trouve parfois de meilleures solutions que GTSD. TS domine TSD, ce qui laisse penser que de meilleurs résultats pourraient être obtenus si la valeur du paramètre $\gamma_{\max}$ était augmentée : cela permettrait à TSD d'atteindre les meilleures solutions trouvées par TS avant de déclencher la diversification pour changer la région explorée au sein de l'espace des solutions.

### 4.5.4 Tests sur le $m$-PTVP

Le tableau 4.6 présente nos résultats sur le $m$-PTVP en utilisant les instances de type A, B, P et VRPNC avec $m > 1$, tandis que le tableau 4.7 met en lumière les variantes TS, TSD, GTS et GTSD de notre algorithme de recherche taboue. Pour chaque valeur de $m$, seules sont retenues les instances qui respectent la condition de faisabilité exprimée par l'équation (3.3), c'est pourquoi le nombre d'instances disponibles diminue lorsque $m$ augmente. Les ratios $\delta_{\%}$ ont été calculés par rapport aux meilleures bornes supérieures connues *UB*, dont l'optimalité n'a pas encore été démontrée pour les instances les plus grandes (voir le chapitre 6 sur l'approche polyédrale).

L'heuristique $m$CW trouve peu de solutions réalisables, à peine 44 sur 177 cas, et plus $m$ augmente, plus souvent les deux métaheuristiques sont initialisées avec une solution non réalisable. ILS est beaucoup moins efficace que la recherche taboue pour chaque valeur de $m$ et type d'instances. Le tableau 4.8 confirme que cette faible performance n'est pas liée au germe (seed) choisi pour le générateur de nombres aléatoires : plusieurs germes différents

(a) 2-PVP

| | TS | | | TSD | | | GTS | | | GTSD | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $\delta_{\%}$ | $\sigma$ | $T$ | $\delta_{\%}$ | $\sigma$ | $T$ | $\delta_{\%}$ | $\sigma$ | $T$ | $\delta_{\%}$ | $\sigma$ | $T$ |
| 19 instances* | **2,20** | 0,89 | 94,58 | 2,45 | 0,80 | 44,63 | 2,44 | 0,95 | 25,07 | 2,26 | 0,67 | 61,42 |

* st70, eil-76 -101, pr-76 -107 -136 -226 -264, rat99, Kro-B100 -C100 -D100 -E100 -A150 -A200, lin105, ch130, d198, a280

(b) 3-PVP

| | TS | | | TSD | | | GTS | | | GTSD | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $\delta_{\%}$ | $\sigma$ | $T$ | $\delta_{\%}$ | $\sigma$ | $T$ | $\delta_{\%}$ | $\sigma$ | $T$ | $\delta_{\%}$ | $\sigma$ | $T$ |
| 10 instances* | 1,65 | 0,44 | 32,20 | 1,67 | 0,43 | 41,02 | **1,54** | 0,37 | 50,22 | 1,71 | 0,55 | 31,89 |

* eil-76 -101, pr76, Kro-B100 -C100 -D100 -E100, pr-107 -124 -136

(c) 4-PVP

| | TS | | | TSD | | | GTS | | | GTSD | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $\delta_{\%}$ | $\sigma$ | $T$ | $\delta_{\%}$ | $\sigma$ | $T$ | $\delta_{\%}$ | $\sigma$ | $T$ | $\delta_{\%}$ | $\sigma$ | $T$ |
| 6 instances* | **1,45** | 0,68 | 47,48 | 1,56 | 0,80 | 36,04 | 1,54 | 0,53 | 37,48 | 1,52 | 0,57 | 30,90 |

* st70, pr-76 -107 -124, Kro-B100 -D100

TABLE 4.5 – Influence des différents composants du (G)TS(D) pour le $m$-PVP ($\lambda = 1$)

ont été appliqués à des instances VRPNC pour le 2-PTVP sans constater d'amélioration significative, l'ILS est resté moins efficace que le GTSD pourtant déterministe. Le $b$-couplage parfait relaxé sélectionne en moyenne 19% des arêtes du graphe initial.

La figure 4.2 montre l'évolution de la recherche taboue au cours du temps pour des instances de type VRPNC. Pour tracer ces courbes nous avons fait imprimer dans un fichier toutes les itérations au cours desquelles les coûts des meilleures solutions sont améliorés ainsi que les temps de calcul correspondant, puis nous avons reconstitué l'évolution du meilleur coût en fonction du temps par tranche de 5 secondes et aggrégé les résultats des 7 instances de type VRPNC. La figure 4.2 ainsi obtenue suggère qu'en moyenne la domination de GTSD sur les trois autres variantes de la recherche taboue se renforce lorsque $m$ augmente. Cette remarque est confirmée par la plupart des tableaux de résultats : excepté pour le 4-PTVP, l'écart du GTSD aux meilleures solutions connues est le plus faible. De plus, ses résultats peuvent être significativement améliorés si on autorise le paramètre $\gamma_{\max}$ à prendre une valeur différente par problème : par exemple pour le 2-PTVP, $\delta_{\%}$ est réduit de 0,98% à 0,91% si le paramètre de diversification $\gamma_{\max}$ est réduit de $2n$ à $3n/2$. Notons qu'il est normal que les chiffres de la figure 4.2 et ceux des tableaux ne soient pas identiques bien qu'ils reflètent les mêmes phénomènes, car la figure montre l'évolution de $\delta_{\%}$ en fonction du temps (en moyenne sur les 7 instances VRPNC) alors que les tableaux de résultats montrent les écarts entre $UB$ et les coûts des meilleures solutions trouvées par chaque métaheuristique, ainsi les durées moyennes respectives pour les atteindre.

(a) 2-PTVP

| | $m$CW | | | ILS | | | GTSD | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $\delta_{\%}$ | $\sigma$ | *Nbf* | $\delta_{\%}$ | $\sigma$ | $T$ | $\delta_{\%}$ | $\sigma$ | $T$ | *Bp* |
| A (26) | 15,29 | 4,66 | 8 | 4,15 | 1,54 | 3,02 | 1,14 | 0,87 | 7,88 | 0,21 |
| B (23) | 12,64 | 5,96 | 10 | 2,10 | 1,48 | 3,07 | 0,72 | 0,66 | 8,86 | 0,24 |
| P (19) | 9,61 | 2,95 | 8 | 3,85 | 1,82 | 3,41 | 1,02 | 0,74 | 10,09 | 0,19 |
| VRPNC (7) | 12,00 | 2,58 | 3 | 3,57 | 1,80 | 54,32 | 1,03 | 0,78 | 68,44 | 0,10 |

(b) 3-PTVP

| | $m$CW | | | ILS | | | GTSD | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $\delta_{\%}$ | $\sigma$ | *Nbf* | $\delta_{\%}$ | $\sigma$ | $T$ | $\delta_{\%}$ | $\sigma$ | $T$ | *Bp* |
| A (25) | - | - | 0 | 3,92 | 1,60 | 5,37 | 0,82 | 0,86 | 14,81 | 0,32 |
| B (22) | 15,27 | 6,44 | 3 | 2,77 | 1,93 | 7,33 | 1,18 | 1,01 | 14,90 | 0,28 |
| P (14) | 6,45 | 1,27 | 3 | 3,23 | 0,97 | 5,78 | 0,76 | 0,36 | 14,08 | 0,20 |
| VRPNC (7) | 15,21 | 5,99 | 3 | 3,49 | 1,46 | 74,07 | 0,98 | 0,68 | 47,44 | 0,14 |

(c) 4-PTVP

| | $m$CW | | | ILS | | | GTSD | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $\delta_{\%}$ | $\sigma$ | *Nbf* | $\delta_{\%}$ | $\sigma$ | $T$ | $\delta_{\%}$ | $\sigma$ | $T$ | *Bp* |
| P (8) | 6,35 | 0,28 | 2 | 2,02 | 0,81 | 8,50 | 0,45 | 0,20 | 4,75 | 0,21 |
| VRPNC (6) | 8,96 | - | 1 | 3,52 | 1,63 | 110,83 | 0,53 | 0,30 | 54,92 | 0,14 |

(d) $5, 6, 7$-PTVP

| | $m$CW | | | ILS | | | GTSD | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $\delta_{\%}$ | $\sigma$ | *Nbf* | $\delta_{\%}$ | $\sigma$ | $T$ | $\delta_{\%}$ | $\sigma$ | $T$ | *Bp* |
| P (11) | 6,38 | - | 1 | 1,76 | 1,33 | 29,35 s | 0,39 | 0,19 | 18,53 s | 0,15 |
| VRPNC (10) | 9,10 | - | 1 | 2,71 | 0,89 | 165,88 s | 0,59 | 0,57 | 159,13 s | 0,17 |

TABLE 4.6 – Résultats de $m$CW, ILS et GTSD appliqués au $m$-PTVP : $m > 1$

(a) 2-PTVP

| | TS | | | TSD | | | GTS | | | GTSD | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $\delta_{\%}$ | $\sigma$ | $T$ | $\delta_{\%}$ | $\sigma$ | $T$ | $\delta_{\%}$ | $\sigma$ | $T$ | $\delta_{\%}$ | $\sigma$ | $T$ |
| A (26) | 1,11 | 0,63 | 9,37 | 0,92 | 0,60 | 9,53 | **0,81** | 0,56 | 8,55 | 1,15 | 0,85 | 7,76 |
| B (23) | 1,10 | 1,14 | 10,27 | 0,71 | 0,56 | 11,39 | **0,68** | 0,73 | 8,53 | 0,72 | 0,66 | 8,86 |
| P (19) | 1,08 | 0,83 | 10,28 | **1,03** | 0,61 | 7,78 | 1,16 | 0,86 | 8,44 | **1,02** | 0,74 | 10,09 |
| VRPNC (7) | 1,48 | 0,67 | 39,87 | 1,28 | 0,64 | 55,50 | 1,25 | 0,76 | 47,23 | **1,03** | 0,78 | 68,44 |
| Moyenne | 1,20 | 0,82 | 17,45 | **0,98** | 0,60 | 21,05 | **0,97** | 0,73 | 18,20 | **0,98** | 0,76 | 23,79 |

(b) 3-PTVP

| | TS | | | TSD | | | GTS | | | GTSD | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $\delta_{\%}$ | $\sigma$ | $T$ | $\delta_{\%}$ | $\sigma$ | $T$ | $\delta_{\%}$ | $\sigma$ | $T$ | $\delta_{\%}$ | $\sigma$ | $T$ |
| A (25) | 1,28 | 1,20 | 14,14 | 1,05 | 0,79 | 12,97 | 1,07 | 0,88 | 11,30 | **0,84** | 0,85 | 13,77 |
| B (22) | 1,94 | 1,91 | 15,51 | 1,12 | 0,91 | 17,27 | 1,47 | 1,35 | 14,51 | **1,02** | 0,98 | 14,86 |
| P (14) | 0,97 | 0,45 | 14,23 | **0,75** | 0,33 | 13,13 | 0,88 | 0,49 | 0,17 | **0,76** | 0,36 | 14,08 |
| VRPNC (7) | 1,06 | 0,54 | 69,59 | **0,97** | 0,76 | 85,80 | 1,13 | 0,76 | 67,50 | **0,98** | 0,68 | 47,44 |
| Moyenne | 1,31 | 1,02 | 28,37 | 0,97 | 0,70 | 32,29 | 1,14 | 0,87 | 23,37 | **0,90** | 0,72 | 22,54 |

(c) 4-PTVP

| | TS | | | TSD | | | GTS | | | GTSD | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $\delta_{\%}$ | $\sigma$ | $T$ | $\delta_{\%}$ | $\sigma$ | $T$ | $\delta_{\%}$ | $\sigma$ | $T$ | $\delta_{\%}$ | $\sigma$ | $T$ |
| P (8) | **0,32** | 0,24 | 19,28 | 0,37 | 0,33 | 11,32 | 0,43 | 0,19 | 12,33 | 0,45 | 0,20 | 4,75 |
| VRPNC (6) | 0,69 | 0,31 | 111,30 | 0,56 | 0,30 | 104,92 | **0,51** | 0,34 | 158,50 | **0,53** | 0,29 | 54,92 |
| Moyenne | 0,50 | 0,27 | 65,29 | **0,46** | 0,31 | 58,12 | **0,47** | 0,26 | 85,41 | 0,49 | 0,24 | 29,83 |

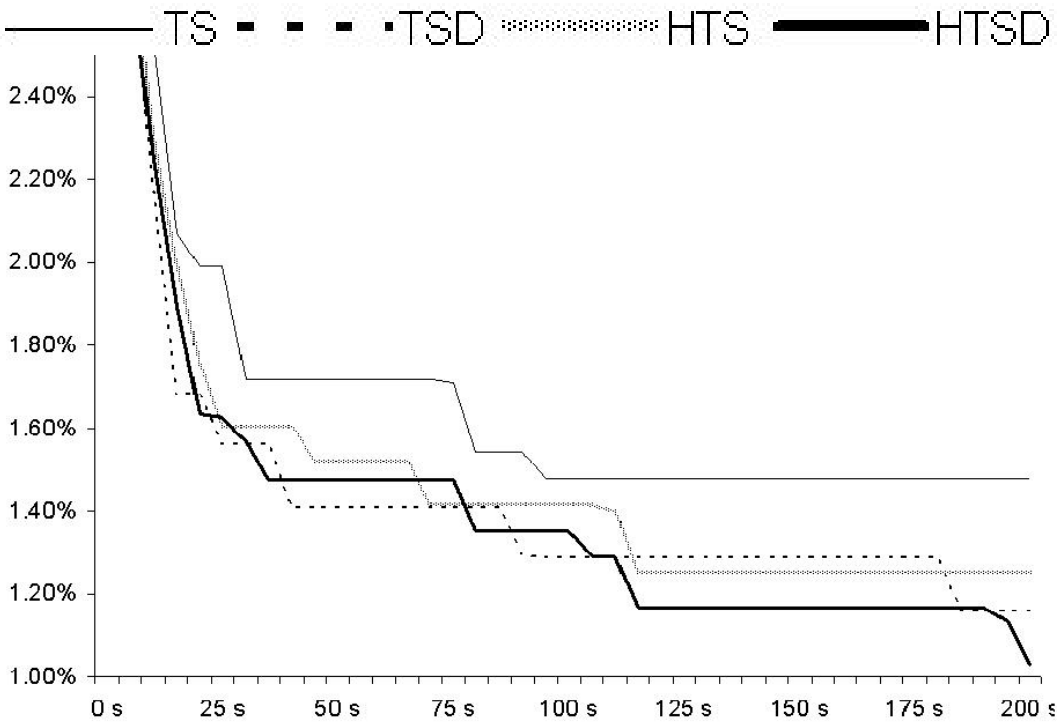
(d) $5, 6, 7$-PTVP

| | TS | | | TSD | | | GTS | | | GTSD | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $\delta_{\%}$ | $\sigma$ | $T$ | $\delta_{\%}$ | $\sigma$ | $T$ | $\delta_{\%}$ | $\sigma$ | $T$ | $\delta_{\%}$ | $\sigma$ | $T$ |
| P (11) | 0,40 | 0,24 | 63,84 | 0,44 | 0,31 | 55,04 | **0,34** | 0,24 | 61,18 | 0,40 | 0,24 | 37,53 |
| VRPNC (10) | 1,08 | 0,92 | 208,73 | 0,76 | 0,43 | 187,46 | 0,57 | 0,47 | 235,14 | **0,50** | 0,46 | 181,40 |
| Moyenne | 0,69 | 0,58 | 168,20 | 0,60 | 0,37 | 121,25 | **0,45** | 0,35 | 148,16 | **0,45** | 0,35 | 109,40 |

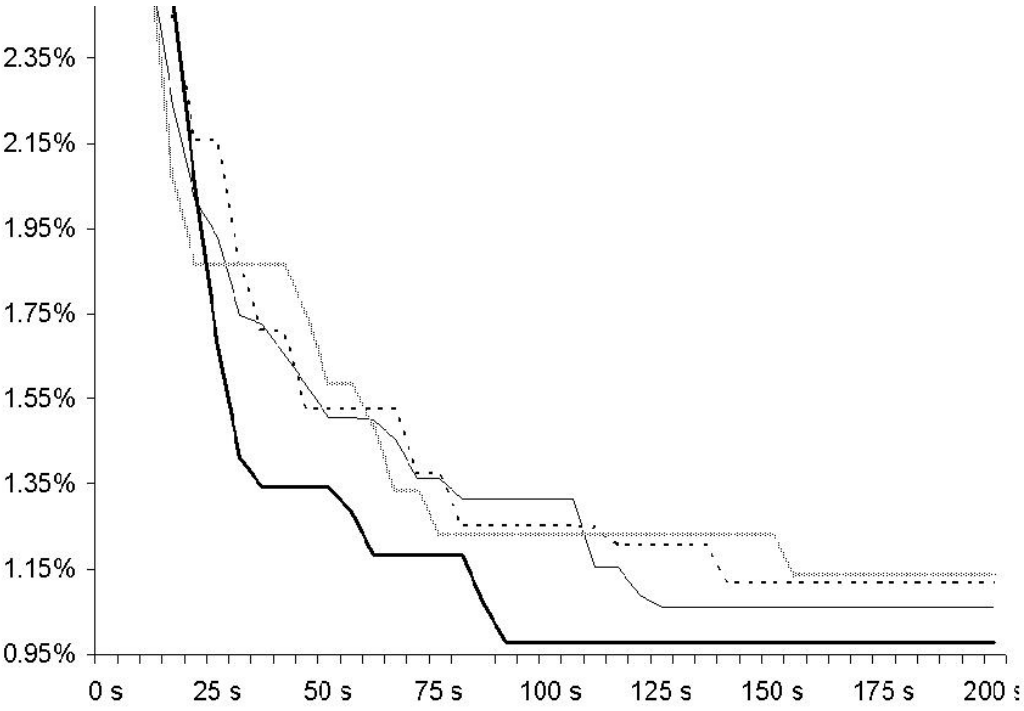
TABLE 4.7 – Influence des différents composants du (G)TS(D) pour le $m$-PTVP : $m > 1$

| $\delta$ avec le germe | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| VRPNC (7) | 3,86 | 3,91 | 4,15 | 3,61 | 4,13 |

TABLE 4.8 – Robustesse de l'ILS testée sur les instances VRPNC pour le 2-PTVP




(a) $m = 2$




(b) $m = 3$

FIGURE 4.3 – Evolution de $\delta$ en fonction du temps pour les instances VRPNC

## 4.6 Conclusion

Ce chapitre introduit une heuristique constructive et deux métaheuristiques pour le problème de tournées de véhicules $m$-péripatétiques ($m$-PTVP). Ce problème consiste à extraire d'un graphe complet des tournées de véhicules disjointes de coût total minimal sur $m$ périodes telles que chaque client soit visité exactement une fois par période. Il peut être considéré comme une généralisation de deux problèmes NP-difficiles connus : le VRP = 1-PTVP et le $m$-PVP, correspondant au cas mono-véhicule.

L'heuristique constructive proposée généralise l'algorithme de Clarke et Wright pour le VRP. La première métaheuristique est une recherche locale itérative (ILS). La deuxième métaheuristique est un algorithme de recherche partiellement taboue, guidée et diversifiée (GTSD). Les deux métaheuristiques utilisent des mouvements dont le but est d'insérer dans la solution courante des arêtes choisies qui ne sont pas utilisées. Les tests numériques ont montré leur efficacité sur le $m$-PTVP ainsi que sur les cas particuliers du VRP et du $m$-PVP. Les résultats montrent également que la diversification et le guidage avec la solution du $b$-couplage introduit au chapitre précédent améliorent la vitesse et la performance de la recherche taboue, faisant de GTSD une meilleure métaheuristique que l'ILS.

Ces métaheuristiques ont fait l'objet de présentations aux conférences internationales *Optimization* [98] et *MatHeuristics* [100] puis d'un chapitre de livre dans la collection *Operations Research / Computer Science Interfaces Series* [105]. Les bonnes bornes supérieures obtenues ont été utilisées au sein de l'approche de type génération de colonnes du chapitre 5, ainsi que la l'approche polyédrale du chapitre 6.

# Chapitre 5

# Approche de type génération de colonnes pour le $m$-PTVP

## 5.1 Introduction

Rappelons que le $m$-PTVP consiste à trouver des tournées réalisables de coût total minimal sur $m$ périodes telles qu'aucune arête ne soit utilisée plus d'une fois. Dans chaque période, chaque client est servi une seule fois, chaque véhicule effectue au maximum une tournée et chacune de ces tournées débute puis se termine au dépôt avec une charge totale (somme des demandes des clients visités) inférieure ou égale à la capacité des véhicules. Les chapitres précédents étaient consacrés aux bornes inférieures polynomiales et aux métaheuristiques.

Dans ce chapitre nous proposons une borne inférieure plus élaborée pour le $m$-PTVP, basée sur la recherche de bonnes solutions duales de la relaxation linéaire de la formulation de partitionnement agrégée, à l'aide de deux heuristiques duales ascendantes (ou *dual ascent*). La première heuristique (H1) est basée sur l'approximation des tournées par des tournées non élémentaires appelées $\mathfrak{q}$-tournées et l'utilisation d'une approche de type génération de colonnes pour pallier au nombre exponentiel de variables. Une tournée normale de demande totale $q$ n'autorise pas de "boucles", alors que pour une $\mathfrak{q}$-tournée cette contrainte est relaxée. Si la présence des boucles est autorisée, elle est cependant pénalisée par la relaxation lagrangienne. Les $\mathfrak{q}$-tournées pouvant être générées avec un algorithme pseudo-polynomial de programmation dynamique, alors l'heuristique résultante est rapide et efficace. Cette dernière n'a pas besoin de faire appel à Cplex® ou tout autre solveur. En fait chaque problème-maitre obtenu après la génération des $\mathfrak{q}$-tournées à coût réduit négatif est résolu en estimant les meilleures valeurs des variables duales. Puis, les violations des contraintes de degré sont quantifiées et utilisées pour corriger (par la relaxation lagrangienne) les valeurs attribuées aux variables duales, jusqu'à ce que les améliorations deviennent négligeables ou après un nombre prédéfini d'itérations. A ce moment-là on revient au problème principal et on essaie de générer de nouvelles $\mathfrak{q}$-tournées à coût réduit

négatif pour les insérer dans le problème-maitre et recommencer la résolution. Lorsqu'aucune de ces $\mathfrak{q}$-tournées n'a pu être générée, alors la solution trouvée est une borne inférieure de notre $m$-PTVP. La deuxième heuristique (H2) prend le relais après deux itérations consécutives de H1 sans générer de $\mathfrak{q}$-tournées à coût réduit négatif. Elle utilise des procédures similaires à celles de H1 mais génère des tournées élémentaires au lieu de $\mathfrak{q}$-tournées. Ses variables duales doivent être initialisées par H1, pour réduire ses besoins en temps de calcul et espace-mémoire, puisque la génération des tournées élémentaires de coût réduit négatif est un problème NP-complet (Garey et Johnson [53]). H1 est également à la base des règles de dominance utilisées au sein de H2 pour optimiser la génération de tournées élémentaires.

Cette approche est très efficace sur les problèmes de tournées de véhicules comme le montrent les articles de Baldacci et Mingozzi [11], Baldacci *et al.* [10] et Christofides *et al.* [24] car elle permet souvent d'obtenir de très bonnes bornes inférieures en un temps de calcul limité. L'adaptation au $m$-PTVP est toutefois loin d'être évidente. C'est pourquoi, après avoir présenté les formulations mathématiques et les relaxations utilisées, nous nous attardons dans ce chapitre sur les principes de base de l'application de la méthode au $m$-PTVP avant de détailler les différentes procédures et analyser les résultats.

## 5.2 Formulations mathématiques

Les heuristiques duales pour le $m$-PTVP s'appliquent sur le dual d'une relaxation du modèle de partitionnement (PP) déjà présenté au chapitre 3 et rappelé par les équations (5.1) à (5.5) ci-après. Rappelons que $\mathscr{R}$ est l'ensemble des tournées réalisables extractibles du graphe $G$ et $x_r^k$ la variable de décision égale à 1 si et seulement si la tournée $r \in \mathscr{R}$ de coût $c_r$ est utilisée au cours de la période $k$. Une tournée réalisable débute au dépôt, visite un sous-ensemble de clients une et une seule fois chacun avant de retourner au dépôt, avec une demande totale ou charge qui n'excède pas la capacité $Q$. Son coût est égal à la somme des coûts des arêtes qui la composent. Dans la formulation (PP), $\mathscr{R}_i$ désigne l'ensemble des tournées qui traversent le nœud $i \in V$ tandis que $\mathscr{R}(e)$ est l'ensemble des tournées qui traversent l'arête $e \in E$.

$$\text{(PP)} \quad \min \sum_{k \in \mathbb{M}} \sum_{r \in \mathscr{R}} c_r x_r^k \tag{5.1}$$

s. c.

$$\sum_{r \in \mathscr{R}_i} x_r^k = 1, \qquad \forall k \in \mathbb{M}, \forall i \in V' \tag{5.2}$$

$$\sum_{r \in \mathscr{R}} x_r^k \geq \lambda, \qquad \forall k \in \mathbb{M} \tag{5.3}$$

$$\sum_{r \in \mathscr{R}(e)} \sum_{k \in \mathbb{M}} x_r^k \leq 1, \qquad \forall e \in E \tag{5.4}$$

$$x_r^k \in \{0,1\}, \qquad \forall r \in \mathscr{R}, \forall k \in \mathbb{M} \tag{5.5}$$

La fonction-objectif minimise la somme des coûts des tournées utilisées. Les contraintes (5.2) imposent que chaque client soit servi par une seule tournée au cours de chaque période, tandis que les contraintes (5.3) imposent que les tournées choisies au cours de chaque période $k \in \mathbb{M}$ soient en nombre suffisant, c'est à dire supérieur ou égal à $\lambda$ calculé suivant l'équation (3.2). Les contraintes péripatétiques (5.4) empêchent les arêtes d'être utilisées plus d'une fois au cours des $m$ périodes. Enfin, les contraintes (5.5) définissent des variables binaires. Rappelons que cette formulation possède un nombre exponentiel de variables, ce qui la rend inaccessible aux solveurs.

La relaxation agrégée (RPP) de ce modèle est obtenue après agrégation sur $k \in \mathbb{M}$ des contraintes (5.2) à (5.5) du modèle (PP), puis changement de variable $y_r = \sum_{k \in \mathbb{M}} x_r^k$. La variable de décision $y_r$ est égale à 1 si la tournée $r \in \mathscr{R}$ est choisie au sein de la solution de (RPP), et 0 sinon.

$$\text{(RPP)} \quad \min \sum_{r \in \mathscr{R}} c_r y_r \tag{5.6}$$

s. c.

$$\sum_{r \in \mathscr{R}_i} y_r = m, \qquad \forall i \in V' \tag{5.7}$$

$$\sum_{r \in \mathscr{R}} y_r \geq m\lambda \tag{5.8}$$

$$\sum_{r \in \mathscr{R}(e)} y_r \leq 1, \qquad \forall e \in E \tag{5.9}$$

$$y_r \in \{0, 1\}, \qquad r \in \mathscr{R} \tag{5.10}$$

L'avantage de cette formulation est de diviser par $m$ le nombre de variables et de contraintes à gérer. L'inconvénient est d'obtenir une solution optimale dont le coût n'est qu'une borne inférieure pour le $m$-PTVP. Sa fonction-objectif minimise toujours la somme des coûts des tournées choisies. Les contraintes (5.7) imposent que chaque client soit servi par $m$ tournées au total, tandis que les contraintes (5.8) imposent que le nombre total de tournées choisies soit suffisant. Les contraintes péripatétiques (5.9) empêchent les arêtes d'être utilisées plus d'une fois au sein de l'ensemble des tournées choisies. Enfin, les contraintes (5.10) s'assurent que chaque variable soit binaire. Ce problème ainsi que sa relaxation linéaire restent difficiles à traiter y compris pour des instances de petites tailles à cause du nombre exponentiel de variables.

Le dual de la relaxation linéaire de (RPP) est (DRPP) illustré par les équations (5.11) à (5.15), où $V(r)$ est l'ensemble des noeuds de la tournée $r$ et $E(r)$ est l'ensemble des arêtes utilisées par la tournée $r$. Pour déduire (DRPP) de (RPP), en plus de la relaxation linéaire, nous associons les variables $u_i$ aux contraintes (5.7), $u_0$ à la contrainte (5.8) et $v_e$ aux contraintes (5.9).

$$\text{(DRPP)} \quad \max f(u, v) = \left( m \sum_{i \in V'} u_i + m\lambda u_0 + \sum_{e \in E} v_e \right) \tag{5.11}$$

s. c.

$$\sum_{i \in V(r)} u_i + \sum_{e \in E(r)} v_e \leq c_r, \qquad \forall r \in \mathscr{R} \tag{5.12}$$

$$u_i \in \mathbb{R}, \qquad \forall i \in V \tag{5.13}$$

$$u_0 \geq 0 \tag{5.14}$$

$$v_e \leq 0, \qquad \forall e \in E \tag{5.15}$$

La fonction-objectif (5.11) est la somme pondérée des variables duales. Les contraintes (5.12) imposent que la somme des variables duales des clients et des arêtes de n'importe quelle tournée n'excède pas le coût de cette tournée. Enfin, les contraintes (5.13) à (5.15) imposent tout simplement le domaine de définition de ces variables duales. Puisque (DRPP) est le dual de la relaxation linéaire de (RPP), le coût de toute solution de (DRPP) est une borne inférieure pour (RPP) et par conséquent pour la formulation initiale (PP) de notre $m$-PTVP. La méthode des $\mathfrak{q}$-tournées a pour but d'estimer au mieux les variables duales $u_i$ et $v_e$ pour maximiser $f(u, v)$ tout en respectant les contraintes (5.12) à (5.15). Le respect de ces contraintes garantit que les valeurs affectées aux variables duales forment une solution réalisable de (DRPP) et donc une borne inférieure du $m$-PTVP, tandis que la maximisation de $f(u, v)$ permet de s'approcher de l'optimum. Par contre, cette formulation n'est pas utilisable telle quelle à cause du nombre exponentiel de tournées $|\mathscr{R}|$.

Soit $V(r)$ l'ensemble des nœuds de la tournée $r$, $V'(r) = V(r) \setminus \{0\}$ et $E(r)$ l'ensemble de ses arêtes. Chaque variable $y_r$ peut être scindée en $|V'(r)|$ variables $\xi_r^i \in \{0, 1\}, i \in V'$ égales à 1 si et seulement si $i \in V'(r)$. Cette scission se fait suivant les équations (5.16) où $a_i(r)$ est le nombre de traversées du client $i$ par la tournée $r$ (donc $a_i(r) = 0$ ou 1) et $q_r = \sum_{i \in V'(r)} q_i = \sum_{i \in V'} a_i(r) q_i$ est la charge de la tournée $r$.

$$y_r = \sum_{i \in V'} a_i(r) \frac{q_i}{q_r} \xi_r^i \tag{5.16}$$

L'expression (5.16) impose que si $y_r = 1$ alors $\xi_r^i = 1, \forall i \in V'(r)$, de même si $y_r = 0$ alors $\xi_r^i = 0, \forall i \in V'(r)$. De plus, nous ajoutons au modèle résultant les contraintes $\sum_{r \in \mathscr{R}_i} \xi_r^i = m, \forall i \in V'$, pour préciser que chaque client $i \in V'$ doit être couvert par $m$ tournées et ainsi obtenir le modèle (RPP2).

$$\text{(RPP2)} \quad \min \sum_{r \in \mathscr{R}} c_r \left( \sum_{i \in V'} a_i(r) \frac{q_i}{q_r} \xi_r^i \right) \tag{5.17}$$

s. c.

$$\sum_{r\in\mathscr{R}} a_i(r)\left(\sum_{j\in V'} a_j(r)\frac{q_j}{q_r}\xi_r^j\right) = m, \qquad \forall i \in V' \tag{5.18}$$

$$\sum_{r\in\mathscr{R}}\left(\sum_{i\in V'} a_i(r)\frac{q_i}{q_r}\xi_r^i\right) \geq m\lambda \tag{5.19}$$

$$\sum_{r\in\mathscr{R}(e)}\left(\sum_{i\in V'} a_i(r)\frac{q_i}{q_r}\xi_r^i\right) \leq 1, \qquad \forall e \in E \tag{5.20}$$

$$\sum_{r\in\mathscr{R}_i} \xi_r^i = m, \qquad \forall i \in V' \tag{5.21}$$

$$\xi_r^i \in \{0,1\}, \qquad \forall i \in V', \forall r \in \mathscr{R} \tag{5.22}$$

Nous pouvons à présent dualiser les contraintes (5.18) à (5.20) en leur associant les pénalités respectives $\lambda_i \in \mathbb{R}, i \in V'$, $\lambda_0 \geq 0$ et $\mu_e \leq 0, e \in E$. Le modèle obtenu est (RPP$(\lambda,\mu)$).

$$(\text{RPP}(\lambda,\mu)) \quad \min \sum_{i\in V'}\left(\sum_{r\in\mathscr{R}_i}(c_r - \sum_{j\in V(r)}\lambda_j - \sum_{e\in E(r)}\mu_e)\frac{q_i}{q_r}\xi_r^i + m\lambda_i\right) + m\lambda\lambda_0 + \sum_{e\in E}\mu_e \tag{5.23}$$

s. c.

$$\sum_{r\in\mathscr{R}_i} \xi_r^i = m, \qquad \forall i \in V' \tag{5.24}$$

$$\xi_r^i \in \{0,1\}, \qquad i \in V', \forall r \in \mathscr{R} \tag{5.25}$$

En conséquence, le problème (RPP$(\lambda,\mu)$) peut être décomposé en $n$ sous-problèmes, un par client $i \in V'$, dont la solution optimale est donnée par les équations (5.26).

$$\xi_{r'}^i = \begin{cases} 1 & \text{si } \; r' \in \mathscr{R}_i^* \\ 0 & \text{sinon} \end{cases} \tag{5.26}$$

sachant que $\forall r' \in \mathscr{R}_i^*$ et $\forall r \in \mathscr{R}_i \setminus \mathscr{R}_i^*$ :

$$q_i(c_{r'} - \sum_{i\in V(r')}\lambda_i - \sum_{e\in E(r')}\mu_e)/q_{r'} \leq q_i(c_r - \sum_{i\in V(r)}\lambda_i - \sum_{e\in E(r)}\mu_e)/q_r \tag{5.27}$$

$$\mathscr{R}_i^* \subseteq \mathscr{R}_i \tag{5.28}$$

$$|\mathscr{R}_i^*| = m \tag{5.29}$$

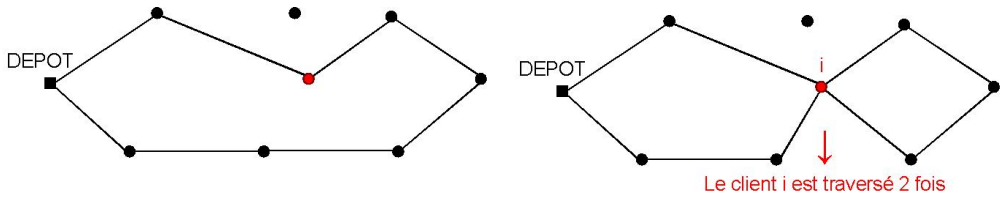
Le théorème 3 (page 98) montre comment construire une solution réalisable de (DRPP) à partir d'une solution optimale de (RPP$(\lambda,\mu)$). Toutefois, le nombre exponentiel de tournées $|\mathscr{R}|$ rend les problèmes (DRPP) et (RPP$(\lambda,\mu)$) difficiles à traiter, c'est pourquoi ils seront résolus à l'aide des heuristiques duales H1 et H2.

## 5.3 Principes de l'heuristique duale H1

L'efficacité de la première heuristique duale H1 pour notre $m$-PTVP repose sur trois idées-clés présentées ci-après : l'approximation de tournées par des $\mathfrak{q}$-tournées, la génération progressive de ces $\mathfrak{q}$-tournées et l'estimation progressive des meilleures valeurs des variables duales. Il s'agit de résultats combinant la théorie de la dualité, la relaxation Lagrangienne et le *dual ascent*.

### 5.3.1 Approximer $\mathscr{R}$ avec l'ensemble $\overline{\mathscr{R}}$ des $\mathfrak{q}$-tournées

Une tournée réalisable est définie comme un cycle qui débute et se termine au dépôt et dont la demande totale des clients visités ne dépasse pas la capacité $Q$ des véhicules. Ce cycle doit être élémentaire, puisque chaque client visité ne doit être traversé qu'une seule fois. Une $\mathfrak{q}$-tournée est une tournée pas forcément élémentaire obtenue lorsque cette contrainte est relaxée et que les cycles non élémentaires sont autorisés comme l'illustre la figure 5.1. Toute tournée élémentaire est donc également une $\mathfrak{q}$-tournée alors que l'inverse est faux. Le concept de $\mathfrak{q}$-tournées a été introduit dans l'article de Christofides *et al.* [23] sous le terme "$\mathfrak{q}$-*routes*" et utilisé comme substitut aux vraies tournées. Elles sont ainsi nommées parce que leur génération par une procédure pseudo-polynomiale est effectuée en fonction d'une demande totale souhaitée $q$. Leur efficacité résulte de la combinaison de deux éléments : (1) les $\mathfrak{q}$-tournées sont plus rapides à générer par programmation dynamique que les vraies tournées, (2) une pénalisation (lagrangienne) des violations des contraintes de degré ou nombre de traversées des clients élimine la plupart des traversées multiples de clients au sein des $\mathfrak{q}$-tournées qui tendent alors vers de vraies tournées.




(a) $\mathfrak{q}$-tournée élémentaire (b) $\mathfrak{q}$-tournée non élémentaire

FIGURE 5.1 – Exemples de $\mathfrak{q}$-tournées

Notons $\overline{(5.6)}$ à $\overline{(5.10)}$ les équations résultant de l'approximation de $\mathscr{R}$ par $\overline{\mathscr{R}}$ au sein des équations (5.6) à (5.10) et notons $(\overline{\text{DRPP}})$ le problème résultant de l'introduction des $\mathfrak{q}$-tournées dans le modèle (DRPP). Le coût de chacune des solutions de $(\overline{\text{DRPP}})$ est donc une borne inférieure de $(\overline{\text{RPP}})$ et donc du $m$-PTVP.

### 5.3.2 Résoudre ($\overline{\text{RPP}}$) par génération de $\mathfrak{q}$-tournées

Le modèle mathématique ($\overline{\text{RPP}}$) possède un nombre exponentiel de variables (ou colonnes). Le principe de la génération de colonnes est de résoudre de tels problèmes en ne gardant qu'un sous-ensemble limité de colonnes et en augmentant progressivement la taille de ce sous-ensemble pour atteindre une solution optimale du problème initial sans avoir généré toutes les colonnes. Il s'inspire du constat selon lequel à l'optimum, la plupart des variables sont hors base et très souvent la plupart d'entre elles sont nulles, c'est-à-dire qu'un (petit) sous-ensemble de variables peut être seul pris en compte pour résoudre le problème.

Notre heuristique duale H1 basée sur la génération de $\mathfrak{q}$-tournées consiste à résoudre à chaque itération $t$ le problème-maitre $(\overline{\text{DRPP}})^{(t)}$ obtenu avec un sous-ensemble de $\mathfrak{q}$-tournées $\overline{\mathscr{R}}^{(t)} \subset \overline{\mathscr{R}}$, puis générer des $\mathfrak{q}$-tournées à coût réduit négatif. Si aucune $\mathfrak{q}$-tournée à coût réduit négatif ne peut être générée, alors la solution trouvée est réalisable pour le problème initial ($\overline{\text{DRPP}}$), son coût est une borne inférieure valide de ($\overline{\text{RPP}}$) et donc de (PP). Sinon, les meilleures $\mathfrak{q}$-tournées à coût réduit négatif sont ajoutées à $\overline{\mathscr{R}}^{(t)}$ pour obtenir $\overline{\mathscr{R}}^{(t+1)}$ et le nouveau problème-maitre $(\overline{\text{DRPP}})^{(t+1)}$ est résolu dans le but de trouver une solution valide ou améliorer la précédente.

L'utilisation de $\mathfrak{q}$-tournées au lieu de tournées élémentaires est l'une des clés de l'efficacité de H1, car générer des tournées à coût réduit négatif constitue un problème NP-complet, comme l'ont rappelé Garey et Johnson [53].

### 5.3.3 Résoudre chaque problème-maitre sans utiliser de solveur

Chaque problème-maitre $(\overline{\text{DRPP}})^{(t)}$, identifié à l'itération $t$ et dont l'ensemble des $\mathfrak{q}$-tournées est $\overline{\mathscr{R}}^{(t)}$, n'est pas résolu à l'aide d'un solveur tel que Cplex®, mais plutôt avec une procédure dite de *dual ascent* qui consiste à estimer et affiner progressivement les valeurs des variables duales. Son application au $m$-PTVP est basée sur les équations (5.26) et les résultats du théorème 3.

Soit $V(e)$ les deux nœuds connectés par l'arête $e \in E$, soit $q_e = \sum_{i \in V(e) \setminus \{0\}} \frac{q_i}{2}$ la charge affectée à l'arête $e \in E$ et soit $\sigma_i$ l'ensemble des arêtes incidentes au nœud $i \in V$. L'idée sous-jacente au théorème 3 est de répartir le coût pénalisé $c_r(\lambda, \mu)$ de chaque $\mathfrak{q}$-tournée $r$ de manière proportionnelle sur chacune des arêtes $e \in E(r)$ qui la composent. La répartition proportionnelle est effectuée à l'aide du ratio des demandes ou ratio des charges $\frac{q_e}{q_r}$. Ainsi on peut dire que chaque arête $e \in E(r)$ contribue au coût $c_r(\lambda, \mu)$ de la $\mathfrak{q}$-tournée $r$ à la hauteur de $\frac{q_e}{q_r} c_r(\lambda, \mu)$. Etant donné un ensemble de $\mathfrak{q}$-tournées, il est donc possible d'estimer une contribution minimale de chaque arête $e$ égale à $b_e$ calculée par l'équation (5.36) et désigner $rb_e$ la $\mathfrak{q}$-tournée d'où provient cette contribution minimale. Conformément aux équations (5.35) et (5.34), il suffit alors d'affecter à chaque client $i$ la moyenne $\sigma_i$ des $m$ plus faibles contributions des arêtes incidentes à $i$ pour avoir une estimation de la participation par période du client $i$ au coût de la solution.

**Théorème 3** *Etant données des pénalités $\lambda_i$ affectées aux clients $i \in V'$, $\lambda_0$ affectée au dépôt et $\mu_e$ affectée aux arêtes $e \in E$, une solution réalisable de $(\overline{\text{DRPP}})^{(t)}$ dual de $(\overline{\text{RPP}})^{(t)}$, est donnée par les équations (5.30) à (5.32),*

$$u_0 = \lambda_0 \tag{5.30}$$

$$u_i = \frac{1}{m} \sum_{e \in B_i} b_e + \lambda_i, \qquad \forall i \in V' \tag{5.31}$$

$$v_e = \min \left\{ 0, b_e - \frac{1}{2} \sum_{i \in V(e)} \sigma_i \right\} + \mu_e, \qquad \forall e \in E \tag{5.32}$$

*dont les termes sont les suivants :*

- *les pénalités $\lambda_i$, $\lambda_0$ et $\mu_e$, qui peuvent être considérées comme étant associées aux relaxations lagrangiennes des contraintes $\overline{(5.7)}$, $\overline{(5.8)}$ et $\overline{(5.9)}$ comme illustré par le modèle (RPP($\lambda, \mu$))*

$$\lambda_i \in \mathbb{R} \quad , \lambda_0 \geq 0 \quad \text{et} \quad \mu_e \leq 0 \tag{5.33}$$

- *l'estimation de la contribution $\sigma_i$ du client $i \in V'$ au coût d'une période de la solution*

$$\sigma_i = \frac{1}{m} \sum_{e \in B_i} b_e \quad \text{et} \quad \sigma_0 = 0 \tag{5.34}$$

- *le sous-ensemble $B_i$ des $m$ arêtes de plus petite contribution et incidentes au client $i \in V'$*

$$\max_{e \in B_i} b_e \leq \min_{e \in \delta_i \setminus B_i} b_e \quad \text{et} \quad |B_i| = m \tag{5.35}$$

- *l'estimation de la contribution $b_e$ de l'arête $e \in E$ au coût de la solution courante et la $\mathfrak{q}$-tournée associée $rb_e$*

$$b_e = q_e \frac{c_{rb_e}(\lambda, \mu)}{q_{rb_e}} \quad \text{et} \quad \frac{c_{rb_e}(\lambda, \mu)}{q_{rb_e}} = \min_{r \in \overline{\mathscr{R}}^{(t)}(e)} \left\{ \frac{c_r(\lambda, \mu)}{q_r} \right\} \tag{5.36}$$

- *le coût pénalisé $c_r(\lambda, \mu)$ de chaque $\mathfrak{q}$-tournée $r \in \overline{\mathscr{R}}^{(t)}$*

$$c_r(\lambda, \mu) = c_r - \sum_{i \in V(r)} \lambda_i - \sum_{e \in E(r)} \mu_e \tag{5.37}$$

- *la charge $q_r$ de chaque $\mathfrak{q}$-tournée $r \in \overline{\mathscr{R}}^{(t)}$*

$$q_r = \sum_{i \in V'(r)} q_i \tag{5.38}$$

**Preuve** La preuve de ce théorème consiste à démontrer que la contrainte $\overline{(5.12)}$ est toujours vérifiée si les variables duales sont définies selon les équations (5.30) à (5.32), car les autres contraintes $\overline{(5.13)}$ à $\overline{(5.15)}$ sont implicitement satisfaites. A partir des équations (5.30) à (5.32), pour chaque $\mathfrak{q}$-tournée $r$ le terme de gauche de l'équation $\overline{(5.12)}$ devient :

$$\sum_{i\in V(r)} u_i + \sum_{e\in E(r)} v_e = \sum_{i\in V'(r)} \sigma_i + \sum_{i\in V(r)} \lambda_i + \sum_{e\in E(r)} \mu_e \tag{5.39}$$

Définissons $\widetilde{E}(r)$ comme le sous-ensemble des arêtes $e = (i_e, j_e)$ de la $\mathfrak{q}$-tournée $r \in \overline{\mathscr{R}}^{(t)}$ qui vérifient $\frac{1}{2}(\sigma_{i_e} + \sigma_{j_e}) > b_e$. Pour chaque $\mathfrak{q}$-tournée $r \in \overline{\mathscr{R}}^{(t)}$, on distingue donc deux cas à étudier, selon que $\widetilde{E}(r)$ soit vide ou pas.

– Si $\widetilde{E}(r) = \emptyset$ :

Dans ce cas, d'après l'équation (5.32) on obtient l'égalité (5.40) et cette $\mathfrak{q}$-tournée $r$ vérifie l'inégalité (5.41).

$$v_e = \mu_e, \qquad \forall e \in E_r \tag{5.40}$$

$$\sum_{i\in V'(r)} \sigma_i = \sum_{e=(i_e,j_e)\in E(r)} \frac{1}{2}(\sigma_{i_e} + \sigma_{j_e}) \leq \sum_{e\in E(r)} b_e \tag{5.41}$$

A partir de l'équation (5.36), on en déduit :

$$b_e \leq q_e \frac{c_r(\lambda,\mu)}{q_r}, \qquad \forall e \in E(r) \tag{5.42}$$

Puisque $\sum_{e\in E(r)} q_e = q_r$, à partir des inégalités (5.41) et (5.42), on en déduit :

$$\sum_{i\in V'(r)} \sigma_i \leq \sum_{e\in E(r)} q_e \frac{c_r(\lambda,\mu)}{q_r} = c_r(\lambda,\mu) \tag{5.43}$$

On peut donc déduire l'inégalité (5.44) ci-après à partir des équations (5.37), (5.39), (5.40) et (5.43) ; ce qui permet de conclure que la solution définie à l'itération $t$ par les équations (5.30) à (5.32) respecte la contrainte $\overline{(5.12)}$ quel que soit la $\mathfrak{q}$-tournée $r \in \overline{\mathscr{R}}^{(t)}$ qui vérifie $\widetilde{E}(r) = \emptyset$

$$\sum_{i\in V(r)} u_i + \sum_{e\in E(r)} v_e \leq c_r(\lambda,\mu) + \sum_{i\in V(r)} \lambda_i + \sum_{e\in E(r)} \mu_e = c_r \tag{5.44}$$

– Si $\widetilde{E}(r) \neq \emptyset$ :

Rappelons que $\frac{1}{2}(\sigma_{i_e} + \sigma_{j_e}) > b_e$ pour toute arête $\forall e = (i_e, j_e) \in \widetilde{E}(r)$. A partir de (5.32), on obtient :

$$\sum_{e\in E(r)} v_e = \sum_{e\in E(r)} \mu_e + \sum_{e\in \widetilde{E}(r)} b_e - \sum_{e\in \widetilde{E}(r)} \frac{1}{2}(\sigma_{i_e} + \sigma_{j_e}) \tag{5.45}$$

De plus,

$$\sum_{i\in V'(r)} \sigma_i = \sum_{e\in E(r)\setminus\tilde{E}(r)} \frac{1}{2}(\sigma_{i_e}+\sigma_{j_e}) + \sum_{e\in\tilde{E}(r)} \frac{1}{2}(\sigma_{i_e}+\sigma_{j_e}) \tag{5.46}$$
$$\leq \sum_{e\in E(r)\setminus\tilde{E}(r)} b_e + \sum_{e\in\tilde{E}(r)} \frac{1}{2}(\sigma_{i_e}+\sigma_{j_e})$$

A partir des équations (5.39), (5.45) et (5.46) on obtient donc l'inégalité (5.47).

$$\sum_{i\in V(r)} u_i + \sum_{e\in E(r)} v_e \leq \sum_{e\in E(r)} b_e + \sum_{i\in V(r)} \lambda_i + \sum_{e\in E(r)} \mu_e \tag{5.47}$$

En combinant les inégalités (5.47) et (5.42), on en déduit l'équation (5.48) qui montre que la solution définie à l'itération $t$ par les équations (5.30) à (5.32) respecte la contrainte $\overline{(5.12)}$ quelle que soit la $\mathfrak{q}$-tournée $r \in \overline{\mathscr{R}}^{(t)}$ qui vérifie $\tilde{E}(r) \neq \emptyset$.

$$\sum_{i\in V(r)} u_i + \sum_{e\in E(r)} v_e \leq c_r(\lambda,\mu) + \sum_{i\in V(r)} \lambda_i + \sum_{e\in E(r)} \mu_e = c_r \tag{5.48}$$

L'étude de ces deux cas démontre que les équations (5.30) à (5.32) permettent effectivement de construire une solution réalisable de $(\overline{\text{DRPP}})^{(t)}$, puisque n'importe quelle $\mathfrak{q}$-tournée $r \in \overline{\mathscr{R}}^{(t)}$ a un coût supérieur ou égal à la somme des variables duales de ses clients et de ses arêtes. La preuve de la validité du théorème 3 ayant été faite, la résolution de chaque problème-maitre correspond à l'optimisation des valeurs affectées aux pénalités $\lambda_i$ et $\mu_e$. La procédure d'optimisation de ces pénalités sera présentée dans la section 5.5.

Le théorème 3 constitue donc la base de notre méthode, car c'est grâce à lui que nous pouvons concevoir un algorithme de résolution qui a le double avantage de ne pas nécessiter de solveur et d'affiner progressivement les valeurs des variables duales, ce qui garantit l'efficacité mais aussi la stabilité de l'heuristique H1.

## 5.4 Génération des meilleures $\mathfrak{q}$-tournées

A chaque itération $t$, les $\mathfrak{q}$-tournées $r \notin \overline{\mathscr{R}}^{(t)}$ de meilleur coût réduit $\tilde{c}_r$, calculé par l'équation (5.49) ci-après, sont obtenues après fusions des paires de meilleurs $\mathfrak{q}$-chemins générés par programmation dynamique. On appelle $\mathfrak{q}$-chemin de charge $q$ un chemin pas forcément élémentaire du dépôt à un client et dont la somme des demandes des clients visités est égale à $q \leq Q$. La fusion de deux $\mathfrak{q}$-chemins permet donc d'obtenir une $\mathfrak{q}$-tournée, cycle qui se débute et se termine au dépôt.

$$\tilde{c}_{e=(i_e,j_e)} = c_e - \frac{1}{2}u_{i_e} - \frac{1}{2}u_{j_e} - v_e,\ \forall e \in E \quad \text{et} \quad \tilde{c}_r = \sum_{e\in E(r)} \tilde{c}_e,\ \forall r \in \overline{\mathscr{R}} \tag{5.49}$$

Seules les $\mathfrak{q}$-tournées $r$ qui vérifient $\tilde{c}_r \leq \gamma$ peuvent être rajoutées à $\overline{\mathscr{R}}^{(t)}$ pour obtenir $\overline{\mathscr{R}}^{(t+1)}$ sachant que $\gamma$ est une valeur-seuil égale à $\infty$ lors de l'initialisation ($t = 0$), puis égale à 0 dès que $t \geq 1$, de manière à n'insérer dans $\overline{\mathscr{R}}^{(t\geq 1)}$ que des $\mathfrak{q}$-tournées à coût réduit négatif. Un client traversé plus d'une fois par un même $\mathfrak{q}$-chemin voit sa demande $q_i$ être comptabilisée autant de fois dans la charge totale du $\mathfrak{q}$-chemin. Ainsi il existe toujours une tournée élémentaire capable de servir les mêmes clients que ceux d'une $\mathfrak{q}$-tournée quelconque, puisque la somme de leurs demandes n'excède jamais $Q$.

### 5.4.1 Génération de $\mathfrak{q}$-chemins

Soient $\phi^1_{iq}$ et $\phi^2_{iq}$ respectivement le meilleur $\mathfrak{q}$-chemin et le deuxième meilleur $\mathfrak{q}$-chemin qui se terminent chez le client $i$, chacun avec une demande totale ou charge $q$. Dans les paragraphes qui suivent nous utiliserons le terme "$\mathfrak{q}$-chemin de type 1" pour le premier et "$\mathfrak{q}$-chemin de type 2" pour le deuxième. $f(\phi^1_{iq})$ est le coût de $\phi^1_{iq}$ égal à la somme des coûts des arêtes utilisées, tandis que $\pi(\phi^1_{iq})$ précise le prédécesseur de $i$ dans ce $\mathfrak{q}$-chemin de type 1. De même, $f(\phi^2_{iq})$ est le coût de $\phi^2_{iq}$ égal à la somme des coûts des arêtes utilisées, tandis que $\pi(\phi^2_{iq})$ précise le prédécesseur de $i$ dans ce $\mathfrak{q}$-chemin de type 2.

Après l'initialisation au cours de laquelle $f(\phi^1_{iq}) = \infty$ quel que soit $i \in V'$ et $q \in [1, Q]$, les coûts des $\mathfrak{q}$-chemins de type 1 sont obtenus par programmation dynamique à l'aide des formules de récurrence (5.50) et (5.51). L'algorithme pseudo-polynomial résultant contient deux boucles imbriquées, la première faisant varier $q$ de $q_i+1$ à $Q$ alors que dans la seconde $i$ varie de 1 à $n$. En effet, le meilleur $\mathfrak{q}$-chemin d'extrémité $i$ et de charge $q$ résulte soit d'une connection directe entre $i$ et le dépôt si $q = q_i$, soit de l'ajout de $i$ à la fin de l'un des meilleurs $\mathfrak{q}$-chemins de charge $q - q_i$, s'ils existent. Le prédécesseur de $i$ dans le nouveau $\mathfrak{q}$-chemin est stocké dans la variable $\pi(\phi^1_{iq})$.

$$f(\phi^1_{iq_i}) = \tilde{c}_{0i}, \qquad \forall i \in V' \tag{5.50}$$

$$f(\phi^1_{iq}) = \min_{\forall j \in V} \{f(\phi^1_{j,q-q_i}) + \tilde{c}_{ji}\}, \qquad \forall i \in V', \forall q \in [q_i + 1, Q] \tag{5.51}$$

Une fois les $\mathfrak{q}$-chemins de type 1 trouvés pour tous les couples "$q \in [1, Q], i \in V'$", ceux de type 2 sont recherchés avec une procédure similaire à la précédente : l'initialisation stipule que $f(\phi^2_{iq}) = \infty$ quel que soit $i \in V'$ et $q \in [1, Q]$ avant l'application de la programmation dynamique basée sur les formules de récurrence (5.52) et (5.53) où $i$ varie de 1 à $n$ et $q$ varie de $q_i + 1$ à $Q$. Ces formules sont très similaires aux formules (5.50) et (5.51), sauf que l'on s'assure en plus qu'aucun $\mathfrak{q}$-chemin de type 2 ne soit une copie d'un $\mathfrak{q}$-chemin de type 1. Pour cela, tout nouveau $\mathfrak{q}$-chemin $\phi^2_{iq}$ ne peut résulter de l'ajout de $i$ à la fin du meilleur $\mathfrak{q}$-chemin de charge $q - q_i$ se terminant au nœud $j \in V$ que si $j \neq \pi(\phi^1_{iq})$. Dans le cas contraire, $i$ ne peut être ajouté qu'au deuxième meilleur $\mathfrak{q}$-chemin se terminant en $j$. L'option choisie est mémorisée au sein de la variable $\tau(\phi^2_{iq})$, égale à 1 dans le premier cas

et 2 dans le second, conformément à la formule (5.54).

$$f(\phi^2_{iq_i}) = \tilde{c}_{0i}, \qquad \forall i \in V' \tag{5.52}$$

$$f(\phi^2_{iq}) = \min_{\forall j \in V} \begin{cases} f(\phi^1_{j,q-q_i}) + \tilde{c}_{ji} & \text{si } j \neq \pi(\phi^1_{iq}) \\ f(\phi^2_{j,q-q_i}) + \tilde{c}_{ji} & \text{sinon} \end{cases}, \qquad \forall i \in V', \forall q \in [q_i+1, Q] \tag{5.53}$$

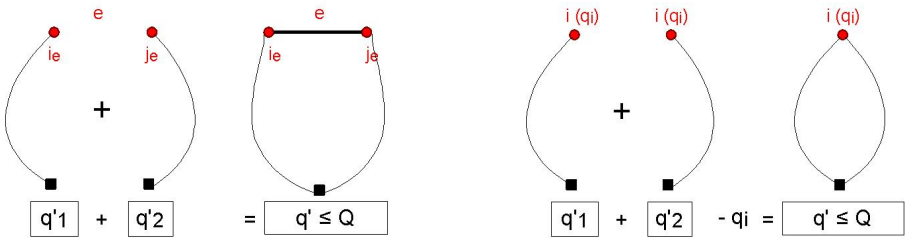
$$\tau(\phi^2_{iq}) = \begin{cases} 1 & \text{si } \pi(\phi^1_{iq}) \neq \pi(\phi^2_{iq}) \\ 2 & \text{sinon} \end{cases}, \qquad \forall i \in V', \forall q \in [q_i+1, Q] \tag{5.54}$$

La complexité de la génération de $\mathfrak{q}$-chemins est de l'ordre de $O(Qn(n+1)) = O(Qn^2)$ puisque les tests sont effectués $\forall q \in [q_i+1, Q]$, $\forall i \in V'$ et $\forall j \in V$. En pratique, à l'exception du cas où toutes les demandes sont identiques, comme pour le $m$-PVP par exemple, les clients pouvant être atteints par un $\mathfrak{q}$-chemin de charge $q \in [1, Q]$ sont en nombre restreint. Pour accélérer les calculs, il est donc utile d'implémenter un vecteur $InQ_q$ affecté à chacune des valeurs potentielles de $q$ (nombres entiers entre 1 et $Q$) et destiné à stocker la liste des clients qui ont pu être atteints avec par un ou plusieurs $\mathfrak{q}$-chemins de charge $q$. Seuls ces clients seront pris en compte comme prédécesseurs potentiels de $i$ lorsqu'il faudra prolonger un $\mathfrak{q}$-chemin de charge $q$ pour atteindre $i$ avec une charge $q + q_i$. Si pour chaque valeur de $q$ au maximum $\overline{n}$ clients peuvent être atteints, cela réduit la complexité de la procédure à $O(Qn\overline{n})$ au lieu de $O(Qn^2)$, ce qui produit une amélioration significative dès lors que $\overline{n} << n$.

N'importe quel $\mathfrak{q}$-chemin jusque-là généré implicitement peut être reconstitué en remontant la liste des prédécesseurs. S'il s'agit d'un $\mathfrak{q}$-chemin de type 2, il faut utiliser $\tau(\phi)$ pour retrouver les clients qui le constituent, sous la forme $(i, \pi(\phi^2_{iq}), \pi(\phi^{\tau(\phi^2_{iq})}_{\pi(\phi^2_{iq}),q-q_i}), \ldots, 0)$. En revanche, la reconstitution d'un $\mathfrak{q}$-chemin de type 1 est plus simple, puisqu'il ne résulte que du prolongement de $\mathfrak{q}$-chemins de type 1. Le résultat obtenu est $(i, \pi(\phi^1_{iq}), \pi(\phi^1_{\pi(\phi^1_{iq}),q-q_i}), \ldots, 0)$.

### 5.4.2 Fusion de paires de $\mathfrak{q}$-chemins

Les $\mathfrak{q}$-tournées sont obtenues en fusionnant les $\mathfrak{q}$-chemins deux par deux. Nous avons dans un premier temps utilisé une fusion *par arêtes*, illustrée par la figure 5.2(a). Elle permet de déduire les meilleures $\mathfrak{q}$-tournées qui traversent les arêtes $e \in E$ et de charge totale $q$. Utilisée dans la première version de notre procédure, elle aurait dû permettre d'obtenir une variété de $\mathfrak{q}$-tournées de coût minimum, mais le nombre important de $\mathfrak{q}$-tournées ainsi obtenues a conduit à des temps de calculs dissuasifs. La deuxième version de notre procédure a donc délaissé ce type de fusion pour utiliser une fusion *par clients*, illustrée par la figure 5.2(b).




(a) Fusion par l'arête $e$ (b) Fusion par le client $i$

FIGURE 5.2 – Fusion d'une paire de $\mathfrak{q}$-chemins pour obtenir une $\mathfrak{q}$-tournée

### Version 1 : fusion par les arêtes

Pour chaque arête $e = (i_e, j_e) \in E$, la meilleure $\mathfrak{q}$-tournée $\Phi_{e=(i_e,j_e),q}$ qui le traverse avec une charge totale $q$ résulte de la fusion de la meilleure paire de $\mathfrak{q}$-chemins qui atteignent $i_e$ et $j_e$ avec des charges respectives $q_1$ et $q_2$ telles que $q = q_1 + q_2$. Son coût est donc obtenu par l'équation (5.55) et il nécessite la génération préalable de l'ensemble des meilleurs $\mathfrak{q}$-chemins suivant la procédure déjà décrite plus haut. La condition $\pi(\phi^1_{i_e,q_1}) \neq \pi(\phi^1_{j_e,q_2})$ permet d'éviter de créer une boucle à l'extrémité de la tournée générée.

$$f(\Phi_{eq}) = \min_{q_{i_e} \leq q_1 \leq q - q_{j_e}} \begin{cases} f(\phi^1_{i_e,q_1}) + f(\phi^1_{j_e,q_2}) + c_{i_e,j_e} & \text{si } \pi(\phi^1_{i_e,q_1}) \neq \pi(\phi^1_{j_e,q_2}) \\ c_{i_e,j_e} + \min \begin{cases} f(\phi^1_{i_e,q_1}) + f(\phi^2_{j_e,q_2}) \\ f(\phi^2_{i_e,q_1}) + f(\phi^1_{j_e,q_2}) \end{cases} & \text{sinon} \end{cases} \tag{5.55}$$

La meilleure $\mathfrak{q}$-tournée n'est pas générée explicitement, mais ses caractéristiques $\gamma^1(\Phi_{eq})$, $\gamma^2(\Phi_{eq})$ et $q^1(\Phi_{eq})$ sont sauvegardées. Les deux premières identifient les types respectifs des $\mathfrak{q}$-chemins de la paire assemblée alors que la troisième est la charge du premier $\mathfrak{q}$-chemin. Il est évident que la charge du deuxième $\mathfrak{q}$-chemin est égale à $q - q^1(\Phi_{e,q})$. Ces caractéristiques permettront de reconstituer la $\mathfrak{q}$-tournée $(\Phi_{e,q})$ en reconstituant et concaténant les deux $\mathfrak{q}$-chemins qui la composent. Dans le pire des cas, le nombre de $\mathfrak{q}$-tournées identifiées est de l'ordre de $O(Q|E|) = O(Qn^2)$.

### Version 2 : fusion par les clients

La deuxième version de la procédure d'assemblage des $\mathfrak{q}$-chemins vise à réduire le nombre d'opérations effectuées. Pour cela, le principe est de générer les meilleures tournées qui traversent chaque client $i$, au lieu des arêtes $e$. Il s'agit de trouver la paire de $\mathfrak{q}$-chemins qui se termine en $i$ et dont la somme des charges est $q \in [1, Q]$, et ce à l'aide de la formule de récurrence (5.56). Les caractéristiques $\gamma^1(\Phi_{iq})$, $\gamma^2(\Phi_{iq})$ et $q^1(\Phi_{iq})$ sont également sauvegardées pour pouvoir reconstituer la $\mathfrak{q}$-tournée lorsque nécessaire. La condition $\pi(\phi^1_{i,q_1}) \neq \pi(\phi^1_{i,q_2=q-q_1+q_i})$ permet d'éviter de créer une boucle à l'extrémité de la tournée

générée.

$$\Phi_{i,q} = \min_{q_i \leq q_1 \leq \frac{q+q_i}{2}} \begin{cases} f(\phi^1_{i,q_1}) + f(\phi^1_{i,q_2=q-q_1+q_i}) & \text{si } \pi(\phi^1_{i,q_1}) \neq \pi(\phi^1_{i,q_2=q-q_1+q_i}) \\ \min \begin{cases} f(\phi^1_{i,q_1}) + f(\phi^2_{i,q_2=q-q_1+q_i}) \\ f(\phi^2_{i,q_1}) + f(\phi^1_{i,q_2=q-q_1+q_i}) \end{cases} & \text{sinon} \end{cases} \tag{5.56}$$

Le principal avantage de ce type de fusion est la réduction d'un facteur $n$ du nombre de $\mathfrak{q}$-tournées générées, désormais de l'ordre de $O(Q|V'|) = O(Qn)$. De plus, la symétrie inhérente à cette fusion permet d'effectuer deux fois moins de tests ou comparaisons, car les valeurs de $q_1$ et $q_2$ peuvent être interverties sans effet sur la $\mathfrak{q}$-tournée générée.

### 5.4.3 Ajout des meilleures $\mathfrak{q}$-tournées dans $\overline{\mathscr{R}}^{(t)}$

A l'itération $t$, suite à l'exécution de l'algorithme de génération et de fusion des $\mathfrak{q}$-chemins, pour chaque client $i$ et pour chaque quantité $q$, la meilleure tournée $\Phi_{iq}$ qui traverse $i$ avec une charge totale $q$ est désormais connue. Pour obtenir $\overline{\mathscr{R}}^{(t+1)}$, nous ajoutons donc à $\overline{\mathscr{R}}^{(t)}$ les $\Delta_{\max}$ $\mathfrak{q}$-tournées $r \notin \overline{\mathscr{R}}^{(t)}$ aux coûts réduits les plus faibles et qui vérifient $\tilde{c}_r \leq \gamma$. Rappelons que $\gamma$ est une valeur-seuil égale à 0 excepté lors de l'initialisation où elle vaut alors $\gamma^0 = \infty$ tandis que $\Delta_{\max}$ vaut $\Delta^0_{\max}$. Ainsi, si un client $i \in V'$ ne possède que $\Delta < \Delta_{\max}$ $\mathfrak{q}$-tournées $q^1(\Phi_{iq})$ qui respectent le seuil $\gamma$, alors seules ces $\Delta$ $\mathfrak{q}$-tournées sont insérées dans $\overline{\mathscr{R}}^{(t+1)}$. Dans le cas contraire, seules les $\Delta_{\max}$ meilleures sont insérées. Une fois $\overline{\mathscr{R}}^{(t+1)}$ finalisée, une solution du problème $(\overline{\text{DRPP}})^{(t+1)}$ résultant est trouvée grâce au théorème 3 et améliorée avec la mise à jour lagrangienne des pénalités décrite dans la section 5.5.

Si quel que soit le client $i \in V'$ il n'existe aucune $\mathfrak{q}$-tournée $r \notin \overline{\mathscr{R}}^{(t)}$ qui vérifie $\tilde{c}_r \leq \gamma$, alors $\overline{\mathscr{R}}^{(t+1)} = \overline{\mathscr{R}}^{(t)}$ et $(\overline{\text{DRPP}})^{(t+1)} = (\overline{\text{DRPP}})^{(t)}$. La solution courante de $(\overline{\text{DRPP}})^{(t)}$ constitue une solution réalisable du problème initial $(\overline{\text{DRPP}})$ et sa valeur $f(u, v)$ est alors une borne inférieure de notre $m$-PTVP. Cette dernière est donc sauvegardée, puis la procédure d'optimisation de solutions de problèmes-maitres du théorème 3 et paragraphe 5.5 est appliquée à $(\overline{\text{DRPP}})^{(t+1)}$ dans l'optique d'améliorer $f(u, v)$. La nouvelle solution obtenue peut toutefois ne pas être une solution réalisable du problème initial $(\overline{\text{DRPP}})$, dans ce cas de nouvelles $\mathfrak{q}$-tournées $r \notin \overline{\mathscr{R}}^{(t+1)}$ à coûts réduits négatifs seront trouvées.

L'algorithme général s'arrête après deux itérations successives sans ajout de $\mathfrak{q}$-tournée à coût réduit négatif, ou alors si le nombre maximum d'itérations $t_{\max}$ est atteint.

## 5.5 Mise à jour lagrangienne des pénalités $\lambda_i$ et $\mu_e$

Rappelons qu'à chaque itération $t$, quelles que soient des pénalités $\lambda_i \in \mathbb{R}$ affectées aux clients $i \in V'$, $\lambda_0 \geq 0$ affectée au dépôt et $\mu_e \leq 0$ affectées aux arêtes $e \in E$, grâce au théorème 3 (page 98) nous sommes désormais capables d'en déduire une solution

réalisable du problème-maitre $(\overline{\text{DRPP}})^{(t)}$. La mise à jour lagrangienne de nos pénalités a donc pour but de corriger progressivement leurs valeurs, initialement nulles, afin d'améliorer la solution qui en sera déduite et la rendre réalisable pour le $(\overline{\text{DRPP}})$ initial de manière à obtenir une borne inférieure du $m$-PTVP. Cette correction s'effectue à travers une méthode de sous-gradient qui tend à minimiser les violations des contraintes de degré des clients et les contraintes de péripatéticité sur les arêtes.

Les contraintes de degré des clients stipulent que chaque client doit être visité exactement $m$ fois au total et donc être traversé par $m$ tournées. Les contraintes de péripatéticité imposent qu'aucune arête ne soit utilisée plus d'une fois. La mise à jour lagrangienne nécessite donc d'identifier le degré $deg_i$ de chaque client $i$ ou $deg_0$ du dépôt, défini comme le nombre de fois où ils sont traversés, ainsi que le nombre d'utilisations $slk_e$ de chaque arête $e$ au sein de la solution déduite des vecteurs de pénalités $\lambda$ et $\mu$.

### 5.5.1 Evaluation de $deg_i$ et $slk_e$

Pour calculer les degrés $deg_i$ des clients $i \in V'$ et les nombres d'utilisation $slk_e$ des arêtes $e \in E$ au sein la solution courante, il suffit donc de retrouver les $\mathfrak{q}$-tournées dont une fraction du coût a été pris en compte pour l'évaluation des $\sigma_i$, et de cumuler les fractions de degrés des clients et de nombres d'utilisations des arêtes qui les constituent. Soit $a_i(r)$ le nombre de fois que le client $i$ est traversé par la $\mathfrak{q}$-tournée $r$ et soit $d_e(rb_e)$ le nombre de fois que l'arête $e$ est utilisée par la $\mathfrak{q}$-tournée $r$, l'évaluation des valeurs de $deg_i$ et $slk_e$ au sein de la solution déduite des vecteurs pénalités $\lambda$ et $\mu$ par le théorème 3 peut être formalisée par les équations (5.57) et (5.58).

$$deg_i = \sum_{j \in V'} \sum_{e \in B_j} \frac{q_e}{q_r} a_i(rb_e) \tag{5.57}$$

$$slk_e = \sum_{j \in V'} \sum_{e \in B_j} \frac{q_e}{q_r} d_e(rb_e) \tag{5.58}$$

### 5.5.2 Mise à jour de $\lambda_i$ et $\mu_e$ en fonction de $deg_i$ et $slk_e$

Soit $UB$ la meilleure borne supérieure fournie et $\epsilon$ un paramètre. A chaque itération $h$ de la méthode de sous-gradients qui optimise les vecteurs de pénalités, leurs valeurs courantes $\lambda^{(h)}$ et $\mu^{(h)}$ sont mises à jour à l'aide des équations (5.59), (5.60) et (5.61). Les nouvelles valeurs $\lambda^{(h+1)}$ et $\mu^{(h+1)}$ servent ensuite à construire une nouvelle solution de $(\overline{\text{DRPP}})^{(t)}$ au cours l'étape $h+1$, à partir de laquelle seront calculés les nouveaux degrés $deg_i$ et nombres d'utilisations $slk_e$ pour en déduire $\lambda^{(h+2)}$ et $\mu^{(h+2)}$. Par conséquent, au fur à mesure des étapes, la solution proposée tend à mieux respecter les contraintes de degré des clients ainsi que le nombre maximum d'utilisation des arêtes.

$$\lambda_i^{(t+1)} = \lambda_i^{(t)} - \epsilon.step.(deg_i - m), \qquad \forall i \in V' \tag{5.59}$$

$$\lambda_0^{(t+1)} = \max\{\lambda_0^{(t)} - \epsilon.step.(deg_0 - m), 0\} \tag{5.60}$$

$$\mu_e^{(t+1)} = \min\{0, \mu_e^{(t)} - \epsilon.step.(slk_e - 1)\}, \qquad \forall e \in E \tag{5.61}$$

où

$$step = \frac{UB - f(u,v)}{\sqrt{\sum_{i \in V}(deg_i - m)^2 + \sum_{e \in E}(slk_e - 1)^2}} \tag{5.62}$$

Le paramètre $\epsilon$ est initialisé à $\epsilon_0$ et réduit de 25% toutes les $h^\epsilon$ étapes sans amélioration de la meilleure borne inférieure trouvée jusque là. A chaque itération $t$, l'optimisation par la méthode de sous-gradients de la solution de $(\overline{\text{DRPP}})^{(t)}$ s'arrête dès que le nombre maximum d'étapes est atteint ou alors qu'il y a convergence, c'est à dire dès que $UB = f(u,v)$, ou $deg_i = m, \forall i \in V'$. Le nombre maximum d'étapes est $\max\{h_{\max}, \theta h_{\max+}\}$ où $h_{\max}$ et $h_{\max+}$ sont des paramètres prédéfinis et $\theta$ est une variable binaire égale à 1 si la meilleure solution trouvée lors de l'itération $t-1$ a produit une borne inférieure au $m$-PTVP et 0 sinon. L'algorithme résultant est capable de prolonger l'optimisation lorsqu'une borne inférieure de $m$-PTVP est trouvée et limiter les étapes sinon.

## 5.6 Récapitulatif de l'heuristique duale H1

L'algorithme 3 récapitule la structure de notre heuristique duale H1 de calcul de bornes inférieures basée sur la génération des $\mathfrak{q}$-tournées appliquée au $m$-PTVP. L'initialisation consiste à mettre à zéro les valeurs des pénalités $\lambda_i$ et $\mu_e$, ainsi que les variables duales $u_i$ et $v_e$, puis initialiser l'ensemble des $\mathfrak{q}$-tournées avec les $\Delta^0_{\max}$ meilleures $\mathfrak{q}$-tournées incidentes à chaque client $i \in V'$. Les procédures de résolution des problèmes-maitres (lignes 5 à 9) ont été présentées dans les sections 5.4 et 5.5. La génération des $\mathfrak{q}$-tournées de meilleurs coûts réduits (lignes 10 à 11) a été décrite dans la section 5.3. Si aucune $\mathfrak{q}$-tournée à coût réduit négatif n'est trouvée à l'itération $t$, alors les variables duales courantes forment une solution réalisable de $(\overline{\text{DRPP}})$ initial, le coût associé $f(u,v)$ est une borne inférieure valide de notre $m$-PTVP et peut être sauvegardé s'il dépasse la meilleure borne inférieure trouvée jusque-là. La procédure s'arrête après deux itérations consécutives sans ajout de nouvelles $\mathfrak{q}$-tournées dans le problème-maitre.

## 5.7 Génération de tournées élémentaires pour H2

Il est possible d'améliorer la borne inférieure obtenue à la fin de l'heuristique H1 en générant des tournées non élémentaires à partir des variables duales qui ont produit cette borne inférieure. L'heuristique duale H1+H2 résultant exécute H1 une première fois, puis

**Algorithm 3** Heuristique duale H1

*ImporterDonnées*$(G, Q, t_{\max}, h_{\max}, h_{\max+})$ ;
$\theta := 0$ ; $t := 0$ ; $ok := 0$ ; $LB := 0$ ; $\overline{\mathscr{R}} := \overline{\mathscr{R}}_{init}$
$\forall i \in V$, $\lambda_i := 0$ et $u_i := 0$ ; $\forall e \in E$, $\mu_e := 0$ et $v_e := 0$
**répéter**
  **pour** $h = 1$ **à** $\max\{h_{\max}, \theta h_{\max+}\}$ **faire**
    *MiseàjourCoûtsPénalisés*$(G, c(\lambda, \mu), \lambda, \mu)$ --→ (eq. 5.37)
    *CalculVariablesDuales*$(u, v, c(\lambda, \mu))$ --→ (eq. 5.30 à 5.32)
    *MiseàjourPénalités*$(\lambda, \mu)$ --→ (eq. 5.59 à 5.61)
  **fin pour**
  *MiseàjourCoûtsRéduitsdesArêtes*$(G, \widetilde{c}, u, v)$ --→ (eq. 5.49)
  *GénérerMeilleures_$\mathfrak{q}$-tournées*$(G, \Phi, \widetilde{c}, Q)$ --→ (eq. 5.56)
  **si** *il existe des $\mathfrak{q}$-tournées à coût réduit négatif* **alors**
    $\overline{\mathscr{R}} := \overline{\mathscr{R}} \cup \{$*quelques $\mathfrak{q}$-tournées*$\}$
    $\theta := 0$ ; $ok := 0$ ; $t := t + 1$
  **sinon**
    $\theta := 1$ ; $ok := ok{+}1$ ; $t := t + 1$
  **fin si**
  **si** $\theta = 1$ **alors**
    // Une borne inférieure valide a été trouvée
    **si** $f(u, v) > LB$ **alors**
      $LB := f(u, v)$
      $u_{best} := u$ ; $v_{best} := v$
    **fin si**
  **fin si**
**jusqu'à** $(ok = 2)$ ou $(t = t_{\max})$
**renvoyer** $LB$ ; $u_{best}$ ; $v_{best}$

remplace la procédure de génération des meilleures $\mathfrak{q}$-tournées de ce dernier par la procédure de génération des meilleures tournées élémentaires avant de le re-exécuter. Cette dernière ne peut fonctionner que si les variables duales fournies sont déjà de bonne qualité car sinon elle requiert des temps de calculs élevés et un espace-mémoire important ; en effet, le problème de génération de tournées élémentaires à coût réduit négatif est NP-complet, puisqu'il équivaut à rechercher un cycle de coût inférieur ou égal à une valeur $\gamma_2 = 0$ sur un graphe où certains coûts sont négatifs, ce qui est NP-complet comme l'ont rappelé Garey et Johnson [53].

Détaillée dans l'article de Christofides *et al.* [24]. la procédure de génération de tournées élémentaires, résumée dans cette section, admet en entrée des valeurs de variables duales ainsi qu'un graphe complet et produit en sortie les $\Delta_2$ meilleures tournées dont le coût réduit est inférieur au seuil $\gamma_2$ donné. Le principe est le suivant : n'importe quelle tournée élémentaire, c'est à dire sans boucle, peut être obtenue en assemblant deux chemins élémentaires $P$ et $\overline{P}$ qui vérifient les quatre conditions ci-après, où $V'(P)$ est l'ensemble des clients de $P$, $z_P$ son client final, $q_P = \sum_{i \in V'(P)} q_i$ sa charge et $f(P)$ son coût :

1. $P$ et $\overline{P}$ ont le même client final $i$ : $z_P = z_{\overline{P}} = i$
2. $P$ et $\overline{P}$ sont disjoints à l'exception de $i$ : $V'(P) \cap V'(\overline{P}) = \{i\}$
3. La capacité des véhicules est respectée : $q_P + q_{\overline{P}} \leq Q + q_i$
4. Il n'y a pas de différence importante de charge entre $P$ et $\overline{P}$ : $\min\{q_P, q_{\overline{P}}\} \geq \frac{q_P + q_{\overline{P}} - q_i}{2}$ et $\max\{q_P, q_{\overline{P}}\} \leq \frac{q_P + q_{\overline{P}} - q_i}{2} + q_i$

Notons $LB(f(P))$ une borne inférieure du coût $f(R)$ de n'importe quelle tournée $R$ obtenue en combinant $P$ de coût $f(P)$ avec n'importe quel autre chemin élémentaire $\overline{P}$ en respectant les conditions décrites ci-dessus. Sa valeur peut être estimée à l'aide de l'équation (5.63), où l'on approxime le coût minimal de $\overline{P}$ par le coût minimal de tout $\mathfrak{q}$-chemin pouvant compléter $P$.

$$LB(f(P)) = f(P) + \min_{q_{z_P} \leq q \leq Q + q_P + q_{z_P}} \begin{cases} f(\phi^1_{z_P,q}) & \text{si } \pi(\phi^1_{z_P,q}) \notin V(P) \\ f(\phi^2_{z_P,q}) & \text{sinon} \end{cases} \tag{5.63}$$

Pour trouver les meilleures tournées élémentaires de coût réduit inférieur ou égal à $\gamma_2$, la première étape consistera donc à générer l'ensemble $\mathscr{P}$ d'au maximum $\Delta_2$ chemins élémentaires qui vérifient chacun les quatre conditions suivantes :

1. Les coûts sont les meilleurs et inférieurs ou égaux à $\gamma_2$ : $\max_{P \in \mathscr{P}}\{LB(f(P))\} \leq \min_{P \notin \mathscr{P}}\{LB(f(P)), \gamma_2\}$
2. Au maximum $\Delta_2$ chemins se trouvent dans $\mathscr{P}$ : $|\mathscr{P}| \leq \Delta_2$
3. La charge reste limitée : $\forall P \in \mathscr{P} : q_P \leq \frac{Q}{2} + q_{z_P}$
4. Aucun chemin de $\mathscr{P}$ n'est dominé : $\forall P \in \mathscr{P}, \nexists P' \in \mathscr{P} : V'(P') = V'(P)$ et $f(P') \leq f(P)$.

Une fois l'ensemble $\mathscr{P}$ généré, les tournées élémentaires de coût inférieur ou égal à $\gamma_2$ sont obtenues en combinant les paires de chemins $(P, \overline{P}) \subset \mathscr{P}$ se terminant chez le même client

$i$, sans client commun autre que $i$ et telles que $q_P + q_{\overline{P}} \leq Q + q_i$ et $f(P) + f(\overline{P}) \leq \gamma_2$. Parmi les tournées obtenues, il ne faut garder que les tournées non dominées, c'est à dire toute tournée $R$ telle qu'il n'existe aucune autre tournée $R'$ qui vérifie $V'(R') = V'(R)$ et $f(R') \leq f(R)$. Les procédures de génération des chemins élémentaires et fusions en tournées élémentaires de coût inférieur à $\gamma_2$ sont résumées ci-après.

### 5.7.1 Génération de chemins élémentaires

Cette procédure manipule trois ensembles de chemins élémentaires $\mathscr{T}$, $\mathscr{P}_P$ et $\mathscr{P}$. $\mathscr{T}$ regroupe les chemins dits "temporaires". Chacun d'eux peut être transféré vers l'ensemble définitif $\mathscr{P}$, ou alors prolongé vers un client qui ne lui appartient pas encore. $\mathscr{P}_P$ est l'ensemble des chemins qui peuvent être obtenus après un prolongement de $P \in \mathscr{T}$.

A chaque itération, le chemin $P^* \in \mathscr{T}$ qui minimise $LB$ est transféré de $\mathscr{T}$ à $\mathscr{P}$, car il vérifie $LB(f(P*)) = \min_{P \in \mathscr{T}}\{LB(f(P))\}$. Ensuite, l'ensemble $\mathscr{P}_{P^*}$ est généré avant d'être réduit. La réduction de $\mathscr{P}_{P^*}$ consiste à en supprimer tous les chemins dont la borne inférieure est supérieure au seuil $\gamma_2$ et tous les chemins qui sont dominés par un autre chemin dans $\mathscr{T} \cup \mathscr{P}$. Cela revient à supprimer de $\mathscr{P}_{P^*}$ tous les chemins $P'$ qui vérifient la condition $LB(f(P')) \geq \gamma_2$ ou la condition $\exists P'' \in \mathscr{T} \cup \mathscr{P} : z_{P''} = z_{P'}, V'(P'') = V'(P')$, et $LB(f(P'')) \leq LB(f(P'))$. De même, on supprime de $\mathscr{T}$ tout chemin dominé par un chemin de $\mathscr{P}_{P^*}$. Les deux ensembles résultants peuvent alors être fusionnés, en posant $\mathscr{T} = \mathscr{T} \cup \mathscr{P}_{P^*}$, et une nouvelle itération peut commencer.

Pour initialiser cette procédure nous définissons $P_0 = (0)$ un chemin contenant uniquement le dépôt, puis $f(P_0) = 0$, $z_{P_0} = 0$, $\mathscr{T} = \{P_0\}$ et $\mathscr{P} = \emptyset$. Elle s'arrête dès que $\mathscr{T} = \emptyset$ ou $|\mathscr{P}| = \Delta$. Pour éviter les erreurs de mémoire (*out of memory*) la taille maximale de $\mathscr{T}$ est limité à un paramètre *NSTATB*. Si cette valeur de l'ordre de 10000000 est dépassée, la procédure s'arrête prématurément, l' instance n'est donc pas résolue et la borne inférieure fournie par notre algorithme de génération des $\mathfrak{q}$-tournées n'est pas améliorée. Dans le cas contraire, l'ensemble final $\mathscr{P}$ est transmis à la procédure de fusion décrite ci-après.

### 5.7.2 Fusions de paires chemins élémentaires en tournées

La procédure de fusion de chemins élémentaires en tournées utilise l'ensemble $\mathscr{P}$ des chemins déjà générés pour construire un ensemble $\mathscr{T}$ des paires de chemins $(P, \overline{P})$ de coût $f(P) + f(\overline{P})$, qui sont choisis dans $\mathscr{P}$ et qui ont un même client final. Les meilleures tournées obtenues à partir des paires de chemins sont stockées dans l'ensemble final $|\mathscr{B}|$.

Ainsi, à chaque itération, la paire $(P^*, \overline{P^*})$ de coût minimal est extraite de $\mathscr{T}$. Si elle forme une tournée réalisable de coût inférieur à $\gamma_2$ alors la tournée correspondante est ajoutée à $\mathscr{B}$, sinon elle est rejetée.

Cette procédure, initialisée avec $|\mathscr{B}| = \emptyset$, s'arrête dès que $|\mathscr{B}| = \Delta_2$ ou que $\mathscr{T} = \emptyset$.

## 5.8 Résultats Numériques

### 5.8.1 Instances et implémentation

Nous avons testé nos heuristiques duales H1 et H1+H2 sur le cas particulier du VRP ou 1-PTVP avant de les appliquer au $m$-PTVP avec $m > 1$. Rappelons que H1 est basée sur la génération de $\mathfrak{q}$-tournées, tandis que H1+H2 cherche à améliorer la borne inférieure produite par H1 en générant des tournées élémentaires. Toutes les deux ne sont pas adaptées au $m$-PVP car ce dernier ne comprend aucune contrainte de capacité. Les instances utilisées sont donc de type A, B, P et VRPNC de 19 à 101 nœuds ; pour plus de détails les concernant, se référer au chapitre 3.

L'heuristique H1 est programmée en C tandis que H2 est obtenue en remplaçant la génération de $\mathfrak{q}$-tournées de H1 par une adaptation de la procédure GENROUTE de Baldacci *et al.* [10] pour le VRP, capable de produire les meilleures tournées élémentaires à partir de variables duales données. Son implémentation est toutefois loin d'être triviale puisque GENROUTE est programmée en Fortran et doit communiquer avec le reste de nos procédures en C. Elle doit en plus subir certaines modifications avant d'être applicable au $m$-PTVP, par exemple l'introduction des variables duales $v_e$ liées aux arêtes, peu pertinentes pour le VRP mais essentielles pour le $m$-PTVP. Le PC utilisé est un Intel Core 2 Duo de 1,80 GHz et 2 Go de mémoire.

Les résultats obtenus sont présentés sous forme agrégée dans les tableaux 5.1 et 5.8.3, afin d'en simplifier la lecture et l'analyse. Le détail des résultats est disponible en annexe. Le nombre d'instances utilisées par type est précisé dans la première colonne de chaque tableau. Soit UB la meilleure borne supérieure trouvée par les métaheuristiques du chapitre 4, la colonne $\theta_{\%}$ d'une procédure disons $x$ montre la moyenne des ratios $\frac{100LBx}{UB}$ pour chaque type d'instances, sachant que LB$x$ est la valeur de la borne inférieure fournie par $x$. Les colonnes $\min - \max$ présentent les valeurs minimales et maximales de $\theta_{\%}$ alors que $\sigma$ est son écart-type et $T$ le temps de calcul en secondes. Enfin, $n_i$ et $n_o$ désignent respectivement le nombre d'instances pour lesquelles H1 a atteint le nombre maximum d'itérations et le nombre d'instances pour lesquelles l'adaptation de GENROUTE appelée par H2 a atteint la taille mémoire limite *NSTATB*. Comme expliqué au paragraphe 5.7.1, si *NSTATB* est atteint alors H2 s'arrête par manque d'espace-mémoire. Dans ce cas les valeurs de $\theta_{\%}$ et $\sigma$ prises en compte pour H1+H2 sont celles de H1.

Les paramètres appliqués à nos deux heuristiques duales sont les suivants :

- Communs
  - nbre max. d'itérations : $t_{\max} = 500$
  - nbre max. d'étapes (sous-gradient) : $h_{\max} = 1, h_{\max+} = 100$
  - nbre max. d'itérations avant mise à jour de $\epsilon$ : $h^{\epsilon} = 25$
  - taille mémoire maximale allouée : $NSTATB = 12000000$
  - valeur maximale du coût réduit des ($\mathfrak{q}$-)tournées à ajouter : $\gamma^0 = \infty, \gamma = 0$
- Spécifiques à H1
  - coefficient initial (sous-gradient) : $\epsilon_0 = 1,5$

nbre max. de $\mathfrak{q}$-tournées par client à ajouter : $\Delta^0_{\max} = m, \Delta_{\max} = m$

– Spécifiques à H2

paramètre initial de sous-gradient : $\epsilon_0 = 0,2$

nbre max. de tournées élémentaires par clients à ajouter : $\Delta^0_2 = 3000, \Delta_2 = 400$

### 5.8.2 Résultats sur le VRP

Le tableau 5.1 illustre la performance des bornes inférieures produites par nos heuristiques duales sur le cas particulier du VRP. A titre indicatif, la colonne $LB_{vrp}$ montre les résultats des $\mathfrak{q}$-tournées de Baldacci *et al.* [10] pour le VRP, ces derniers ayant à ce jour la meilleure procédure de calcul de bornes inférieures pour le VRP. Celle-ci sera donc comparée à H1. Rappelons enfin que les bornes supérieures UB utilisées pour les calculs des ratios $\frac{100LBi}{UB}, \forall i \in \{1,2,3\}$ ont été prouvées optimales pour les instances de type A, B et P, soit par Fukasawa *et al.* [52], soit par Baldacci *et al.* [10].

| | $LB_{vrp}$ | H1 | | | | | H1+H2 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Type | $\theta_{\%}$ | $\theta_{\%}$ | $\sigma$ | $\min - \max$ | $T$ | $n_i$ | $\theta_{\%}$ | $\sigma$ | $\min - \max$ | $T$ | $n_o$ |
| A (27) | 96,5 | 96,10 | 1,44 | 91,67 - 98,85 | 8,83 | 0 | 97,93 | 0,75 | 95,73 - 99,63 | 37,74 | 0 |
| B (23) | - | 90,42 | 2,92 | 85,89 - 95,63 | 11,41 | 0 | 93,00 | 3,45 | 85,89 - 97,46 | 48,98 | 9 |
| P (23) | 98,2 | 97,55 | 1,31 | 93,56 - 98,64 | 40,52 | 0 | 98,66 | 0,91 | 96,31 - 100,00 | 59,39 | 0 |
| VRPNC (7) | - | 96,41 | 1,13 | 94,78 - 97,47 | 241,00 | 0 | 97,66 | 1,57 | 94,83 - 99,79 | 273,00 | 2 |

Note : Baldacci *et al.* [10] n'ont pas traité les instances VRPNC et ont inséré des coupes au cours de la résolution des instances B

TABLE 5.1 – Résultats de H1 et H1+H2 sur le cas particulier du VRP = 1-PTVP

Les résultats montrent que H1 et $LB_{vrp}$ obtiennent des résultats similaires sur le VRP, comme attendu. H1 fonctionne moins bien sur les instances fortement clusterisées (type B) que sur les autres. H2 améliore de 1% à 2,5% la borne inférieure produite par H1. Nos procédures convergent efficacement puisqu'aucune instance n'a nécessité un nombre d'itérations proche du maximum autorisé. Deux instances VRPNC et neuf instances de type B n'ont pas pu être traités par H2 faute de mémoire. Pour les instances de type B, cela peut s'expliquer par la relative faiblesse de la borne fournie par H1.

### 5.8.3 Résultats sur le $m$-PTVP

Le tableau 5.8.3 illustre la performance des bornes inférieures produites par nos heuristiques duales sur le cas particulier du VRP. A titre indicatif, la colonne LB3 montre les résultats du $b$-couplage parfait présenté au chapitre 3, dont l'efficacité croît lorsque $m$ augmente.

Les résultats obtenus montrent que contrairement à LB3, nos algorithmes sont plus efficaces lorsque $m$ prend de petites valeurs, en particulier $m = 2$. Dans ce dernier cas, H1

(a) 2-PTVP

| | $LB3$ | H1 | | | | | H1+H2 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Type | $\theta_{\%}$ | $\theta_{\%}$ | $\sigma$ | $\min - \max$ | $T$ | $n_i$ | $\theta_{\%}$ | $\sigma$ | $\min - \max$ | $T$ | $n_o$ |
| A (26) | 91,89 | 96,18 | 1,19 | 93,63 - 98,27 | 6,92 | 0 | 97,26 | 0,95 | 94,78 - 98,96 | 57,43 | 0 |
| B (23) | 79,22 | 90,93 | 2,79 | 86,40 - 96,07 | 10,74 | 0 | 93,32 | 3,42 | 86,83 - 98,43 | 187,79 | 5 |
| P (19) | 97,18 | 93,97 | 1,46 | 92,95 - 98,57 | 58,63 | 1 | 97,35 | 1,41 | 93,32 - 98,87 | 178,77 | 1 |
| VRPNC (7) | 86,82 | 94,93 | 2,88 | 88,98 - 97,4 | 163,26 | 3 | 96,02 | 1,43 | 93,70 - 97,92 | 201,9 | 3 |

(b) 3-PTVP

| | $LB3$ | H1 | | | | | H1+H2 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Type | $\theta_{\%}$ | $\theta_{\%}$ | $\sigma$ | $\min - \max$ | $T$ | $n_i$ | $\theta_{\%}$ | $\sigma$ | $\min - \max$ | $T$ | $n_o$ |
| A (25) | 97,24 | 96,74 | 1,68 | 91,88 - 98,80 | 23,72 | 5 | 97,05 | 1,65 | 92,19 - 98,89 | 119,01 | 1 |
| B (22) | 92,67 | 92,76 | 4,72 | 75,98 - 97,99 | 23,98 | 8 | 93,54 | 4,72 | 75,98 - 99,16 | 127,76 | 8 |
| P (14) | 98,61 | 97,31 | 3,3 | 86,45 - 99,28 | 86,11 | 8 | 97,47 | 3,35 | 86,45 - 99,53 | 113,93 | 3 |
| VRPNC (7) | 90,27 | 94,42 | 3,59 | 88,58 - 97,76 | 191,43 | 5 | 94,47 | 3,64 | 88,58 - 98,06 | 212,36 | 5 |

(c) 4-PTVP

| | $LB3$ | H1 | | | | | H1+H2 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Type | $\theta_{\%}$ | $\theta_{\%}$ | $\sigma$ | $\min - \max$ | $T$ | $n_i$ | $\theta_{\%}$ | $\sigma$ | $\min - \max$ | $T$ | $n_o$ |
| P (8) | 98,84 | 96,92 | 1,64 | 94,09 - 98,90 | 254,70 | 2 | 97,55 | 2,14 | 94,09 - 99,59 | 288,7 | 3 |
| VRPNC (6) | 92,09 | 90,75 | 4,40 | 84,56 - 96,74 | 171,58 | 5 | 90,75 | 4,40 | 84,56 - 96,74 | 171,58 | 6 |

(d) $5, 6, 7$-PTVP

| | $LB3$ | H1 | | | | | H1+H2 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Type | $\theta_{\%}$ | $\theta_{\%}$ | $\sigma$ | $\min - \max$ | $T$ | $n_i$ | $\theta_{\%}$ | $\sigma$ | $\min - \max$ | $T$ | $n_o$ |
| P (5) | 98,92 | 95,93 | 2,00 | 93,23 - 98,46 | 547,04 | 5 | 96,04 | 2,45 | 93,23 - 99,45 | 626,33 | 4 |
| VRPNC (6) | 94,59 | 87,23 | 7,91 | 73,66 - 96,58 | 304,31 | 9 | 87,23 | 7,91 | 73,66 - 96,58 | 304,31 | 6 |

TABLE 5.2 – Résultats des heuristiques duales H1 et H1+H2 appliquées au $m$-PTVP : $m \in \{2, 3, 4, 5, 6, 7\}$

produit de bonnes bornes inférieures, ce qui permet à H1+H2 de dominer LB3, malgré les neuf instances pour lesquelles la limite de mémoire est atteinte. Toutefois, la génération de tournées élémentaires se fait au prix d'un doublement des temps de calculs, pour une amélioration de 1% à 2% de $\theta_{\%}$. Lorsque $m \geq 3$, nous constatons une forte augmentation du nombre d'instances pour lesquelles un nombre d'itérations supérieur au maximum prédéfini est requis, ce qui veut dire que la croissance de la borne inférieure est de plus en plus souvent arrêtée avant qu'il n'y ait eu convergence, ce qui explique les moins bonnes performances de H1 par rapport à LB3. Les performances de H1+H2 s'en ressentent puisque le nombre d'instances qui n'ont pas pu être traitées croît également. Nous envisageons donc d'effectuer des tests supplémentaires avec des paramètres différents selon la valeur de $m$, tels que le nombre maximum d'itérations autorisées ou le nombre maximum de $\mathfrak{q}$-tournées à ajouter, dans le but d'étendre les bonnes performances de H1 et H1+H2, déjà vérifiée pour $m \in \{2, 3\}$, à toutes les valeurs de $m$. En tout cas, on constate que LB3 et H1+H2 sont complémentaires.

## 5.9 Conclusion

Ce chapitre présente une borne inférieure plus élaborée que celles du chapitre 3 et basée sur la recherche de bonnes solutions duales de la relaxation linéaire de la formulation de partitionnement aggrégée du problème de tournées de véhicules $m$-péripatétiques ($m$-PTVP). Rappelons que ce problème consiste à extraire d'un graphe complet des tournées de véhicules disjointes de coût total minimal sur $m$ périodes telles que chaque client soit visité exactement une fois par période. Il peut être considéré comme une généralisation de deux problèmes NP-difficiles connus : le VRP = 1-PTVP et le $m$-PVP, correspondant au cas mono-véhicule.

Le calcul des meilleures valeurs de variables duales est effectué à l'aide de deux heuristiques duales basées sur (1) l'approximation des tournées par des $\mathfrak{q}$-tournées, lesquelles peuvent être générées à l'aide d'un algorithme pseudo-polynomial et (2) l'utilisation d'une approche de type génération de colonnes pour pallier au nombre exponentiel de variables (3) la résolution des différents problèmes-maitres, obtenus après génération et ajout des ($\mathfrak{q}$-)tournées à coût réduit négatif, en optimisant les valeurs des variables duales sans faire appel à un quelconque solveur tel que Cplex®, mais en quantifiant les violations des contraintes de degré pour corriger (par relaxation lagrangienne) les valeurs affectées. L'heuristique duale H1 s'arrête après deux itérations consécutives sans ajout de $\mathfrak{q}$-tournées et H2 prend le relai, en essayant de générer des tournées élémentaires à coût réduit négatif. Les résultats numériques montrent que cette approche domine les bornes polynomiales précédemment présentées lorsque $m$ prend de petites valeurs. Pour les autres, les besoins en nombre d'itérations ou en mémoire vive empêchent d'obtenir de meilleurs résultats. Sachant que les bornes inférieures polynomiales sont très efficaces lorsque $m$ croît et moins lorsque $m$ prend de petites valeurs, nous sommes donc capable de produire bonnes bornes inférieures de $m$-PTVP quel que soit $m$.

Cette approche a été présentée lors du récapitulatif des travaux de thèse au sein de la conférence nationale *ROADEF2009* [104].

# Chapitre 6

# Approche polyédrale pour le $m$-PTVP

## 6.1 Introduction

Rappelons que le $m$-PTVP consiste à trouver des tournées réalisables de coût total minimal sur $m$ périodes, telles qu'aucune arête ne soit utilisée plus d'une fois. Dans chaque période, chaque client est servi une seule fois, chaque véhicule effectue au maximum une tournée et chacune de ces tournées débute puis se termine au dépôt avec une charge totale (somme des demandes des clients visités) inférieure ou égale à la capacité des véhicules. Les chapitres précédents étaient consacrés aux métaheuristiques, aux bornes inférieures polynomiales et à une approche de type génération de colonne.

Ce chapitre se focalise sur l'adaptation de l'approche polyédrale au $m$-PTVP. Deux méthodes de branchements et coupes (*branch-and-cut*) pour ce problème sont proposées, basées respectivement sur la formulation (PA) présentée au chapitre 3 et sur une relaxation de ce modèle. Le principe de telles méthodes, déjà évoqué dans l'état de l'art, consiste à résoudre le problème en continu, à ajouter les contraintes valides violées et à créer des sous-problèmes disjoints tels que leur union contienne toutes les solutions réalisables du problème initial. Notre objectif lors du développement de cette approche est d'évaluer la qualité des solutions trouvées par les métaheuristiques du chapitre 4 en améliorant significativement les bornes inférieures du chapitre 3, si possible jusqu'à l'optimum. Les algorithmes sont implémentés à l'aide du logiciel Cplex® et nous ont amenés à modifier profondément la bibliothèque de fonctions de détection de coupes violées CVRPSEP de Lysgaard pour le VRP, afin d'obtenir une nouvelle bibliothèque pour le $m$-PTVP.

La suite de ce chapitre est organisée en sept sections. La première rappelle le fonctionnement des algorithmes de séparation et coupes ainsi que les modèles mathématiques utilisés. Les quatre suivantes détaillent les inégalités valides utilisées, ainsi que leurs procédures de séparation. Celle d'après est consacrée aux règles de branchement et de parcours de l'arbre de résolution. Enfin les résultats numériques font l'objet de la dernière section.

## 6.2 Présentation de la méthode

Un algorithme de branchements et coupes correspond à une procédure de recherche arborescente par séparation et évaluation où la borne inférieure en minimisation de chaque sous-problème est renforcée à l'aide de méthodes de coupes. Ce genre de méthodes apporte généralement de bons résultats sur les problèmes de tournées, voir par exemple les articles de Letchford *et al.* [80] et Fukasawa *et al.* [52]. Le principe est expliqué au chapitre 2, mais nous allons tout de même en rappeler les grandes lignes avant de détailler les modèles mathématiques utilisés pour le $m$-PTVP.

### 6.2.1 Principe

Une méthode de coupes résout un programme en nombres entiers disons $P$ comme une séquence de programmes linéaires. Au départ, certaines contraintes de $P$ sont relaxées, notamment les contraintes d'intégrité, pour obtenir $P_0$. Si la solution optimale $s^*_{P_0}$ de $P_0$ est réalisable pour $P$, alors elle est aussi optimale pour $P$. Dans le cas contraire, il doit exister une ou plusieurs contraintes valides pour $P$ qui sont violées par $s^*_{P_0}$. Ces dernières, appelées coupes, sont alors ajoutées au programme linéaire $P_0$ avant la prochaine résolution. La procédure se poursuit jusqu'à ce que $s^*_{P_0}$ soit réalisable pour $P$, ou alors qu'il ne soit plus possible de générer des contraintes violées. Dans le premier cas on obtient une solution optimale du problème d'origine $P$. Dans le second cas le coût de $s^*_{P_0}$ est une borne inférieure et une méthode de branchement peut prendre le relais.

Le branchement consiste à diviser un problème $P_0$ en deux (ou plusieurs) sous-problèmes dont les ensembles des solutions réalisables forment un recouvrement, idéalement une partition, de l'ensemble des solutions de $P_0$. Ainsi, en résolvant tous les sous-problèmes et en prenant la meilleure solution trouvée, on est assuré d'avoir résolu $P_0$. Si pour un des sous-problèmes la solution obtenue par la méthode de coupes est optimale, on arrête son exploration. Sinon, on le sépare en deux (ou plusieurs) sous-problèmes, et ainsi de suite. La procédure s'arrête lorsque tous les sous-problèmes ont été résolus. La meilleure solution trouvée est alors optimale.

Une arborescence générant tous les sous-problèmes reviendrait à une énumération complète des solutions. En pratique, heureusement, un grand nombre de sous-problèmes sont éliminés grâce à la borne inférieure $LB$ aussi appelé évaluation. Si on a déjà trouvé une solution de coût $F$ et qu'en un nœud $P_k$ on a $LB(P_k) \geq F$, alors on peut abandonner l'exploration de la branche partant de $P_k$, car elle ne permettra pas d'obtenir une solution meilleure que celle que l'on connaît déjà.

Il est possible d'arrêter prématurément la procédure de branchements et coupes lorsque les temps de calculs ou l'espace mémoire requis deviennent trop importants. Dans ce cas le coût de la meilleure solution linéaire trouvée n'est qu'une borne inférieure. Le choix des formulations mathématiques utilisées est donc très important pour trouver la meilleure borne inférieure possible, et idéalement la solution optimale. Il est également possible d'obtenir

une borne supérieure de $P$ en ajoutant des contraintes d'intégrité à l'un des sous-problèmes générés au cours de la recherche, avant de résoudre le problème résultant avec une méta-heuristique.

### 6.2.2 Formulations mathématiques utilisées

Les travaux publiés sur le VRP et le $m$-PVP présentés dans l'état de l'art montrent que l'approche polyédrale est efficace sur les modèles mathématiques pour lesquels on peut partir d'une relaxation qui (1) n'a pas un nombre exponentiel de contraintes (2) ne nécessite pas l'ajout d'un nombre exponentiel de contraintes avant que ses solutions n'atteignent des valeurs proches de l'optimum. C'est pour ces raisons que nos deux algorithmes de branchements et coupes pour le $m$-PTVP concernent des modèles mathématiques basés sur les arêtes sans indices de véhicules. Ces derniers sont présentés avec leurs relaxations initiales respectives dans les paragraphes qui suivent.

#### Modèle (PA)

La première procédure de branchements et coupes $\mathrm{BC_{PA}}$ résout le $m$-PTVP formulé par le modèle à 2 indices suivant, déjà présenté au chapitre 3 (se référer à la section 3.2.3 pour le descriptif du rôle des différentes contraintes).

$$\text{(PA)} \quad \min \sum_{k\in\mathbb{M}} \sum_{e\in E} c_e x_e^k \tag{6.1}$$

s. c.

$$\sum_{e\in\delta(i)} x_e^k = 2, \qquad \forall k \in \mathbb{M}, \forall i \in V' \tag{6.2}$$

$$\sum_{e\in\delta(0)} x_e^k \geq 2\lambda, \qquad \forall k \in \mathbb{M} \tag{6.3}$$

$$\sum_{e\in\delta(S)} x_e^k \geq 2r(S), \qquad \forall S \subseteq V', S \neq \varnothing, \forall k \in \mathbb{M} \tag{6.4}$$

$$\sum_{k\in\mathbb{M}} x_e^k \leq 1, \qquad \forall e \in E \tag{6.5}$$

$$x_e^k \in \{0,1\}, \qquad \forall e \in E, \forall k \in \mathbb{M} \tag{6.6}$$

Au départ de la procédure $\mathrm{BC_{PA}}$, les contraintes d'intégrité (6.6) et les $m2^n$ contraintes de capacité (6.4) sont relaxées pour obtenir ($\mathrm{PA_0}$), défini par les équations (6.7) à (6.11). Ce dernier constitue alors la relaxation initiale.

$$\text{(PA}_0\text{)} \quad \min \sum_{k\in\mathbb{M}} \sum_{e\in E} c_e x_e^k \tag{6.7}$$

s. c.

$$\sum_{e\in\delta(i)} x_e^k = 2, \qquad \forall k \in \mathbb{M}, \forall i \in V' \tag{6.8}$$

$$\sum_{e\in\delta(0)} x_e^k \geq 2\lambda, \qquad \forall k \in \mathbb{M} \tag{6.9}$$

$$\sum_{k\in\mathbb{M}} x_e^k \leq 1, \qquad \forall e \in E \tag{6.10}$$

$$0 \leq x_e^k \leq 1, \qquad \forall e \in E, \forall k \in \mathbb{M} \tag{6.11}$$

La relaxation initiale ($\mathrm{PA}_0$) contient un nombre de variables de l'ordre de $mn^2$ et des nombres de contraintes respectivement de l'ordre de $mn$, $m$ et $n^2$. La fonction-objectif à minimiser (6.7) correspond toujours au coût total des arêtes qui sont utilisées au cours des $m$ périodes. Les contraintes (6.8), (6.9) et (6.10) imposent respectivement les degrés des clients, du dépôt, et l'utilisation au maximum 1 fois de chacune des arêtes au cours des $m$ périodes. Enfin, les contraintes (6.11) s'assurent que les variables de décision réelles $x_e^k$ varient entre 0 et 1.

Les coupes appliquées à la relaxation initiale lors de la procédure $\mathrm{BC}_{\mathrm{PA}}$ auront donc deux fonctions : (1) trouver une solution entière (2) ajouter des contraintes de capacité de manière progressive pour que la solution finale devienne réalisable pour (PA) et donc pour le $m$-PTVP, bien avant que toutes les $m2^n$ contraintes soient insérées.

### Modèle RPA ou PA agrégé

La deuxième procédure de branchements et coupes pour le $m$-PTVP résout le problème obtenu après agrégation sur $k \in \mathbb{M}$ des contraintes du modèle (PA). Concrètement, (RPA) est obtenu après agrégation des contraintes (6.2) à (6.6) sur $k$, puis changement de variable $X_e = \sum_{k\in\mathbb{M}} x_e^k$. Notons que les contraintes (6.5) deviennent redondantes avec (6.6) une fois agrégées, elles peuvent donc être supprimées. La variable de décision $X_e$ est binaire et égale à 1 si et seulement si l'arête $e$ a été choisie au sein de la solution de (RPA).

$$\text{(RPA)} \quad \min \sum_{e\in E} c_e X_e \tag{6.12}$$

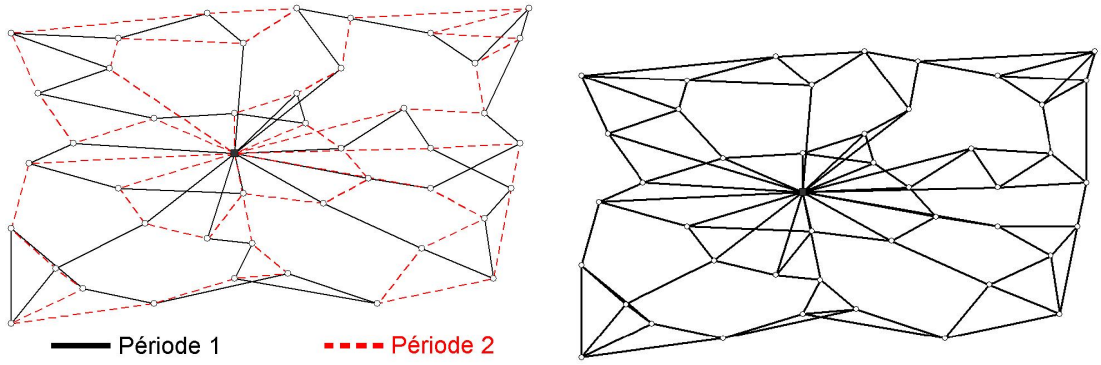
s. c.

$$\sum_{e\in\delta(i)} X_e = 2m, \qquad \forall i \in V' \tag{6.13}$$

$$\sum_{e\in\delta(0)} X_e \geq 2m\lambda, \tag{6.14}$$

$$\sum_{e\in\delta(S)} X_e \geq 2mr(S), \qquad \forall S \subseteq V', S \neq \varnothing \tag{6.15}$$

$$X_e \in \{0,1\}, \qquad \forall e \in E \tag{6.16}$$




(a) Sol. opt. de (PA) (2-PTVP) : coût = 1148 (b) Sol. opt. de (RPA) : coût = 1146

FIGURE 6.1 – Comparaison de solutions optimales de (PA) et (RPA) pour $m = 2$

L'avantage est de diviser par $m$ le nombre de variables et de contraintes à gérer. L'inconvénient est d'obtenir une solution optimale dont le coût n'est qu'une borne inférieure pour le $m$-PTVP. En effet, comme l'illustre la figure 6.1, toute solution du problème agrégé (RPA) n'est qu'un sous-ensemble d'arêtes "en vrac", qui ne sont pas réparties suivant les $m$ périodes du $m$-PTVP et ne correspond pas forcément à une solution réalisable. Même dans les rares cas où ce sous-ensemble correspond exactement à une solution réalisable et donc optimale du $m$-PTVP, le partitionner en $m$ périodes reste NP-difficile comme nous l'avons montré avec le théorème 2 (page 59). Malgré toutes ces limitations, la formulation à un indice (RPA) reste utile car plus simple à résoudre et manipuler, non seulement grâce au nombre réduit de variables, mais également grâce aux contraintes agrégées qui réduisent le nombre d'inégalités violées à identifier. Des bornes inférieures de bonne qualité devraient donc être obtenues dans un temps de calcul raisonnable.

Au départ de la procédure de branchements et coupes $\mathrm{BC}_{\mathrm{RPA}}$ pour ce modèle (RPA), les contraintes d'intégrité (6.16) et les $2^n$ contraintes de capacité (6.15) sont relaxées pour obtenir la relaxation initiale ($\mathrm{RPA}_0$), définie par les équations (6.17)-(6.20).

$$(\mathrm{RPA}_0) \quad \min \sum_{e \in E} c_e X_e \tag{6.17}$$

s. c.

$$\sum_{e \in \delta(i)} X_e = 2m, \qquad \forall i \in V' \tag{6.18}$$

$$\sum_{e \in \delta(0)} X_e \geq 2m\lambda, \tag{6.19}$$

$$0 \leq X_e \leq 1, \qquad \forall e \in E \tag{6.20}$$

Cette relaxation initiale, qui rappelle le modèle relaxé du $b$-couplage du chapitre 3,

contient un nombre de variables de l'ordre de $n^2$ et des nombres de contraintes respectivement de l'ordre de $n$, 1 et $n^2$. La fonction-objectif à minimiser (6.17) correspond toujours au coût total des arêtes qui sont choisies. Les contraintes (6.18), (6.19) et (6.20) imposent respectivement les degrés des clients, du dépôt et le domaine de définition des variables de décision réelles $X_e$.

Les coupes appliquées à ($\text{RPA}_0$) auront donc deux fonctions : (1) trouver une solution entière (2) ajouter les contraintes de capacité de manière progressive pour que la solution finale devienne réalisable pour (RPA) bien avant que toutes les $2^n$ contraintes soient insérées.

## 6.3 Coupes pour la procédure $\text{BC}_{\text{PA}}$

Puisque le $m$-PTVP peut être considéré comme une généralisation du VRP, nous avons généralisé des familles d'inégalités dont l'efficacité pour le VRP a été démontrée. Cette généralisation peut se faire en remarquant que les arêtes utilisées par une solution du $m$-PTVP au cours d'une période $k \in \mathbb{M}$ forment une solution réalisable du CVRP. Ce principe, déjà exploité au chapitre 3 pour construire le modèle de partitionnement (PS) basé sur les *solutions-VRP*, nous autorise donc à détecter les violations de contraintes de capacité, de multi-étoiles et de peigne au sein de chacune des $m$ périodes, à quelques exceptions-près également évoquées dans les paragraphes qui suivent.

### 6.3.1 Contraintes de capacité

Les contraintes de capacité (6.21), qui empêchent les sous-tours, correspondent tout simplement aux contraintes (3.7) ou (6.4) initialement relaxées dans le modèle initial ($\text{PA}_0$). Rappelons que ces dernières, déjà détaillées au paragraphe 3.2.3 et illustrées par la figure 3.3, imposent que n'importe quel sous-ensemble $S \subseteq V'$ de clients soit traversé par au moins $2r(S)$ arêtes, afin qu'un nombre suffisant de véhicules puisse servir ces clients. La procédure de séparation de ces coupes est détaillée dans la section 6.5.

$$\sum_{e \in \delta(S)} x_e^k \geq 2r(S), \qquad \forall S \subseteq V', S \neq \varnothing, \quad \forall k \in \mathbb{M} \tag{6.21}$$

Les travaux publiés sur le VRP comportent d'autres types de contraintes de capacité, plus fortes mais plus difficiles à séparer. La différence se situe au niveau de la définition et la valeur du terme de droite de l'inégalité. Les coupes de capacité renforcées évaluent le nombre minimum de véhicules nécessaires pour satisfaire la demande de $S \subset V'$ en résolvant le problème de bin-packing correspondant : considérant les clients $i \in S$ comme des objets de taille $q_i$, le problème de bin-packing calcule le nombre minimum $B(S)$ de boîtes de taille $Q$ nécessaires pour ranger les $|S|$ objets. Leur adaptation au $m$-PTVP (6.22) n'est toutefois pas utilisée au sein de notre procédure $\text{BC}_{\text{PA}}$ car le sous-problème de

bin-packing est encore NP-difficile.

$$\sum_{e\in\delta(S)} x_e^k \geq 2B(S), \qquad \forall S \subseteq V', S \neq \varnothing, \quad \forall k \in \mathbb{M} \tag{6.22}$$

Enfin, il existe des contraintes de capacité dites généralisées pour le VRP (voir par exemple l'article de Cornuejols et Harche [29] ou même la thèse d'Augerat[8]) qui sont plus fortes que (6.22) et sont basées sur le nombre exact de véhicules utilisés. Ces contraintes ne sont pas adaptées au $m$-PTVP car le nombre de tournées n'est pas imposé et peut varier d'une période à l'autre. C'est également le cas des contraintes d'hypotour d'Augerat [8].

### 6.3.2 Contraintes de peigne

Des contraintes de peigne sont souvent utilisées pour résoudre le TSP (voir par exemple Chvátal [25] ou Grötschel et Padberg [60]). Araque [3] a proposé une nouvelle variante pour le VRP qui généralise et domine les précédentes, ainsi que les contraintes de 2-couplage d'Edmonds [44]. Cette variante est valable au cours de chacune des $m$ périodes du $m$-PTVP. Pour définir une contrainte de peigne, nous avons besoin de définir un manche $H \subset V'$ et $t$ dents $\{T_1, ..T_t\} \subset V'$ qui vérifient les quatre conditions suivantes :

1. $t \geq 2$
2. chaque dent a au moins un client en commun avec le manche : $H \cap T_j \neq \emptyset, \forall j \in [1 \dots t]$
3. chaque dent a au moins un client hors du manche : $T_j \setminus H \neq \emptyset, \forall j \in [1 \dots t]$
4. deux dents ne peuvent se croiser qu'au sein du manche : $\forall \{i, j\} \subset [1 \dots t] : T_i \cap T_j \subset H$ ou $T_i \cap T_j \cap H = \emptyset$.

Les contraintes de peigne renforcées pour le $m$-PTVP et déduites des travaux de Araque [3] sont données par les équations (6.23). Leur objectif est d'accélérer les algorithmes de résolution en éliminant des solutions trouvées les structures illustrées par la figure 6.2.

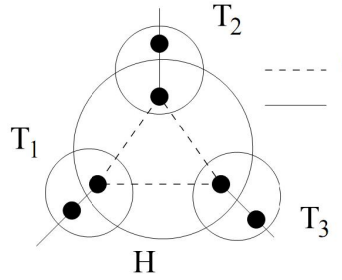
$$\sum_{e\in\delta(H)} x_e^k + \sum_{e\in\delta(T_j)} \sum_{j=1}^{t} x_e^k \geq S(H, T_1, .., T_t) + 1, \quad \forall k \in \mathbb{M} \tag{6.23}$$

où

- $S(H, T_1, .., T_t) = \sum_{j=1}^{t} (\tilde{r}(T_j \cap H) + \tilde{r}(T_j \setminus H) + \tilde{r}(T_j))$ avec $\tilde{r}(S) = \begin{cases} r(V \setminus S) \text{ si } 0 \in S \\ r(S) \text{ sinon} \end{cases}$
- $S(H, T_1, .., T_t)$ est impair.

### 6.3.3 Contraintes de multi-étoiles

Les contraintes de multi-étoiles ont été originellement définies par Araque *et al.* [4], pour le VRP avec demande unitaire, avant d'être étendues au cas général par Letchford *et al.* [80]. Soit $(S : S')$ le sous-ensemble d'arêtes ayant une extrémité dans $S \subseteq V$ et l'autre




$$\sum_{e\in\delta(H)} x_e^k + \sum_{e\in\delta(T_j)} \sum_{j=1}^{t} x_e^k = 3 + 2 \times 3 = 9$$

$$S(H, T_1, T_2, T_3) + 1 = (1+1+1)+(1+1+1)+(1+1+1) + 1 = 10$$

$$\Downarrow$$

$$\sum_{e\in\delta(H)} x_e^k + \sum_{e\in\delta(T_j)} \sum_{j=1}^{t} x_e^k < S(H, T_1, .., T_t) + 1$$

$$\Downarrow$$

VIOLATION !

FIGURE 6.2 – Détection de coupes de peigne avec demandes unitaires et $Q = 7$

extremité dans $S' \subseteq V$. Définissons un noyau $N \subset V'$, un ensemble $S \subseteq \{V' \setminus N\}$ de clients-satellites et un ensemble $C \subset N$ de clients-connecteurs. Les inégalités (6.24), (6.25) et (6.26) présentent les trois principaux types de contraintes de multi-étoiles appliquées à chaque période du $m$-PTVP.

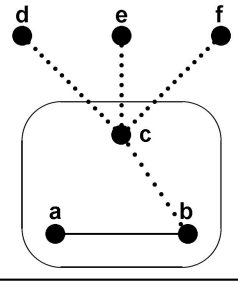
### Multi-étoiles partielles

Les contraintes de multi-étoiles partielles proposées par Araque *et al.* [4] généralisées pour le $m$-PTVP se déclinent en quatre variantes comme le montre l'équation (6.24). La figure 6.3 illustre un exemple de détection de ces coupes avec des demandes unitaires.

$$\beta \sum_{e\in(N:N)} x_e^k + \sum_{e\in(C:S)} x_e^k \leq \gamma, \quad \forall k \in \mathbb{M} \tag{6.24}$$

où

- si $Q \geq 3$, $|N| \bmod Q = 0$ et $|C| = 1$ alors $\beta = 2$ et $\gamma = 2(|N| - r(N))$
- si $Q \geq 4$, $|N| \bmod Q = 1$ et $|C| = 2$ alors $\beta = 2$ et $\gamma = 2(|N| - r(N) + 1)$
- si $Q \geq 4$, $|N| \bmod Q = 0$ et $|C| = 3$ alors $\beta = 3$ et $\gamma = 3(|N| - r(N))$
- si $Q \geq 4$, $|N| \bmod Q = 0$ et $|C| = 3$ alors $\beta = 2$ et $\gamma = 2(|N| - r(N)) + 1$




$$N = \{a, b, c\}\,;\; S = \{d, e, f\}\,;\; C = \{c\}\,;\; \beta = 2\,;\; \gamma = 2(3 - \tfrac{3}{3}) = 4$$

$$\beta \sum_{e\in(N:N)} x_e^k + \sum_{e\in(C:S)} x_e^k = 2 \times \tfrac{3}{2} + \tfrac{3}{2} = 4,5$$

$$\Downarrow$$

$$\beta \sum_{e\in(N:N)} x_e^k + \sum_{e\in(C:S)} x_e^k > \gamma$$

$$\Downarrow$$

VIOLATION !

FIGURE 6.3 – Détection de coupes de multi-étoiles partielles avec $Q = 3$

**Multi-étoiles homogènes**

Appelons $p$ le plus petit entier compris entre 1 et $|S|$ pour lequel n'importe quel sous-ensemble $W \subseteq S$ de $p$ clients ($|W| = p$) vérifie $r(N \cup W) > r(N)$. Les contraintes (6.25) dérivent des contraintes de multi-étoiles homogènes de Letchford *et al.* [80].

$$\alpha \sum_{e \in (N:N)} x_e^k + \beta \sum_{e \in (N:S)} x_e^k \leq \gamma, \quad \forall k \in \mathbb{M} \tag{6.25}$$

où

- si $p-1 > 2r(N)$ alors $\alpha = |S| - p + 1$, $\beta = 1$ et $\gamma = (|S| - p + 1)(|N| - r(N)) + p - 1$
- si $p-1 \leq 2r(N)$ alors $\alpha = |S| - p + 1$, $\beta = 1$ et $\gamma = (|S| - p + 1)(|N| - r(N)) + p - 1$
- si $p-1 \leq 2r(N)$ alors $\alpha = 2r(N) - p + 3$, $\beta = -1$ et $\gamma = (2r(N) - p + 3)(|N| - r(N)) + p - 1$

**Multi-étoiles généralisées**

Les contraintes (6.26) dérivent de l'inégalité proposée par Gouveia [59], qui a l'avantage de nécessiter une procédure de séparation polynomiale.

$$Q \sum_{e \in (N:N)} x_e^k + \sum_{j \in \{V' \setminus N\}} \sum_{e \in (N:j)} q_j x_e^k \leq Q|N| - \sum_{i \in N} q_i, \quad \forall k \in \mathbb{M} \tag{6.26}$$

## 6.4 Coupes pour la procédure $\mathrm{BC_{RPA}}$

Rappelons que le modèle à un indice (RPA) est obtenu après agrégation des contraintes de (PA) sur $k$. Par conséquent, des coupes valides pour (RPA) peuvent être obtenues en sommant sur $k$ les coupes de capacité, de peigne et de multi-étoiles dont la validité a été prouvée pour le modèle (PA).

### 6.4.1 Contraintes agrégées de capacité

Les contraintes (6.27) résultent de la somme sur $k$ des contraintes de capacité (6.21) et dominent les contraintes de sous-tours utilisées par Duchenne *et al.* [41] pour résoudre le $m$-PVP.

$$\sum_{e \in \delta(0)} X_e \geq 2m\lambda \tag{6.27}$$

### 6.4.2 Contraintes agrégées de peigne

Les contraintes (6.28) résultent de la somme sur $k$ des contraintes de peigne renforcées (6.23) et dominent les contraintes de peigne fortes proposées par Duchenne *et al.* [41] pour

résoudre le $m$-PVP. En effet, ces dernières résultent de l'agrégation des contraintes définies par Grötschel et Padberg [60] pour le TSP, lesquelles sont dominées par les contraintes d'Araque [3] que nous avons agrégées.

$$\sum_{e\in\delta(H)} X_e + \sum_{e\in\delta(T_j)} \sum_{j=1}^{t} X_e \geq S(H,T_1,..,T_t) + m \tag{6.28}$$

où

- $S(H,T_1,..,T_t) = \sum_{j=1}^{t}(\tilde{r}(T_j\cap H)+\tilde{r}(T_j\setminus H)+\tilde{r}(T_j))$ avec $\tilde{r}(S) = \begin{cases} mr(V\setminus S) \text{ si } 0\in S \\ mr(S) \text{ sinon} \end{cases}$
- $S(H,T_1,..,T_t)$ est impair

### 6.4.3 Contraintes agrégées multi-étoiles

Les contraintes (6.29), (6.30), et (6.31) résultent de la somme sur $k$ des contraintes de multi-étoiles respectivement partielles (6.24), homogènes (6.25) et généralisées (6.26).

- Multi-étoiles partielles :

$$\alpha \sum_{e\in(N:N)} X_e + \beta \sum_{e\in(C:S)} X_e \leq m\gamma \tag{6.29}$$

- Multi-étoiles homogènes :

$$\alpha \sum_{e\in(N:N)} X_e + \beta \sum_{e\in(N:S)} X_e \leq m\gamma \tag{6.30}$$

- Multi-étoiles généralisées :

$$Q \sum_{e\in(N:N)} X_e + \sum_{j\in\{V'\setminus N\}} \sum_{e\in(N:j)} q_j X_e \leq mQ|N| - m\sum_{i\in N} q_i \tag{6.31}$$

avec $\alpha, \beta$ et $\gamma$ dépendant de $|N|$, $|S|$ et $|C|$ comme décrits au paragraphe 6.3.3.

## 6.5 Séparation des inégalités valides du modèle (PA)

Nous utilisons les procédures de séparation des coupes de capacité, de peigne et de multi-étoiles de la bibliothèque de fonctions CVRPSEP [84] développée par Lysgaard [80] pour le VRP, que nous appliquons à chacune des $m$ périodes des solutions fractionnaires du $m$-PTVP trouvées au cours de la procédure de branchements et coupes $\mathrm{BC_{PA}}$ puisqu'elles doivent être des solutions réalisables de VRP. Pour chaque solution $x^*$ du programme linéaire pour laquelle des coupes sont recherchées, on construit le graphe-support $G^k(x^*)$ de la période $k$. Il s'agit du graphe composé de l'ensemble des nœuds $i \in V$ et de l'ensemble des arêtes utilisées au cours de la période $k$ par la solution $x^*$ : $x_e^{*k} > 0$. C'est sur ce

graphe $G^k(x^*)$ que sont recherchées les inégalités violées avec CVRPSEP comme le montre l'algorithme 4. Notons que les coupes trouvées au cours d'une période sont appliquées par symétrie à toutes les $m$ périodes du modèle pour éliminer totalement les violations correspondantes.

**Algorithm 4** Séparation de coupes sur le modèle (PA) grâce aux fonctions de CVRPSEP

```
1: Coupes := ∅ ; Model := PA ; stop := false
2: répéter
3:    Optimiser(Model, x*)
4:    pour k := 1 to m faire
5:       RecupGrapheSupport({e : x*_e^k > 0}, G^k(x*))
6:       CVRPSEP_CoupesCapacite(G^k(x*), Coupes)
7:       si Coupes := ∅ alors
8:          CVRPSEP_CoupesPeigne(G^k(x*), Coupes)
9:          si Coupes := ∅ alors
10:            CVRPSEP_CoupesMultiEtoiles(G^k(x*), Coupes)
11:            stop := (Coupes = ∅)
12:         fin si
13:      fin si
14:      Model := Model ∪ Coupes
15:   fin pour
16: jusqu'à stop
17: renvoyer x*.
```

### 6.5.1 Séparation des contraintes de capacité

La procédure de séparation des coupes de capacité est composée d'une heuristique principale et de trois heuristiques secondaires appelées lorsque l'heuristique principale ne trouve aucune coupe à ajouter. L'heuristique principale s'applique sur le graphe-support $G^k(x^*)$ défini plus haut tandis que les trois autres s'appliquent normalement sur un graphe contracté $G^k_c(x^*)$ construit à partir de $G^k(x^*)$.

#### Heuristique principale

L'heuristique principale recherche les composantes connexes qui existent au sein du graphe-support sans dépôt $G(x^*) \setminus \{0\}$. Pour chacune de ces composantes $S_c$, la demande totale et le nombre d'arêtes incidentes sont identifiées, afin de vérifier l'existence de violations des contraintes de capacité en posant $S = S_c$, puis $S = G^k(x^*) \setminus \{S_c\}$ au sein des équations (6.21). Notons que cette procédure ne garantit de trouver une coupe (si elle existe) que dans le cas où $x^*$ est entière, comme l'a fait remarquer Lysgaard au sujet du VRP. Si cette heuristique ne trouve pas de coupe à ajouter, alors on fait appel aux trois autres heuristiques qui s'appliquent sur un graphe contracté $G^k_c(x^*)$ déduit de $G^k(x^*)$.

**Principe de la contraction de graphe**

La contraction de graphe inspirée du VRP consiste à remplacer toute paire de nœuds connectée par une arête $e = (i, j)$ vérifiant la condition (6.32) par un super-nœud $t$ de demande $q_t = q_i + q_j$. Les arêtes reliant les nœuds disparus à un autre nœud $l$ sont remplacées par une arête $e = (t, l)$ de valeur $x^{*k}_{e=(t,l)} = x^{*k}_{e_1=(i,l)} + x^{*k}_{e_2=(j,l)}$ et s'il existe au moins une violation dans le graphe original alors il y aura au moins une violation au sein du graphe contracté.

$$x^{*k}_e \geq 1 \tag{6.32}$$

Lysgaard généralise cette condition pour le VRP en remplaçant un sous-ensemble de nœuds quelconque $S$ par un super-nœud $s$ de demande $q_s = \sum_{i \in S} q_i$ lorsque les conditions de contraction sans perte de coupes (6.33) et (6.34) sont respectées.

$$\sum_{e \in S} x^{*k}_e \leq 2, \tag{6.33}$$

$$\sum_{e \in R} x^{*k}_e \geq 2, \quad \forall R \subset S \tag{6.34}$$

Ces conditions sont valables pour chacune des $m$ périodes du $m$-PTVP qui sont des solutions réalisables de VRP, comme expliqué dans la section 3.2.4.

**Heuristiques secondaires**

La seconde heuristique identifie à partir du graphe contracté $G^k_c(x^*)$ les violations de contraintes de capacité dites fractionnaires, où $r(S)$ est remplacée par la borne inférieure $\frac{\sum_{i \in S} q_i}{Q}$. Ces inégalités, plus faibles que les contraintes initiales (6.21), peuvent être identifiées en temps polynomial en résolvant un problème de flot maximum sur un graphe auxiliaire $G_A$ construit à partir de $G^k_c(x^*)$ de la manière suivante :

1. copier $G^k_c(x^*)$ dans $G_A$ et y ajouter un dépôt final $n + 1$
2. affecter aux arêtes connectant deux clients dans $G_A$ la même valeur que dans $G^k_c(x^*)$
3. affecter aux arêtes de $G_A$ connectant un client à un dépôt $(0, i)$ et $(i, n+1)$ les valeurs respectives $\max\{0, x^{*k}_{e=(0,i)} - q_i/Q\}$ et $\max\{0, q_i/Q - x^{*k}_{e=(0,i)}\}$.

Le graphe $G_A$ ainsi obtenu n'a pas de coût négatif et le calcul de sa coupe minimale $F_{G_A}$, qui sépare 0 de $n+1$, s'effectue en temps polynomial. McCormick *et al.* [86] ont montré que $F_{G^k_c(x^*)}$ se déduit de $F_{G_A}$ en retirant les arêtes connectées à $n + 1$. La relation entre leurs coûts respectifs s'écrit donc $C_{F_{G_A}} = C_{F_{G^k_c(x^*)}} + \Delta$ où $\Delta = \sum_{i \in V'} \max\{0, q_i/Q - x^{*k}_{e=(0,i)}\}$. Par conséquent, si $C_{F_{G_A}} < \Delta$ alors $F_{G^k_c(x^*)}$ définie une inégalité de capacité fractionnaire non respectée. Si $C_{F_{G_A}} \geq \Delta$ alors il n'y a aucune violation de contraintes de capacité fractionnaires. Si $0 \leq C_{F_{G_A}} < \Delta + 2$ alors bien que toutes les contraintes fractionnaires soient respectées, il peut exister des violations des contraintes de capacité initialement définies (6.21).

La troisième heuristique est un algorithme constructif glouton initialisé avec un super-nœud du graphe contracté $G_c^k(x^*)$. Cette procédure étend $S$ initialement réduit à un super-nœud en le fusionnant à chaque itération avec un autre super-nœud jusqu'à qu'il ne soit plus possible d'étendre $S$ sans obtenir un sous-ensemble déjà généré. Après chaque itération, la procédure vérifie l'existence de violation de contraintes de capacité par $S$. Puis le super-nœud $j$ à fusionner avec $S$ est choisi comme étant celui pour lequel $S \cup \{j\}$ n'a pas été précédemment généré et dont la variable d'écart de l'équation (6.21) est minimale. Lorsqu'aucune fusion n'est plus possible, l'algorithme est réinitialisé à partir d'un autre super-nœud.

La quatrième et dernière heuristique passe en revue toutes les coupes de capacité générées depuis le début de l'optimisation, dans le but de les appliquer au graphe-support contracté $G_c^k(x^*)$. Pour chacune de ces inégalités, définie sur un ensemble de clients $S$, cette procédure commence par remplacer $S$ par le plus petit ensemble de super-nœuds de $G_c^k(x^*)$ qui contient $S$. Elle considère ensuite les super-nœuds de cet ensemble par ordre non-décroissant de demande et retire tous ceux dont le retrait fait décroître la valeur de la variable d'écart de l'équation (6.21). A partir de là, l'ensemble est manipulé en retirant, ajoutant ou remplaçant des super-nœuds par d'autres tant que la valeur d'écart continue de décroître. Au final, la coupe obtenue est vérifiée avant d'être ajoutée au modèle et l'heuristique recommence avec une autre des inégalités trouvées précédemment.

### 6.5.2 Séparation des contraintes de peigne

La procédure de séparation des contraintes de peigne utilisée par Lysgaard pour le VRP et que nous avons généralisée pour le $m$-PTVP domine celle d'Augerat *et al.* [9]. Elle s'applique à un graphe contracté $\hat{G}_c^k(x^*)$ déduit de $G^k(x^*)$. Cette procédure recherche les manches potentiels pour lesquels elle identifie ensuite les violations de contraintes de peigne. Les coupes trouvées sont renforcées avant d'être rapportées, mais si aucune n'est trouvée alors une procédure polynomiale identifie les violations de contraintes de 2-matching d'Edmonds [44]. Rappelons que ces dernières sont un cas particulier obtenu lorsque $S(H, T_1, \dots T_t) = 3t$, $|T_j| = 2$ et $\forall\{i,j\} \subset \{1 \dots m\} : T_i \cap T_j = \emptyset$. Le graphe contracté $\hat{G}_c^k(x^*)$ résulte de la contraction des ensembles $S$ de deux ou trois clients tels que $\sum_{\{i,j\}\subset S} x^{*k}_{e=(i,j)} = 2$ et qui vérifient au moins une des deux conditions (6.35) ou (6.36).

$$\exists i \in \{V' \setminus S\} : \sum_{i \in S} x^{*k}_{e=(i,j)} = 1 \tag{6.35}$$

$$\sum_{i \in S} x^{*k}_{e=(0,i)} = 1 \quad \text{et} \quad 2r(V' \setminus S) = 2r(V') = \sum_{e \in \delta(0)} x^{*k}_e \tag{6.36}$$

L'identification des manches potentiels consiste à supprimer toutes les arêtes de $\hat{G}_c^k(x^*)$ et à les y réintroduire par ordre non-décroissant de $|x^{*k}_e - \frac{1}{2}|$, en considérant chaque composante connexe ou biconnexe créée comme un manche potentiel. Cela revient en fait à

supprimer toutes les arêtes de $\hat{G}_c^k(x^*)$ sauf celles qui vérifient $\varepsilon \leq x_e^{*k} \leq 1-\varepsilon$, et ce (implicitement) pour toutes les valeurs de $\varepsilon$ telles que $0 < \varepsilon \leq \frac{1}{2}$. Rappelons qu'un graphe biconnexe est un graphe qui reste connexe après suppression de n'importe lequel de ses nœuds et toutes les arêtes qui y sont connectées.

L'inégalité de 2-matching qui minimise la valeur de la variable d'écart est recherchée pour chaque manche potentiel du graphe contracté $\hat{G}_c^k(x^*)$. Ses dents ne peuvent être en intersection qu'au dépôt, et on peut en déduire une inégalité de peigne valide pour le graphe original $G^k(x^*)$. Ensuite les dents sont modifiées pour renforcer chaque contrainte de peigne trouvée. A cet effet, elles sont agrandies une par une en y ajoutant un nœud à la fois de manière gloutonne. Si aucune coupe valide n'est trouvée, alors l'algorithme pseudo-polynomial de séparation exacte des contraintes de 2-matching est appliqué au graphe contracté $\hat{G}_c^k(x^*)$. Cet algorithme peut produire de nouveaux manches potentiels, sur lesquels la procédure précédemment décrite pourra alors être appliquée pour trouver les violations de coupes de peigne renforcées.

### 6.5.3 Séparation des contraintes de multi-étoiles

La procédure de séparation des contraintes de multi-étoiles de la bibliothèque de fonction CVRPSEP que nous avons généralisée pour le $m$-PTVP est composée des heuristiques polynomiales développées par Letchford *et al.* pour détecter les coupes de multi-étoiles partielles et homogènes ainsi que l'heuristique polynomiale de Gouveia pour détecter les coupes de multi-étoiles généralisées.

L'heuristique de Letchford *et al.* [79] recherche les potentiels noyaux $N$, satellites $S$ et connecteurs $C$ avant d'appliquer une procédure dite *procédure-polygone* pour trouver les violations d'inégalités de multi-étoiles. Chaque noyau $N$ se réduit initialement à un client et s'agrandit à chaque itération par ajout du client $j$ qui vérifie $0 < \sum_{i \in N} x_{e=(i,j)}^{*k} = \max_{l \in \{V' \setminus N\}} \{\sum_{i \in N} x_{e=(i,l)}^{*k} - x_{e=(0,l)}^{*k}\}$. On obtient un noyau potentiel s'il reste des clients qui ne peuvent pas y être insérés. Sinon, la procédure est réinitialisée à partir d'un autre client. Une variante de cette procédure consiste à agrandir chaque noyau avec le client qui minimise la valeur de la variable d'écart de l'équation (6.26). Cette seconde variante est utilisée lorsque la première échoue.

Une fois que l'ensemble des noyaux potentiels a été identifié, les satellites potentiels sont créés également de manière gloutonne : pour chaque noyau $N$, l'ensemble des satellites $S$ est initialisé avec l'ensemble des clients hors de $N$ tels que $\{i \in V' \setminus N : \sum_{j \in N} x_{e=(i,j)}^{*k} > 0\}$. Puis de manière itérative on enlève le satellite $i$ ayant une valeur minimale de $\sum_{j \in N} x_{e=(i,j)}^{*k}$. Pour chaque paire $(N, S)$ la procédure-polygone trouve des nouvelles coupes multi-étoiles à ajouter. Cette procédure approxime en effet la projection du polygone du VRP dans un espace à deux dimensions ayant pour axes $\kappa_1 = \sum_{\{i,j\} \subset N} x_{e=(i,j)}^{*k}$ et $\kappa_2 = \sum_{(i \in C) \cap (j \in S)} x_{e=(i,j)}^{*k}$. Pour ce faire, elle calcule $UB_{\kappa_1}(\kappa_2)$ la borne supérieure de la valeur de $\kappa_1$ en fonction de la valeur de $\kappa_2$, laquelle varie de 0 à $UB_{\kappa_2}$. Il ne reste ensuite plus qu'à calculer l'enveloppe convexe du polygone dans l'espace $(\kappa_2, UB_{\kappa_2})$. Les meilleurs algorithmes pour un tel calcul,

tels que le parcours de Graham par exemple, sont en $O(n \log n)$ pour $n$ points.

L'heuristique de séparation des contraintes de multi-étoiles homogènes est appelée lorsque la précédente échoue. A partir de la liste des nucleus potentiels $N$ déjà identifiés par l'heuristique précédente et de l'ensemble des connecteurs $C$ initialement vide, les étapes suivantes sont appliquées :

1. Fixer $p := \min\{4, |N| - 1\}$
2. Insérer dans $C$ les $p$ clients de $N$ ayant les plus grandes valeurs de $\sum_{i \in \{V' \setminus N\}} x^{*k}_{e=(i,j)}$
3. $S$ est l'ensemble des clients de $V' \setminus N$ connectés à $C$ : $\{i \in V' \setminus N : \sum_{j \in C} x^{*k}_{e=(i,j)} > 0\}$
4. Utiliser la procédure-polygone pour trouver inégalités violées
5. Si $|S| > 2$ retirer de $S$ le client $i$ ayant la plus petite valeur de $\sum_{j \in C} x^{*k}_{e=(i,j)}$ et retour à l'étape 4
6. $p := p - 1$. Si $p > 2$ retourner à l'étape 2, sinon stop.

Enfin, l'heuristique de séparation des coupes de multi-étoiles généralisées est polynomiale. En effet, Letchford *et al.* [79] ont montré qu'en définissant $t_i$ comme une variable binaire égale à 1 si $i \in S$ et 0 sinon, les contraintes de multi-étoiles généralisées peuvent se reformuler en (6.37), ce qui transforme le problème de séparation de coupes en problème de minimisation d'une fonction quadratique en $t$ dont tous les termes quadratiques sont négatifs. Ce problème peut être converti en un problème polynomial de flot maximum / coupe minimale et résolu avec la méthode de Picard et Ratliff [111] par exemple.

$$\sum_{i \in N} A_i t_i + \sum_{\{i,j\} \subset N} t_i (1 - t_j) B_{ij} \geq 0 \tag{6.37}$$

$$\text{avec } A_i = \left( x^{*k}_{e=(0,i)} \left( 1 - \frac{q_i}{Q} \right) - \sum_{j \in V' \setminus \{i\}} \frac{q_i x^{*k}_{e=(i,j)}}{Q} \right) \text{ et } B_{ij} \left( x^{*k}_{e=(i,j)} \left( 1 - \frac{q_i + q_j}{Q} \right) \right).$$

## 6.6 Séparation des inégalités valides du modèle (RPA)

La bibliothèque de fonctions de détection de coupes CVRPSEP de Lysgaard a été modifiée pour détecter les coupes à ajouter au modèle (RPA) du $m$-PTVP. Par exemple, le membre de droite des inégalités testées est désormais lesté du coefficient $m$ à l'instar des contraintes agrégées (6.27), (6.28) et (6.31). La nouvelle bibliothèque (MPCVRPSEP) admet donc en entrée le graphe-support $G(X^*)$ de la solution $X^*$ du modèle agrégé (RPA) et produit en sortie les coupes à ajouter. Le graphe-support est composé de l'ensemble des nœuds $i \in V$ et de l'ensemble des arêtes $e$ telles que $X^*_e > 0$. C'est sur ce graphe $G(X^*)$ que seront recherchées les inégalités violées avec MPCVRPSEP comme le montre l'algorithme 5. En plus de détecter les violations de contraintes agrégées de capacité, de peigne et de multi-étoiles, la bibliothèque de fonctions MPCVRPSEP a également la caractéristique de désactiver toutes les procédures de contraction de graphes, car les conditions de contraction sans perte ne sont plus vérifiées en ce qui concerne (RPA), comme expliqué ci-après.

**Algorithm 5** Séparation de coupes sur (RPA) grâce aux fonctions de MPCVRPSEP

*Coupes* := ∅ ; *Model* := RPA ; *stop* := *false*
**répéter**
  *Optimiser*(*Model*, $X^*$)
  *RecupGrapheSupport*($\{e : X^*_e > 0\}, G(X^*)$)
  *MPCVRPSEP_CoupesCapacite*($G(X^*)$, *Coupes*)
  **si** *Coupes* := ∅ **alors**
    *MPCVRPSEP_CoupesPeigne*($G(X^*)$, *Coupes*)
    **si** *Coupes* := ∅ **alors**
      *MPCVRPSEP_CoupesMultiEtoiles*($G(X^*)$, *Coupes*)
      *stop* := (*Coupes* = ∅)
    **fin si**
  **fin si**
  *Model* := *Model* ∪ *Coupes*
**jusqu'à** *stop*
**renvoyer** $X^*$.

Les procédures de contraction de graphes-supports généralement utilisées pour le VRP sont basées sur le principe selon lequel contracter des arêtes de valeur supérieure ou égale à 1 réduit la taille du graphe-support sans perdre la possibilité de détecter des contraintes violées. Ce principe n'est plus vrai en ce qui concerne le $m$-PTVP, car comme l'illustre la figure (6.4), un graphe contracté $G_c(X^*)$ peut ne présenter aucune violation alors qu'il est évident que des violations existent au sein du graphe-support original $G(X^*)$.

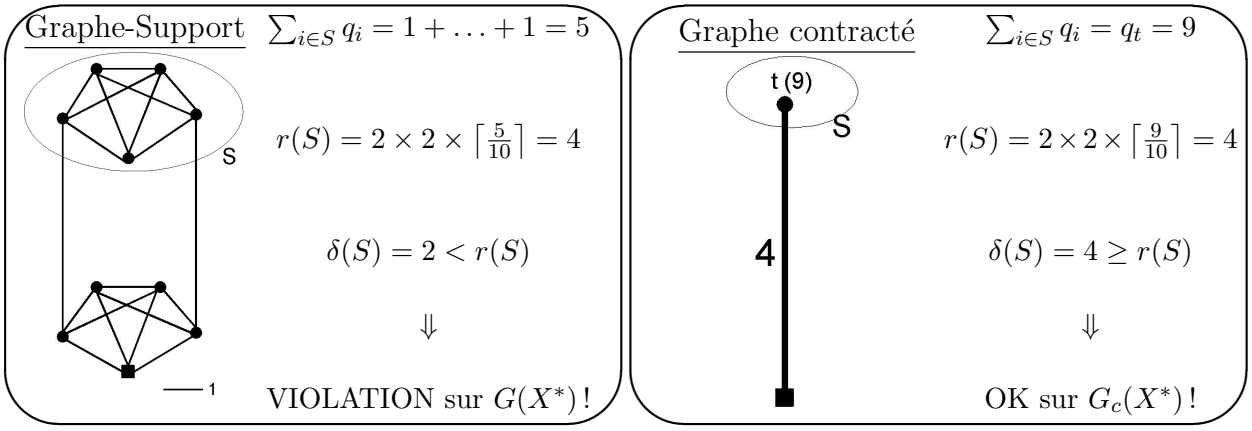



FIGURE 6.4 – Perte de coupes potentielles lors de la contraction du graphe $G(X^*)$

## 6.7 Parcours et branchement sur les arbres

Afin de définir complètement les méthodes de séparation et de coupes $\mathrm{BC_{PA}}$ et $\mathrm{BC_{RPA}}$, il est nécessaire de préciser les règles qui ont été choisies pour le parcours de l'arbre de

recherche de ces algorithmes. Le premier point concerne l'ordre de traitement des nœuds de cet arbre. Nous traitons en priorité le problème dont la solution a le coût le plus faible. Ceci permet d'améliorer rapidement la meilleure borne inférieure connue pour avoir la meilleure borne inférieure possible à la fin de l'algorithme si celui-ci n'a pas pu aller à son terme faute de temps. L'autre règle utilisée consiste à ne pas sonder un nœud et donc "tuer" la branche correspondante si la valeur de sa fonction-objectif est supérieure ou égale à la meilleure borne supérieure connue. Cette règle permet de ne pas traiter un nœud si cela n'a aucune chance d'améliorer le meilleur coût connu. Enfin nous utilisons la règle de branchement qui consiste à choisir en priorité la variable la plus proche de 0,5. Cette règle a fait ses preuves sur des problèmes de tournées de véhicules déjà traités dans la littérature.

## 6.8 Résultats numériques

### 6.8.1 Instances et implémentation

Les algorithmes de branchements et coupes $\mathrm{BC_{PA}}$ et $\mathrm{BC_{RPA}}$ ont été testés sur le 2-PTVP et le 3-PTVP avec les quatre types d'instances A, B, P et VRPNC de 19 à 101 nœuds. Pour plus de détails sur ces instances, se référer au chapitre 3. Les algorithmes ont été codés en C et tous les programmes linéaires ont été résolus à l'aide de CPLEX 10.1 qui gère également les arbres en suivant les règles de parcours et de branchement que nous avons prédéfinies. L'implémentation n'est pas triviale et requiert des connaissances informatiques spécifiques pour récupérer directement les résultats du programme linéaire dans l'application C par le biais d'une DLL (Dynamic Link Library - bibliothèque du logiciel), ainsi que modifier la bibliothèque de fonctions du CVRPSEP codée en C++ et la relier au programme principal pour identifier les coupes valides du $m$-PTVP. Notons également que malgré le fait que les procédures d'agrégation de graphe soient valides pour l'algorithme $\mathrm{BC_{PA}}$, nous les avons désactivées ici pour une comparaison équitable avec $\mathrm{BC_{RPA}}$ puisque ce dernier n'inclut aucune agrégation.

Le PC utilisé est un Pentium de 3,39 GHz et 1 Go de mémoire. Les deux algorithmes sont arrêtés après 10800 secondes (3 heures). Les colonnes suivantes sont utilisées dans les tableaux de résultats 6.1 à 6.14 :

1. $UB_{\mathrm{meta}}$ : meilleure borne supérieure initialement donnée à $\mathrm{BC_{PA}}$ et $\mathrm{BC_{RPA}}$ (trouvée par les métaheuristiques du chapitre 4)
2. $UB_{\mathrm{PA}}$ (resp. $UB_{\mathrm{RPA}}$) : meilleure borne supérieure trouvée par $\mathrm{BC_{PA}}$ (resp. $\mathrm{BC_{RPA}}$)
3. $LB_{\mathrm{PA}}$ (resp. $LB_{\mathrm{RPA}}$) : meilleure borne inférieure trouvée par $\mathrm{BC_{PA}}$ (resp. $\mathrm{BC_{RPA}}$)
4. $\theta\%$ : $\frac{100LB_{\mathrm{PA}}}{UB_{\mathrm{PA}}}$ (ou $\frac{100LB_{\mathrm{RPA}}}{UB_{\mathrm{PA}}}$).
5. $T$ : durée en secondes
6. $CAP$ : nombre de coupes de capacité générées
7. $PEI$ : nombre de coupes de peigne générées
8. $ETO$ : nombre de coupes de multi-étoiles générées

| | $m$ | $UB_{\text{meta}}$ | $UB_{\text{PA}}$ | $LB_{\text{PA}}$ | $\theta\%$ | $T$ | $CAP$ | $PEI$ | $ETO$ |
|---|---|---|---|---|---|---|---|---|---|
| A-n32-k5 | 2 | 1806 | 1806 | * 1806 | 100,00 | 2,3 | 184 | 0 | 0 |
| A-n33-k5 | 2 | 1501 | 1501 | * 1501 | 100,00 | 50,77 | 724 | 90 | 258 |
| A-n33-k6 | 2 | 1653 | 1647 | * 1647 | 100,00 | 2648,48 | 3032 | 872 | 7784 |
| A-n34-k5 | 2 | 1690 | 1689 | * 1689 | 100,00 | 991,47 | 2232 | 626 | 2460 |
| A-n36-k5 | 2 | 1833 | 1818 | 1813 | 99,74 | 10800,2 | 3184 | 3480 | 8036 |
| A-n37-k5 | 2 | 1583 | 1574 | * 1574 | 100,00 | 484,55 | 2456 | 530 | 2260 |
| A-n37-k6 | 2 | 2040 | 2040 | 1996 | 97,84 | 3601,22 | 3958 | 1538 | 8722 |
| A-n38-k5 | 2 | 1655 | 1655 | 1627 | 98,29 | 3600,69 | 4192 | 1978 | 5994 |
| A-n39-k5 | 2 | 1809 | 1809 | 1777 | 98,21 | 10800,14 | 6908 | 2506 | 17862 |
| A-n39-k6 | 2 | 1929 | 1915 | 1899 | 99,18 | 10800,05 | 6742 | 3814 | 11728 |
| A-n44-k7 | 2 | 2065 | 2065 | 2008 | 97,26 | 10803,91 | 4450 | 1234 | 26828 |
| A-n45-k6 | 2 | 2101 | 2101 | 2017 | 95,98 | 10800,5 | 4688 | 998 | 15640 |
| A-n45-k7 | 2 | 2453 | 2453 | 2385 | 97,21 | 10800,76 | 9416 | 536 | 22612 |
| A-n46-k7 | 2 | 2100 | 2100 | 2052 | 97,71 | 10807,42 | 5312 | 784 | 17682 |
| A-n48-k7 | 2 | 2346 | 2346 | 2301 | 98,10 | 10800,66 | 5706 | 1652 | 16662 |
| A-n53-k7 | 2 | 2241 | 2241 | 2187 | 97,61 | 10800,86 | 11002 | 1328 | 12130 |
| A-n54-k7 | 2 | 2516 | 2516 | 2405 | 95,60 | 10802,25 | 11008 | 722 | 12316 |
| A-n55-k9 | 2 | 2366 | 2366 | 2316 | 97,87 | 10800,16 | 10004 | 1274 | 16332 |
| A-n60-k9 | 2 | 2895 | 2895 | 2780 | 96,04 | 10801,03 | 13608 | 274 | 11486 |
| A-n61-k9 | 2 | 2311 | 2311 | 2196 | 95,00 | 10803,76 | 15876 | 1102 | 9652 |
| A-n62-k8 | 2 | 2791 | 2791 | 2682 | 96,08 | 10803,02 | 13440 | 316 | 9240 |
| A-n63-k9 | 2 | 3454 | 3454 | 3331 | 96,44 | 3601,64 | 9996 | 194 | 4026 |
| A-n63-k10 | 2 | 2844 | 2844 | 2735 | 96,17 | 3605,3 | 11446 | 84 | 5226 |
| A-n64-k9 | 2 | 3040 | 3040 | 2922 | 96,11 | 3601,19 | 10690 | 234 | 4186 |
| A-n65-k9 | 2 | 2611 | 2611 | 2510 | 96,13 | 10801,14 | 11284 | 968 | 9742 |
| A-n69-k9 | 2 | 2602 | 2602 | 2507 | 96,36 | 10810,38 | 11204 | 1018 | 7386 |
| A-n80-k10 | 2 | 3829 | 3829 | 3687 | 96,30 | 10855,08 | 15590 | 336 | 3576 |
| Moyenne | 2 | | | | 97,60 | 7625,15 | 7716 | 1055 | 9994 |

TABLE 6.1 – Résultats de $BC_{PA}$ sur instances de type A pour le 2-PTVP

### 6.8.2 Résultats de $\text{BC}_{\text{PA}}$

Les tableaux 6.1 à 6.6 présentent les résultats obtenus par l'algorithme de branchements et coupes $\text{BC}_{\text{PA}}$ qui résout la formulation (PA) du $m$-PTVP. Par rapport à celui de la borne inférieure polynomiale LB3 présentée au chapitre 3, le ratio $\theta_{\%}$ de $LB_{\text{PA}}$ augmente en moyenne de 5 à 10 points. A présent l'écart entre les meilleures bornes inférieures et les meilleures bornes supérieures est en moyenne ramené à 1,9% pour $m = 2$ et 2,1% pour $m = 3$. Un tiers des instances sont résolues optimalement, en général celles de petites tailles (par exemple $n \leq 37$ pour le type A), et le nombre décroît lorsque $m$ augmente. Pour le reste des instances, la limite de temps choisie de 10800 secondes est insuffisante pour atteindre la solution optimale. En moyenne, la meilleure solution trouvée par les métaheuristiques du chapitre 4 est améliorée de 1% par l'algorithme $\text{BC}_{\text{PA}}$.

### 6.8.3 Résultats de $\text{BC}_{\text{RPA}}$

Les tableaux 6.9 à 6.14 présentent les résultats obtenus par l'algorithme de branchements et coupes $\text{BC}_{\text{RPA}}$ qui résout la formulation relaxée (RPA) du $m$-PTVP. A l'exception de 7 cas sur 274, les instances sont résolues optimalement. En moyenne, l'algorithme $\text{BC}_{\text{RPA}}$ produit des résultats inférieurs à ceux de $\text{BC}_{\text{RPA}}$ d'à peine 0,5% pour un temps de calculs cent fois inférieur. Ces temps de calculs deviennent d'ailleurs négligeables lorsque $m$ aug-

| | $m$ | $UB_{\text{meta}}$ | $UB_{\text{PA}}$ | $LB_{\text{PA}}$ | $\theta\%$ | $T$ | $CAP$ | $PEI$ | $ETO$ |
|---|---|---|---|---|---|---|---|---|---|
| A-n32-k5 | 3 | 3368 | 3340 | * 3340 | 100,00 | 43,9 | 2523 | 54 | 132 |
| A-n33-k5 | 3 | 2684 | 2669 | * 2669 | 100,00 | 82,7 | 5634 | 99 | 318 |
| A-n34-k5 | 3 | 3010 | 2987 | 2976 | 99,63 | 10802,4 | 26844 | 5415 | 13908 |
| A-n36-k5 | 3 | 3258 | 3229 | * 3229 | 100,00 | 3669,4 | 12486 | 2649 | 3525 |
| A-n37-k5 | 3 | 2774 | 2751 | * 2751 | 100,00 | 109,0 | 6180 | 159 | 453 |
| A-n37-k6 | 3 | 3754 | 3754 | 3629 | 96,67 | 10800,5 | 31590 | 2208 | 15714 |
| A-n38-k5 | 3 | 2900 | 2851 | * 2851 | 100,00 | 7815,3 | 21426 | 2031 | 6045 |
| A-n39-k5 | 3 | 3133 | 3133 | 3060 | 97,67 | 10801,2 | 20664 | 2895 | 12852 |
| A-n39-k6 | 3 | 3468 | 3417 | * 3417 | 100,00 | 3123,8 | 16308 | 1515 | 5436 |
| A-n44-k7 | 3 | 3558 | 3558 | 3487 | 98,00 | 10801,2 | 25020 | 1869 | 13041 |
| A-n45-k6 | 3 | 3856 | 3856 | 3583 | 92,91 | 10800,8 | 26115 | 3033 | 7008 |
| A-n48-k7 | 3 | 4158 | 4158 | 4103 | 98,68 | 10801,9 | 36015 | 894 | 10083 |
| A-n53-k7 | 3 | 3908 | 3908 | 3817 | 97,67 | 10801,5 | 26751 | 1407 | 6255 |
| A-n54-k7 | 3 | 4168 | 4168 | 4017 | 96,37 | 10801,2 | 21720 | 1503 | 8181 |
| A-n55-k9 | 3 | 4177 | 4177 | 4085 | 97,80 | 10812,1 | 27819 | 1491 | 14205 |
| A-n60-k9 | 3 | 4834 | 4834 | 4733 | 97,91 | 10801,5 | 23367 | 924 | 10926 |
| A-n61-k9 | 3 | 4035 | 4035 | 3794 | 94,03 | 10962,1 | 23172 | 447 | 4278 |
| A-n62-k8 | 3 | 4769 | 4769 | 4649 | 97,48 | 10809,9 | 26946 | 693 | 6393 |
| A-n63-k9 | 3 | 5888 | 5888 | 5680 | 96,47 | 11051,5 | 23028 | 966 | 4752 |
| A-n63-k10 | 3 | 4967 | 4967 | 4823 | 97,10 | 10873,2 | 27999 | 495 | 6753 |
| A-n64-k9 | 3 | 5271 | 5271 | 5150 | 97,70 | 3601,9 | 23718 | 717 | 2814 |
| A-n65-k9 | 3 | 4420 | 4420 | 4275 | 96,71 | 10804,7 | 24537 | 633 | 7683 |
| A-n69-k9 | 3 | 4457 | 4457 | 4316 | 96,84 | 10806,9 | 24972 | 576 | 4515 |
| A-n80-k10 | 3 | 6454 | 6454 | 6309 | 97,75 | 35,4 | 21657 | 0 | 0 |
| Moyenne | 3 | | | | 97,81 | 7992,24 | 21937 | 1361 | 6886 |

TABLE 6.2 – Résultats de $BC_{PA}$ sur instances de type A pour le 3-PTVP

| | $m$ | $UB_{\text{meta}}$ | $UB_{\text{PA}}$ | $LB_{\text{PA}}$ | $\theta\%$ | $T$ | $CAP$ | $PEI$ | $ETO$ |
|---|---|---|---|---|---|---|---|---|---|
| B-n31-k5 | 2 | 1387 | 1387 | * 1387 | 100,00 | 25,95 | 1028 | 12 | 554 |
| B-n34-k5 | 2 | 1746 | 1745 | * 1745 | 100,00 | 192,86 | 1688 | 208 | 1066 |
| B-n35-k5 | 2 | 2020 | 2020 | * 2016 | 99,80 | 3,05 | 676 | 12 | 0 |
| B-n38-k6 | 2 | 1688 | 1687 | * 1687 | 100,00 | 22,81 | 868 | 24 | 232 |
| B-n39-k5 | 2 | 1190 | 1190 | * 1190 | 100,00 | 26,95 | 446 | 4 | 542 |
| B-n41-k6 | 2 | 1772 | 1772 | 1738 | 98,05 | 10800,31 | 8694 | 1044 | 30276 |
| B-n43-k6 | 2 | 1609 | 1590 | * 1590 | 100,00 | 8165 | 10266 | 572 | 15362 |
| B-n44-k7 | 2 | 1913 | 1901 | 1895 | 99,71 | 10800,92 | 5248 | 628 | 17286 |
| B-n45-k5 | 2 | 1601 | 1601 | 1586 | 99,03 | 10800,41 | 6534 | 3380 | 9424 |
| B-n45-k6 | 2 | 1531 | 1531 | 1483 | 96,84 | 10801,03 | 7590 | 1104 | 25324 |
| B-n50-k7 | 2 | 1572 | 1571 | * 1571 | 100,00 | 11,48 | 828 | 6 | 178 |
| B-n50-k8 | 2 | 2730 | 2730 | 2653 | 97,19 | 10806,27 | 12516 | 222 | 19896 |
| B-n51-k7 | 2 | 2140 | 2136 | * 2136 | 100,00 | 111,56 | 2224 | 144 | 552 |
| B-n52-k7 | 2 | 1602 | 1594 | * 1594 | 100,00 | 27,61 | 1396 | 32 | 156 |
| B-n56-k7 | 2 | 1506 | 1502 | 1494 | 99,48 | 10800,31 | 8048 | 832 | 13876 |
| B-n57-k7 | 2 | 2382 | 2382 | 2356 | 98,89 | 3600,41 | 10276 | 570 | 9550 |
| B-n57-k9 | 2 | 3314 | 3314 | 3286 | 99,15 | 10833,3 | 7878 | 154 | 22156 |
| B-n63-k10 | 2 | 3194 | 3194 | 3102 | 97,12 | 10801,45 | 10540 | 428 | 13964 |
| B-n64-k9 | 2 | 1868 | 1868 | 1818 | 97,32 | 10801,14 | 16784 | 1222 | 12214 |
| B-n66-k9 | 2 | 2833 | 2833 | 2751 | 97,09 | 10815,55 | 13626 | 1428 | 12022 |
| B-n67-k10 | 2 | 2222 | 2222 | 2190 | 98,54 | 10800,98 | 11288 | 482 | 16432 |
| B-n68-k9 | 2 | 2663 | 2663 | 2597 | 97,52 | 10801,16 | 14722 | 384 | 11134 |
| B-n78-k10 | 2 | 2613 | 2613 | 2552 | 97,67 | 5714,02 | 20912 | 154 | 2030 |
| Moyenne | 2 | | | | 98,84 | 6415,85 | 7569 | 567 | 10184 |

TABLE 6.3 – Résultats de $BC_{PA}$ sur instances de type B pour le 2-PTVP

| | $m$ | $UB_{\text{meta}}$ | $UB_{\text{PA}}$ | $LB_{\text{PA}}$ | $\theta\%$ | $T$ | $CAP$ | $PEI$ | $ETO$ |
|---|---|---|---|---|---|---|---|---|---|
| B-n31-k5 | 3 | 2607 | 2603 | * 2603 | 100,00 | 17,7 | 2265 | 24 | 93 |
| B-n34-k5 | 3 | 2903 | 2892 | 2890 | 99,93 | 3602,9 | 13602 | 1140 | 6129 |
| B-n35-k5 | 3 | 3222 | 3205 | * 3205 | 100,00 | 33,6 | 2946 | 45 | 66 |
| B-n38-k6 | 3 | 2860 | 2860 | * 2849 | 99,62 | 82,7 | 6423 | 66 | 108 |
| B-n39-k5 | 3 | 2076 | 2018 | * 2018 | 100,00 | 41,0 | 2133 | 99 | 282 |
| B-n41-k6 | 3 | 3015 | 3015 | 2939 | 97,48 | 10801,6 | 26307 | 1521 | 16125 |
| B-n43-k6 | 3 | 2733 | 2733 | 2683 | 98,17 | 10802,1 | 27120 | 3012 | 14634 |
| B-n44-k7 | 3 | 3217 | 3173 | * 3173 | 100,00 | 4266,7 | 12396 | 672 | 5790 |
| B-n45-k5 | 3 | 2666 | 2666 | 2578 | 96,70 | 10827,4 | 24303 | 3672 | 10578 |
| B-n45-k6 | 3 | 2585 | 2585 | 2497 | 96,60 | 10800,4 | 20679 | 1515 | 13038 |
| B-n50-k7 | 3 | 2616 | 2606 | * 2606 | 100,00 | 9,8 | 1005 | 24 | 0 |
| B-n50-k8 | 3 | 4488 | 4488 | 4316 | 96,17 | 10874,8 | 26922 | 606 | 9753 |
| B-n51-k7 | 3 | 3542 | 3542 | 3456 | 97,57 | 10820,7 | 29091 | 1929 | 10341 |
| B-n52-k7 | 3 | 2709 | 2676 | * 2676 | 100,00 | 2656,7 | 16500 | 627 | 5391 |
| B-n56-k7 | 3 | 2427 | 2427 | 2398 | 98,81 | 10801,6 | 19365 | 1839 | 15267 |
| B-n57-k7 | 3 | 4127 | 4127 | 3843 | 93,12 | 10801,2 | 20454 | 903 | 3189 |
| B-n57-k9 | 3 | 5490 | 5490 | 5447 | 99,21 | 10801,1 | 27942 | 609 | 13494 |
| B-n63-k10 | 3 | 5553 | 5553 | 5475 | 98,60 | 10801,6 | 27357 | 459 | 10440 |
| B-n64-k9 | 3 | 3286 | 3286 | 3054 | 92,94 | 10841,1 | 24036 | 1461 | 3312 |
| B-n66-k9 | 3 | 4696 | 4696 | 4547 | 96,83 | 10804,1 | 20907 | 882 | 10935 |
| B-n67-k10 | 3 | 3693 | 3693 | 3605 | 97,62 | 10810,3 | 31197 | 345 | 6783 |
| B-n68-k9 | 3 | 4336 | 4336 | 4143 | 95,55 | 10832,0 | 22896 | 363 | 6912 |
| B-n78-k10 | 3 | 4234 | 4234 | 4096 | 96,74 | 435,7 | 23187 | 0 | 0 |
| Moyenne | 3 | | | | 97,90 | 7068,11 | 18654 | 948 | 7072 |

TABLE 6.4 – Résultats de $BC_{PA}$ sur instances de type B pour le 3-PTVP

| | $m$ | $UB_{\text{meta}}$ | $UB_{\text{PA}}$ | $LB_{\text{PA}}$ | $\theta\%$ | $T$ | $CAP$ | $PEI$ | $ETO$ |
|---|---|---|---|---|---|---|---|---|---|
| P-n19-k2 | 2 | 501 | 500 | * 500 | 100,00 | 12,27 | 316 | 56 | 94 |
| P-n20-k2 | 2 | 511 | 511 | * 511 | 100,00 | 40,44 | 372 | 298 | 212 |
| P-n21-k2 | 2 | 511 | 507 | * 507 | 100,00 | 3,06 | 230 | 16 | 24 |
| P-n22-k2 | 2 | 512 | 511 | * 511 | 100,00 | 12,48 | 306 | 136 | 32 |
| P-n40-k5 | 2 | 1072 | 1065 | * 1065 | 100,00 | 1745,08 | 3182 | 1894 | 3330 |
| P-n45-k5 | 2 | 1170 | 1148 | * 1148 | 100,00 | 903,36 | 3428 | 866 | 1440 |
| P-n50-k10 | 2 | 1549 | 1549 | 1504 | 97,07 | 10800,25 | 12750 | 1296 | 18110 |
| P-n50-k7 | 2 | 1257 | 1257 | 1247 | 99,22 | 10801,03 | 7730 | 2314 | 14758 |
| P-n50-k8 | 2 | 1424 | 1424 | 1332 | 93,54 | 10809,19 | 9242 | 1948 | 17498 |
| P-n51-k10 | 2 | 1678 | 1678 | 1611 | 96,02 | 10801,83 | 14520 | 918 | 14880 |
| P-n55-k10 | 2 | 1543 | 1543 | 1515 | 98,17 | 10801,59 | 10782 | 848 | 19660 |
| P-n55-k7 | 2 | 1286 | 1286 | 1271 | 98,83 | 10801,03 | 12114 | 2108 | 12568 |
| P-n55-k8 | 2 | 1294 | 1294 | 1271 | 98,22 | 10806,7 | 11334 | 2330 | 16264 |
| P-n60-k10 | 2 | 1659 | 1659 | 1599 | 96,35 | 10800,69 | 11378 | 1210 | 16736 |
| P-n65-k10 | 2 | 1761 | 1761 | 1710 | 97,10 | 10800,75 | 13180 | 456 | 13680 |
| P-n70-k10 | 2 | 1826 | 1826 | 1735 | 95,03 | 10801 | 12906 | 654 | 10060 |
| P-n76-k4 | 2 | 1385 | 1385 | 1352 | 97,62 | 10808,42 | 13006 | 1808 | 4194 |
| P-n76-k5 | 2 | 1440 | 1440 | 1405 | 97,57 | 10801,36 | 13836 | 1424 | 4128 |
| P-n101-k4 | 2 | 1581 | 1581 | 1557 | 98,45 | 1371,63 | 10542 | 1474 | 456 |
| Moyenne | 2 | | | | 98,06 | 7038,01 | 8482 | 1161 | 8849 |

TABLE 6.5 – Résultats de $BC_{PA}$ sur instances de type P pour le 2-PTVP

| | $m$ | $UB_{\text{meta}}$ | $UB_{\text{PA}}$ | $LB_{\text{PA}}$ | $\theta\%$ | $T$ | $CAP$ | $PEI$ | $ETO$ |
|---|---|---|---|---|---|---|---|---|---|
| P-n19-k2 | 3 | 882 | 881 | * 879 | 99,77 | 64,7 | 1668 | 537 | 282 |
| P-n20-k2 | 3 | 897 | 897 | * 893 | 99,55 | 24,4 | 1293 | 210 | 168 |
| P-n21-k2 | 3 | 898 | 898 | * 898 | 100,00 | 23,6 | 963 | 126 | 30 |
| P-n22-k2 | 3 | 906 | 901 | * 901 | 100,00 | 51,6 | 2283 | 564 | 231 |
| P-n40-k5 | 3 | 1869 | 1853 | * 1853 | 100,00 | 4990,0 | 24339 | 1719 | 5229 |
| P-n45-k5 | 3 | 2063 | 2063 | 2037 | 98,74 | 10802,4 | 29526 | 3339 | 9240 |
| P-n50-k7 | 3 | 2203 | 2203 | 2177 | 98,82 | 10801,7 | 34779 | 936 | 12621 |
| P-n55-k7 | 3 | 2244 | 2244 | 2225 | 99,15 | 10804,0 | 40782 | 1023 | 5367 |
| P-n55-k8 | 3 | 2259 | 2233 | 2225 | 99,64 | 10803,8 | 27909 | 1815 | 9390 |
| P-n65-k10 | 3 | 3029 | 3029 | 2976 | 98,25 | 10801,6 | 31971 | 711 | 9015 |
| P-n70-k10 | 3 | 3159 | 3159 | 3026 | 95,79 | 7781,2 | 24987 | 252 | 1572 |
| P-n76-k4 | 3 | 2374 | 2374 | 2317 | 97,60 | 163,4 | 15435 | 0 | 0 |
| P-n76-k5 | 3 | 2464 | 2464 | 2413 | 97,93 | 10802,1 | 15972 | 981 | 1284 |
| P-n101-k4 | 3 | 2702 | 2702 | 2658 | 98,37 | 29,8 | 7470 | 0 | 0 |
| Moyenne | 3 | | | | 98,83 | 5567,45 | 18527 | 872 | 3888 |

TABLE 6.6 – Résultats de $BC_{PA}$ sur instances de type P pour le 3-PTVP

| | $m$ | $UB_{\text{meta}}$ | $UB_{\text{PA}}$ | $LB_{\text{PA}}$ | $\theta\%$ | $T$ | $CAP$ | $PEI$ | $ETO$ |
|---|---|---|---|---|---|---|---|---|---|
| VRPNC1 | 2 | 1198,38 | 1198,38 | 1175,12 | 98,06 | 10803,34 | 5348 | 3864 | 8962 |
| VRPNC2 | 2 | 1846,10 | 1846,10 | 1760,43 | 95,36 | 10806,03 | 11822 | 356 | 6348 |
| VRPNC3 | 2 | 1871,83 | 1871,83 | 1814,32 | 96,93 | 101,70 | 14918 | 0 | 0 |
| VRPNC4 | 2 | 2381,38 | 2381,38 | 2254,66 | 94,68 | 51,22 | 9406 | 0 | 0 |
| VRPNC11 | 2 | 2318,86 | 2318,86 | 1963,66 | 84,68 | 32,02 | 12970 | 0 | 0 |
| VRPNC12 | 2 | 1821,71 | 1810,58 | 1810,40 | 99,99 | 10146,61 | 16056 | 662 | 7608 |
| Moyenne | 2 | | | | 94,95 | 5323,49 | 11753 | 814 | 3820 |

TABLE 6.7 – Résultats de $BC_{PA}$ sur instances de type VRPNC pour le 2-PTVP

| | $m$ | $UB_{\text{meta}}$ | $UB_{\text{PA}}$ | $LB_{\text{PA}}$ | $\theta\%$ | $T$ | $CAP$ | $PEI$ | $ETO$ |
|---|---|---|---|---|---|---|---|---|---|
| VRPNC1 | 3 | 2127,77 | 2127,77 | 2089,43 98,20 | 10822,69 | 30045 | 3435 | 7023 | |
| VRPNC2 | 3 | 3184,81 | 3184,81 | 3056,23 95,96 | 887,45 | 23136 | 15 | 543 | |
| VRPNC3 | 3 | 3174,74 | 3174,74 | 3109,72 97,95 | 48,31 | 13797 | 0 | 0 | |
| VRPNC4 | 3 | 4023,86 | 4023,86 | 3898,58 96,89 | 101,05 | 23466 | 0 | 0 | |
| VRPNC12 | 3 | 3000,82 | 3000,82 | 2977,40 99,22 | 832,09 | 26346 | 3 | 105 | |
| Moyenne | 3 | | | | 97,64 | 2538,32 | 23358 | 691 | 1534 |

TABLE 6.8 – Résultats de $BC_{PA}$ sur instances de type VRPNC pour le 3-PTVP

| | $m$ | $UB_{\text{meta}}$ | $UB_{\text{RPA}}$ | $LB_{\text{RPA}}$ | $\theta\%$ | $T$ | $CAP$ | $PEI$ | $ETO$ |
|---|---|---|---|---|---|---|---|---|---|
| A-n32-k5 | 2 | 1806 | 1806 | * 1770 | 98,01 | 0,28 | 22 | 0 | 0 |
| A-n33-k5 | 2 | 1501 | 1501 | * 1421 | 94,67 | 0,22 | 60 | 0 | 0 |
| A-n33-k6 | 2 | 1653 | 1634 | * 1634 | 98,85 | 2,61 | 43 | 0 | 31 |
| A-n34-k5 | 2 | 1690 | 1675 | * 1675 | 99,11 | 3,78 | 66 | 0 | 164 |
| A-n36-k5 | 2 | 1833 | 1808 | * 1808 | 98,64 | 1,88 | 27 | 0 | 184 |
| A-n37-k5 | 2 | 1583 | 1583 | * 1564 | 98,80 | 0,86 | 38 | 0 | 0 |
| A-n37-k6 | 2 | 2040 | 1995 | * 1995 | 97,79 | 4,28 | 142 | 0 | 66 |
| A-n38-k5 | 2 | 1655 | 1615 | * 1615 | 97,58 | 2,13 | 95 | 0 | 11 |
| A-n39-k5 | 2 | 1809 | 1774 | * 1774 | 98,07 | 6,7 | 246 | 0 | 288 |
| A-n39-k6 | 2 | 1929 | 1895 | * 1895 | 98,24 | 5,47 | 122 | 0 | 23 |
| A-n44-k7 | 2 | 2065 | 2018 | * 2018 | 97,72 | 288,72 | 207 | 0 | 1342 |
| A-n45-k6 | 2 | 2101 | 2013 | * 2013 | 95,81 | 50,95 | 203 | 0 | 796 |
| A-n45-k7 | 2 | 2453 | 2391 | * 2391 | 97,47 | 181,97 | 480 | 0 | 1129 |
| A-n46-k7 | 2 | 2100 | 2067 | * 2067 | 98,43 | 336,84 | 215 | 0 | 1144 |
| A-n48-k7 | 2 | 2346 | 2308 | * 2308 | 98,38 | 158,5 | 311 | 0 | 594 |
| A-n53-k7 | 2 | 2241 | 2183 | * 2183 | 97,41 | 6,58 | 408 | 0 | 128 |
| A-n54-k7 | 2 | 2516 | 2416 | * 2416 | 96,03 | 448,39 | 1329 | 0 | 1511 |
| A-n55-k9 | 2 | 2366 | 2312 | * 2312 | 97,72 | 7,77 | 592 | 0 | 50 |
| A-n60-k9 | 2 | 2895 | 2788 | * 2788 | 96,30 | 223,44 | 997 | 0 | 1068 |
| A-n61-k9 | 2 | 2311 | 2199 | * 2199 | 95,15 | 34,52 | 945 | 0 | 166 |
| A-n62-k8 | 2 | 2791 | 2720 | 2687 | 96,28 | 3601,5 | 1001 | 0 | 5371 |
| A-n63-k9 | 2 | 3454 | 3359 | 3335 | 96,54 | 3600,02 | 817 | 0 | 5341 |
| A-n63-k10 | 2 | 2844 | 2743 | * 2743 | 96,45 | 2829,59 | 856 | 0 | 2387 |
| A-n64-k9 | 2 | 3040 | 2934 | * 2934 | 96,51 | 1285,59 | 784 | 0 | 2033 |
| A-n65-k9 | 2 | 2611 | 2515 | * 2515 | 96,32 | 90,11 | 893 | 0 | 257 |
| A-n69-k9 | 2 | 2602 | 2509 | * 2509 | 96,43 | 26,2 | 1027 | 0 | 217 |
| A-n80-k10 | 2 | 3829 | 3710 | 3694 | 96,48 | 3615,2 | 1418 | 0 | 5552 |
| Moyenne | 2 | | | | 97,32 | 622,74 | 494 | 0 | 1106 |

TABLE 6.9 – Résultats de $BC_{RPA}$ sur instances de type A pour le 2-PTVP

mente car le nombre de coupes valides à ajouter diminue fortement. Enfin, aucune coupe de peigne n'a été détectée dans la résolution des instances choisies.

## 6.9 Conclusion

Ce chapitre introduit deux approches polyédrales pour la résolution du problème de tournées de véhicules $m$-péripatétiques ($m$-PTVP). Ce problème consiste à extraire d'un graphe complet des tournées de véhicules disjointes de coût total minimal sur $m$ périodes telles que chaque client soit visité exactement une fois par période. Il peut être considéré comme une généralisation de deux problèmes NP-difficiles connus que sont le VRP, correspondant au cas mono-période 1-PTVP, et le $m$-PVP, correspondant au cas mono-véhicule.

Les deux méthodes de branchements et coupes proposées sont basées respectivement sur la formulation (PA) présentée au chapitre 2 et une relaxation de ce modèle. Nous avons identifié parmi les inégalités efficaces pour le VRP celles qui sont généralisables au $m$-PTVP, montré les limites de celles qui ne le sont pas, et adapté les procédures de séparation en conséquence. Ces inégalités dominent souvent celles utilisées pour résoudre le $m$-PVP. Les deux algorithmes obtenus nous ont amenés à modifier profondément la bibliothèque de fonctions de détection de coupes violées CVRPSEP de Lysgaard pour le

| | $m$ | $UB_{\text{meta}}$ | $UB_{\text{RPA}}$ | $LB_{\text{RPA}}$ | $\theta\%$ | $T$ | $CAP$ | $PEI$ | $ETO$ |
|---|---|---|---|---|---|---|---|---|---|
| A-n32-k5 | 3 | 3368 | 3368 | * 3340 | 100,00 | 0,33 | 0 | 0 | 0 |
| A-n33-k5 | 3 | 2684 | 2672 | * 2669 | 100,00 | 1,03 | 0 | 0 | 0 |
| A-n34-k5 | 3 | 3010 | 3010 | * 2976 | 99,63 | 0,19 | 0 | 0 | 0 |
| A-n36-k5 | 3 | 3258 | 3258 | * 3229 | 100,00 | 0,28 | 2 | 0 | 0 |
| A-n37-k5 | 3 | 2774 | 2774 | * 2751 | 100,00 | 0,22 | 0 | 0 | 0 |
| A-n37-k6 | 3 | 3754 | 3641 | * 3629 | 96,67 | 0,67 | 0 | 0 | 0 |
| A-n38-k5 | 3 | 2900 | 2900 | * 2851 | 100,00 | 0,31 | 3 | 0 | 0 |
| A-n39-k5 | 3 | 3133 | 3133 | * 3060 | 97,67 | 0,69 | 29 | 0 | 0 |
| A-n39-k6 | 3 | 3468 | 3468 | * 3416 | 99,97 | 0,25 | 0 | 0 | 0 |
| A-n44-k7 | 3 | 3558 | 3487 | * 3487 | 98,00 | 1,94 | 6 | 0 | 0 |
| A-n45-k6 | 3 | 3856 | 3856 | * 3583 | 92,92 | 1,49 | 10 | 0 | 16 |
| A-n45-k7 | 3 | 4138 | 4138 | * 4073 | 98,43 | 0,41 | 23 | 0 | 0 |
| A-n46-k7 | 3 | 3703 | 3703 | * 3686 | 99,54 | 0,41 | 1 | 0 | 0 |
| A-n48-k7 | 3 | 4158 | 4158 | * 4103 | 98,68 | 0,45 | 2 | 0 | 0 |
| A-n53-k7 | 3 | 3908 | 3908 | * 3817 | 97,67 | 0,63 | 14 | 0 | 0 |
| A-n54-k7 | 3 | 4168 | 4024 | * 4014 | 96,31 | 2,17 | 158 | 0 | 1 |
| A-n55-k9 | 3 | 4177 | 4085 | * 4085 | 97,80 | 6,23 | 0 | 0 | 0 |
| A-n60-k9 | 3 | 4834 | 4734 | * 4734 | 97,93 | 9,73 | 42 | 0 | 10 |
| A-n61-k9 | 3 | 4035 | 4035 | * 3794 | 94,03 | 0,84 | 10 | 0 | 0 |
| A-n62-k8 | 3 | 4769 | 4650 | * 4650 | 97,50 | 6,17 | 33 | 0 | 4 |
| A-n63-k9 | 3 | 5888 | 5888 | * 5680 | 96,47 | 1,59 | 9 | 0 | 3 |
| A-n63-k10 | 3 | 4967 | 4824 | * 4824 | 97,12 | 5,44 | 6 | 0 | 0 |
| A-n64-k9 | 3 | 5271 | 5271 | * 5150 | 97,70 | 1,5 | 0 | 0 | 0 |
| A-n65-k9 | 3 | 4420 | 4420 | * 4275 | 96,72 | 4,17 | 18 | 0 | 0 |
| A-n69-k9 | 3 | 4457 | 4316 | * 4316 | 96,84 | 9,7 | 28 | 0 | 27 |
| A-n80-k10 | 3 | 6454 | 6314 | * 6314 | 97,83 | 22,34 | 12 | 0 | 30 |
| Moyenne | 3 | | | | 97,90 | 3,05 | 16 | 0 | 4 |

TABLE 6.10 – Résultats de $BC_{RPA}$ sur instances de type A pour le 3-PTVP

| | $m$ | $UB_{\text{meta}}$ | $UB_{\text{RPA}}$ | $LB_{\text{RPA}}$ | $\theta\%$ | $T$ | $CAP$ | $PEI$ | $ETO$ |
|---|---|---|---|---|---|---|---|---|---|
| B-n31-k5 | 2 | 1387 | 1387 | * 1387 | 100 | 1,72 | 61 | 0 | 68 |
| B-n34-k5 | 2 | 1746 | 1746 | * 1740 | 99,66 | 1,08 | 87 | 0 | 26 |
| B-n35-k5 | 2 | 2020 | 2020 | * 2016 | 99,80 | 0,42 | 16 | 0 | 0 |
| B-n38-k6 | 2 | 1688 | 1688 | * 1679 | 99,47 | 0,59 | 105 | 0 | 34 |
| B-n39-k5 | 2 | 1190 | 1189 | * 1189 | 99,92 | 2,16 | 72 | 0 | 26 |
| B-n41-k6 | 2 | 1772 | 1734 | * 1733 | 97,80 | 1,66 | 181 | 0 | 2 |
| B-n43-k6 | 2 | 1609 | 1579 | * 1579 | 98,14 | 6,17 | 116 | 0 | 78 |
| B-n44-k7 | 2 | 1913 | 1892 | * 1892 | 98,90 | 14,67 | 318 | 0 | 168 |
| B-n45-k5 | 2 | 1601 | 1579 | * 1579 | 98,63 | 1,83 | 77 | 0 | 45 |
| B-n45-k6 | 2 | 1531 | 1482 | * 1482 | 96,8 | 4,58 | 164 | 0 | 106 |
| B-n50-k7 | 2 | 1572 | 1572 | * 1571 | 99,94 | 1,05 | 85 | 0 | 7 |
| B-n50-k8 | 2 | 2730 | 2652 | * 2652 | 97,14 | 189,24 | 779 | 0 | 934 |
| B-n51-k7 | 2 | 2140 | 2140 | * 2131 | 99,58 | 0,63 | 45 | 0 | 0 |
| B-n52-k7 | 2 | 1602 | 1593 | * 1589 | 99,19 | 1,58 | 113 | 0 | 4 |
| B-n56-k7 | 2 | 1506 | 1506 | * 1492 | 99,07 | 4,58 | 172 | 0 | 85 |
| B-n57-k7 | 2 | 2382 | 2382 | * 2346 | 98,49 | 8,36 | 423 | 0 | 67 |
| B-n57-k9 | 2 | 3314 | 3293 | * 3293 | 99,37 | 1713,03 | 793 | 0 | 3467 |
| B-n63-k10 | 2 | 3194 | 3092 | * 3092 | 96,81 | 14,39 | 424 | 0 | 383 |
| B-n64-k9 | 2 | 1868 | 1820 | * 1815 | 97,16 | 4 | 228 | 0 | 28 |
| B-n66-k9 | 2 | 2833 | 2747 | * 2747 | 96,96 | 170,23 | 728 | 0 | 438 |
| B-n67-k10 | 2 | 2222 | 2188 | * 2188 | 98,47 | 56,3 | 522 | 0 | 231 |
| B-n68-k9 | 2 | 2663 | 2599 | * 2598 | 97,56 | 42,97 | 1315 | 0 | 371 |
| B-n78-k10 | 2 | 2613 | 2557 | * 2557 | 97,86 | 838,47 | 1734 | 0 | 1155 |
| Moyenne | 2 | | | | 98,68 | 133,90 | 372 | 0 | 336 |

TABLE 6.11 – Résultats de $BC_{RPA}$ sur instances de type B pour le 2-PTVP

| | $m$ | $UB_{\text{meta}}$ | $UB_{\text{RPA}}$ | $LB_{\text{RPA}}$ | $\theta\%$ | $T$ | $CAP$ | $PEI$ | $ETO$ |
|---|---|---|---|---|---|---|---|---|---|
| B-n31-k5 | 3 | 2607 | 2603 | * 2603 | 100,00 | 1,63 | 0 | 0 | 0 |
| B-n34-k5 | 3 | 2903 | 2903 | * 2890 | 99,93 | 0,34 | 4 | 0 | 0 |
| B-n35-k5 | 3 | 3222 | 3222 | * 3205 | 100,00 | 0,34 | 10 | 0 | 0 |
| B-n38-k6 | 3 | 2860 | 2860 | * 2849 | 99,62 | 0,25 | 4 | 0 | 0 |
| B-n39-k5 | 3 | 2076 | 2076 | * 2018 | 100,00 | 0,33 | 9 | 0 | 0 |
| B-n41-k6 | 3 | 3015 | 3015 | * 2817 | 93,43 | 0,36 | 6 | 0 | 0 |
| B-n43-k6 | 3 | 2733 | 2733 | * 2683 | 98,17 | 0,34 | 14 | 0 | 0 |
| B-n44-k7 | 3 | 3217 | 3217 | * 3170 | 99,91 | 0,41 | 9 | 0 | 0 |
| B-n45-k5 | 3 | 2666 | 2579 | * 2578 | 96,70 | 5,31 | 38 | 0 | 12 |
| B-n45-k6 | 3 | 2585 | 2500 | * 2500 | 96,71 | 37,16 | 40 | 0 | 69 |
| B-n50-k7 | 3 | 2616 | 2616 | * 2606 | 100,00 | 0,5 | 16 | 0 | 0 |
| B-n50-k8 | 3 | 4488 | 4320 | * 4320 | 96,26 | 7,66 | 40 | 0 | 29 |
| B-n51-k7 | 3 | 3542 | 3464 | * 3464 | 97,80 | 6,34 | 34 | 0 | 4 |
| B-n52-k7 | 3 | 2709 | 2709 | * 2676 | 100,00 | 0,55 | 7 | 0 | 0 |
| B-n56-k7 | 3 | 2427 | 2402 | * 2400 | 98,89 | 2,91 | 41 | 0 | 10 |
| B-n57-k7 | 3 | 4127 | 4127 | * 3843 | 93,12 | 1 | 37 | 0 | 0 |
| B-n57-k9 | 3 | 5490 | 5451 | * 5447 | 99,21 | 1,25 | 18 | 0 | 0 |
| B-n63-k10 | 3 | 5553 | 5553 | * 5475 | 98,60 | 0,88 | 8 | 0 | 0 |
| B-n64-k9 | 3 | 3286 | 3286 | * 3054 | 92,94 | 1,14 | 81 | 0 | 0 |
| B-n66-k9 | 3 | 4696 | 4574 | * 4548 | 96,85 | 4,39 | 59 | 0 | 13 |
| B-n67-k10 | 3 | 3693 | 3693 | * 3605 | 97,62 | 0,86 | 25 | 0 | 0 |
| B-n68-k9 | 3 | 4336 | 4144 | * 4144 | 95,57 | 4,91 | 99 | 0 | 5 |
| B-n78-k10 | 3 | 4234 | 4098 | * 4098 | 96,79 | 34,42 | 208 | 0 | 151 |
| Moyenne | 3 | | | | 97,74 | 4,93 | 35 | 0 | 13 |

TABLE 6.12 – Résultats de $BC_{RPA}$ sur instances de type B pour le 3-PTVP

| | $m$ | $UB_{\text{meta}}$ | $UB_{\text{RPA}}$ | $LB_{\text{RPA}}$ | $\theta\%$ | $T$ | $CAP$ | $PEI$ | $ETO$ |
|---|---|---|---|---|---|---|---|---|---|
| P-n19-k2 | 2 | 501,00 | 501,00 | * 488 | 97,60 | 0,19 | 3 | 0 | 0 |
| P-n20-k2 | 2 | 511,00 | 511,00 | * 505 | 98,83 | 0,17 | 0 | 0 | 57 |
| P-n21-k2 | 2 | 511,00 | 511,00 | * 503 | 99,21 | 0,16 | 0 | 0 | 76 |
| P-n22-k2 | 2 | 512,00 | 512,00 | * 508 | 99,41 | 0,34 | 6 | 0 | 85 |
| P-n40-k5 | 2 | 1072,00 | 1062,00 | * 1062 | 99,72 | 2,84 | 9 | 0 | 15 |
| P-n45-k5 | 2 | 1170,00 | 1146,00 | * 1146 | 99,83 | 1,98 | 3 | 0 | 50 |
| P-n50-k10 | 2 | 1549,00 | 1549,00 | * 1502 | 96,97 | 0,80 | 4 | 0 | 0 |
| P-n50-k7 | 2 | 1257,00 | 1257,00 | * 1246 | 99,12 | 1,70 | 24 | 0 | 49 |
| P-n50-k8 | 2 | 1424,00 | 1333,00 | * 1333 | 93,61 | 7,91 | 65 | 0 | 88 |
| P-n51-k10 | 2 | 1678,00 | 1611,00 | * 1611 | 96,01 | 4,69 | 64 | 0 | 9 |
| P-n55-k10 | 2 | 1543,00 | 1517,00 | * 1517 | 98,31 | 16,38 | 7 | 0 | 108 |
| P-n55-k7 | 2 | 1286,00 | 1271,00 | * 1271 | 98,83 | 6,03 | 14 | 0 | 43 |
| P-n55-k8 | 2 | 1294,00 | 1294,00 | * 1271 | 98,22 | 1,45 | 32 | 0 | 120 |
| P-n60-k10 | 2 | 1659,00 | 1659,00 | * 1599 | 96,38 | 2,48 | 26 | 0 | 111 |
| P-n65-k10 | 2 | 1761,00 | 1761,00 | * 1710 | 97,10 | 1,66 | 10 | 0 | 58 |
| P-n70-k10 | 2 | 1826,00 | 1748,00 | * 1740 | 95,29 | 781,23 | 175 | 0 | 2026 |
| P-n76-k4 | 2 | 1385,00 | 1385,00 | * 1352 | 97,62 | 1,20 | 35 | 0 | 0 |
| P-n76-k5 | 2 | 1440,00 | 1440,00 | * 1406 | 97,64 | 10,78 | 114 | 0 | 1939 |
| P-n101-k4 | 2 | 1581,00 | 1558,00 | * 1557 | 98,48 | 18,17 | 9 | 0 | 1130 |
| Moyenne | 2 | | | | 97,80 | 45,27 | 32 | 0 | 314 |

TABLE 6.13 – Résultats de $BC_{RPA}$ sur instances de type P pour le 2-PTVP

| | $m$ | $UB_{\text{meta}}$ | $UB_{\text{RPA}}$ | $LB_{\text{RPA}}$ | $\theta\%$ | $T$ | $CAP$ | $PEI$ | $ETO$ |
|---|---|---|---|---|---|---|---|---|---|
| P-n19-k2 | 3 | 882 | 881 | * 879 | 99,77 | 0,42 | 2 | 0 | 0 |
| P-n20-k2 | 3 | 897 | 897 | * 893 | 99,55 | 0,09 | 0 | 0 | 0 |
| P-n21-k2 | 3 | 898 | 898 | * 898 | 100,00 | 0,09 | 0 | 0 | 0 |
| P-n22-k2 | 3 | 906 | 906 | * 901 | 100,00 | 0,11 | 0 | 0 | 0 |
| P-n40-k5 | 3 | 1869 | 1869 | * 1853 | 100,00 | 0,77 | 0 | 0 | 0 |
| P-n45-k5 | 3 | 2063 | 2063 | * 2037 | 98,74 | 1,33 | 0 | 0 | 0 |
| P-n50-k7 | 3 | 2203 | 2177 | * 2177 | 98,82 | 3,67 | 0 | 0 | 0 |
| P-n55-k7 | 3 | 2244 | 2244 | * 2225 | 99,15 | 0,52 | 0 | 0 | 0 |
| P-n55-k8 | 3 | 2259 | 2259 | * 2225 | 99,64 | 0,53 | 0 | 0 | 0 |
| P-n65-k10 | 3 | 3029 | 2976 | * 2976 | 98,25 | 7,8 | 0 | 0 | 0 |
| P-n70-k10 | 3 | 3159 | 3027 | * 3027 | 95,82 | 8,67 | 0 | 0 | 0 |
| P-n76-k4 | 3 | 2374 | 2322 | * 2318 | 97,64 | 2,41 | 0 | 0 | 0 |
| P-n76-k5 | 3 | 2464 | 2426 | * 2413 | 97,93 | 7,42 | 0 | 0 | 0 |
| P-n101-k4 | 3 | 2702 | 2702 | 2658 | 98,37 | 7,97 | 0 | 0 | 0 |
| Moyenne | 3 | | | | 98,84 | 2,99 | 0 | 0 | 0 |

TABLE 6.14 – Résultats de $BC_{RPA}$ sur instances de type P pour le 3-PTVP

| | $m$ | $UB_{\text{meta}}$ | $UB_{\text{RPA}}$ | $LB_{\text{RPA}}$ | $\theta\%$ | $T$ | $CAP$ | $PEI$ | $ETO$ |
|---|---|---|---|---|---|---|---|---|---|
| VRPNC1 | 2 | 1198,38 | 1174,68 | * 1174,68 | 98,02 | 5,88 | 20 | 0 | 356 |
| VRPNC2 | 2 | 1846,10 | 1766,57 | 1766,39 | 95,68 | 2305,33 | 190 | 0 | 4220 |
| VRPNC3 | 2 | 1871,83 | 1817,68 | * 1817,68 | 97,11 | 39,88 | 228 | 0 | 794 |
| VRPNC5 | 2 | 2947,26 | 2947,26 | 2614,86 | 88,62 | 90,80 | 7118 | 0 | 0 |
| VRPNC11 | 2 | 2318,86 | 2318,86 | 2145,67 | 92,53 | 189,95 | 4052 | 0 | 5965 |
| VRPNC12 | 2 | 1821,71 | 1808,82 | * 1808,82 | 99,90 | 57,05 | 493 | 0 | 886 |
| Moyenne | 2 | | | | 95,33 | 448,15 | 2017 | 0 | 2037 |

TABLE 6.15 – Résultats de $BC_{RPA}$ sur instances de type VRPNC pour le 2-PTVP

| | $m$ | $UB_{\text{meta}}$ | $UB_{\text{RPA}}$ | $LB_{\text{RPA}}$ | $\theta\%$ | $T$ | $CAP$ | $PEI$ | $ETO$ |
|---|---|---|---|---|---|---|---|---|---|
| VRPNC1 | 3 | 2127,77 | 2127,77 | 2089,43 * 98,20 | 0,44 | 0 | 0 | 0 | |
| VRPNC2 | 3 | 3184,81 | 3184,81 | 3056,23 * 95,96 | 1,25 | 0 | 0 | 0 | |
| VRPNC3 | 3 | 3174,74 | 3110,16 | 3110,16 * 97,97 | 31,56 | 2 | 0 | 2 | |
| VRPNC4 | 3 | 4023,86 | 3910,66 | 3898,58 * 96,89 | 45,72 | 0 | 0 | 0 | |
| VRPNC5 | 3 | 4913,56 | 4588,32 | 4587,87 93,37 | 1066,67 | 62 | 0 | 1322 | |
| VRPNC11 | 3 | 3706,39 | 3541,1 | 3438,69 92,78 | 10802,61 | 1138 | 0 | 4193 | |
| VRPNC12 | 3 | 3000,82 | 2998,71 | 2998,42 99,92 | 441,38 | 96 | 0 | 609 | |
| Moyenne | 3 | | | | 96,44 | 1769,95 | 185 | 0 | 875 |

TABLE 6.16 – Résultats de $BC_{RPA}$ sur instances de type VRPNC pour le 3-PTVP

VRP afin d'obtenir une nouvelle bibliothèque pour le $m$-PTVP. Nous faisons appel au solveur Cplex® pour résoudre les problèmes linéaires identifiés après ajout des coupes valides. Les résultats numériques montrent une augmentation de 5% à 10% des ratios entre les meilleures bornes inférieures et supérieures connues jusque-là, et environ un tiers des instances ont été résolues optimalement.

Cette approche a été présentée lors du récapitulatif des travaux de thèse au sein de la conférence nationale *ROADEF* [104].

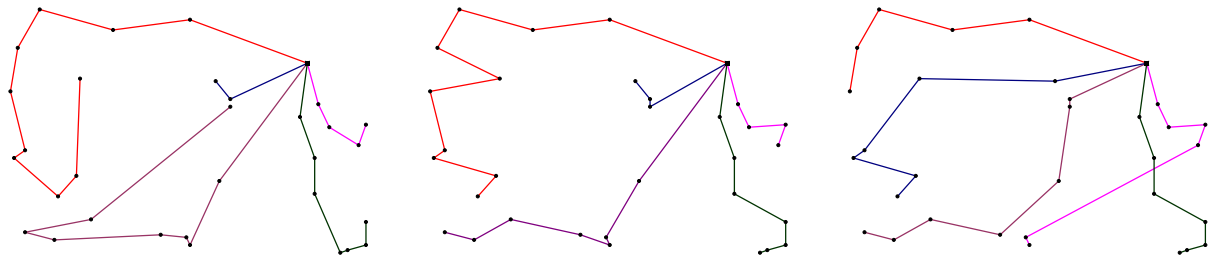
# Chapitre 7

# Le PTVCC : formulation, métaheuristique et bornes inférieures

## 7.1 Introduction

Dans ce chapitre nous étudions le Problème de Tournées de Véhicules Cumulatives avec contraintes de Capacité (ou PTVCC). Etant donné un graphe non orienté complet, le PTVCC consiste à déterminer un ensemble de tournées de véhicules qui minimise la somme des dates d'arrivées chez les clients, au lieu du traditionnel coût total, tout en respectant la capacité des véhicules. Chaque client est servi une seule fois, chaque véhicule effectue au maximum une tournée. Chacune de ces tournées débute puis se termine au dépôt, avec une charge totale (somme des demandes des clients visités) inférieure ou égale à la capacité des véhicules. Le PTVCC généralise le problème du réparateur itinérant (ou *travelling repairman problem* = TRP) en y ajoutant des contraintes de capacité et une flotte homogène de véhicules. Il appartient donc à la classe des problèmes de logistique humanitaire décrite au paragraphe 2.5.

A notre connaissance, seuls Campbell *et al.* [20] se sont intéressés à un problème similaire dans leur article sur le transport de denrées vitales après une catastrophe naturelle : le min-avg VRP qui minimise la date moyenne d'arrivée chez un client. Il est donc équivalent à notre PTVCC si sa fonction-objectif est multipliée par le nombre de clients à servir. Campbell *et al.* ont proposé une heuristique d'insertion appliquée à des instances simplifiées (demandes unitaires, capacité des véhicules modifiée, etc.). Ils ont comparé les résultats avec ceux obtenus pour le VRP classique et pour le min-max VRP, et montré l'intérêt d'une fonction-objectif basée sur la moyenne des dates d'arrivées chez les clients lors de la résolution de problèmes de logistique humanitaire. Rappelons que, contrairement à la logistique commerciale classique axée sur la rentabilité et le profit, la logistique humanitaire cherche à minimiser les souffrances et pertes en vies humaines et doit en plus prendre en compte des notions telles que l'impartialité et l'équité, voir par exemple l'article de Van Wassenhover [130].



(a) Solution optimale de CVRP : coût recalculé du PTVCC = 3470

(b) Solution optimale de OVRP : coût recalculé du PTVCC = 2512

(c) Meilleure solution connue du PTVCC : coût = 2192

FIGURE 7.1 – Comparaison des solutions optimales de CVRP et OVRP adaptées au PTVCC avec la meilleure solution obtenue par notre métaheuristique sur le PTVCC pour A-n32-k5

Le PTVCC diffère du min-max VRP d'Arkin *et al.* [7] car ce dernier ne minimise que la date d'arrivée maximale chez les clients, et donc la date d'arrivée chez le dernier client de chaque tournée, alors que le PTVCC tient compte des dates d'arrivées des clients intermédiaires de chaque tournée, ce qui peut varier considérablement selon si la plupart des clients sont servis tôt ou tard dans leur tournée. Le PTVCC diffère également du VRP cumulatif de Kara et Yetis [68] (ou CumVRP) qui utilise une fonction-objectif basée sur la consommation en énergie égale à la somme des longueurs des arêtes pondérées par la charge des véhicules qui les traversent. Enfin, il est évident que notre problème diffère du problème de tournées de véhicules ouvertes (Open VRP ou OVRP) traité par exemple par Letchford *et al.* [80] où l'objectif est de minimiser le coût comme dans un VRP classique, mais sans retour au dépôt après avoir servi le dernier client de chaque tournée.

La figure 7.1 illustre à quel point des solutions optimales du VRP et du OVRP peuvent être différentes de notre meilleure solution connue du PTVCC, sur l'instance A-n32-k5 classique en optimisation de tournées de véhicules. Pour faciliter la comparaison avec le OVRP où les véhicules ne reviennent pas au dépôt, nous n'avons pas dessiné les arêtes de retour au dépôt pour le PTVCC (7.1(c)) d'autant plus que ces arêtes n'interviennent pas dans le calcul du coût. Pour (7.1(a)), nous considérons une solution optimale du VRP (coût = 784) et en déduisons une solution du PTVCC (coût = 3470) en enlevant une des arêtes incidentes au dépôt dans chaque tournée. Comme le montre la figure 7.1, les solutions des trois problèmes peuvent varier significativement. Cette figure montre également que les tournées d'une bonne solution de PTVCC ont tendance à être équilibrées.

Dans la suite de ce chapitre, nous étudions une formulation mathématique du PTVCC et une métaheuristique. Nous identifions également plusieurs propriétés spécifiques au problème qui permettent d'améliorer l'efficacité des algorithmes présentés et de déduire des bornes inférieures simples.

## 7.2 Formulation mathématique et bornes inférieures

### 7.2.1 Modèle mathématique

Le PTVCC est défini sur un graphe complet et non orienté $G = (V, E, W)$ où $V = \{0, ..., n, n+1\}$ représente l'ensemble des nœuds, $V' = V \backslash \{0, n+1\}$ étant l'ensemble des clients, tandis que les nœuds 0 et $n+1$ correspondent au dépôt. $E$ est l'ensemble des arêtes et un poids $w_{ij} = w_{ji} \in W$ est affecté à chaque arête $(i, j) \in E$ pour représenter la distance ou le temps de parcours entre les nœuds $i$ et $j$, l'inégalité triangulaire étant supposée vérifiée. Une flotte de $R$ véhicules de capacité $Q$ est disponible au dépôt afin de satisfaire la demande $q_i > 0$ de chacun des clients $i \in V'$. Sachant qu'une tournée est effectuée par un seul véhicule et que chaque client est servi exactement une fois, l'objectif du PTVCC est d'identifier un ensemble des tournées qui minimise la somme des dates d'arrivées chez les clients, tout en respectant les contraintes de capacité : la somme des demandes des clients visités par une tournée ne peut pas excéder la capacité $Q$ des véhicules.

Soit $t_i^k$ la date d'arrivée du véhicule $k$ chez le client $i$ et $x_{ij}^k$ une variable binaire égale à 1 si et seulement si le véhicule $k$ traverse l'arête $(i, j)$ de $i$ à $j$. En s'inspirant de l'approche de Kallehauge *et al.* [67] appliquée au VRP avec fenêtres de temps, il est possible de formuler le PTVCC comme suit :

$$\min F = \sum_{k=1}^{R} \sum_{i \in V'} t_i^k \tag{7.1}$$

sous les contraintes

$$\sum_{i \in V} x_{ij}^k = \sum_{i \in V} x_{ji}^k, \qquad \forall j \in V', \forall k \in [1 \dots R] \tag{7.2}$$

$$\sum_{k=1}^{R} \sum_{j \in V} x_{ij}^k = 1, \qquad \forall i \in V' \tag{7.3}$$

$$\sum_{i \in V'} \sum_{j \in V} x_{ij}^k q_i \leq Q, \qquad \forall k \in [1 \dots R] \tag{7.4}$$

$$\sum_{j \in V} x_{0j}^k = 1, \qquad \forall k \in [1 \dots R] \tag{7.5}$$

$$\sum_{i \in V} x_{i,n+1}^k = 1, \qquad \forall k \in [1 \dots R] \tag{7.6}$$

$$t_i^k + w_{ij} \leq t_j^k + (1 - x_{ij}^k)T, \qquad \forall i \in V \setminus \{n+1\}, \forall j \in V, \forall k \in [1 \dots R] \tag{7.7}$$

$$t_i^k \geq 0, \qquad \forall i \in V, \forall k \in [1 \dots R] \tag{7.8}$$

$$x_{ij}^k \in \{0, 1\}, \qquad \forall i \in V, \forall j \in V, i \neq j, \forall k \in [1 \dots R] \tag{7.9}$$

La particularité de ce modèle est sa fonction-objectif (7.1) à minimiser, égale à la somme des dates d'arrivées chez les clients, tandis que les contraintes à respecter restent

classiques pour un problème de tournées de véhicules. Les contraintes (7.2) précisent qu'un véhicule qui arrive chez un client doit ensuite le quitter. Les contraintes (7.3) signifient que chaque client est servi par une seule tournée. Les contraintes de capacité (7.4) stipulent que la somme des demandes des clients servis par chaque tournée est inférieure ou égale à la capacité des véhicules. Les contraintes (7.5) et (7.6) assurent que le dépôt se trouve au début et à la fin de chaque tournée. Enfin, les contraintes (7.7), détaillées ci-après, calculent les dates d'arrivées chez chaque client et empêchent la formation de sous-tours, grâce à $T$, défini comme une grande constante positive. Un sous-tour est un sous-ensemble de clients connectés entre eux mais déconnectés du dépôt.

La figure 7.2 illustre l'élimination des sous-tours par les contraintes (7.7) qui fonctionnent comme celles développées par Miller *et al.* pour le problème de voyageur de commerce : si $i \in V \setminus \{n+1\}$ précède $j \in V$ au sein de la tournée $k$, alors $x_{ij}^k = 1$ et la contrainte associée devient $t_i^k + w_{ij} \leq t_j^k$, qui engendre une violation en cas de sous-tour. Au contraire, si $i \in V \setminus \{n+1\}$ ne précède pas $j \in V$ au sein de la tournée $k$, alors $x_{ij}^k = 0$ et la contrainte devient $t_i^k + w_{ij} \leq t_j^k + T$, qui est trivialement vérifiée puisque $T \gg t_i^k + w_{ij}$.

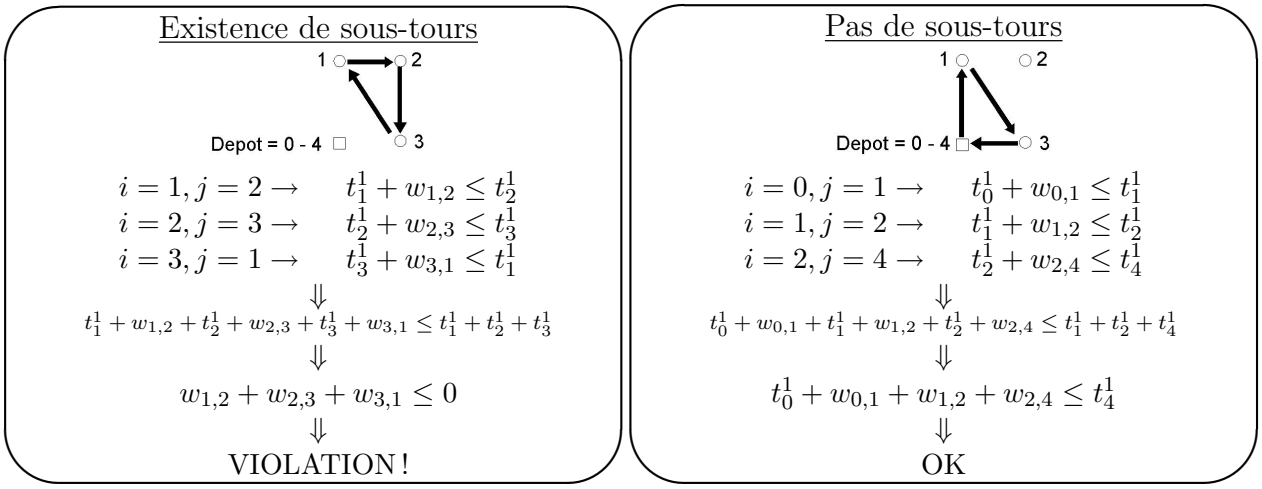



FIGURE 7.2 – Détection de sous-tours par les contraintes (7.7) avec $n = 3$

**Propriété 1 ($P1$) : le TRP ne donne pas de borne inférieure pour le PTVCC**

La figure 7.3 illustre une différence importante entre le PTVCC et le VRP classique. Pour ce dernier, à cause de l'inégalité triangulaire, il est préférable qu'un véhicule visite un maximum de clients avant de retourner au dépôt. Pour le PTVCC au contraire, le phénomène inverse se produit. Lorsque le nombre $R$ de véhicules disponibles est supérieur ou égal au nombre $n$ de clients, alors la solution optimale du PTVCC est trivialement obtenue en connectant chaque client directement au dépôt, afin de servir chacun d'eux le plus tôt possible. Lorsque $R < n$, certains clients doivent être regroupés au sein de la même tournée et servis par le même véhicule, ce qui retarde des dates d'arrivées et donc augmente la valeur de la fonction-objectif. Par conséquent, le coût des solutions du PTVCC augmente

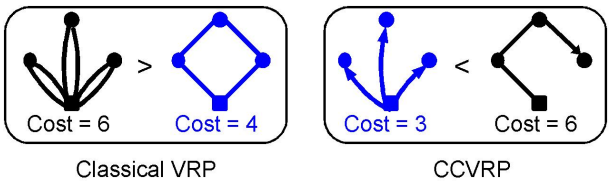



FIGURE 7.3 – PTVCC vs VRP

lorsque $R$ décroît et vice-versa. C'est pour cette raison que le TRP, cas particulier pour lequel $R = 1$, ne produit pas de borne inférieure pour le PTVCC, alors que le TSP produit une borne inférieure pour le VRP classique.

*P*2 : **les solutions optimales du PTVCC utilisent exactement** $\min(R, n)$ **véhicules**

Les solutions optimales du PTVCC utilisent tous les véhicules disponibles, puisque le coût des solutions du PTVCC augmente lorsque $R$ décroît et vice-versa. En effet, toute solution ayant moins de $R$ tournées peut être améliorée en déplaçant tout simplement le dernier client d'une des tournées qui visitent le plus de clients vers une nouvelle tournée, initialement vide. L'algorithme mémétique décrit par la suite exploite cette propriété et ne considère que les solutions à $R$ tournées (en supposant $R \leq n$).

*P*3 : **coût de l'inverse d'une tournée : pré-calculs**

Contrairement au coût d'une tournée de VRP classique, le coût d'une tournée de PTVCC varie selon le sens de parcours de la tournée. Sans perte de généralité, nous pouvons considérer une tournée $k$ dont la séquence de clients est $(1, 2, \ldots, n_k)$, qui débute et se termine au dépôt (nœud 0 au début et nœud $n_{k+1}$ à la fin). La date d'arrivée $t_j$ au nœud $j$ (client ou dépôt) est donc : $t_0 = 0$ et $t_j = t_{j-1} + w_{j-1,j}, \quad \forall 1 \leq j \leq n_k$. Par conséquent, le coût $F_k$ de la tournée $k$ peut être exprimé comme suit :

$$\begin{aligned} F_k &= \textstyle\sum_{j=1}^{n_k} t_j = t_1 + \ldots + t_{n_k} \\ &= w_{0,1} + (t_1 + w_{1,2}) + \ldots + (t_{n_k-1} + w_{n_k-1,n_k}) \\ &= w_{0,1} n_k + w_{1,2}(n_k - 1) + \ldots + w_{n_k-1,n_k} \times 1 \end{aligned}$$

ce qui correspond à :

$$F_k = \sum_{j=1}^{n_k} (n_k - j + 1) w_{j-1,j} \tag{7.10}$$

Le terme $(n_k - j + 1)$ est appelé dans la suite le *coefficient* du $j^{ème}$ client, égal au nombre de clients visités à partir de ce client. A l'aide de l'équation (7.10), le coût de toute

solution du PTVCC peut être exprimé comme la somme, sur le nombre de clients, du coût de l'arête utilisée pour atteindre chaque client, multiplié par le *coefficient* de ce client. La deuxième borne inférieure $LB_2$ décrite au paragraphe 7.2.2 est basée sur l'estimation de ce coût.

Soit $D_k = \sum_{j=1}^{n_k+1} w_{j-1,j}$ la longueur de la tournée $k$, calculée dans un VRP classique. La figure 7.4 illustre la validité de l'expression (7.11) utilisée pour le calcul du coût de la tournée $\bar{k}$, inverse de $k$. Il suffit de constater que la somme des coûts de la tournée $k$ et de son inverse $\bar{k}$ est égale à $n_k$ fois la distance $D_k$. Sur cette figure le petit carré représente le dépôt tandis que les petits cercles correspondent aux clients visités. On a indiqué auprès de chaque arête le nombre de fois où son coût est compté dans la fonction-objectif.

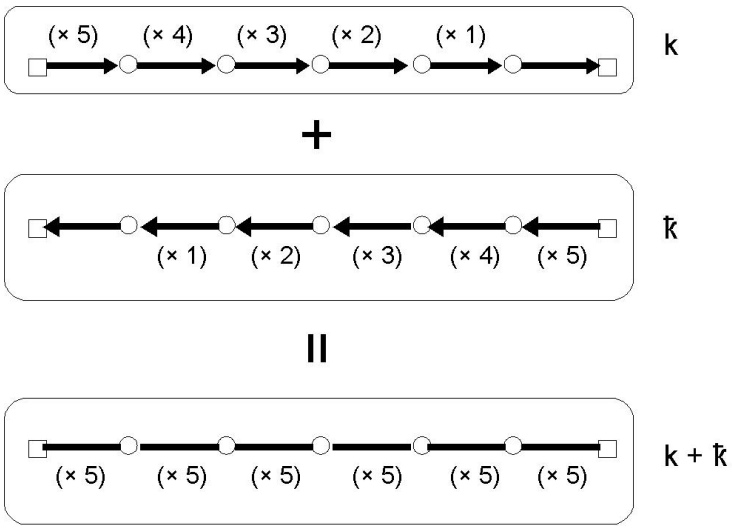



FIGURE 7.4 – Coût d'une tournée $k$ et de son inverse $\bar{k}$

$$F_{\bar{k}} = n_k D_k - F_k \tag{7.11}$$

L'équation (7.11) est particulièrement utile pour réaliser une métaheuristique efficace pour le PTVCC. Puisque le coût d'une tournée ne dépends plus seulement des arêtes utilisées, mais aussi du sens de parcours, il faut pouvoir estimer rapidement si une tournée choisie devrait être inversée ou pas. L'équation (7.11) nous permet de faire ce test en $O(1)$, à condition que le coût et la longueur de la tournée soient déjà calculés. Cette idée développée au paragraphe 7.3.4 a permis d'identifier les quantités à précalculer et à mettre à jour périodiquement pour évaluer des mouvements de recherche locale en $O(1)$. L'idée d'améliorer la complexité à l'aide de pré-calculs a déjà été appliquée avec succès au VRPTW, voir exemple Kindervater *et al.* [70] et Cordone *et al.* [28].

### 7.2.2 Bornes inférieures

**Borne inférieure 1 : flotte illimitée de véhicules**

Puisque le coût des solutions augmente lorsque $R$ décroît et vice-versa (voir propriété $P1$), alors la solution triviale et optimale qui consiste à connecter chaque client directement au dépôt lorsque $R \geq n$ donne une borne inférieure $LB_1$ pour le cas où $R < n$.

$$LB_1 = \sum_{j \in V'} w_{0j} \tag{7.12}$$

**Borne inférieure 2 : estimation du coût de la solution**

La deuxième borne inférieure pour le PTVCC est obtenue en approximant le coût des arêtes et les *coefficients* des clients de l'équation (7.10). Toutes les arêtes du graphe $G$ sont préalablement triées par ordre croissant de coût, puis les arêtes les moins chères sont affectées aux coefficients les plus élevés. Les $n$ *coefficients* non nuls sont quant à eux estimés en faisant l'hypothèse d'une solution équilibrée : quelle que soit la paire de véhicules choisie, la différence entre les nombres de clients servis par ces deux véhicules est toujours inférieure ou égale à 1. En effet, même si la solution PTVCC s'avère déséquilibrée, les *coefficients* seront au mieux égaux et au pire supérieurs aux estimations calculées avec l'hypothèse d'équilibre (voir figure 7.5).

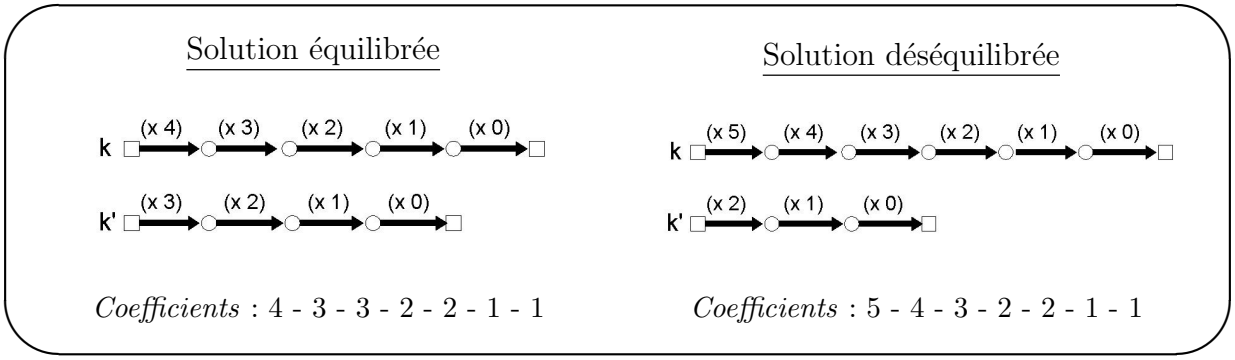



FIGURE 7.5 – Solution équilibrée/déséquilibrée et *coefficients* associés (ici $n = 7$, $R = 2$)

Soit $w_e$ le coût de la $e^{ème}$ arête la moins chère du graphe $G$. La borne inférieure $LB_2$ pour le PTVCC peut être calculée selon la formule (7.13). En cas d'application au TRP (équivalent au PTVCC avec $R = 1$), $LB_2$ se réduit à la borne inférieure proposée pour le TRP dans l'article de Salehipour *et al.* [120].

$$LB_2 = \sum_{e=1}^{n} \left( \left\lceil \frac{R + n - e - (n \bmod R)}{R} \right\rceil \right) w_e \tag{7.13}$$

$LB'_2$ calculée par la formule (7.14) est une version améliorée de $LB_2$, obtenue en différenciant les arêtes connectées au dépôt des arêtes connectant deux clients ensemble. Par conséquent :

$$\begin{aligned} LB'_2 &= \sum_{e=1}^{R} \left( \left\lceil \frac{R+n-e-(n \bmod R)}{R} \right\rceil \right) w'_e \\ &+ \sum_{e=1}^{n-R} \left( \left\lceil \frac{n-e-(n \bmod R)}{R} \right\rceil \right) w''_e \end{aligned} \tag{7.14}$$

où $w'_e$ est le coût de la $e^{ème}$ arête la moins chère connectée au dépôt tandis que $w''_e$ est le coût de la $e^{ème}$ arête la moins chère parmi celles qui relient deux clients, les non-incidentes au dépôt. $LB'_2$ domine évidemment $LB_2$.

## 7.3 Algorithme mémétique

Inspirés par la génétique et de l'évolution naturelle des espèces, les algorithmes génétiques sont des algorithmes d'optimisation qui utilisent des opérateurs de sélection, croisement et mutation pour faire évoluer une population de solutions vers l'optimum. Pour améliorer leur efficacité sur les problèmes combinatoires, ces métaheuristiques doivent souvent être hybridées avec une recherche locale, mécanisme d'intensification, et le résultat est désigné sous le terme d'algorithme mémétique.

L'algorithme que nous avons développé pour le PTVCC est appelé MA dans ce qui suit. La population *Pop* qu'il manipule est composée de $\sigma$ éléments appelés chromosomes, qui représentent des solutions du PTVCC. A chaque itération, deux chromosomes-parents $P_1$ et $P_2$ de *Pop* sont croisés pour obtenir des chromosomes-enfants. Puis, le meilleur enfant qui respecte un certain critère de distance est choisi pour remplacer un de ses 50% plus mauvais chromosomes de *Pop*. Enfin, une recherche locale est appliquée avec une probabilité $p_{ls}$ à l'enfant choisi avant son insertion dans la population. Notre algorithme mémétique s'arrête après un nombre prédéfini d'itérations $It_{\max}$. Il ne nécessite pas de mutation, car le critère de distance garantit une dispersion suffisante de la population et donc la diversification de la recherche. Les paragraphes qui suivent détaillent respectivement :

1. Le codage de solutions en chromosomes et l'évaluation des chromosomes créés
2. La sélection des parents $P_1$, $P_2$ et le croisement appliqué
3. Le remplacement des chromosomes de *Pop*
4. La recherche locale
5. Et enfin, l'initialisation de *Pop* ainsi que le critère de distance utilisé

La section 7.3.6 récapitule l'algorithme complet à la fin.

### 7.3.1 Codage et évaluation des chromosomes

Le choix du codage des solutions sous forme de chromosomes est crucial lors de la création d'un algorithme mémétique efficace. Toute solution faisable du problème doit pouvoir être représentée. Dans notre MA, un chromosome est une séquence ou permutation de $n$ clients sans délimiteurs de tournées. Il peut donc être vu comme une tournée géante de type voyageur de commerce. Il s'agit d'un codage indirect car toute solution réalisable est facilement convertie en chromosome en concaténant les clients de ses tournées, tandis que chaque chromosome représente un sous-ensemble de solutions qui peuvent être obtenues en découpant la séquence de clients en tournées qui respectent la capacité $Q$ des véhicules. Soit $K$ le nombre maximum de clients que peut contenir une tournée réalisable. La procédure optimale de découpage (ou *split*) décrite dans le prochain paragraphe permet d'obtenir en $O(RnK)$ la meilleure solution du PTVCC extractible d'un chromosome $C_u$, lequel sera affecté d'un *coût* $F(C_u)$ égal au coût de cette solution.

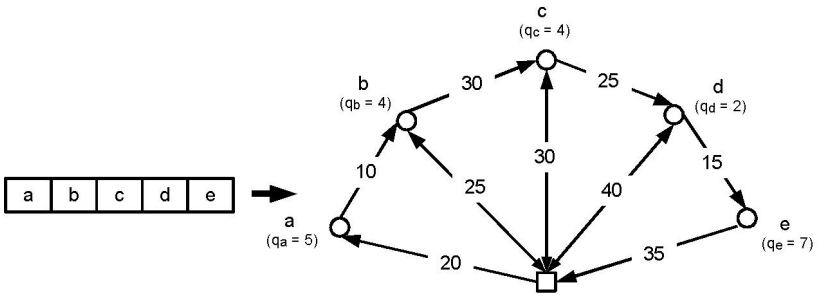
La procédure de découpage dérive de celle utilisée par Prins [112] pour le VRP. Le découpage optimal d'un chromosome sans délimiteurs de tournées est résolu comme un problème de recherche de plus court chemin sur un graphe auxiliaire acyclique orienté $G_A$ dans lequel les nœuds représentent les différentes positions au sein du chromosome tandis que l'existence d'une arête $(i, j)$ dans ce graphe signifie que les clients aux positions $i + 1$ à $j$ dans le chromosome forment une tournée réalisable. Le plus court chemin est trouvé à l'aide de l'algorithme de Bellman pour les graphes acycliques.

La procédure de découpage adaptée au PTVCC et illustrée par la figure 7.6 utilise la forme générale de l'algorithme de Bellman pour trouver en $R$ itérations le plus court chemin d'au plus $R$ arcs de 0 à $n$ dans $G_A$, dont sera déduit la meilleure solution ayant exactement $R$ tournées, conformément à la propriété $P2$ (page 145). Il s'agit d'une méthode de programmation dynamique basée sur les relations de récurrence (7.15) dans lesquelles $T_i^k$ désigne le label du nœud $i$ (coût du plus court chemin du nœud 0 de $G_A$ au nœud $i$) à l'itération $k$, tandis que le $P_{ki}$ est le prédécesseur de $i$ dans ce plus court chemin d'au plus $k$ arcs.

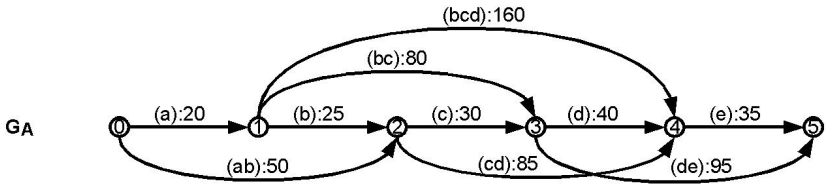
$$\begin{cases} \forall i \in \{1 \dots n\} : T_i^0 = +\infty \\ T_0^0 = 0 \\ \forall k > 0 : \begin{cases} T_j^k = \min\{T_i^{k-1} + w_{ij} | i \text{ prédécesseur de } j\} \\ P_{kj} = i | T_j^k = T_i^{k-1} + w_{ij} \end{cases} \end{cases} \tag{7.15}$$

On peut montrer facilement l'invariant suivant : à l'itération $k$, les labels correspondent aux coûts des plus courts chemins d'origine 0 ayant au plus $k$ arcs. A l'itération $R$, on a donc un plus court chemin d'au plus $R$ arcs, c'est à dire un découpage optimal du chromosome en $R$ tournées au maximum. Grâce à la propriété $P2$ (page 145), on peut même prédire que le chemin optimal de 0 à $n$ aura exactement $R$ arcs. La solution de PTVCC correspondante est obtenue comme l'illustre la sous-figure 7.6(e) en retraçant le chemin de $n$, $P_{(R,n)}$, $P_{(R-1,P_{(R,n)})}$, ..., à 0. Son coût est égal à $T_n^R$.

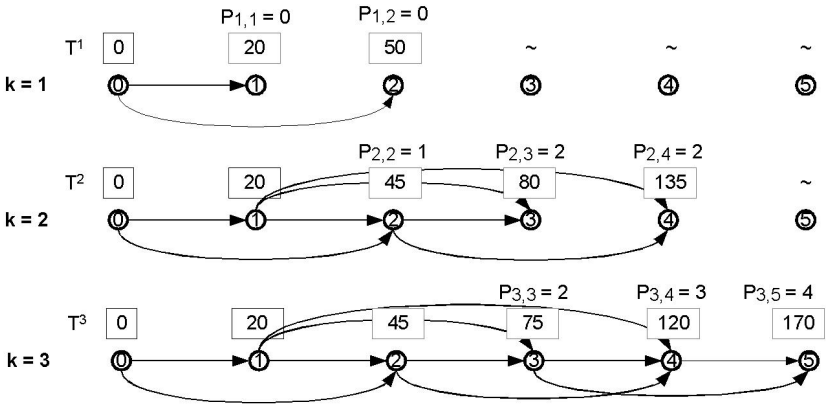
L'implémentation efficace de la procédure de découpage permet de ne pas générer explicitement le graphe $G_A$. A chaque itération $k$, seules les deux dernières lignes $k - 1$ et $k$



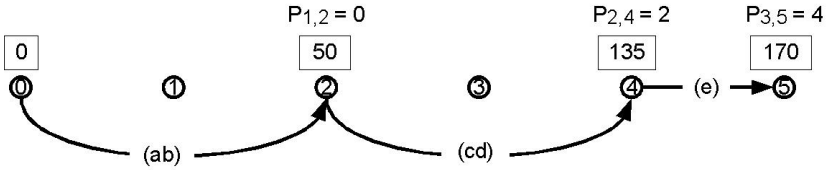

(a) Chromosome *abcde*



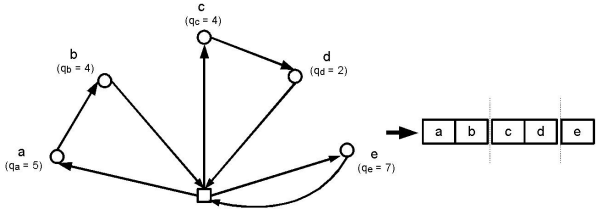

(b) Graphe auxiliaire $G_A$ déduit de *abcde*




(c) Algorithme de Bellman généralisé appliqué sur $G_A$




(d) Solution déduite de $G_A$ : *coût* = 170




(e) Découpage déduit de *abcde* : coût = 170

FIGURE 7.6 – Découpage optimal du chromosome *abcde* avec $Q = 10$ et $R = 3$

du tableau des labels $T_i^k$ sont conservées, représentées respectivement par les vecteurs $V$ et $W : \forall i \in \{1 \dots n\} : V_i = T_i^{k-1}$ et $W_i = T_i^k$. Nous stockons également $M_i$ (respectivement $N_i$) le nombre d'arcs utilisés pour atteindre le nœud $i$ à l'itération $k$ (respectivement $k-1$). Enfin, les termes *charge*, *distance* et *coût* représentent la somme des demandes des clients, la longueur depuis le dépôt et le coût de la tournée en cours de construction. Il est nécessaire de connaître la longueur de cette tournée pour mettre à jour son coût en $O(1)$ suite à un ajout de client (se référer par exemple à la formule de calcul du terme "$\Delta F_{k'}$" de l'équation (7.17)).

L'algorithme 6 récapitule notre procédure de découpage de chromosome de PTVCC. Sa complexité est $O(RnK)$ à cause de ses trois boucles imbriquées avec $O(R)$, $O(n)$ et $O(K)$ itérations respectives. Rappelons que $K$ est le nombre maximum de clients que peut contenir une tournée réalisable, et donc le nombre maximum d'itérations effectuées à l'intérieur de la dernière boucle avant que la charge de la tournée en cours de construction ne dépasse la capacité de véhicule ($charge \geq Q$). L'extraction de la solution de PTVCC du chromosome découpé est effectuée à l'aide de l'algorithme 7 de complexité $O(R)$.

### 7.3.2 Initialisation de la population *Pop* et critère de distance

L'algorithme mémétique utilise une population *Pop* avec un nombre prédéfini $\sigma$ de chromosomes, triés par ordre croissant de coût. Cette population est initialisée de la manière suivante : le premier chromosome résulte du codage de la solution de l'heuristique du plus proche voisin appliquée au problème. Le deuxième chromosome résulte du codage de la solution obtenue après application de la recherche locale décrite au paragraphe 7.3.4 sur la solution de l'heuristique du plus proche voisin. Les $\sigma - 2$ chromosomes restants sont des permutations aléatoires des $n$ clients.

Afin d'assurer une dispersion suffisante de la population et par conséquent la diversification de la recherche, la différence de coût entre deux chromosomes quelconques de *Pop* doit toujours être supérieure ou égale à une valeur prédéfinie $\Gamma$. En conséquence, un enfant obtenu par croisement ne peut remplacer un élément de *Pop* que si l'écart entre son coût et celui des autres chromosomes de la population, à l'exception de l'élément qui sera remplacé, est supérieur ou égal à $\Gamma$. Le critère de distance garantit une diversification suffisante de la population et rend donc inutile le recours à la mutation. La seule exception à cette règle vise à assurer l'élitisme : on accepte toujours un chromosome dont le coût est inférieur à celui de tous les chromosomes présents dans *Pop*. Cela revient à toujours accepter une solution qui améliore la meilleure solution $Pop_1$ obtenue jusque-là.

### 7.3.3 Sélection et croisement de deux parents $P_1$ et $P_2$

Un tournoi binaire sélectionne chacun des deux parents à croiser : deux chromosomes de *Pop* sont aléatoirement sélectionnés, celui qui possède le meilleur *coût* devient le premier parent $P_1$, puis la procédure se répète pour choisir le deuxième parent $P_2$.

**Algorithm 6** Procédure de découpage pour évaluer le coût $F(S)$ du chromosome $S$

$V_0 := 0$ ; $N_0 := 0$
**pour** $i := 1$ **à** $n+1$ **faire**
  $V_i = \infty$ ; $N_i = \infty$
**fin pour**
Copier $V$ dans $W$
Copier $N$ dans $M$
**pour** $k := 1$ **à** $R$ **faire**
  $i := 1$
  **répéter**
    $charge := 0$ ; $distance := 0$ ; $coût := 0$ ; $j := i$
    **répéter**
      $charge := charge + q_{S_j}$
      **si** $i = j$ **alors**
        $coût := w_{0,S_j}$
        $distance := w_{0,S_j}$
      **sinon**
        $coût := coût + distance + w_{S_{j-1},S_j}$
        $distance := distance + w_{S_{j-1},S_j}$
      **fin si**
      **si** $charge \leq Q$ **alors**
        **si** $V_{i-1} + coût < W_j$ **alors**
          $W_j := V_{i-1} + coût$
          $M_j := N_{i-1} + 1$
          $P_{M_j,j} := i - 1$
        **fin si**
        $j := j + 1$
      **fin si**
    **jusqu'à** $(j > n)$ ou $(charge \geq Q)$
    $i := i + 1$
  **jusqu'à** $(V_{i-1} = \infty)$ ou $(i > n)$
  Copier $W$ dans $V$
  Copier $M$ dans $N$
**fin pour**
$F(S) := V_n$

Le croisement a pour but de découvrir de nouvelles solutions en combinant des chromosomes ; en général deux parents $P_1$ et $P_2$ génèrent deux enfants $C_1$ et $C_2$. Notre croisement pour le PTVCC se base sur le croisement classique OX (ou order crossover), connu pour être performant pour le TSP et donc appliquable aux chromosomes sans délimiteurs de tournées. La figure 7.7 en donne un exemple.

Il fonctionne comme suit. Deux points de coupures $1 \leq i < j \leq n$ sont sélectionnés aléatoirement. La sous-séquence $P_1(i)...P_1(j)$ est copiée dans le chromosome-enfant $C_1$ aux mêmes positions, puis $C_1$ est complété en parcourant $P_2$ de manière cyclique à partir de

**Algorithm 7** Extraction de la solution du PTVCC suite au découpage du chromosome $S$

1: Initialiser la solution *Sol* (liste vide de $R$ tournées)
2: $t = 1$ ; $j := n$ ; $maxarc := N_j$
3: **répéter**
4: $\quad i := P_{maxarc,j}$
5: $\quad$ Remplacer la $t^{th}$ tournée de *Sol* par $T = (S_{i+1}, S_{i+2}, ..., S_j)$
6: $\quad t := t + 1$ ; $j := i$ ; $maxarc := maxarc - 1$
7: **jusqu'à** $i = 0$
8: **renvoyer** *Sol*

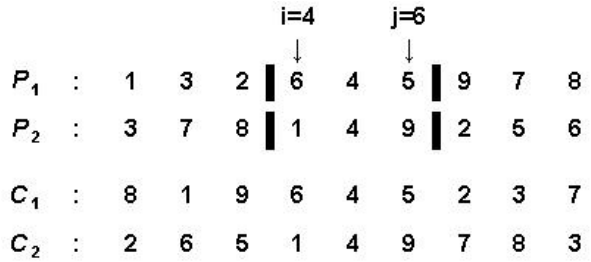



FIGURE 7.7 – Exemple de croisement OX

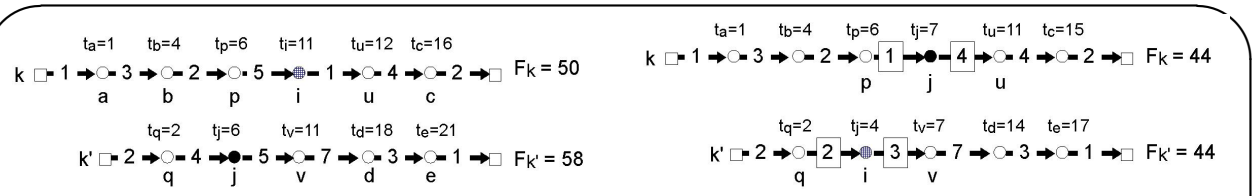
la même position $j + 1$ et en ajoutant dans $C_1$ les clients qui n'y sont pas déjà présents. La procédure permet d'obtenir le chromosome-enfant $C_2$ mais en permutant les rôles de $P_1$ et $P_2$ : la sous-séquence $P_2(i)...P_2(j)$ est copiée exactement et $P_1$ est balayé circulairement pour compléter $C_2$.

Le sens de parcours étant important pour le PTVCC, le croisement adapté doit également considérer les chromosomes obtenus si l'un ou les deux parents sont inversés tout en gardant les mêmes points de coupure. Au total, $x = 8$ enfants potentiels sont générés au cours de chaque itération et évalués avec la procédure de découpage précédemment décrite. Le vecteur $C$ des chromosomes-enfants est ensuite trié par ordre croissant de coût.

### 7.3.4 Recherche locale : mouvements et pré-calculs

La recherche locale est appliquée avec une probabilité $p_{ls}$ sur la meilleure solution de PTVCC obtenue après le découpage du chromosome-enfant $C_u$ candidat au remplacement d'un élément de *Pop*. La nouvelle solution obtenue est reconvertie en chromosome en concaténant ses différentes tournées. Le MA utilise une recherche locale qui combine 3 types de mouvements : l'échange de 2 clients, le déplacement d'un client et les mouvements 2-opt. En plus de ces 3 mouvements classiques entre 1 ou 2 tournées, notre recherche locale teste aussi l'inversion des tournées obtenues. A chaque itération, le meilleur mouvement est identifié et appliqué à la solution courante.

L'obligation de prendre en compte la possible inversion des tournées modifiées après




(a) Avant inversion de $i$ et $j$ (b) Après inversion de $i$ et $j$

D'après la formule (7.19), l'échange des clients $i$ et $j$ devrait entraîner un gain $\Delta F = \Delta F_k + \Delta F'_k$ avec $\Delta F_k = 5(6-3) + 1(6-4) - 1(6-3) - 4(6-4) = 6$ et $\Delta F'_k = 4(5-1) + 5(5-2) - 2(5-1) - 3(5-2) = 14$ ; ce que confirme la figure (b) ci-dessus, car le coût de la tournée $k$ passe de 50 à 44 tandis que le coût de la tournée $k'$ passe de 58 à 44.

FIGURE 7.8 – Calcul en $O(1)$ de l'impact de l'échange des clients $i$ et $j$

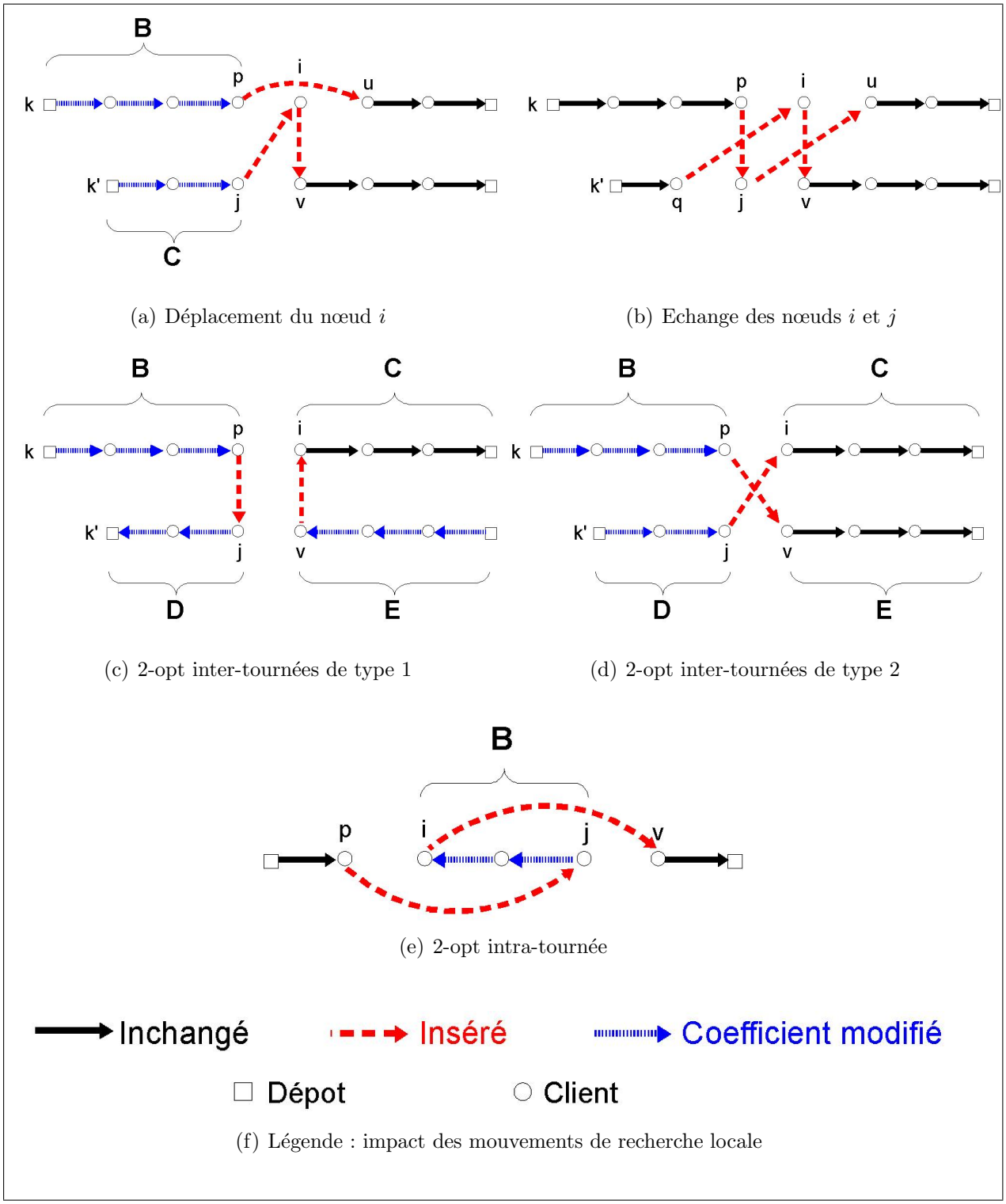
chaque mouvement a un impact important sur la recherche locale mise en place. Pour un VRP classique, la variation de coût suite à un mouvement est la simple différence entre le coût des arêtes insérées et le coût des arêtes retirées de la solution, calculable en $O(1)$. Pour le PTVCC, le calcul est plus compliqué. Par exemple, déplacer un client $i$ de sa tournée $k$ à une autre tournée $k'$ réduit les dates d'arrivées chez les clients initialement situés après $i$ dans la tournée $k$ et retarde les dates d'arrivées chez les clients qui seront situés après $i$ dans la tournée $k'$. Dans le pire des cas, les dates d'arrivées de $n-1$ clients devront être recalculées pour chaque mouvement potentiel, conduisant à des temps de calculs plus élevés en $O(n)$. Grâce à la propriété $P3$ (page 145) et des pré-calculs opportuns, nous avons pu déduire des formules pour évaluer la variation de coût de chaque mouvement potentiel en $O(1)$ et obtenir ainsi une recherche locale efficace.

### Pré-calculs nécessaires

Soient $D_k$, $F_k$ et $n_k$ la longueur, le coût et le nombre de clients servis par la tournée $k$. De même, soient $DN_i$, $FN_i$ la longueur et le coût le long de la tournée jusqu'au client $i$, et $r_i$ son rang. Ces valeurs doivent être pré-calculées pour évaluer en $O(1)$ le gain potentiel $\Delta F$ de chaque mouvement sur le coût de la solution (voir par exemple la figure 7.8). Ce gain est positif en cas d'amélioration.

De même, pour effectuer les tests de capacité en $O(1)$ pour les mouvements qui impliquent deux tournées, nous avons besoin de pré-calculer pour chaque client $i$ de la tournée $k$ les valeurs $q_i^+$ et $q_i^-$ égales respectivement au total des demandes depuis le dépôt jusqu'au client $i$ ($q_i^+ = q_i + \sum_{j \in V' | j \text{ est servi avant } i} q_j$), et à la capacité disponible restante après $i$ dans la tournée $k$ (il est évident que $q_i^+ + q_i^- = Q$). Enfin, $charge_k$ stocke le total des demandes des clients de la tournée $k$.

La figure 7.9, aidée de sa légende 7.9(f), illustre l'impact des différents mouvements de recherche locale détaillés dans les paragraphes qui suivent. Pour cela nous devrons à chaque fois identifier 6 nœuds : $i$ de la tournée $k$, $j$ de la tournée $k'$, $p$ et $q$ leurs prédécesseurs respectifs et finalement $u$ et $v$ leurs successeurs respectifs.




(a) Déplacement du nœud $i$

(b) Echange des nœuds $i$ et $j$

(c) 2-opt inter-tournées de type 1

(d) 2-opt inter-tournées de type 2

(e) 2-opt intra-tournée

(f) Légende : impact des mouvements de recherche locale

FIGURE 7.9 – Mouvements de recherche locale

**Déplacement d'un nœud**

La figure 7.9(a) illustre l'impact du déplacement d'un nœud $i$ de sa tournée $k$ vers une position située entre $j$ et $v$ de la tournée $k'$. Ce mouvement n'est autorisé que si le test de capacité de l'équation (7.16) est réussi. Notons que le terme $|k - k'|$ permet de prendre en compte le cas où le nœud $i$ se déplace au sein de sa propre tournée ($k = k'$). Si le test de capacité est réussi, alors au sein de la tournée $k$ les arêtes $(p, i)$ et $(i, u)$ sont supprimées, $(p, u)$ est insérée et les coefficients des clients de la sous-séquence $B$ doivent être décrémentés au sein de la fonction-objectif puisque le nombre de clients qui les suivent a été réduit d'une unité. De même, au sein de la tournée $k'$, les arêtes $(j, i)$ et $(i, v)$ sont supprimées, $(j, v)$ est insérée et les coefficients des clients de la sous-séquence $C$ sont à incrémenter.

Le coût total de ce déplacement est évalué par l'équation (7.17). Le terme $\gamma$ permet à la formule de rester valide lorsque $i$ et $j$ sont sélectionnés dans la même tournée $k = k'$.

$$|k - k'|charge_{k'} + q_i \leq Q|k - k'| \tag{7.16}$$

$$\Delta F = \Delta F_k + \Delta F_{k'} - \gamma \times \min(1 - |k - k'|, 0) \tag{7.17}$$

$$\text{où } \begin{cases} \Delta F_k = w_{pi}(n_k - r_p) + w_{iu}(n_k - r_i) - w_{pu}(n_k - r_i) + DN_p \\ \Delta F_{k'} = w_{jv}(n_{k'} - r_j) - w_{ji}(n_{k'} - r_j + 1) - w_{iv}(n_{k'} - r_j) - DN_j \\ \text{Si } \ r_j > r_i \ \text{ alors } \ \gamma = w_{iu} - w_{pu} \ \text{ sinon } \ \gamma = w_{ji} + w_{iv} - w_{jv} \end{cases}$$

**Echange de 2 nœuds**

La figure 7.9(b) illustre l'impact de l'échange de deux nœuds $i$ et $j$ des tournées $k$ et $k'$. Ce mouvement n'est autorisé que si le test de capacité de l'équation (7.18) est réussi. Notons que le terme $|k - k'|$ permet de prendre en compte le cas où l'échange a lieu au sein de la même propre tournée ($k = k'$). Si le test de capacité est réussi, alors au sein de la tournée $k$ les arêtes $(p, i)$ et $(i, u)$ sont remplacées par $(p, j)$ et $(j, u)$. De même, dans la tournée $k'$, $(q, j)$ et $(j, v)$ sont remplacées par $(q, i)$ et $(i, v)$. Le coût total de ce mouvement est évalué par l'équation (7.19).

$$|k - k'| \times \max\{charge_k - q_i + q_j, charge_{k'} + q_i - q_j\} \leq Q|k - k'| \tag{7.18}$$

$$\Delta F = \Delta F_k + \Delta F_{k'} \tag{7.19}$$

$$\text{où } \begin{cases} \Delta F_k = w_{pi}(n_k - r_p) + w_{iu}(n_k - r_i) - w_{pj}(n_k - r_p) - w_{ju}(n_k - r_i) \\ \Delta F_{k'} = w_{qj}(n_{k'} - r_q) + w_{jv}(n_{k'} - r_j) - w_{qi}(n_{k'} - r_q) - w_{iv}(n_{k'} - r_j) \end{cases}$$

**2-opt intra-tournée**

La figure 7.9(e) illustre l'impact d'un mouvement 2-opt effectué au sein d'une tournée $k = k'$. Aucun test de capacité n'est nécessaire puisqu'une seule tournée est impliquée.

L'équation (7.20) évalue le coût de ce mouvement en $O(1)$. Les arêtes $(p,i)$ et $(j,v)$ sont supprimées tandis que $(p,j)$ et $(i,v)$ sont insérées. La sous-séquence $B$ est inversée et le coût d'une telle inversion est évalué par la procédure $\Delta FlipChain$ détaillée par l'algorithme 8 ci-après. Cette procédure s'inspire largement de la propriété $P3$ concernant le calcul du coût d'inversion d'une tournée en $O(1)$.

$$\Delta F = \Delta F_k \ + \Delta FlipChain_{i,j} \tag{7.20}$$

$$\text{où } \Delta F_k = w_{pi}(n_k - r_p) + w_{jv}(n_k - r_j) - w_{pj}(n_k - r_p) - w_{iv}(n_k - r_j)$$

**Algorithm 8** Calcul de $\Delta FlipChain_{i,j}$ en $O(1)$

$\omega := FN_j - FN_i(r_j - r_i)$
**si** $j = n+1$ **alors**
  **si** $i = 0$ **alors**
    $\omega := \omega + (DN_j - DN_i)$
  **sinon**
    $\omega := \omega + DN_i$
  **fin si**
**fin si**
$\Delta FlipChain_{i,j} := 2\omega - (r_j - r_i + 1)(DN_j - DN_i)$
**si** $(i(j-n-1) = 0)$ et $(i+j-n-1 \neq 0)$ **alors**
  $\Delta FlipChain_{i,j} := \Delta FlipChain_{i,j} + (DN_j - DN_i)$
**fin si**
**renvoyer** $\Delta FlipChain_{i,j}$

### 2-opt inter-tournées

Nous avons identifié deux variantes type 1 et type 2 du 2-opt entre 2 tournées $k$ et $k'$. Dans les deux cas, nous aurons besoin de la procédure $\Delta FChain_{i,j}$ détaillée par l'algorithme 9, qui estime en $O(1)$ le coût d'une sous-séquence $[i,j]$ à partir des valeurs de $DN_i$, $DN_j$, $FN_i$ et $FN_j$, $\forall i \in V, \forall j \in V$.

La figure 7.9(c) et l'équation (7.22) illustrent l'impact du mouvement 2-opt de type 1 entre deux tournées $k$ et $k'$. Ce mouvement n'est autorisé que si le test de capacité de l'équation (7.21) est réussi. Dans ce cas, au sein de la tournée $k$, l'arête $(p,i)$ et la sous-séquence $C$ sont supprimées tandis que $(p,j)$ et l'inverse de la sous-séquence $D$ sont insérés. Les coefficients des clients de la sous-séquence $B$ sont augmentés de $|D| - |C|$ au sein de la fonction-objectif. Au sein de la tournée $k'$, l'arête $(j,v)$ et la sous-séquence $D$ disparaissent, la sous-séquence $E$ est inversée, et enfin l'arête $(v,i)$ et la sous-séquence $C$ sont insérées.

$$\max\{q_p^+ + q_j^+, q_i^- + q_i + q_v^- + q_v\} \leq Q \tag{7.21}$$

$$\Delta F = \Delta F_k + \Delta F_{k'} \tag{7.22}$$

$$\text{où} \begin{cases} \Delta F_k = F_k - (FN_p + DN_p r_j + w_{pj} r_j + FN_j - \Delta FlipChain_{0,j}) \\ \Delta F_{k'} = F_{k'} - (FChain_{v,n+1} - \Delta FlipChain_{v,n+1} + (D_{k'} - DN_v)(n_k - r_p) + w_{vi}(n_k - r_p) \\ \qquad\qquad + FChain_{i,n+1}) \end{cases}$$

**Algorithm 9** Calcul de $FChain_{i,j}$ en $O(1)$

1: $FChain_{i,j} := (FN_j - FN_i) - DN_i(r_j - r_i)$
2: **si** $(i = 0)$ et $(j \neq n+1)$ **alors**
3: $\quad FChain_{i,j} := FChain_{i,j} + DN_i$
4: **fin si**
5: **renvoyer** $FChain_{i,j}$

La figure 7.9(d) et l'équation (7.24) illustrent l'impact du mouvement 2-opt de type 2 entre deux tournées $k$ et $k'$. Ce mouvement n'est autorisé que si le test de capacité de l'équation (7.23) est réussi. Dans ce cas, au sein de la tournée $k$, l'arête $(p, i)$ et la sous-séquence $C$ sont supprimées tandis que $(p, v)$ et la sous-séquence $E$ sont insérées. Les coefficients des clients de la sous-séquence $B$ sont augmentés de $|E| - |C|$ au sein de la fonction-objectif. De même, au sein de la tournée $k'$, l'arête $(j, v)$ et la sous-séquence $E$ disparaissent, tandis que l' arête $(j, i)$ et la sous-séquence $C$ sont insérées. Les coefficients de la sous-séquence $D$ sont augmentés de $|C| - |E|$ au sein de la fonction-objectif.

$$\max\{q_p^+ + q_v^- + q_v, q_j^+ + q_i^- + q_i\} \leq Q \tag{7.23}$$

$$\Delta F = \Delta F_k + \Delta F_{k'} \tag{7.24}$$

$$\text{où} \begin{cases} \Delta F_k = DN_p(n_k - r_p - n_{k'} + r_j) - w_{pv}(n_{k'} - r_j) + FChain_{p,n+1} - FChain_{v,n+1} \\ \Delta F_{k'} = DN_j(n_{k'} - r_j - n_k + r_p) - w_{ji}(n_k - r_p) + FChain_{j,n+1} + FChain_{i,n+1} \end{cases}$$

### 7.3.5 Remplacement

Notre MA est un algorithme incrémental, car il ne construit pas des tableaux de populations successives (générations), mais remplace à chaque itération un élément de *Pop*. Cet élément est remplacé par l'un des $x = 8$ chromosomes-enfants créés par croisement. Rappelons que le vecteur $C$ des chromosomes-enfants est trié par ordre croissant de coût. La procédure de remplacement est donc la suivante : le meilleur enfant $C_u$ (initialement $u = 1$) est sélectionné comme candidat à l'insertion dans *Pop*. La recherche locale est appliquée à $C_u$ avec une probabilité $p_{ls}$. Ensuite, un chromosome $E$ est aléatoirement choisi dans la pire moitié de *Pop* (triée par ordre croissant de coût). $C_u$ remplace $E$ dans *Pop* si le critère de distance ou alors le critère d'élitisme est vérifié. Rappelons que le critère de distance spécifie que l'écart de coût entre $C_u$ et les autres éléments de *Pop* doit être supérieur ou égal à $\Gamma$. Le critère d'élitisme quant à lui spécifie que $C_u$ doit être accepté s'il a le meilleur coût connu jusque-là. Si aucun de ces deux critères n'est vérifié, alors $C_u$ est rejeté et $C_{u+1}$ devient candidat à l'insertion dans *Pop*. Dans le cas contraire, $C_u$ remplace $E$ puis est déplacé au sein de *Pop* à travers un tri par insertion pour rétablir le tri de *Pop*.

### 7.3.6 Récapitulatif de notre algorithme mémétique pour le PTVCC

L'algorithme 10 récapitule notre MA pour résoudre le PTVCC. Les différents composants ont été présentés dans les sections précédentes.

**Algorithm 10** Algorithme mémétique pour le PTVCC (MA)

Initialiser la population $Pop$ de $\sigma$ chromosomes
Evaluer chaque chromosome de $Pop$ et obtenir leur coût ($F$) avec la procédure de découpage
Trier $Pop$ par ordre croissant de *coût*
**pour** $It = 1$ **à** $It_{\max}$ **faire**
  Sélectionner deux parents $P_1, P_2$ de $Pop$, chacun grâce à un tournoi binaire
  Appliquer le croisement OX adapté à $P_1$ et $P_2$ pour obtenir l'ensemble $C$ des $x$ chromosomes-enfants
  Evaluer chaque enfant de $C$ et obtenir son coût avec la procédure de découpage
  Trier $C$ par ordre croissant de *coût*
  Choisir un nombre aléatoire $rand$ entre 0 et 1
  $u := 0$
  **répéter**
    $u := u + 1$
    **si** $rand \leq p_{ls}$ **alors**
      $S :=$ solution du $PTVCC$ extraite après découpage de $C_u$
      Appliquer la recherche locale sur $S$
      Convertir $S$ en chromosome $C_u$
    **fin si**
    Choisir un chromosome $E$ dans la pire moitié de $Pop$
    **si** $(F(C_u) < F(Pop(1)))$ ou $(|F(y) - F(C_u)| \geq \Gamma, \forall y \in Pop \setminus \{E\})$ **alors**
      Remplacer $E$ par $C_u$
      Déplacer $C_u$ pour rétablir le tri de $Pop$
    **fin si**
  **jusqu'à** $(u = x)$ ou (remplacement effectué)
**fin pour**
**renvoyer** $Pop_1$

## 7.4 Résultats numériques

Le PTVCC étant un problème nouveau, à notre connaissance aucune instance spécifique au problème n'a été publiée à ce jour. Dans un premier temps, nos algorithmes sont donc testés sur le cas particulier du TRP ($R = 1$) et comparés à la seule autre métaheuristique pour ce problème, le GRASP+VND (*Greedy Randomized Adaptive Procedure + Variable Neighborhood Descent*) de Salehipour *et al.* [120]. Dans un deuxième temps, nos algorithmes sont appliqués sur des instances classiques du VRP pour obtenir les premières bornes supérieures et inférieures du PTVCC avec $R > 1$. Enfin, l'évolution du coût des solutions en fonction du nombre de véhicules disponibles est étudiée.

### 7.4.1 Instances et implémentation

En l'absence d'instances spécifiques pour le PTVCC, nous avons utilisé les instances VRPNC de 50 à 199 clients initialement proposées par Christofides *et al.* [23] pour le VRP et disponibles à l'adresse internet [115]. Les 7 instances sans limite sur la longueur maximale de tournées ont été traitées en considérant une flotte de taille minimale $R = R_{min}$ égale au nombre minimal de véhicules nécessaire pour satisfaire les contraintes de capacité, comme lors d'un VRP classique. Ce nombre est tout simplement égal à $R_{min} = \left\lceil \sum_{i \in S} \frac{q_i}{Q} \right\rceil$ pour la plupart des instances et $R_{min} = \left\lceil \sum_{i \in S} \frac{q_i}{Q} \right\rceil + 1$ dans certains cas. En ce qui concerne le cas particulier du TRP, nous avons utilisé les instances dont Salehipour *et al.* [120] se sont servis pour tester leur métaheuristique.

Nos algorithmes sont tous codés en C sur un Pentium D de 3.40 GHz avec 1 Go de mémoire RAM sous Windows XP. Les algorithmes mémétiques utilisent une population de $\sigma = 15$ chromosomes avec une différence minimale de coût $\Gamma = 0.5$ et un nombre d'itérations $It_{\max} = 20000$. Deux versions de MA ont été testées : $\mathrm{MA}_2$ qui utilise un simple opérateur de croisement OX et génère $x = 2$ enfants par croisement ; et $\mathrm{MA}_1$ qui teste si aucun, l'un ou les deux parents devraient être inversés au début du croisement et donc génère $x = 8$ enfants à chaque fois.

### 7.4.2 Cas particulier du TRP

La seule autre métaheuristique pour le TRP est le GRASP+VND de Salehipour *et al.* [120] qui existe en version normale ($\mathrm{GV}_n$) et version rapide ($\mathrm{GV}_r$). Ses auteurs ont généré 120 instances symétriques de 10 à 500 nœuds et comparé les résultats de leur métaheuristique avec ceux de l'heuristique du plus proche voisin et leur borne inférieure simple $LB_{trp}$ basée sur les arbres recouvrants. Les calculs ont été effectués sur un PC de 2,4 GHz et 512 Mo RAM.

Afin d'évaluer nos algorithmes, nous comparons notre $\mathrm{MA}_1$ à $\mathrm{GV}_n$, notre $\mathrm{MA}_2$ à $\mathrm{GV}_r$ et notre borne inférieure $\mathrm{LB}_{\mathrm{new}} = \max(\mathrm{LB}_1, \mathrm{LB}'_2)$ avec $\mathrm{LB}_{\mathrm{trp}}$. Notons que $\mathrm{MA}_1$ et $\mathrm{MA}_2$ ne sont exécutés qu'une seule fois, comme $\mathrm{GV}_r$, contrairement à $\mathrm{GV}_n$ dont les résultats sont obtenus après 10 exécutions. Les tableaux 7.1 et 7.2 montrent les résultats obtenus sur les mêmes instances. Conformément au format utilisé par Salehipour *et al.* [120], les en-têtes suivants sont communs aux deux tableaux : $n$ représente le nombre de nœuds y compris le dépôt ; $NbI$ est le nombre d'instances générées pour chaque valeur de $n$ ; $\left\lceil \frac{meta}{NN} \right\rceil$ % (équation (7.25)) représente l' amélioration apportée par une métaheuristique par rapport à l'heuristique du plus proche voisin (NN), plus élevée pour les meilleures métaheuristiques ; $\left\lceil \frac{LB}{meta} \right\rceil$ % (équation (7.25)) est l'écart entre le résultat d'une métaheuristique et une borne inférieure $LB$, plus faible pour les meilleures bornes inférieures ; $T$ représente la durée de calcul d'une métaheuristique jusqu'à obtention de sa meilleure solution ; et enfin, $cT$ est une estimation de la durée de calcul en secondes de $\mathrm{MA}_1$ et $\mathrm{MA}_2$ si ces algorithmes étaient

exécutés sur un PC identique à celui de Salehipour *et al.* [120]. Cette dernière estimation est calculée à l'aide du comparatif des vitesses d'ordinateurs publié par Dongarra [37].

$$\begin{bmatrix} i \\ j \end{bmatrix} \% = \frac{100(j-i)}{j} \quad \text{avec} \quad i < j \quad \text{dans les tables 7.1 et 7.2.} \tag{7.25}$$

| | | Résultats de [120] | | | Nos résultats | | | |
|---|---|---|---|---|---|---|---|---|
| *NbI* | $n$ | $\begin{bmatrix} GV_n \\ NN \end{bmatrix}$ % | $\begin{bmatrix} LB_{trp} \\ GV_n \end{bmatrix}$ % | $T$ | $\begin{bmatrix} MA_1 \\ NN \end{bmatrix}$ % | $\begin{bmatrix} LB_{new} \\ MA_1 \end{bmatrix}$ % | $T$ | $cT$ |
| 20 | 10 | **2,44** | 33,04 | 0,00 | 2,43 | 26,68 | 0,00 | 0,00 |
| 20 | 20 | 9,86 | 39,63 | 0,04 | **10,11** | 32,49 | 0,00 | 0,01 |
| 20 | 50 | **9,67** | 45,41 | 3,21 | 9,36 | 37,58 | 1,06 | 1,44 |
| 20 | 100 | 11,56 | 43,40 | 101,27 | **11,95** | 36,48 | 61,56 | 93,26 |
| 20 | 200 | 11,33 | 42,32 | 3741,05 | **12,81** | 34,57 | 619,19 | 938,16 |
| 20 | 500 | 8,11 | 50,73 | 91470,78 | **13,85** | 36,31 | 10697,92 | 16208,7 |
| Moyenne | | 8,83 | 42,42 | 15886,06 | **10,09** | 34,02 | 1896,62 | 2873,6 |

TABLE 7.1 – Résultats des versions *normales* des algorithmes pour le $TRP$

| | | Résultats de [120] | | | Nos résultats | | | |
|---|---|---|---|---|---|---|---|---|
| *NbI* | $n$ | $\begin{bmatrix} GV_r \\ NN \end{bmatrix}$ % | $\begin{bmatrix} LB_{trp} \\ GV_r \end{bmatrix}$ % | $T$ | $\begin{bmatrix} MA_2 \\ NN \end{bmatrix}$ % | $\begin{bmatrix} LB_{new} \\ MA_2 \end{bmatrix}$ % | $T$ | $cT$ |
| 20 | 10 | 2,14 | 33,48 | 0,00 | **2,43** | 26,68 | 0,00 | 0,00 |
| 20 | 20 | 8,19 | 42,43 | 0,00 | **10,11** | 32,49 | 0,01 | 0,01 |
| 20 | 50 | 6,85 | 50,07 | 0,27 | **9,32** | 37,00 | 0,56 | 0,84 |
| 20 | 100 | 10,24 | 45,53 | 9,73 | **11,92** | 36,50 | 25,06 | 37,96 |
| 20 | 200 | 10,69 | 43,37 | 356,73 | **12,70** | 34,69 | 234,02 | 354,58 |
| 20 | 500 | 10,70 | 46,49 | 7192,94 | **13,44** | 36,63 | 3541,30 | 5365,52 |
| Moyenne | | 8,14 | 43,56 | 1259,95 | **9,99** | 34,00 | 633,49 | 959,82 |

TABLE 7.2 – Résultats des versions *rapides* des algorithmes pour le $TRP$

Le tableau 7.1 montre que $MA_1$ améliore les résultats de $GV_n$ de $1,2\%$ en moyenne tout en étant beaucoup plus rapide, même après conversion des temps de calculs. Les bornes inférieures sont également améliorées : l'écart aux bornes supérieures obtenues par les métaheuristiques, qui varie de 33% à 50% dans [120], varie désormais de 26% à 37%. Le tableau 7.2 montre que $MA_2$ produit de meilleures solutions que $GV_r$. En ce qui concerne les temps de calcul, les deux métaheuristiques sont très rapides sur les petites instances de 10 à 20 nœuds, $MA_2$ est plus lent que $GV_r$ sur les instances médianes (50 à 100 nœuds), mais plus rapide sur les plus grandes instances (200 à 500 nœuds).

*Pour résumer, l'algorithme mémétique MA décrit dans ce chapitre est désormais la meilleure métaheuristique publiée sur le TRP, bien qu'il ne soit pas spécialement conçu pour ce cas particulier.*

### 7.4.3 PTVCC

Le tableau 7.3 évalue la performance des deux versions $MA_1$ et $MA_2$ de notre algorithme mémétique sur les instances VRPNC avec un nombre de véhicules minimal $R = R_{\min} > 1$. Les en-têtes de ce tableau sont les suivants : *Pb* contient le numéro du problème VRPNC, $n$ est le nombre de nœuds y compris le dépôt ; $BS$ est la valeur de meilleure borne supérieure obtenue avec différents réglages ; $NN$ est le coût donné par l'heuristique du plus proche voisin ; $\left[\begin{smallmatrix}(meta)heur\\ BS\end{smallmatrix}\right]\%$ (équation (7.26)) représente l'écart de coût entre la solution obtenue par une heuristique ou métaheuristique et la meilleure borne supérieure connue, dont la valeur devrait être faible pour les meilleures (méta)heuristiques ; $LB_i\%$ (équation (7.27)) est le ratio entre une borne inférieure et la meilleure borne supérieure connue, élevé pour les meilleures bornes inférieures ; et enfin, $T$ représente la durée de calcul en secondes de la métaheuristique jusqu'à obtention de sa meilleure solution.

$$\left[\begin{matrix} i \\ BS \end{matrix}\right]\% = \frac{100(i - BS)}{BS}, \qquad \text{with } \; i \geq BS \;\text{ in table 7.3} \tag{7.26}$$

$$LB_i\% = \frac{100 \times LB_i}{BS}, \qquad \text{with } \; LB_i < BS, \quad i \in \{1, 2\} \tag{7.27}$$

| | | | | Lower bounds | | NN | $MA_1$ | | $MA_2$ | |
|---|---|---|---|---|---|---|---|---|---|---|
| Pb | $n$ | $R$ | $BS$ | $LB_1\%$ | $LB_2\%$ | $\left[\begin{smallmatrix}NN\\ BS\end{smallmatrix}\right]\%$ | $\left[\begin{smallmatrix}MA_1\\ BS\end{smallmatrix}\right]\%$ | $T$ | $\left[\begin{smallmatrix}MA_2\\ BS\end{smallmatrix}\right]\%$ | $T$ |
| 1 | 50 | 5 | 2230,35 | 53,86 | 84,02 | 23,69 | **0,00** | 15,80 | **0,00** | 6,77 |
| 2 | 75 | 10 | 2426,12 | 74,83 | 76,73 | 39,79 | **0,18** | 59,77 | 3,61 | 4,25 |
| 3 | 100 | 8 | 4045,42 | 61,67 | 72,87 | 38,02 | **0,68** | 168,78 | **0,68** | 65,27 |
| 4 | 150 | 12 | 4987,52 | 73,79 | 71,41 | 32,64 | **0,18** | 1069,56 | 0,21 | 109,30 |
| 5 | 199 | 17 | 5817,75 | 82,58 | 66,78 | 39,39 | **0,69** | 1617,23 | 0,70 | 971,36 |
| 11 | 120 | 7 | 7317,98 | 83,62 | 36,43 | 22,07 | **0,00** | 270,72 | 1,36 | 97,31 |
| 12 | 100 | 10 | 3558,92 | 81,08 | 65,57 | 30,16 | **0,01** | 103,83 | **0,01** | 16,11 |
| Moyenne | | | | 73,06 | 67,69 | 32,25 | **0,25** | 472,24 | 0,94 | 181,48 |

TABLE 7.3 – Résultats du PTVCC sur les instances VRPNC

Le tableau 7.3 montre que les deux versions de l'algorithme mémétique sont efficaces sur le PTVCC : toutes les deux améliorent l'heuristique du plus proche voisin de plus de 30% et produisent des solutions à moins de 1% des meilleures solutions connues. $MA_1$ est plus rapide que $MA_2$, mais les résultats de $MA_1$ sont meilleurs. Les bornes inférieures simples

produisent aussi des résultats satisfaisants : en moyenne $\max(LB_1\%, LB_2\%) = 79,24\%$ et quatre instances sur sept ont une borne inférieure qui excède 80% de $BS$. Il est important de mentionner qu'aucun algorithme de calcul de bornes inférieures ne domine l'autre. Leurs efficacités respectives dépendent du type d'instance traitée. Par exemple, une instance avec de nombreux groupes de clients éloignés du dépôt aura de nombreuses arêtes de faible coût connectant des paires de clients entre eux et seulement de longues arêtes connectant les différents clients au dépôt, et donc probablement $LB_1 > LB_2$. A l'inverse, pour une instance dont les clients sont uniformément répartis tout autour du dépôt, il est plus probable que $LB_1 < LB_2$, car il y aura de nombreuses petites arêtes pour connecter les clients au dépôt et seulement de longues arêtes connectant les clients entre eux.

### Impact du nombre de véhicules $R$

D'après la propriété $P2$, les meilleures solutions du PTVCC utilisent tous les véhicules disponibles et le coût des solutions diminue lorsque le nombre de véhicules augmente. Nous avons voulu quantifier cette diminution en évaluant le ratio entre la borne supérieure obtenue avec une taille de flotte minimale ($R = R_{min}$) et la borne supérieure obtenue avec des véhicules supplémentaires ($R > R_{min}$). Le graphe 7.10 illustre l'évolution de ce ratio sur les instances VRPNC. L'allure de la courbe montre que la contribution des véhicules supplémentaires diminue au fur et à mesure que leur nombre augmente : les 2 premiers véhicules ajoutés réduisent la borne supérieure de 10%, alors que les 3 véhicules suivants n'apportent que 5% de gain à peine, et ces contributions ne cessent de décroître.

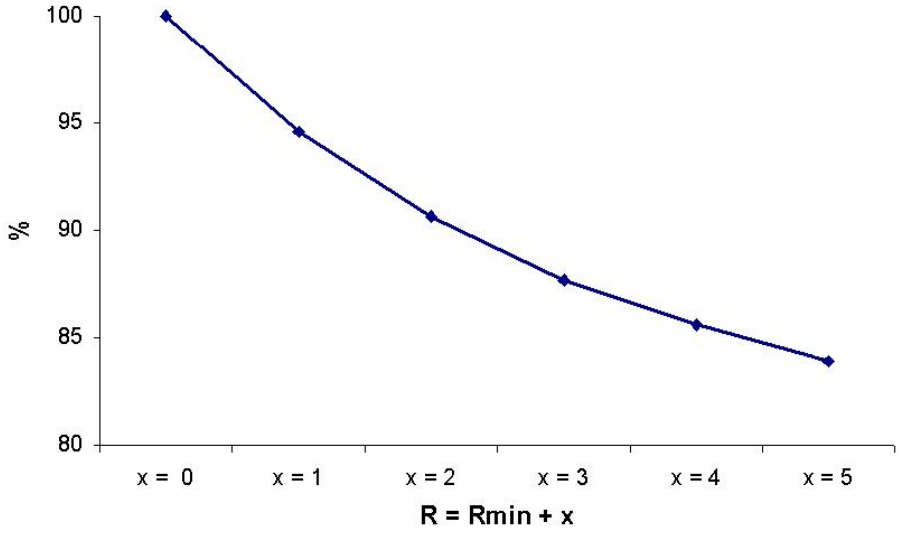



FIGURE 7.10 – Evolution du coût des solutions en fonction de $R$

**Robustesse de l'algorithme mémétique**

Les algorithmes mémétiques comportent des composants non déterministes tels que l'initialisation de la population, la sélection des parents à croiser et le choix des chromosomes à remplacer. Il est donc important de s'assurer que les résultats obtenus ne varient pas de manière importante en fonction de l'argument ou *germe* (seed) utilisé pour générer la séquence de nombres pseudo-aléatoires nécessaires, car jusqu'à présent nos algorithmes ont toujours été exécutés une seule fois. Le tableau 7.4 illustre la robustesse des algorithmes $MA_1$ et $MA_2$ à travers les moyennes ($\overline{\left[\frac{MA}{BS}\right]}\%$) et surtout les écart-types ($\sigma$) obtenus avec 10 *germes* différents sur les instances VRPNC. L'écart-type moyen est inférieur à 0,5% pour les deux algorithmes, ils sont donc robustes car peu dépendants de la séquence de nombres pseudo-aléatoires choisie. Il n'est donc pas nécessaire de lancer plusieurs exécutions de nos algorithmes comme cela se fait parfois pour les méthodes à population, car le gain potentiel (0,5%) ne justifie pas la perte de temps occasionnée.

Table 7.4 – Robustesse des algorithmes mémétiques

| | $MA_1$ | | $MA_2$ | |
|---|---|---|---|---|
| Pb | $\overline{\left[\frac{MA_1}{BS}\right]}\%$ | $\sigma$ | $\overline{\left[\frac{MA_2}{BS}\right]}\%$ | $\sigma$ |
| 1 | 0,90 | 0,93 | 0,37 | 0,61 |
| 3 | 0,82 | 0,19 | 0,78 | 0,19 |
| 4 | 0,59 | 1,05 | 0,27 | 0,64 |
| 5 | 0,58 | 0,29 | 0,69 | 0,30 |
| 11 | 1,02 | 0,41 | 0,88 | 0,85 |
| 12 | 0,01 | 0,01 | 0,01 | 0,01 |
| Moy | 0,65 | 0,48 | 0,50 | 0,43 |

## 7.5 Conclusion

Ce chapitre a présenté un modèle mathématique et la première métaheuristique pour le Problème de Tournées de Véhicules Cumulatives avec contraintes de Capacité (ou PTVCC), ainsi que deux bornes inférieures simples. En y ajoutant les contraintes de capacité et une flotte de véhicules, le PTVCC généralise le problème du réparateur itinérant (TRP), variante du TSP dont l'objectif est de minimiser la somme des dates d'arrivées chez les clients, au lieu de l'habituelle longueur ou coût total. Ce type de fonction-objectif est adapté lors de la modélisation de la distribution de denrées vitales par exemple, ou la répartition de premiers secours suite à une catastrophe naturelle, lorsque la priorité est donnée à la satisfaction du besoin du client.

La métaheuristique proposée est un algorithme mémétique efficace (MA) dont les trois points forts sont le croisement OX adapté, la procédure généralisée de découpage de chromosome et les pré-calculs déduits des propriétés que nous avons identifiées. Les propriétés

et pré-calculs ont rendu possible l'évaluation en $O(1)$ des mouvements de recherche locale, au prix d'une implémentation non triviale. Les résultats numériques montrent qu'en plus de produire de bons résultats sur le PTVCC, le MA est désormais la meilleure métaheuristique pour le TRP, bien qu'il ne soit pas spécifiquement conçu pour cela. Les bornes inférieures simples atteignent 80% des meilleures bornes supérieures.

Cet algorithme, les propriétés identifiées et les bornes inférieures ont fait l'objet de présentations aux conférences *EUME* [101] et *EvoTransLog* [102], puis d'une publication dans la revue *Computers and Operations Research* [103]. Un développement possible serait d'étendre notre MA à d'autres problèmes de tournées de véhicules avec fonction-objectif cumulative tels que le PTVCC multi-tournées où chaque véhicule peut effectuer plusieurs tournées. Ce problème serait encore plus difficile à traiter car la date d'arrivée chez un client doit prendre en compte la durée des tournées déjà effectuées par le même véhicule.

# Chapitre 8

# Conclusion générale

Les travaux de recherche développés dans cette thèse portent sur la résolution de deux problèmes NP-difficiles de transport qui intègrent des contraintes ou des objectifs de maîtrise des risques : le problème de tournées de véhicules $m$-péripatétiques ($m$-PTVP) qui concerne la logistique de sécurité, notamment le transport de fonds, et le problème de tournées de véhicules cumulatives avec contraintes de capacité (PTVCC) qui concerne la logistique humanitaire, notamment le secours des populations victimes de catastrophes naturelles.

Le $m$-PTVP a fait l'objet de la majeure partie de nos travaux. Il consiste à définir des tournées de véhicules sans arêtes communes de coût total minimal sur $m$ périodes telles que chaque client soit visité exactement une fois par période. Ses applications, outre le transport de fonds, comprennent par exemple la définition de tournées de gardiennage ou encore le design des parcours de véhicules auto-guidés (ou *Automated Guided Vehicle* = AGV) au sein d'usines automatisées. D'un point de vue théorique, il peut être considéré comme une généralisation deux problèmes NP-difficiles connus que sont le VRP, correspondant au cas mono-période 1-PTVP et le $m$-PVP, correspondant au cas mono-véhicule. C'est pourquoi les tests numériques des algorithmes et méthodes proposées ont été effectuées non seulement sur le $m$-PTVP en général, mais aussi sur les cas particuliers du VRP et du $m$-PVP.

Après avoir décrit les données et contraintes spécifiques au problème, nous avons démontré des résultats sur sa complexité, notamment le fait que le problème demeure NP-difficile lorsque $m$ croît, avant de proposer différentes formulations mathématiques. Puis des bornes inférieures polynomiales ont résulté de la relaxation de certaines contraintes du $m$-PTVP pour obtenir un $b$-couplage parfait ou $m$ 1-arbres recouvrants de coût total minimal. L'analyse numérique a montré que la performance de ces bornes simples s'améliore lorsque $m$ augmente.

Ensuite, nous avons développé deux métaheuristiques pour obtenir des solutions de bonne qualité pour un temps de calcul raisonnable. Il s'agit d'un algorithme de recherche locale itérative et d'un algorithme de recherche taboue diversifiée et guidée à l'aide d'une solution du $b$-couplage parfait. Les deux algorithmes sont initialisés à partir d'une adap-

tation de l'heuristique de Clarke et Wright. Ils utilisent des mouvements basés sur l'idée d'insérer dans la solution courante des arêtes qui n'y sont pas utilisées. L'analyse numérique a montré que la diversification de la recherche taboue et son guidage par la solution du $b$-couplage parfait améliorent la performance de l'algorithme, faisant de lui la meilleure des deux métaheuristiques dans la plupart des cas. Les coûts des meilleures solutions trouvées par les métaheuristiques ont servi de bornes supérieures initiales pour améliorer la vitesse et l'efficacité des approches polyédrales et de type génération de colonnes, en réduisant la taille de l'arbre de recherche ou en ajustant le pas des relaxations lagrangiennes.

Notre algorithme de génération de bornes inférieures basée sur une approche de type génération de colonnes estime les meilleures valeurs des variables duales de la relaxation linéaire d'une formulation aggrégée du $m$-PTVP, notamment à l'aide de deux heuristiques duales basées sur (1) l'approximation des tournées de véhicules par des $\mathfrak{q}$-tournées, lesquelles peuvent être générées à l'aide d'un algorithme pseudo-polynomial, (2) l'utilisation d'une approche de type génération de colonnes pour pallier au nombre exponentiel de variables et (3) la résolution des différents problèmes-maitres, obtenus après génération et ajout des ($\mathfrak{q}$-)tournées à coût réduit négatif, en optimisant les valeurs des variables duales sans faire appel à un quelconque solveur tel que Cplex, mais en quantifiant les violations des contraintes de degré pour corriger (par relaxation lagrangienne) les valeurs affectées. Les résultats numériques montrent que les bornes inférieures obtenues ainsi sont meilleures lorsque $m$ prend de petites valeurs, ce qui rend cette approche complémentaire avec les bornes inférieures polynomiales dont l'efficacité augmente avec $m$. Ainsi des bornes inférieures de bonne qualité peuvent toujours être obtenues quel que soit $m$ du $m$-PTVP.

Enfin, deux approches polyédrales ont été proposées pour le $m$-PTVP, deux méthodes de branchements et coupes basées respectivement sur une formulation du $m$-PTVP et sur une relaxation de cette formulation. Nous avons identifié parmi les inégalités efficaces pour le VRP celles qui sont généralisables au $m$-PTVP, montré les limites de celles qui ne le sont pas, et adapté les procédures de séparation en conséquence. Les deux algorithmes obtenus nous ont amenés à modifier profondément la bibliothèque de fonctions de détection de coupes violées CVRPSEP de Lysgaard pour le VRP afin d'obtenir une nouvelle bibliothèque pour le $m$-PTVP. Le solveur Cplex a été utilisé pour résoudre les problèmes linéaires identifiés après ajout des coupes valides.

Le PTVCC a également été étudié. Il consiste à définir les tournées de véhicules qui minimisent la somme des dates d'arrivées chez les clients tout en respectant les contraintes de capacité des véhicules. Il modélise les situations où la priorité est de satisfaire le besoin du client au plus tôt, notamment lors de la distribution de denrées vitales ou de premiers secours suite à une catastrophe naturelle.

La métaheuristique proposée pour trouver de bonnes solutions de ce problème est un algorithme mémétique (MA) qui manipule une population de chromosomes qui représentent des solutions du problème à l'aide d'un codage indirect sans délimiteur de tournées. Son efficacité repose sur l'adaptation de la procédure de découpage optimal de chromosome et l'utilisation de pré-calculs déduits des propriétés du PTVCC que nous avons identifiées. Les propriétés et pré-calculs ont en effet rendu possible l'évaluation en $O(1)$ des mouvements de

recherche locale, au prix d'une implémentation non triviale. Ces propriétés ont également permis de déduire des formules de calcul de bornes inférieures.

Les résultats numériques montrent qu'en plus de produire de bons résultats sur le PTVCC, le MA est désormais la meilleure métaheuristique publiée à ce jour pour le cas particulier du problème du réparateur itinérant (= PTVCC mono-véhicule = TRP), bien qu'il ne soit pas spécifiquement conçu pour cela.

Les travaux de recherche développés dans cette thèse ont déjà donné lieu à deux publications, l'une dans la revue internationale *Computers and Operations Research* et l'autre dans un numéro spécial de la collection *Operations Research / Computer Science Interfaces Series*, six conférences internationales dont quatre avec actes publiés et une conférence nationale. Notons en plus qu'un article est en cours de révision pour la revue internationale *4OR*.

Tous ces travaux ont nécessité un développement informatique important, des évaluations de complexité, la définition et l'usage de pré-calculs et de structures de données complexes. Les algorithmes ont été programmés en langage Delphi, C et C++. Les méthodes exactes codées en C et en C++ ont en plus nécessité l'implémentation d'appels à des solveurs externes à partir du programme informatique par le biais de DLLs (Dynamic Link Libraries - bibliothèques du logiciel), ainsi que des appels à des fonctions en fortran. Enfin, les réglages de paramètres ont souvent nécessité de nombreux tests.

Des extensions possibles de travaux présentés dans cette thèse incluent la prise en compte de contraintes de fenêtres de temps dans le $m$-PTVP et de multi-tournées dans le PTVCC, afin de rapprocher encore plus les deux problèmes traités des applications pratiques qu'ils modélisent. En ce qui concerne le $m$-PTVP, deux types de contraintes de fenêtres de temps sont envisagées, inclusives et exclusives. Les premières visent à varier les horaires de livraison d'une période à l'autre pour réduire les risques de braquage chez le client au moment de la livraison, alors que les secondes visent à garder des horaires de livraisons stables au cours des $m$ périodes pour faciliter la mise en place d'une routine de fonctionnement chez les employés du client. Dans ce dernier cas par exemple un supermarché pourra exiger de recevoir les billets et pièces nécessaires à son fonctionnement en début de semaine entre 6h et 7h, alors que dans le premier cas il faudrait éviter de livrer en 6h et 7h si la livraison de la période précédente s'était déjà faite dans ce créneau horaire. En ce qui concerne le PTVCC, il serait intéressant d'étudier le cas multi-tournées où chaque véhicule peut effectuer plusieurs tournées. Le problème résultant est encore plus difficile à traiter car la date d'arrivée chez un client doit prendre en compte la durée des tournées déjà effectuées par le même véhicule. Toutefois cela permettrait de prendre en compte le cas où peu de véhicules sont disponibles par rapport à la demande totale, ce qui se produit parfois dans la logistique humanitaire, lorsque les ressources sont inférieures aux besoins.

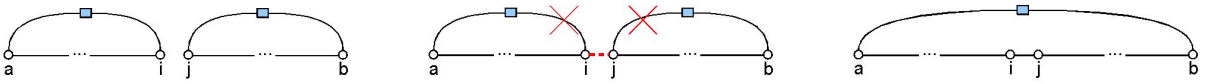
# Annexes

- **A-1 : Théorème : En l'absence de contraintes de capacité, le $m$-PTVP se réduit au $m$-PVP**

- **A-2 : Résultats numériques des algorithmes développés**

# A-1 : Théorème : En l'absence de contraintes de capacité, le $m$-PTVP se réduit au $m$-PVP

Il est évident que le VRP se réduit au TSP si les contraintes de capacité sont ignorées. En effet, si une solution contient plus d'une tournée, alors il est possible de fusionner ces tournées deux par deux. Fusionner deux tournées $t_1$ et $t_2$ consiste à supprimer une arête de retour au dépôt au sein de chaque tournée, puis insérer l'arête qui relie les deux clients qui ne sont plus connectés au dépôt, comme le montre la figure 8.1. L'inégalité triangulaire garantit que le coût d'une solution obtenue après chaque fusion est inférieur ou égal à celui de la solution avant fusion, puisque $c_{i,j} \leq c_{0,i} + c_{0,j}$.




(a) Avant fusion (b) Fusion possible (c) Après fusion

FIGURE 8.1 – Fusion de deux tournées

Le même principe peut être appliqué au $m$-PTVP avec $m > 1$ puisque les tournées peuvent être fusionnées deux par deux au sein de chacune des $m$ périodes. Toutefois, puisqu'aucune arête ne peut être utilisée plus d'une fois au sein d'une solution, la fusion de deux tournées d'une période $k^1 \in [1...m]$ ne peut être appliquée que si l'arête à insérer (pour connecter les deux clients) n'est pas déjà utilisée au sein d'une autre période $k^2 \in [1...m] \setminus \{k^1\}$. **Notons $H(m)$ la proposition suivante : quelle que soit la solution de $m$-PTVP, il est toujours possible de fusionner deux par deux les tournées de chaque période $k \in [1...m]$.** Montrer que $\forall m > 1$, en l'absence de contraintes de capacité le $m$-PTVP se réduit au $m$-PVP revient donc à montrer que $\forall m > 1$, $H(m)$ est vraie.

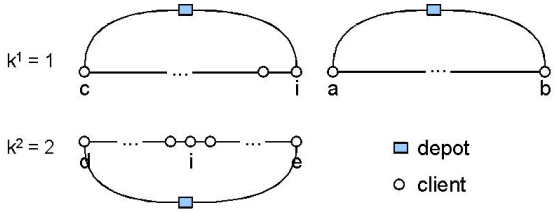
Notre démonstration se décompose en deux parties : nous montrerons d'abord que $H(2)$ est vraie, puis nous montrerons que si $H(m)$ est vraie, alors $H(m+1)$ est forcément vraie.

## A-1.1 : Montrer que $H(2)$ est vraie

Il s'agit de montrer que 2 tournées quelconques d'une période du 2-PTVP peuvent toujours être fusionnées pour en obtenir une seule.

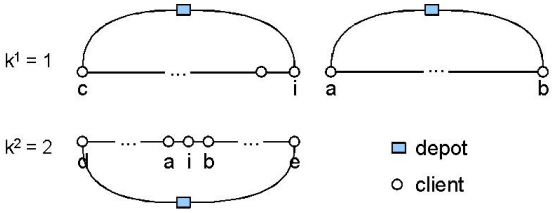
La figure 8.2 illustre comment fusionner n'importe quelle paire de tournées au sein d'une période d'une solution de 2-PTVP, et ce en moins de 5 itérations. Lors de l'initialisation, une arête $(i, j)$ est choisie pour connecter deux clients situés aux extrémités des deux tournées d'une période $k^1$ (avec $j = a$ sur la figure 8.2). Si la solution courante n'utilise pas déjà $(i, j)$, alors $(i, j)$ est insérée au sein de la période $k^1$ pour relier $i$ et $j$ tandis que $(0, i)$ et $(0, j)$ sont supprimées. Par contre, si $(i, j)$ était déjà utilisée au sein d'une période $k^2 \neq k^1$, alors une procédure d'échange est appliquée aux arêtes $(0, i)$, $(0, j)$ et $(i, j)$ : $(i, j)$ est désormais utilisé au sein de la période $k^2$ tandis que $(0, i)$ et $(0, j)$ sont affectées à la

**0 - Initialisation**



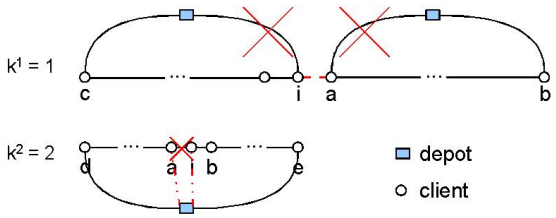

La figure ci-dessus représente les deux tournées 1 et 2 à fusionner au sein de la période $k^1 = 1$, ainsi que la tournée 3 qui contient $i$ au sein la période $k^2 = 2$ ($i$ étant le dernier client de la tournée 1).
L'objectif est donc de montrer que si les contraintes de capacité sont ignorées, il est toujours possible de fusionner les tournées 1 et 2, tout en conservant ou en réduisant le coût.

**1 - Insérer $((i, a))$ ou $(i, b)$**



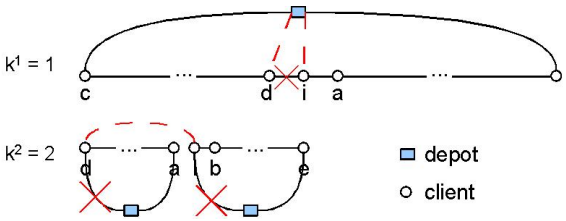

Si $(i, a)$ n'est pas utilisée au sein de la période $k^2$, alors $(i, a)$ peut être insérée pour fusionner les tournées 1 et 2. Si $(i, b)$ n'est pas utilisée au sein de la période $k^2$, alors $(i, b)$ peut être insérée pour fusionner les tournées 1 et 2.
Le seul cas restant, représenté par la figure ci-dessus, est donc celui où les arêtes $(i, a)$ et $(i, b)$ sont déjà utilisées. C'est ce cas qui sera étudié par la suite.

**2 - Echanger $(0, i)$, $(0, a)$ et $(i, a)$**



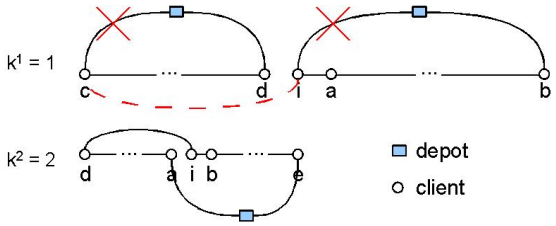

Si $(i, a)$ et $(i, b)$ sont déjà utilisées au sein de la période $k^2$, alors un échange d'arêtes peut être effectué entre les périodes $k^1$ et $k^2$ : l'arête $(i, a)$ sera désormais utilisée au sein de la période $k^1$, tandis que les arêtes $(0, i)$ et $(0, a)$ seront désormais utilisées au sein de la période $k^2$.
L'échange ne modifie pas le coût de la solution, car l'ensemble de toutes les arêtes utilisées reste le même. Reste donc à refusionner les tournées de la période $k^2$.

**3 - Insérer $(i, d)$ ou Echanger $(0, i)$, $(0, d)$ et $(i, d)$**



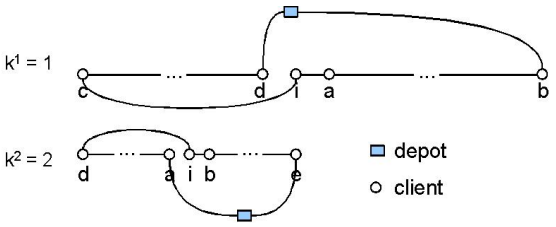

Si $(i, d)$ n'est pas utilisée au sein de la période $k^1$, alors $(i, d)$ peut être insérée pour fusionner deux tournées de la période $k^2$. Par contre, si $(i, d)$ est utilisée (comme l'illustre la figure ci-dessus), alors l'échange d'arêtes suivant peut être appliqué : $(0, d)$ et $(0, i)$ sont désormais utilisées au sein de la période $k^1$ tandis que l'arête $(d, i)$ est désormais utilisée au sein de la période $k^2$.

**4 - Insérer $(i, c)$**




L'arête $(i, c)$ ne peut pas être déjà utilisée au sein de la période $k^2$ puisque les arêtes $(i, a)$ et $(i, b)$ sont utilisées. L'arête $(i, c)$ peut donc être insérée pour fusionner deux tournées de la période $k^1$.

**5 - Solution dans le pire des cas**




On constate donc qu'il est toujours possible de fusionner deux tournées d'une période d'un 2-PTVP, car il est impossible que les arêtes $(i, a)$, $(i, b)$, $(i, d)$ et $(i, c)$ soient toutes déjà utilisées au sein de la solution courante ($(i, 0)$ étant déjà utilisée au sein de la période $k^1$).

FIGURE 8.2 – Fusion de deux tournées de la période $k^1 = 1$ d'une solution de 2-PTVP

période $k^1$. Après chaque échange, $j$ devient le premier (ou le dernier) client de sa tournée au sein de la période $k^2$, tandis que $j'$ désigne le dernier (ou le premier) client de cette même tournée ($j \neq j'$). Les échanges sont arrêtés si $(i, j')$ n'est pas utilisée, et on pose $j' = j$ avant d'insérer $(i, j)$ en remplacement de $(0, i)$ et $(0, j)$. Sinon, on pose $j = j'$ avant d'échanger les périodes des arêtes $(0, i)$, $(0, j)$ et $(i, j)$. Sachant qu'aucune arête ne doit être impliquée dans plus d'un échange, il faut donc au plus 5 itérations pour fusionner n'importe quelle paire de tournées d'une solution de 2-PTVP, comme l'illustre la figure 8.2.

## A-1.2 : Montrer que si $H(m)$ est vraie, alors $H(m+1)$ est vraie

Il s'agit de montrer que si $H(m)$ est vraie, alors il est toujours possible de fusionner deux tournées quelconques au sein d'une même période d'une solution de $(m+1)$-PTVP.

Notons :
- $S$ une solution de $(m+1)$-PTVP
- $\alpha_i^k$ le 1er client visité par la tournée qui traverse $i$ au sein de la période $k$ de $S$
- $\omega_i^k$ le dernier client visité par la tournée qui traverse $i$ au sein de la période $k$ de $S$
- $p_i^k$ le client qui précède $i$ au sein de la période $k$ de $S$
- $s_i^k$ le client qui suit $i$ au sein de la période $k$ de $S$
- $E(S)$ l'ensemble des arêtes utilisées par $S$
- $E(S^k)$ l'ensemble des arêtes utilisées au sein de la période $k$ de $S$
- $S^{\notin(k)}$ la solution obtenue en supprimant de $S$ toutes les tournées de la période $k$. Alors $E(S^{\notin(k)}) = E(S) \setminus E(S^k)$ et $S^{\notin(k)}$ est une solution de $m$-PTVP.

Supposons donc que $H(m)$ est vraie et que l'on souhaite fusionner deux tournées d'une période $k^1 \in [1...m+1]$ d'une solution $S$ de $(m+1)$-PTVP. Soit $i$ le dernier client servi par la première tournée et soient $j$ et $z$ deux clients respectivement servis en premier et en dernier par la 2ème tournée. Montrons que dans tous les cas, $H(m+1)$ est vraie.

**Cas 1 : $(i, j) \notin E(S)$.** Au sein de la période $k^1$, les arêtes $(0, i)$ et $(0, j)$ peuvent être remplacées par $(i, j)$ ; donc $H(m+1)$ est vraie.

**Cas 2 : $(i, j) \in E(S)$ et $(i, z) \notin E(S)$.** Au sein de la période $k^1$, les arêtes $(0, i)$ et $(0, w)$ peuvent être remplacées par $(i, w)$ ; donc $H(m+1)$ est vraie.

**Cas 3 : $(i, j) \in E(S)$ et $(i, z) \in E(S)$.** $\forall k^2 \in [1...m+1] \setminus \{k^1\}$, au sein d'une même période de $S^{\notin(k^2)}$ les tournées peuvent être fusionnées deux par deux, puisque $S^{\notin(k^2)}$ est une solution de $m$-PTVP et que $H(m)$ est vraie. Notons $(i, a)$ la nouvelle arête qui sera insérée dans $E(S^{\notin(k^2)})$ suite à la fusion des deux tournées de la période $k^1$.

- **Cas 3.1 : $(i, j) \in E(S)$, $(i, z) \in E(S)$ et $(i, a) \notin E(S^{k^2})$**
  Puisque $(i, a)$ n'est pas déjà utilisée, prendre en compte ou ignorer la période $k^2$ ne fait aucune différence. Pour fusionner les deux tournées de la période $k^1$, il suffit d'appliquer à $S$ les mêmes échanges d'arêtes que ceux appliqués sur $S^{\notin(k^2)}$, puis insérer $(i, a)$ en remplacement de $(0, i)$ et $(0, a)$. Donc $H(m+1)$ est vraie.
- **Cas 3.2 : $(i, j) \in E(S)$, $(i, z) \in E(S)$ et $(i, a) \in E(S^{k^2})$**

- Si $a = p_i^{k^2}$ et $(i, \alpha_a^{k^2}) \notin E(S)$, alors après avoir appliqué à $S$ les mêmes échanges d'arêtes que ceux appliqués sur $S^{\notin(k^2)}$ ; il suffit d'échanger les arêtes $(0, i)$, $(0, a)$ et $(i, a)$, puis de connecter les tournées de la période $k^2$ avec l'arête $(i, \alpha_a^{k^2})$ pour que $H(m+1)$ est vraie.
- De même, si $a = s_i^{k^2}$ et $(i, \omega_a^{k^2}) \notin E(S)$, alors après avoir appliqué à $S$ les mêmes échanges d'arêtes que ceux appliqués sur $S^{\notin(k^2)}$ ; il suffit d'échanger les arêtes $(0, i)$, $(0, a)$ et $(i, a)$, puis de connecter les tournées de la période $k^2$ avec l'arête $(i, \omega_a^{k^2})$ pour que $H(m+1)$ est vraie.

En résumé, pour que $H(m+1)$ soit fausse, sachant que $H(m)$ est vraie et que $(0, i) \in E(S^{k^1}) \subset E(S)$, il faudrait en plus que :

1. $(i, j) \in E(S)$
2. $(i, z) \in E(S)$
3. $\forall k \in [1...m+1] \setminus \{k^1\}$ :
   - soit $(p_i^k, i) \in E(S)$ et $(\alpha_{p_i^k}^k, i) \in E(S)$
   - soit $(s_i^k, i) \in E(S)$ et $(\omega_{s_i^k}^k, i) \in E(S)$

Toutes ces arêtes incidentes à $i$ et qui doivent être utilisées au sein de $S$ sont au nombre de : $1+1+1+2m = 2m+3$. Or $S$ étant une solution de $(m+1)$-PTVP, il ne peut y avoir que $2(m+1)$ arêtes incidentes à $i$. Il est donc impossible que $H(m+1)$ soit fausse.

## A-1.3 : En déduire que $\forall m \geq 1$, le $m$-PTVP se réduit au $m$-PVP en l'absence de contraintes de capacité

Nous avons donc montré que si $H(m)$ est vraie, alors $H(m+1)$ est forcément vraie. Or $H(1)$ est vraie (cas trivial du VRP), ainsi que $H(2)$ (démontré au paragraphe A-1.1 ; donc $H(m)$ est vraie $\forall m \geq 1$ : en l'absence de contraintes de capacité, $\forall m \geq 1$, le $m$-PTVP se réduit au $m$-PVP.

# A-2: Résultats numériques des algorithmes développés pour le $m$-PTVP

Pour l'analyse des résultats des algorithmes développés au cours de la thèse, lire les chapitres de thèse correspondants.

## A-2.1: Résultats détaillés des métaheuristiques et des bornes inférieures polynômiales

Légende: $m$ = nombre de périodes; UB* = optimum (prouvé); $\text{UB}^{\text{M}}$ = meilleure borne supérieure trouvée par les métaheuristiques; $\text{LB}_{\text{vrp}}$ = borne inférieure produite par Baldacci *et al.*[10]; $m$VRP = borne inférieure égale au produit par $m$ du coût de l'optimum du VRP sur le même graphe; LB1 = borne inférieure égale au coût de $m$ 1-arbres recouvrants à arêtes disjointes; LB2 = borne inférieure obtenue en résolvant le $b$-couplage parfait; CW = heuristique de Clarke et Wright; ILS = métaheuristique de recherche locale itérative; TS = métaheuristique de recherche taboue; GTSD = métaheuristique de recherche taboue guidée et diversifiée; $\Delta\%$ = écart en pourcentage entre le résultat d'une métaheuristique et UB*; $\delta\%$ = écart en pourcentage entre le résultat d'une métaheuristique et $\text{UB}^{\text{M}}$; It = nombre d'itérations pour atteindre la meilleure solution; $T$ = durée en secondes pour atteindre la meilleure solution; $Bs$ = Durée de calcul en secondes de la solution du $b$-couplage parfait relaxé utilisée pour guider TS; $Bp$ = ratio entre le nombre d'arêtes utilisées par la solution du $b$-couplage parfait relaxé et le nombre total d'arêtes du graphe.

### A-2.1.1: Bornes inférieures polynômiales, Heuristique et Métaheuristiques pour le 1-PTVP=VRP

| Instance | $m$ | UB* | Bornes_Inferieures | | | CW | ILS | | | TS | | | GTSD | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | $\frac{\text{LB}_{\text{vrp}}}{\text{UB*}}$ % | $\frac{\text{LB1}}{\text{UB*}}$ % | $\frac{\text{LB2}}{\text{UB*}}$ % | $\Delta\%$ | $\Delta\%$ | It | $T$ | $\Delta\%$ | It | $T$ | $\Delta\%$ | It | $T$ | $Bs$ | $Bp$ |
| A-n32-k5 | 1 | 784,00 | – | 59,44 | 79,08 | 7,02 | 5,48 | 41 | 0,00 | 0,00 | 1114 | 0,45 | 0,00 | 444 | 0,16 | 0,08 | 0,11 |
| A-n33-k5 | 1 | 661,00 | – | 65,05 | 80,18 | 7,87 | 0,00 | 6595 | 1,56 | 0,00 | 164 | 0,06 | 0,00 | 169 | 0,05 | 0,07 | 0,12 |
| A-n33-k6 | 1 | 742,00 | – | 63,61 | 82,48 | 4,31 | 0,13 | 818 | 0,19 | 0,00 | 5245 | 2,23 | 0,00 | 3668 | 1,51 | 0,08 | 0,13 |
| A-n34-k5 | 1 | 778,00 | – | 61,44 | 80,59 | 3,98 | 1,41 | 422 | 0,11 | 1,41 | 2069 | 0,91 | 1,41 | 1355 | 0,58 | 0,07 | 0,13 |
| A-n36-k5 | 1 | 799,00 | – | 59,95 | 75,97 | 3,38 | 0,00 | 153 | 0,03 | 0,00 | 2585 | 1,31 | 0,00 | 1889 | 0,92 | 0,07 | 0,13 |
| A-n37-k5 | 1 | 669,00 | 96,10 | 76,68 | 87,67 | 5,38 | 0,30 | 4 | 0,00 | 0,00 | 207 | 0,11 | 0,00 | 214 | 0,11 | 0,07 | 0,08 |
| A-n37-k6 | 1 | 949,00 | 95,90 | 53,53 | 67,44 | 5,06 | 2,32 | 9397 | 2,54 | 0,11 | 9841 | 4,68 | 0,00 | 8054 | 3,87 | 0,07 | 0,15 |
| A-n38-k5 | 1 | 730,00 | 92,50 | 63,84 | 74,93 | 4,79 | 0,55 | 18 | 0,02 | 0,14 | 657 | 0,36 | 0,14 | 85 | 0,03 | 0,06 | 0,07 |
| A-n39-k5 | 1 | 822,00 | 97,50 | 65,09 | 73,97 | 9,49 | 1,95 | 6628 | 2,00 | 0,00 | 821 | 0,45 | 1,22 | 987 | 0,53 | 0,06 | 0,15 |
| A-n39-k6 | 1 | 831,00 | 95,30 | 63,66 | 78,70 | 3,61 | 1,44 | 33 | 0,02 | 0,84 | 1563 | 0,95 | 0,36 | 8197 | 5,06 | 0,06 | 0,08 |
| A-n44-k7 | 1 | 937,00 | 97,90 | 62,22 | 80,68 | – | 1,60 | 341 | 0,14 | 0,00 | 9008 | 6,33 | 0,00 | 5823 | 4,09 | 0,05 | 0,22 |
| A-n45-k6 | 1 | 944,00 | 94,80 | 60,28 | 76,48 | – | 3,28 | 3 | 0,00 | 0,42 | 2279 | 1,61 | 0,42 | 2324 | 1,67 | 0,05 | 0,16 |
| A-n45-k7 | 1 | 1146,00 | 97,80 | 45,72 | 68,11 | 5,15 | 3,84 | 698 | 0,30 | 0,17 | 3565 | 2,78 | 0,17 | 4720 | 3,89 | 0,05 | 0,19 |
| A-n46-k7 | 1 | 914,00 | 98,00 | 60,94 | 80,85 | 2,84 | 0,88 | 232 | 0,09 | 0,33 | 129 | 0,09 | 0,33 | 2760 | 2,23 | 0,06 | 0,09 |
| A-n48-k7 | 1 | 1073,00 | 96,90 | 52,56 | 77,49 | 2,70 | 2,24 | 410 | 0,20 | 1,03 | 2711 | 2,40 | 1,21 | 8009 | 7,55 | 0,05 | 0,13 |
| A-n53-k7 | 1 | 1010,00 | 97,00 | 56,14 | 78,66 | 5,25 | 2,38 | 3034 | 1,78 | 0,00 | 970 | 0,97 | 0,69 | 1043 | 0,97 | 0,05 | 0,09 |
| A-n54-k7 | 1 | 1167,00 | 95,80 | 48,59 | 65,51 | 0,60 | 0,43 | 1 | 0,02 | 0,00 | 7992 | 8,39 | 0,17 | 2788 | 3,00 | 0,05 | 0,11 |
| A-n55-k9 | 1 | 1073,00 | 95,70 | 50,79 | 74,04 | 2,70 | 1,12 | 217 | 0,13 | 0,00 | 2966 | 3,32 | 0,75 | 2734 | 3,10 | 0,05 | 0,08 |
| A-n60-k9 | 1 | 1354,00 | 97,00 | 42,84 | 62,26 | 4,87 | 3,62 | 335 | 0,25 | 2,88 | 3472 | 4,65 | 0,81 | 340 | 0,34 | 0,04 | 0,14 |
| A-n61-k9 | 1 | 1034,00 | 96,40 | 52,61 | 74,08 | 1,84 | 0,68 | 1347 | 1,08 | 0,10 | 1945 | 2,54 | 0,19 | 6665 | 9,67 | 0,04 | 0,07 |
| A-n62-k8 | 1 | 1290,00 | 95,90 | 46,98 | 66,20 | 3,49 | 1,63 | 380 | 0,31 | 1,63 | 1103 | 1,61 | 1,63 | 68 | 0,02 | 0,04 | 0,11 |
| A-n63-k9 | 1 | 1616,00 | 97,30 | 38,61 | 69,12 | – | 1,79 | 2376 | 2,00 | 0,68 | 8744 | 12,11 | 0,68 | 2506 | 3,42 | 0,04 | 0,23 |
| A-n63-k10 | 1 | 1315,00 | 96,50 | 46,24 | 68,40 | 4,26 | 1,90 | 207 | 0,17 | 0,30 | 183 | 0,27 | 0,38 | 334 | 0,27 | 0,04 | 0,11 |
| A-n64-k9 | 1 | 1402,00 | 97,10 | 42,87 | 75,14 | 3,92 | 1,43 | 1626 | 1,44 | 0,29 | 3755 | 5,54 | 0,21 | 2549 | 3,65 | 0,04 | 0,16 |
| A-n65-k9 | 1 | 1174,00 | 96,70 | 51,53 | 77,34 | 6,05 | 2,47 | 5120 | 4,10 | 1,70 | 5442 | 7,96 | 1,96 | 1042 | 1,50 | 0,04 | 0,09 |

| | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A-n69-k9 | 1 | 1159,00 | 96,70 | 59,45 | 79,59 | 2,93 | 2,07 | 879 | 0,86 | 1,64 | 3118 | 5,54 | 1,29 | 9964 | 18,22 | 0,04 | 0,07 |
| A-n80-k10 | 1 | 1763,00 | 97,70 | 41,86 | 74,90 | 4,65 | 4,20 | 3120 | 3,90 | 1,47 | 4743 | 11,06 | 0,45 | 6728 | 16,12 | 0,03 | 0,15 |
| B-n31-k5 | 1 | 672,00 | 99,40 | 41,22 | 75,89 | 0,60 | 0,45 | 2658 | 0,61 | 0,00 | 2463 | 1,01 | 0,00 | 3313 | 1,39 | 0,08 | 0,55 |
| B-n34-k5 | 1 | 788,00 | 98,90 | 39,85 | 73,29 | 0,63 | 0,00 | 1664 | 0,42 | 0,00 | 4 | 0,00 | 0,00 | 3 | 0,00 | 0,07 | 0,31 |
| B-n35-k5 | 1 | 955,00 | 98,90 | 40,52 | 67,12 | 2,51 | 0,42 | 9295 | 2,53 | 0,00 | 583 | 0,28 | 0,00 | 29 | 0,00 | 0,07 | 0,23 |
| B-n38-k6 | 1 | 805,00 | 99,60 | 42,11 | 62,98 | 3,11 | 1,24 | 5 | 0,02 | 0,75 | 2120 | 1,26 | 0,00 | 1114 | 0,66 | 0,06 | 0,24 |
| B-n39-k5 | 1 | 549,00 | – | 56,28 | 60,75 | 2,91 | 2,00 | 1 | 0,00 | 0,18 | 349 | 0,22 | 0,18 | 635 | 0,33 | 0,06 | 0,14 |
| B-n41-k6 | 1 | 829,00 | – | 45,96 | 45,72 | 8,56 | 0,36 | 9835 | 3,84 | 4,34 | 981 | 0,61 | 4,46 | 328 | 0,16 | 0,06 | 0,17 |
| B-n43-k6 | 1 | 742,00 | – | 43,13 | 53,77 | 1,75 | 0,00 | 9828 | 4,06 | 0,13 | 2547 | 1,83 | 0,00 | 154 | 0,08 | 0,06 | 0,17 |
| B-n44-k7 | 1 | 909,00 | 98,90 | 32,67 | 50,00 | 3,08 | 2,31 | 6034 | 2,48 | 0,00 | 1421 | 1,01 | 0,00 | 919 | 0,59 | 0,06 | 0,16 |
| B-n45-k5 | 1 | 751,00 | 99,30 | 60,72 | 65,05 | 0,53 | 0,00 | 1 | 0,00 | 0,00 | 5 | 0,00 | 0,00 | 1 | 0,00 | 0,05 | 0,11 |
| B-n45-k6 | 1 | 678,00 | 99,10 | 46,76 | 64,90 | 5,60 | 5,46 | 1 | 0,00 | 5,31 | 4188 | 3,56 | 5,16 | 4376 | 3,74 | 0,05 | 0,17 |
| B-n50-k7 | 1 | 741,00 | 100,00 | 50,88 | 70,58 | 0,27 | 0,27 | 0 | 0,00 | 0,00 | 14 | 0,02 | 0,00 | 10 | 0,00 | 0,05 | 0,13 |
| B-n50-k8 | 1 | 1312,00 | 99,40 | 35,98 | 59,11 | 4,04 | 0,69 | 4263 | 2,23 | 0,61 | 1777 | 1,69 | 0,61 | 1232 | 1,09 | 0,05 | 0,32 |
| B-n51-k7 | 1 | 1020,00 | 99,20 | 44,12 | 73,63 | 1,57 | 0,20 | 4034 | 2,43 | 0,00 | 2656 | 2,93 | 0,10 | 1938 | 2,08 | 0,05 | 0,16 |
| B-n52-k7 | 1 | 747,00 | 99,70 | 47,52 | 52,41 | 1,74 | 1,20 | 4698 | 2,95 | 0,54 | 1259 | 1,37 | 0,54 | 222 | 0,17 | 0,05 | 0,17 |
| B-n56-k7 | 1 | 707,00 | 98,20 | 46,39 | 47,24 | 4,10 | 2,69 | 3 | 0,02 | 0,42 | 2098 | 2,64 | 0,42 | 703 | 0,72 | 0,04 | 0,16 |
| B-n57-k7 | 1 | 1140,00 | 97,90 | 37,19 | 51,49 | 8,51 | 0,00 | 8 | 0,03 | 0,00 | 11 | 0,02 | 0,00 | 1547 | 1,95 | 0,04 | 0,17 |
| B-n57-k9 | 1 | 1598,00 | 98,70 | 27,97 | 73,56 | 4,13 | 2,63 | 5 | 0,02 | 0,06 | 2774 | 3,51 | 0,06 | 3437 | 4,20 | 0,04 | 0,26 |
| B-n63-k10 | 1 | 1496,00 | 99,60 | 31,08 | 65,51 | 7,02 | 4,75 | 7491 | 6,13 | 3,14 | 4253 | 6,47 | 2,74 | 9053 | 13,48 | 0,04 | 0,21 |
| B-n64-k9 | 1 | 861,00 | 97,40 | 44,48 | 61,85 | 7,32 | 0,93 | 0 | 0,00 | 0,12 | 1540 | 2,48 | 2,21 | 8401 | 12,31 | 0,04 | 0,14 |
| B-n66-k9 | 1 | 1316,00 | 99,70 | 36,70 | 61,78 | 6,91 | 2,51 | 647 | 0,64 | 0,30 | 1342 | 2,20 | 1,75 | 747 | 0,84 | 0,04 | 0,16 |
| B-n67-k10 | 1 | 1032,00 | 99,00 | 43,41 | 64,68 | 6,20 | 4,84 | 7984 | 7,78 | 0,39 | 6708 | 11,64 | 1,07 | 8782 | 15,55 | 0,04 | 0,16 |
| B-n68-k9 | 1 | 1275,00 | 99,80 | 29,96 | 50,71 | 2,67 | 1,25 | 4191 | 4,09 | 1,02 | 288 | 0,53 | 1,02 | 6882 | 12,14 | 0,04 | 0,15 |
| B-n78-k10 | 1 | 1221,00 | 98,00 | 39,72 | 53,28 | 2,29 | 2,21 | 3 | 0,02 | 1,97 | 31 | 0,08 | 1,06 | 7006 | 15,35 | 0,03 | 0,11 |
| P-n19-k2 | 1 | 212,00 | 95,60 | 80,19 | 89,15 | 11,79 | 10,38 | 42 | 0,00 | 0,00 | 1052 | 0,13 | 0,00 | 1137 | 0,13 | 0,12 | 0,15 |
| P-n20-k2 | 1 | 216,00 | 97,10 | 81,48 | 93,06 | 8,33 | 0,00 | 4841 | 0,37 | 0,00 | 184 | 0,02 | 0,00 | 414 | 0,05 | 0,11 | 0,13 |
| P-n21-k2 | 1 | 211,00 | 99,20 | 84,36 | 94,55 | 11,85 | 0,00 | 26 | 0,00 | 0,00 | 7 | 0,00 | 0,00 | 7 | 0,00 | 0,12 | 0,12 |
| P-n22-k2 | 1 | 216,00 | 98,60 | 84,26 | 93,98 | 11,11 | 0,00 | 233 | 0,02 | 0,00 | 13 | 0,02 | 0,00 | 11 | 0,00 | 0,11 | 0,13 |
| P-n22-k8 | 1 | 603,00 | 98,80 | 45,27 | 85,99 | – | 4,64 | 4 | 0,00 | 0,17 | 174 | 0,03 | 0,17 | 729 | 0,14 | 0,14 | 0,61 |
| P-n23-k8 | 1 | 529,00 | 99,80 | 34,40 | 83,93 | – | 2,08 | 35 | 0,00 | 2,08 | 10 | 0,02 | 2,08 | 10 | 0,00 | 0,12 | 0,71 |
| P-n40-k5 | 1 | 458,00 | 99,60 | 77,95 | 93,23 | 10,26 | 6,11 | 265 | 0,09 | 0,00 | 2891 | 1,90 | 0,00 | 1286 | 0,73 | 0,06 | 0,08 |
| P-n45-k5 | 1 | 510,00 | 98,20 | 78,43 | 92,25 | 3,53 | 3,53 | 0 | 0,00 | 0,00 | 727 | 0,58 | 0,00 | 803 | 0,59 | 0,05 | 0,07 |
| P-n50-k10 | 1 | 696,00 | 98,30 | 55,46 | 84,48 | – | 5,89 | 685 | 0,39 | 0,57 | 171 | 0,19 | 0,00 | 9470 | 8,53 | 0,05 | 0,19 |
| P-n50-k7 | 1 | 554,00 | 98,90 | 69,68 | 89,62 | 5,42 | 3,07 | 332 | 0,17 | 0,36 | 106 | 0,11 | 0,36 | 165 | 0,09 | 0,05 | 0,08 |
| P-n50-k8 | 1 | 631,00 | 97,00 | 61,17 | 83,12 | 2,06 | 1,58 | 99 | 0,05 | 0,48 | 1718 | 1,72 | 0,32 | 9059 | 9,05 | 0,05 | 0,10 |
| P-n51-k10 | 1 | 741,00 | 98,40 | 57,09 | 84,35 | 4,72 | 3,78 | 3 | 0,00 | 0,27 | 1563 | 1,50 | 0,13 | 5806 | 5,21 | 0,05 | 0,12 |
| P-n55-k10 | 1 | 694,00 | 97,60 | 58,36 | 85,81 | 5,19 | 2,59 | 63 | 0,05 | 0,86 | 295 | 0,38 | 0,58 | 431 | 0,41 | 0,05 | 0,11 |
| P-n55-k15 | 1 | 945,00 | 97,00 | 42,86 | 83,07 | – | 2,65 | 744 | 0,48 | 0,00 | 7011 | 7,74 | 0,00 | 436 | 0,41 | 0,05 | 0,25 |
| P-n55-k7 | 1 | 568,00 | 97,50 | 71,30 | 89,44 | 8,63 | 5,99 | 243 | 0,17 | 0,70 | 4936 | 5,99 | 1,41 | 2217 | 2,61 | 0,04 | 0,09 |
| P-n55-k8 | 1 | 576,00 | 97,30 | 70,31 | 88,19 | 9,20 | 7,64 | 959 | 0,59 | 0,52 | 2324 | 2,70 | 0,52 | 3848 | 4,27 | 0,04 | 0,09 |
| P-n60-k10 | 1 | 744,00 | 98,70 | 60,22 | 85,48 | 7,12 | 3,90 | 1014 | 0,73 | 0,27 | 824 | 1,11 | 0,54 | 2316 | 2,87 | 0,04 | 0,12 |
| P-n60-k15 | 1 | 968,00 | 98,70 | 47,00 | 85,59 | – | 3,51 | 98 | 0,09 | 0,00 | 5567 | 7,49 | 0,83 | 5778 | 7,80 | 0,05 | 0,21 |
| P-n65-k10 | 1 | 792,00 | 98,60 | 63,13 | 87,37 | 7,07 | 4,42 | 2922 | 2,53 | 1,52 | 3801 | 6,12 | 0,00 | 5431 | 8,49 | 0,04 | 0,10 |
| P-n70-k10 | 1 | 827,00 | 97,70 | 63,72 | 84,10 | – | 3,51 | 2921 | 2,98 | 1,09 | 6662 | 11,70 | 0,24 | 2194 | 3,59 | 0,04 | 0,14 |
| P-n76-k4 | 1 | 593,00 | 98,80 | 90,56 | 95,11 | 12,31 | 3,20 | 4650 | 6,18 | 1,18 | 1152 | 2,43 | 1,01 | 331 | 0,30 | 0,03 | 0,04 |
| P-n76-k5 | 1 | 627,00 | 97,70 | 85,65 | 92,50 | 8,77 | 6,06 | 2688 | 2,98 | 0,64 | 543 | 1,17 | 0,48 | 6081 | 12,22 | 0,03 | 0,05 |
| P-n101-k4 | 1 | 681,00 | 97,90 | 92,07 | 96,18 | 9,25 | 3,08 | 6394 | 13,96 | 0,73 | 7318 | 30,40 | 0,73 | 3074 | 11,86 | 0,02 | 0,04 |
| VRPNC1 | 1 | 524,61 | – | 80,79 | 92,31 | 11,44 | 7,26 | 9940 | 6,10 | 0,00 | 3483 | 3,18 | 0,00 | 4917 | 4,62 | 0,05 | 0,07 |
| VRPNC2 | 1 | 835,26 | – | 65,02 | 83,71 | 7,78 | 4,86 | 9710 | 10,90 | 1,50 | 1321 | 2,70 | 0,57 | 6476 | 13,71 | 0,03 | 0,08 |
| VRPNC3 | 1 | 826,14 | – | 77,19 | 91,23 | 7,35 | 5,89 | 9966 | 21,36 | 0,93 | 1000 | 3,96 | 1,19 | 6551 | 27,00 | 0,02 | 0,05 |
| VRPNC4 | 1 | 1028,42 | – | 68,63 | 85,68 | 10,21 | 8,65 | 9917 | 46,91 | 2,45 | 6433 | 54,60 | 1,08 | 8443 | 73,45 | 0,01 | 0,04 |

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| VRPNC5 | 1 | 1291,29 | – | 59,89 | 77,84 | 8,09 | 5,83 | 9845 | 77,83 | 4,87 | 1612 | 24,13 | 5,33 | 542 | 5,54 | 0,01 | 0,04 |
| VRPNC11 | 1 | 1042,11 | – | 50,51 | 45,34 | 2,50 | 0,46 | 9819 | 27,35 | 0,46 | 12 | 0,06 | 0,46 | 12 | 0,00 | 0,02 | 0,10 |
| VRPNC12 | 1 | 819,56 | – | 60,05 | 74,16 | 1,70 | 0,19 | 9957 | 23,17 | 0,21 | 43 | 0,19 | 0,17 | 31 | 0,00 | 0,02 | 0,06 |

Table 1: $m$*VRP, LB1, LB2, CW, ILS, TS et GTSD pour le 1-PTVP = VRP

### A-2.1.2: Bornes inférieures polynômiales, Heuristique et Métaheuristiques pour le $m$-PVP = $m$-PTVP avec $\lambda = 1$

| Instance | $m$ | UB* | Bornes_Inferieures | | | CW | ILS | | | TS | | | GTSD | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | $\frac{mTSP}{UB^*}$ % | $\frac{LB1}{UB^*}$ % | $\frac{LB2}{UB^*}$ % | Δ% | Δ% | It | $T$ | Δ% | It | $T$ | Δ% | It | $T$ | $Bs$ | $Bp$ |
| St70 | 2 | 1619,00 | 83,38 | 87,03 | 99,20 | 13,03 | 1,73 | 6665 | 33,42 | 1,73 | 3895 | 23,09 | 2,22 | 641 | 2,89 | 0,06 | 0,02 |
| Eil76 | 2 | 1257,00 | 85,60 | 89,63 | 100,00 | 10,18 | 0,88 | 2915 | 17,11 | 1,35 | 460 | 3,31 | 1,59 | 9101 | 61,54 | 0,06 | 0,03 |
| Pr76 | 2 | 266395,00 | 81,20 | 84,78 | 99,43 | 14,72 | 1,90 | 9766 | 57,08 | 1,98 | 3088 | 21,67 | 1,36 | 584 | 2,86 | 0,05 | 0,04 |
| rat99 | 2 | 2928,00 | 82,72 | 93,13 | 99,93 | 11,20 | 1,71 | 8989 | 91,89 | 1,74 | 734 | 8,81 | 2,19 | 1116 | 10,17 | 0,04 | 0,02 |
| KroB100 | 2 | 53600,00 | 82,62 | 90,10 | 99,58 | 14,31 | 2,33 | 8026 | 85,05 | 2,83 | 2996 | 37,60 | 1,96 | 8802 | 104,49 | 0,04 | 0,02 |
| KroC100 | 2 | 51157,00 | 81,12 | 89,85 | 99,41 | 14,67 | 1,83 | 9622 | 102,24 | 2,25 | 4947 | 62,10 | 2,33 | 2674 | 30,51 | 0,04 | 0,01 |
| KroD100 | 2 | 51838,00 | 82,16 | 90,04 | 99,50 | 16,31 | 2,25 | 9731 | 102,96 | 2,65 | 377 | 4,79 | 2,32 | 1166 | 11,39 | 0,04 | 0,02 |
| KroE100 | 2 | 54183,00 | 81,46 | 88,84 | 100,00 | 11,47 | 2,42 | 9165 | 99,30 | 2,51 | 5676 | 71,34 | 2,59 | 4755 | 56,24 | 0,04 | 0,02 |
| Eil101 | 2 | 1453,00 | 86,58 | 89,92 | 100,00 | 6,47 | 1,31 | 6769 | 71,75 | 1,86 | 4226 | 53,32 | 2,20 | 1118 | 11,95 | 0,04 | 0,03 |
| Lin105 | 2 | 36361,00 | 79,09 | 90,65 | 99,43 | 18,29 | 0,78 | 6911 | 77,92 | 0,99 | 7616 | 103,96 | 1,05 | 5612 | 71,87 | 0,04 | 0,04 |
| Pr107 | 2 | 98381,00 | 90,06 | 82,50 | 69,88 | 18,88 | 1,33 | 8520 | 100,87 | 0,98 | 9151 | 128,67 | 0,81 | 6465 | 86,47 | 0,04 | 0,02 |
| Ch130 | 2 | 14854,00 | 82,27 | | 99,60 | 14,17 | 2,65 | 7861 | 141,67 | 2,30 | 1088 | 23,74 | 2,07 | 360 | 3,32 | 0,03 | 0,01 |
| Pr136 | 2 | 220642,00 | 87,72 | 90,47 | 97,98 | 15,70 | 2,87 | 9543 | 189,29 | 3,32 | 606 | 14,34 | 3,76 | 333 | 3,03 | 0,03 | 0,01 |
| kroA150 | 2 | 64249,00 | 82,57 | 90,60 | 99,49 | 11,49 | 1,88 | 9459 | 234,47 | 2,73 | 451 | 13,57 | 2,46 | 4683 | 126,58 | 0,03 | 0,01 |
| d198 | 2 | 35404,00 | 89,14 | | 82,86 | 20,21 | 1,94 | 9747 | 420,36 | 3,64 | 7046 | 365,07 | 1,37 | 8830 | 396,88 | 0,02 | 0,06 |
| kroA200 | 2 | 70871,00 | 82,88 | 91,95 | 99,43 | 12,90 | 3,08 | 9097 | 413,87 | 2,81 | 8955 | 474,66 | 3,15 | 4504 | 222,01 | 0,02 | 0,01 |
| pr226 | 2 | 183537,00 | 87,58 | | 87,96 | 23,19 | 1,70 | 7095 | 408,13 | 3,11 | 110 | 7,30 | 4,10 | 1402 | 61,71 | 0,02 | 0,02 |
| pr264 | 2 | 122941,00 | 79,93 | | 84,81 | 11,47 | 2,30 | 9420 | 744,89 | 2,67 | 2396 | 217,39 | 2,84 | 802 | 29,72 | 0,02 | 0,02 |
| a280 | 2 | 6317,00 | 81,65 | | 100,00 | 15,61 | 4,86 | 9581 | 838,16 | 3,80 | 1698 | 171,66 | 3,34 | 4522 | 395,04 | 0,01 | 0,01 |
| Eil76 | 3 | 2141,00 | 75,39 | 89,95 | 100,00 | 10,23 | 1,35 | 7333 | 61,65 | 1,54 | 292 | 2,98 | 1,07 | 682 | 4,74 | 0,08 | 0,04 |
| Pr76 | 3 | 485719,00 | 66,80 | 83,39 | 100,00 | 12,75 | 1,31 | 9540 | 78,86 | 2,09 | 2033 | 20,67 | 1,25 | 9910 | 94,19 | 0,08 | 0,05 |
| KroB100 | 3 | 96691,00 | 68,70 | 89,66 | 99,93 | 11,38 | 1,52 | 9544 | 144,11 | 1,21 | 2609 | 46,79 | 1,91 | 577 | 5,84 | 0,06 | 0,02 |
| KroD100 | 3 | 91200,00 | 70,05 | 90,77 | 100,00 | 15,32 | 2,48 | 7791 | 113,87 | 1,99 | 799 | 14,51 | 1,86 | 1792 | 25,13 | 0,06 | 0,02 |
| KroC100 | 3 | 92775,00 | 67,09 | 90,70 | 100,00 | 11,95 | 2,40 | 5930 | 87,50 | 1,99 | 4366 | 79,06 | 1,64 | 5457 | 90,79 | 0,06 | 0,02 |
| KroE100 | 3 | 96406,00 | 68,67 | 90,06 | 100,00 | 10,52 | 1,96 | 8711 | 130,37 | 1,50 | 690 | 12,45 | 1,64 | 4230 | 68,31 | 0,06 | 0,02 |
| Eil101 | 3 | 2468,00 | 76,46 | 89,58 | 100,00 | 8,23 | 2,19 | 8588 | 132,21 | 1,66 | 1492 | 26,91 | 1,30 | 8059 | 137,23 | 0,06 | 0,04 |
| Pr107 | 3 | 169298,00 | 78,51 | 85,02 | 78,12 | 18,07 | 1,80 | 9307 | 158,54 | 1,35 | 3556 | 71,37 | 1,62 | 1412 | 20,47 | 0,06 | 0,03 |
| Pr124 | 3 | 265906,00 | 66,60 | 89,38 | 97,31 | 13,73 | 2,40 | 8728 | 204,86 | 2,04 | 754 | 21,08 | 2,28 | 1530 | 37,08 | 0,05 | 0,02 |
| Pr136 | 3 | 383804,00 | 75,64 | 89,97 | 98,76 | 12,96 | 3,00 | 8891 | 248,95 | 2,41 | 4483 | 150,62 | 2,19 | 4413 | 134,22 | 0,04 | 0,02 |
| St70 | 4 | 4340,00 | 62,21 | 89,81 | 100,00 | 9,22 | 1,50 | 9670 | 89,22 | 1,04 | 1977 | 21,08 | 1,22 | 2154 | 20,28 | 0,12 | 0,03 |
| Pr76 | 4 | 738418,00 | 58,59 | 84,29 | 100,00 | 11,33 | 1,85 | 8634 | 90,15 | 2,02 | 2929 | 37,86 | 1,83 | 3459 | 39,14 | 0,11 | 0,06 |
| KroB100 | 4 | 147224,00 | 60,16 | 89,95 | 100,00 | 9,36 | 0,93 | 9244 | 174,44 | 1,52 | 978 | 22,62 | 1,74 | 321 | 2,67 | 0,08 | 0,02 |
| KroD100 | 4 | 139652,00 | 60,99 | 90,92 | 100,00 | 14,38 | 2,07 | 9897 | 189,17 | 1,11 | 6299 | 145,85 | 1,50 | 9421 | 206,28 | 0,08 | 0,02 |
| Pr107 | 4 | 263332,00 | 67,30 | 86,76 | 82,32 | 17,71 | 1,22 | 9625 | 210,07 | 1,04 | 9818 | 254,72 | 1,02 | 5248 | 123,93 | 0,08 | 0,04 |
| Pr124 | 4 | 412429,00 | 57,25 | 89,23 | 98,25 | 11,68 | 2,56 | 8055 | 235,86 | 2,82 | 6750 | 242,19 | 2,64 | 7765 | 253,49 | 0,07 | 0,02 |

Table 2: $m$*VRP, LB1, LB2, CW, ILS, TS et GTSD pour le $m$-PVP

### A-2.1.3: Bornes inférieures polynômiales, Heuristique et Métaheuristiques pour le $m$-PTVP

| Instance | $m$ | $UB^M$ | Bornes_Inferieures | | | CW | ILS | | | TS | | | GTSD | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | $\frac{m\text{VRP}}{UB^M}$ | $\frac{LB1}{UB^M}$ | $\frac{LB2}{UB^M}$ | $\delta\%$ | $\delta\%$ | It | $T$ | $\delta\%$ | It | $T$ | $\delta\%$ | It | $T$ | $Bs$ | $Bp$ |
| A-n32-k5 | 2 | 1806,00 | 86,82 | 63,29 | 98,01 | – | 4,65 | 1424 | 0,58 | 1,61 | 1415 | 1,06 | 1,77 | 1729 | 1,25 | 0,15 | 0,24 |
| A-n33-k5 | 2 | 1501,00 | 88,08 | 71,95 | 94,67 | 21,52 | 4,86 | 539 | 0,22 | 0,07 | 8212 | 6,07 | 0,33 | 4396 | 3,23 | 0,14 | 0,19 |
| A-n33-k6 | 2 | 1653,00 | 89,78 | 68,00 | 96,19 | – | 4,90 | 663 | 0,25 | 0,60 | 6236 | 4,65 | 1,03 | 4722 | 3,48 | 0,15 | 0,31 |
| A-n34-k5 | 2 | 1690,00 | 92,07 | 67,22 | 95,80 | 17,93 | 5,50 | 1194 | 0,52 | 1,12 | 8711 | 6,68 | 0,12 | 258 | 0,16 | 0,14 | 0,19 |
| A-n36-k5 | 2 | 1833,00 | 87,18 | 64,86 | 95,96 | – | 2,45 | 6405 | 3,04 | 1,42 | 673 | 0,61 | 0,16 | 6968 | 6,51 | 0,13 | 0,24 |
| A-n37-k5 | 2 | 1583,00 | 84,52 | 77,39 | 97,03 | – | 2,52 | 2222 | 1,20 | 0,57 | 1930 | 1,97 | 0,57 | 6551 | 6,76 | 0,13 | 0,15 |
| A-n37-k6 | 2 | 2040,00 | 93,04 | 61,37 | 93,19 | – | 4,07 | 1093 | 0,55 | 1,62 | 7840 | 6,61 | 0,39 | 4315 | 3,60 | 0,13 | 0,32 |
| A-n38-k5 | 2 | 1655,00 | 88,22 | 71,84 | 92,75 | – | 5,32 | 811 | 0,47 | 0,66 | 3048 | 2,95 | 0,30 | 6178 | 5,30 | 0,12 | 0,19 |
| A-n39-k5 | 2 | 1809,00 | 90,88 | 70,98 | 90,16 | – | 7,84 | 57 | 0,03 | 1,50 | 314 | 0,33 | 1,49 | 8904 | 8,35 | 0,12 | 0,23 |
| A-n39-k6 | 2 | 1929,00 | 86,16 | 69,41 | 95,54 | – | 2,59 | 915 | 0,56 | 0,93 | 8956 | 9,69 | 1,09 | 9176 | 9,81 | 0,12 | 0,16 |
| A-n45-k6 | 2 | 2101,00 | 89,86 | 68,11 | 91,19 | – | 5,57 | 1809 | 1,40 | 1,14 | 9233 | 12,17 | 3,57 | 6376 | 9,61 | 0,11 | 0,22 |
| A-n45-k7 | 2 | 2453,00 | 93,44 | 52,79 | 88,75 | – | 4,20 | 279 | 0,23 | 1,14 | 8443 | 11,92 | 1,22 | 1845 | 2,42 | 0,11 | 0,29 |
| A-n46-k7 | 2 | 2100,00 | 87,04 | 64,81 | 94,04 | 10,53 | 2,10 | 895 | 0,73 | 0,28 | 4279 | 6,55 | 1,14 | 7026 | 10,80 | 0,10 | 0,15 |
| A-n48-k7 | 2 | 2346,00 | 91,47 | 56,91 | 93,84 | – | 3,07 | 1196 | 1,06 | 0,60 | 428 | 0,75 | 0,51 | 4120 | 6,58 | 0,10 | 0,22 |
| A-n53-k7 | 2 | 2241,00 | 90,14 | 63,05 | 91,39 | – | 8,12 | 370 | 0,41 | 0,45 | 2500 | 4,85 | 1,29 | 4911 | 8,63 | 0,09 | 0,22 |
| A-n54-k7 | 2 | 2516,00 | 92,77 | 55,25 | 83,03 | 7,51 | 2,50 | 2118 | 2,15 | 0,79 | 4288 | 8,32 | 0,40 | 6121 | 11,54 | 0,09 | 0,16 |
| A-n55-k9 | 2 | 2366,00 | 90,70 | 57,40 | 90,61 | – | 2,66 | 4619 | 5,01 | 1,06 | 4595 | 9,56 | 1,23 | 8258 | 16,80 | 0,09 | 0,18 |
| A-n60-k9 | 2 | 2895,00 | 93,54 | 48,88 | 83,23 | 12,47 | 2,87 | 2129 | 2,82 | 0,69 | 8598 | 20,92 | 1,87 | 797 | 1,78 | 0,08 | 0,18 |
| A-n61-k9 | 2 | 2311,00 | 89,49 | 57,90 | 88,42 | – | 3,50 | 2435 | 3,49 | 2,86 | 2241 | 6,26 | 2,51 | 2006 | 5,16 | 0,08 | 0,17 |
| A-n62-k8 | 2 | 2791,00 | 92,44 | 56,00 | 87,62 | 18,27 | 2,65 | 1321 | 1,92 | 0,68 | 6895 | 18,28 | 0,86 | 2091 | 5,15 | 0,08 | 0,17 |
| A-n63-k9 | 2 | 3454,00 | 93,57 | 44,47 | 91,49 | – | 4,05 | 9583 | 13,65 | 1,77 | 9033 | 24,31 | 3,13 | 4094 | 11,17 | 0,07 | 0,36 |
| A-n63-k10 | 2 | 2844,00 | 92,48 | 53,23 | 89,82 | – | 4,47 | 2882 | 4,21 | 1,20 | 8059 | 21,78 | 0,56 | 3101 | 7,88 | 0,08 | 0,25 |
| A-n64-k9 | 2 | 3040,00 | 92,24 | 50,10 | 92,14 | 17,24 | 4,57 | 1213 | 1,81 | 0,86 | 255 | 0,78 | 0,76 | 5012 | 13,51 | 0,07 | 0,25 |
| A-n65-k9 | 2 | 2611,00 | 89,93 | 56,23 | 89,99 | 16,81 | 4,41 | 6736 | 10,20 | 2,45 | 4722 | 13,53 | 1,30 | 5850 | 15,94 | 0,07 | 0,12 |
| A-n69-k9 | 2 | 2602,00 | 89,09 | 63,84 | 91,56 | – | 4,65 | 5279 | 9,14 | 1,42 | 641 | 2,15 | 0,69 | 4709 | 15,07 | 0,07 | 0,12 |
| A-n80-k10 | 2 | 3829,00 | 92,09 | 48,16 | 92,70 | – | 3,89 | 5664 | 12,90 | 1,38 | 9465 | 40,89 | 1,28 | 3489 | 14,34 | 0,06 | 0,24 |
| B-n31-k5 | 2 | 1387,00 | 96,90 | 44,34 | 93,19 | 26,46 | 0,50 | 3767 | 1,42 | 0,43 | 3606 | 2,61 | 0,22 | 8307 | 5,77 | 0,16 | 0,54 |
| B-n34-k5 | 2 | 1746,00 | 90,26 | 48,22 | 93,53 | 9,05 | 1,49 | 4621 | 1,89 | 0,23 | 4617 | 3,49 | 1,66 | 399 | 0,27 | 0,14 | 0,37 |
| B-n35-k5 | 2 | 2020,00 | 94,56 | 43,56 | 80,94 | 4,26 | 0,50 | 210 | 0,09 | 0,10 | 708 | 0,64 | 0,30 | 271 | 0,17 | 0,13 | 0,18 |
| B-n38-k6 | 2 | 1688,00 | 95,38 | 45,38 | 82,11 | 11,85 | 1,07 | 668 | 0,41 | 0,12 | 2678 | 2,90 | 0,18 | 3874 | 3,99 | 0,13 | 0,26 |
| B-n39-k5 | 2 | 1190,00 | 92,27 | 59,33 | 80,80 | 14,63 | 0,16 | 6976 | 4,01 | 0,42 | 2331 | 2,51 | 0,17 | 8979 | 9,98 | 0,12 | 0,11 |
| B-n41-k6 | 2 | 1772,00 | 93,57 | 47,97 | 72,04 | – | 3,27 | 3973 | 2,28 | 1,13 | 9895 | 10,64 | 1,35 | 1817 | 1,87 | 0,12 | 0,27 |
| B-n43-k6 | 2 | 1609,00 | 92,23 | 49,41 | 82,91 | – | 1,43 | 3750 | 2,67 | 0,87 | 8277 | 10,64 | 0,31 | 4890 | 6,44 | 0,11 | 0,22 |
| B-n44-k7 | 2 | 1913,00 | 95,03 | 37,43 | 77,63 | – | 4,70 | 895 | 0,62 | 0,63 | 5632 | 7,30 | 1,41 | 1604 | 2,06 | 0,11 | 0,16 |
| B-n45-k5 | 2 | 1601,00 | 93,82 | 63,46 | 75,80 | 11,31 | 3,00 | 2089 | 1,64 | 0,62 | 8152 | 10,13 | 0,81 | 8413 | 10,69 | 0,10 | 0,09 |
| B-n45-k6 | 2 | 1531,00 | 88,57 | 52,19 | 80,21 | – | 5,62 | 24 | 0,05 | 2,68 | 2500 | 3,95 | 2,55 | 2535 | 4,70 | 0,10 | 0,44 |
| B-n50-k7 | 2 | 1572,00 | 94,27 | 54,32 | 82,44 | 10,05 | 1,27 | 703 | 0,70 | 0,19 | 4977 | 9,24 | 0,32 | 6240 | 11,70 | 0,09 | 0,12 |
| B-n50-k8 | 2 | 2730,00 | 96,12 | 39,30 | 76,37 | – | 1,36 | 2895 | 2,65 | 1,03 | 8179 | 14,13 | 0,51 | 5512 | 9,58 | 0,10 | 0,47 |

| | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| B-n51-k7 | 2 | 2140,00 | 95,42 | 47,01 | 82,71 | – | 1,73 | 1938 | 2,00 | 5,19 | 380 | 0,76 | 0,05 | 5081 | 9,41 | 0,09 | 0,19 |
| B-n52-k7 | 2 | 1602,00 | 93,26 | 51,50 | 72,00 | – | 3,55 | 432 | 0,50 | 0,50 | 922 | 1,90 | 0,19 | 2971 | 5,62 | 0,09 | 0,22 |
| B-n56-k7 | 2 | 1506,00 | 93,89 | 50,80 | 57,11 | 17,47 | 1,19 | 1519 | 1,89 | 0,53 | 3775 | 8,77 | 0,60 | 4465 | 10,03 | 0,08 | 0,18 |
| B-n57-k7 | 2 | 2382,00 | 95,72 | 40,47 | 76,95 | 9,74 | 0,34 | 9854 | 12,14 | 2,06 | 740 | 1,89 | 0,08 | 9536 | 22,09 | 0,08 | 0,16 |
| B-n57-k9 | 2 | 3314,00 | 96,44 | 30,20 | 86,43 | – | 2,11 | 1377 | 1,73 | 0,99 | 8808 | 20,41 | 0,30 | 9071 | 21,11 | 0,08 | 0,38 |
| B-n63-k10 | 2 | 3194,00 | 93,67 | 35,66 | 88,95 | – | 3,73 | 4648 | 6,90 | 2,19 | 5481 | 15,07 | 1,22 | 3959 | 10,61 | 0,08 | 0,29 |
| B-n64-k9 | 2 | 1868,00 | 92,19 | 48,39 | 70,77 | – | 2,79 | 663 | 1,15 | 1,55 | 6762 | 20,89 | 1,55 | 2827 | 7,85 | 0,07 | 0,15 |
| B-n66-k9 | 2 | 2833,00 | 92,91 | 41,58 | 83,53 | – | 1,20 | 3818 | 5,84 | 0,99 | 5339 | 15,16 | 0,78 | 3159 | 8,25 | 0,07 | 0,20 |
| B-n67-k10 | 2 | 2222,00 | 92,89 | 46,53 | 77,21 | – | 3,92 | 4819 | 8,58 | 0,81 | 8480 | 27,13 | 0,36 | 8456 | 26,61 | 0,07 | 0,21 |
| B-n68-k9 | 2 | 2663,00 | 95,76 | 33,31 | 71,18 | 11,57 | 0,98 | 3328 | 5,76 | 0,30 | 4446 | 14,38 | 0,30 | 2320 | 6,91 | 0,07 | 0,18 |
| B-n78-k10 | 2 | 2613,00 | 93,46 | 44,78 | 77,18 | – | 2,41 | 2646 | 5,71 | 1,80 | 7499 | 31,68 | 1,38 | 2188 | 8,13 | 0,06 | 0,14 |
| P-n19-k2 | 2 | 501,00 | 84,63 | 84,43 | 97,41 | 10,98 | 2,99 | 87 | 0,02 | 0,60 | 6286 | 1,17 | 0,20 | 3443 | 1,31 | 0,23 | 0,19 |
| P-n20-k2 | 2 | 511,00 | 84,54 | 86,11 | 98,83 | – | 1,37 | 5 | 0,00 | 0,39 | 4358 | 0,92 | 0,20 | 4358 | 1,31 | 0,22 | 0,27 |
| P-n21-k2 | 2 | 511,00 | 82,58 | 86,30 | 98,43 | 6,46 | 0,98 | 8798 | 1,25 | 0,59 | 389 | 0,09 | 0,59 | 9109 | 0,81 | 0,21 | 0,13 |
| P-n22-k2 | 2 | 512,00 | 84,38 | 86,52 | 98,63 | 9,38 | 5,08 | 7 | 0,00 | 0,78 | 1434 | 0,37 | 0,00 | 1569 | 3,03 | 0,20 | 0,15 |
| P-n40-k5 | 2 | 1072,00 | 85,45 | 78,92 | 98,78 | 10,07 | 4,86 | 1392 | 0,84 | 1,68 | 8412 | 9,63 | 0,84 | 7505 | 3,60 | 0,11 | 0,11 |
| P-n45-k5 | 2 | 1170,00 | 87,18 | 79,74 | 97,52 | 6,07 | 1,62 | 243 | 0,19 | 0,77 | 458 | 0,62 | 0,60 | 6581 | 9,80 | 0,10 | 0,09 |
| P-n50-k10 | 2 | 1549,00 | 89,86 | 59,20 | 96,71 | – | 7,42 | 3983 | 3,67 | 2,26 | 4387 | 6,94 | 0,90 | 6356 | 6,37 | 0,10 | 0,41 |
| P-n50-k7 | 2 | 1257,00 | 88,15 | 72,95 | 98,57 | 13,76 | 3,98 | 1612 | 1,53 | 0,00 | 5890 | 10,44 | 1,83 | 561 | 0,38 | 0,09 | 0,12 |
| P-n50-k8 | 2 | 1424,00 | 88,62 | 64,40 | 92,49 | – | 3,79 | 1206 | 1,17 | 0,98 | 9549 | 17,57 | 1,19 | 1186 | 12,45 | 0,10 | 0,27 |
| P-n51-k10 | 2 | 1678,00 | 88,32 | 58,52 | 94,93 | – | 6,62 | 1054 | 1,05 | 0,00 | 6887 | 11,22 | 2,62 | 6887 | 9,24 | 0,09 | 0,44 |
| P-n55-k10 | 2 | 1543,00 | 89,95 | 62,47 | 97,99 | – | 1,75 | 920 | 1,09 | 0,32 | 440 | 0,95 | 1,36 | 6608 | 0,06 | 0,09 | 0,33 |
| P-n55-k7 | 2 | 1286,00 | 88,34 | 74,96 | 98,41 | – | 3,19 | 1071 | 1,26 | 1,63 | 791 | 1,81 | 1,01 | 4890 | 11,22 | 0,08 | 0,15 |
| P-n55-k8 | 2 | 1294,00 | 89,03 | 74,50 | 97,80 | – | 5,56 | 5668 | 6,29 | 0,77 | 3840 | 7,97 | 1,70 | 3666 | 6,93 | 0,08 | 0,15 |
| P-n60-k10 | 2 | 1659,00 | 89,70 | 63,60 | 96,08 | – | 3,31 | 9836 | 12,22 | 1,08 | 7566 | 17,69 | 1,39 | 4681 | 16,85 | 0,08 | 0,25 |
| P-n65-k10 | 2 | 1761,00 | 89,95 | 66,16 | 96,99 | – | 3,92 | 3152 | 4,85 | 1,93 | 2426 | 6,97 | 0,68 | 3610 | 19,69 | 0,07 | 0,17 |
| P-n70-k10 | 2 | 1826,00 | 90,58 | 66,54 | 94,47 | – | 6,13 | 3232 | 6,27 | 3,17 | 4445 | 16,38 | 2,52 | 3976 | 29,53 | 0,07 | 0,22 |
| P-n76-k4 | 2 | 1385,00 | 85,64 | 90,69 | 97,18 | 13,14 | 3,10 | 3437 | 7,57 | 1,30 | 3201 | 12,50 | 0,29 | 3153 | 29,86 | 0,06 | 0,06 |
| P-n76-k5 | 2 | 1440,00 | 87,08 | 87,22 | 96,91 | – | 4,93 | 1113 | 2,23 | 1,81 | 5617 | 21,15 | 0,76 | 1954 | 3,78 | 0,06 | 0,09 |
| P-n101-k4 | 2 | 1581,00 | 86,15 | 91,90 | 98,26 | 7,02 | 2,53 | 3391 | 13,37 | 0,44 | 6381 | 50,97 | 0,63 | 8765 | 25,58 | 0,04 | 0,04 |
| VRPNC1 | 2 | 1198,38 | 87,55 | 82,47 | 97,29 | – | 6,77 | 9947 | 10,09 | 2,56 | 7195 | 11,93 | 2,50 | 1359 | 2,03 | 0,09 | 0,18 |
| VRPNC2 | 2 | 1846,10 | 90,49 | 68,56 | 94,53 | – | 3,71 | 9390 | 20,22 | 2,03 | 7928 | 31,62 | 0,00 | 6148 | 23,45 | 0,06 | 0,15 |
| VRPNC3 | 2 | 1871,83 | 88,27 | 78,55 | 96,72 | 10,06 | 2,04 | 9934 | 39,09 | 0,65 | 1394 | 10,25 | 0,95 | 1938 | 13,56 | 0,04 | 0,07 |
| VRPNC4 | 2 | 2381,38 | 86,37 | 72,93 | 94,58 | 14,93 | 3,90 | 9671 | 84,44 | 1,14 | 5460 | 88,98 | 0,97 | 7981 | 127,05 | 0,03 | 0,06 |
| VRPNC5 | 2 | 2947,26 | 87,63 | 64,56 | 88,72 | – | 3,82 | 9583 | 141,88 | 1,81 | 3959 | 109,48 | 0,80 | 9678 | 269,29 | 0,02 | 0,07 |
| VRPNC11 | 2 | 2318,86 | 89,88 | 53,47 | 53,79 | 11,02 | 1,01 | 9117 | 45,71 | 1,06 | 651 | 6,12 | 0,59 | 1715 | 13,56 | 0,04 | 0,11 |
| VRPNC12 | 2 | 1821,71 | 89,98 | 62,55 | 82,10 | – | 3,72 | 9442 | 38,80 | 1,09 | 2594 | 20,73 | 1,40 | 4088 | 30,14 | 0,04 | 0,06 |
| A-n32-k5 | 3 | 3368,00 | 69,83 | 63,33 | 99,15 | – | 4,24 | 1541 | 0,81 | 2,05 | 6398 | 6,41 | 0,36 | 4878 | 5,04 | 0,23 | 0,43 |
| A-n33-k5 | 3 | 2684,00 | 73,88 | 72,50 | 99,44 | – | 2,80 | 4244 | 2,29 | 0,26 | 385 | 0,42 | 0,56 | 2592 | 2,79 | 0,22 | 0,29 |
| A-n34-k5 | 3 | 3010,00 | 77,54 | 68,87 | 98,87 | – | 7,01 | 1761 | 1,03 | 0,36 | 7530 | 7,69 | 0,80 | 7753 | 8,08 | 0,20 | 0,37 |
| A-n36-k5 | 3 | 3258,00 | 73,57 | 64,89 | 99,09 | – | 4,54 | 470 | 0,33 | 1,38 | 1229 | 1,53 | 0,86 | 3236 | 3,90 | 0,20 | 0,38 |
| A-n37-k5 | 3 | 2774,00 | 72,35 | 76,50 | 99,17 | – | 2,49 | 1593 | 1,17 | 0,54 | 9565 | 13,59 | 0,61 | 2269 | 3,34 | 0,18 | 0,20 |
| A-n37-k6 | 3 | 3754,00 | 75,84 | 60,52 | 96,67 | – | 1,90 | 3760 | 2,28 | 2,24 | 287 | 0,34 | 0,93 | 4781 | 5,60 | 0,19 | 0,55 |
| A-n38-k5 | 3 | 2900,00 | 75,51 | 75,51 | 98,03 | – | 3,48 | 1391 | 0,92 | 1,14 | 6918 | 8,63 | 0,59 | 8827 | 10,78 | 0,19 | 0,36 |
| A-n39-k5 | 3 | 3133,00 | 78,71 | 72,39 | 96,95 | – | 4,12 | 1798 | 1,25 | 0,67 | 6237 | 8,11 | 0,86 | 5548 | 7,16 | 0,18 | 0,31 |
| A-n39-k6 | 3 | 3468,00 | 71,88 | 67,36 | 98,50 | – | 3,17 | 1718 | 1,34 | 0,34 | 4188 | 6,30 | 0,98 | 4488 | 6,93 | 0,18 | 0,30 |
| A-n45-k6 | 3 | 3856,00 | 73,44 | 66,96 | 91,93 | – | 5,42 | 1479 | 1,69 | 0,00 | 7561 | 15,99 | 0,54 | 8729 | 18,25 | 0,15 | 0,32 |
| A-n45-k7 | 3 | 4138,00 | 83,08 | 53,36 | 98,15 | – | 3,12 | 8585 | 8,57 | 0,41 | 8571 | 16,68 | 0,29 | 9707 | 19,17 | 0,16 | 0,42 |
| A-n46-k7 | 3 | 3703,00 | 74,05 | 64,70 | 99,53 | – | 3,68 | 372 | 0,44 | 1,14 | 7331 | 15,71 | 1,40 | 5367 | 11,86 | 0,15 | 0,36 |
| A-n48-k7 | 3 | 4158,00 | 77,41 | 56,52 | 98,64 | – | 2,50 | 3298 | 3,95 | 1,23 | 4531 | 10,50 | 1,35 | 7207 | 16,37 | 0,15 | 0,30 |
| A-n53-k7 | 3 | 3908,00 | 77,53 | 64,92 | 96,26 | – | 3,20 | 7434 | 10,09 | 1,02 | 2207 | 5,77 | 0,31 | 3555 | 8,72 | 0,13 | 0,23 |
| A-n54-k7 | 3 | 4168,00 | 84,00 | 59,84 | 94,61 | – | 3,07 | 7343 | 10,08 | 1,20 | 8926 | 23,34 | 0,58 | 7348 | 19,31 | 0,13 | 0,23 |

| | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A-n55-k9 | 3 | 4177,00 | 77,07 | 60,78 | 97,77 | – | 5,12 | 3084 | 4,91 | 2,25 | 495 | 1,47 | 0,00 | 9825 | 27,74 | 0,13 | 0,29 |
| A-n60-k9 | 3 | 4834,00 | 84,03 | 51,75 | 95,84 | – | 3,21 | 8954 | 16,72 | 1,03 | 3446 | 12,36 | 0,00 | 4422 | 15,04 | 0,12 | 0,28 |
| A-n61-k9 | 3 | 4035,00 | 76,88 | 60,50 | 93,92 | – | 6,05 | 4076 | 8,25 | 3,77 | 5472 | 21,70 | 3,35 | 8121 | 31,67 | 0,11 | 0,35 |
| A-n62-k8 | 3 | 4769,00 | 81,15 | 59,32 | 97,02 | – | 3,27 | 5585 | 11,28 | 0,19 | 5734 | 22,04 | 1,09 | 6304 | 22,95 | 0,11 | 0,21 |
| A-n63-k9 | 3 | 5888,00 | 82,34 | 46,11 | 96,22 | – | 9,12 | 1594 | 3,48 | 4,83 | 10008 | 39,42 | 0,07 | 6571 | 23,24 | 0,11 | 0,43 |
| A-n63-k10 | 3 | 4967,00 | 79,42 | 55,04 | 97,05 | – | 3,40 | 6395 | 12,76 | 1,05 | 1313 | 5,15 | 0,72 | 5168 | 19,25 | 0,11 | 0,32 |
| A-n64-k9 | 3 | 5271,00 | 79,80 | 52,23 | 97,70 | – | 3,05 | 2538 | 5,21 | 0,40 | 8849 | 34,12 | 0,74 | 5002 | 19,05 | 0,11 | 0,42 |
| A-n65-k9 | 3 | 4420,00 | 79,68 | 59,37 | 96,32 | – | 4,08 | 4787 | 9,50 | 3,28 | 9284 | 37,35 | 3,30 | 2720 | 9,78 | 0,11 | 0,23 |
| A-n69-k9 | 3 | 4457,00 | 78,01 | 65,60 | 96,48 | – | 3,27 | 4045 | 9,75 | 0,65 | 3390 | 15,87 | 0,31 | 1532 | 6,05 | 0,10 | 0,18 |
| A-n80-k10 | 3 | 6454,00 | 81,95 | 50,31 | 97,67 | – | 2,68 | 1854 | 6,22 | 1,45 | 1455 | 9,17 | 0,00 | 6385 | 38,99 | 0,09 | 0,34 |
| B-n31-k5 | 3 | 2607,00 | 77,33 | 54,04 | 99,81 | – | 0,42 | 2836 | 1,50 | 0,46 | 3771 | 3,56 | 0,23 | 8446 | 8,42 | 0,23 | 0,56 |
| B-n34-k5 | 3 | 2903,00 | 81,43 | 51,81 | 99,12 | – | 1,45 | 1929 | 1,06 | 0,24 | 400 | 0,44 | 0,41 | 801 | 0,75 | 0,21 | 0,37 |
| B-n35-k5 | 3 | 3222,00 | 89,06 | 46,78 | 92,35 | – | 0,71 | 1994 | 1,22 | 0,53 | 5280 | 6,10 | 0,62 | 806 | 1,00 | 0,20 | 0,13 |
| B-n39-k5 | 3 | 2076,00 | 79,84 | 63,55 | 93,46 | 13,28 | 0,24 | 2718 | 2,17 | 1,11 | 4270 | 6,55 | 1,75 | 127 | 0,09 | 0,18 | 0,09 |
| B-n41-k6 | 3 | 3015,00 | 82,49 | 55,92 | 93,44 | – | 3,02 | 2498 | 2,12 | 4,71 | 4687 | 7,61 | 0,00 | 8619 | 12,56 | 0,17 | 0,31 |
| B-n43-k6 | 3 | 2733,00 | 81,45 | 57,37 | 97,52 | – | 2,45 | 7352 | 7,08 | 1,65 | 9588 | 17,99 | 1,39 | 2894 | 5,12 | 0,16 | 0,27 |
| B-n44-k7 | 3 | 3217,00 | 84,77 | 39,48 | 90,27 | – | 4,10 | 2526 | 2,47 | 1,43 | 445 | 0,86 | 0,28 | 9366 | 17,41 | 0,16 | 0,24 |
| B-n45-k5 | 3 | 2666,00 | 84,92 | 66,57 | 88,75 | 22,47 | 6,82 | 7594 | 8,30 | 1,32 | 8922 | 19,09 | 2,07 | 6853 | 13,00 | 0,15 | 0,08 |
| B-n45-k6 | 3 | 2585,00 | 78,68 | 56,09 | 88,65 | – | 6,11 | 4006 | 3,93 | 3,48 | 6475 | 13,39 | 1,43 | 5204 | 11,82 | 0,16 | 0,50 |
| B-n50-k7 | 3 | 2616,00 | 84,98 | 58,98 | 94,61 | 10,06 | 1,03 | 5969 | 8,39 | 0,31 | 5088 | 13,17 | 1,72 | 8155 | 22,54 | 0,14 | 0,10 |
| B-n50-k8 | 3 | 4488,00 | 87,70 | 45,96 | 94,36 | – | 2,66 | 2702 | 3,46 | 0,78 | 2588 | 6,26 | 1,45 | 4048 | 9,67 | 0,14 | 0,49 |
| B-n51-k7 | 3 | 3542,00 | 86,47 | 51,64 | 93,45 | – | 4,60 | 1779 | 2,48 | 7,53 | 8021 | 20,69 | 1,24 | 4587 | 10,33 | 0,13 | 0,25 |
| B-n52-k7 | 3 | 2709,00 | 82,82 | 56,02 | 95,12 | – | 6,10 | 2530 | 3,95 | 4,43 | 1863 | 5,48 | 0,33 | 9434 | 26,79 | 0,14 | 0,21 |
| B-n56-k7 | 3 | 2427,00 | 87,46 | 53,90 | 82,64 | – | 2,68 | 4848 | 8,44 | 1,07 | 4569 | 15,37 | 0,37 | 2589 | 7,47 | 0,12 | 0,21 |
| B-n57-k7 | 3 | 4127,00 | 82,91 | 44,17 | 86,65 | – | 0,87 | 9012 | 15,82 | 3,59 | 1869 | 6,58 | 0,58 | 6412 | 20,94 | 0,12 | 0,22 |
| B-n57-k9 | 3 | 5490,00 | 87,32 | 35,92 | 98,01 | – | 2,05 | 866 | 1,59 | 1,24 | 3514 | 11,73 | 0,86 | 5577 | 18,49 | 0,12 | 0,54 |
| B-n63-k10 | 3 | 5553,00 | 80,82 | 37,01 | 97,03 | – | 3,68 | 4043 | 8,46 | 3,62 | 7528 | 29,24 | 0,43 | 819 | 2,50 | 0,11 | 0,34 |
| B-n64-k9 | 3 | 3286,00 | 78,61 | 53,20 | 87,69 | – | 0,70 | 9260 | 21,76 | 0,73 | 7765 | 33,70 | 2,10 | 6431 | 27,68 | 0,11 | 0,32 |
| B-n66-k9 | 3 | 4696,00 | 84,41 | 44,19 | 93,76 | – | 2,10 | 4931 | 10,65 | 1,82 | 2292 | 9,50 | 0,94 | 8322 | 32,17 | 0,10 | 0,33 |
| B-n67-k10 | 3 | 3693,00 | 83,83 | 52,75 | 94,61 | – | 2,60 | 6184 | 15,05 | 3,35 | 7782 | 36,08 | 4,47 | 5961 | 25,80 | 0,11 | 0,24 |
| B-n68-k9 | 3 | 4336,00 | 88,40 | 37,97 | 89,32 | – | 2,03 | 5539 | 13,20 | 0,42 | 5318 | 24,20 | 2,50 | 7083 | 31,56 | 0,10 | 0,22 |
| B-n78-k10 | 3 | 4234,00 | 86,72 | 48,39 | 88,15 | – | 4,45 | 5829 | 18,25 | 1,25 | 8922 | 53,68 | 0,83 | 4190 | 21,62 | 0,09 | 0,15 |
| P-n19-k2 | 3 | 882,00 | 72,11 | 86,40 | 99,32 | – | 2,27 | 4766 | 0,62 | 0,56 | 1189 | 0,27 | 1,02 | 3516 | 0,81 | 0,35 | 0,21 |
| P-n20-k2 | 3 | 897,00 | 72,24 | 86,85 | 99,55 | – | 2,01 | 5368 | 0,81 | 0,89 | 9676 | 2,53 | 0,78 | 8563 | 2,23 | 0,33 | 0,29 |
| P-n21-k2 | 3 | 898,00 | 70,49 | 88,42 | 100,00 | 5,01 | 2,56 | 5832 | 1,03 | 0,67 | 8599 | 2,70 | 0,45 | 222 | 0,06 | 0,31 | 0,15 |
| P-n22-k2 | 3 | 906,00 | 71,52 | 87,53 | 99,45 | 6,95 | 1,99 | 183 | 0,03 | 0,55 | 6247 | 2,11 | 0,11 | 4446 | 1,50 | 0,30 | 0,16 |
| P-n40-k5 | 3 | 1869,00 | 73,52 | 80,31 | 99,14 | – | 3,00 | 1786 | 1,47 | 1,44 | 4503 | 6,88 | 0,37 | 1955 | 3,09 | 0,17 | 0,19 |
| P-n45-k5 | 3 | 2063,00 | 74,17 | 80,56 | 98,72 | – | 2,81 | 348 | 0,36 | 0,34 | 380 | 0,72 | 0,63 | 219 | 0,33 | 0,15 | 0,13 |
| P-n50-k7 | 3 | 2203,00 | 75,45 | 72,63 | 98,82 | – | 3,45 | 3710 | 4,81 | 0,95 | 4907 | 11,90 | 0,68 | 599 | 1,15 | 0,14 | 0,25 |
| P-n55-k7 | 3 | 2244,00 | 75,94 | 74,65 | 99,15 | – | 3,88 | 3891 | 6,63 | 1,34 | 1349 | 4,27 | 0,94 | 8645 | 27,74 | 0,12 | 0,21 |
| P-n55-k8 | 3 | 2259,00 | 76,50 | 74,15 | 98,49 | – | 4,42 | 6307 | 10,06 | 1,02 | 9522 | 27,10 | 0,93 | 4632 | 12,92 | 0,12 | 0,22 |
| P-n65-k10 | 3 | 3029,00 | 78,44 | 65,89 | 98,25 | – | 3,13 | 3239 | 6,97 | 1,68 | 1186 | 4,77 | 0,86 | 4472 | 18,21 | 0,11 | 0,38 |
| P-n70-k10 | 3 | 3159,00 | 78,54 | 65,72 | 95,79 | – | 5,35 | 1743 | 4,74 | 1,67 | 4677 | 22,85 | 1,46 | 6422 | 29,87 | 0,10 | 0,41 |
| P-n76-k4 | 3 | 2374,00 | 74,94 | 90,19 | 97,60 | – | 3,29 | 5867 | 20,05 | 0,59 | 6511 | 34,15 | 0,72 | 1708 | 7,68 | 0,08 | 0,09 |
| P-n76-k5 | 3 | 2464,00 | 76,34 | 86,89 | 97,93 | – | 4,22 | 6732 | 18,41 | 1,25 | 2362 | 12,28 | 1,26 | 1022 | 4,26 | 0,08 | 0,09 |
| P-n101-k4 | 3 | 2702,00 | 75,61 | 91,34 | 98,35 | 7,40 | 2,81 | 870 | 4,91 | 0,59 | 6241 | 66,69 | 0,41 | 7999 | 87,24 | 0,06 | 0,05 |
| VRPNC1 | 3 | 2127,77 | 73,97 | 82,00 | 98,20 | – | 4,45 | 9139 | 13,78 | 1,18 | 4155 | 9,59 | 1,34 | 8682 | 20,37 | 0,13 | 0,18 |
| VRPNC2 | 3 | 3184,81 | 78,68 | 67,62 | 95,97 | – | 3,55 | 9998 | 30,90 | 0,80 | 9832 | 54,20 | 0,00 | 4795 | 25,82 | 0,09 | 0,32 |
| VRPNC3 | 3 | 3174,74 | 78,06 | 78,71 | 97,92 | 9,33 | 0,57 | 9796 | 55,05 | 0,40 | 4678 | 50,23 | 0,57 | 1140 | 10,64 | 0,07 | 0,08 |
| VRPNC4 | 3 | 4023,86 | 76,67 | 73,98 | 96,88 | 14,98 | 3,11 | 9691 | 121,82 | 1,21 | 5308 | 126,52 | 1,54 | 1513 | 25,35 | 0,04 | 0,07 |
| VRPNC5 | 3 | 4913,56 | 78,84 | 67,39 | 93,21 | – | 3,84 | 8006 | 177,83 | 0,96 | 3415 | 144,93 | 0,88 | 3264 | 103,73 | 0,03 | 0,11 |
| VRPNC11 | 3 | 3706,39 | 84,35 | 57,54 | 61,28 | 21,31 | 3,66 | 8422 | 60,58 | 0,77 | 5356 | 72,04 | 0,55 | 8848 | 113,10 | 0,05 | 0,12 |

| | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| VRPNC12 | 3 | 3000,82 | 81,93 | 65,35 | 88,45 | – | 5,22 | 9680 | 58,55 | 2,12 | 2508 | 29,63 | 1,98 | 3343 | 33,06 | 0,07 | 0,08 |
| P-n19-k2 | 4 | 1330,00 | 63,76 | 86,77 | 99,17 | – | 1,36 | 3477 | 0,48 | 0,46 | 1430 | 0,34 | 0,00 | 2566 | 0,61 | 0,47 | 0,37 |
| P-n20-k2 | 4 | 1360,00 | 63,53 | 87,35 | 99,42 | – | 1,69 | 6219 | 1,00 | 0,15 | 445 | 0,13 | 0,44 | 790 | 0,22 | 0,44 | 0,36 |
| P-n21-k2 | 4 | 1366,00 | 61,79 | 88,29 | 99,71 | – | 1,76 | 174 | 0,03 | 0,00 | 6189 | 2,14 | 0,37 | 986 | 0,33 | 0,42 | 0,27 |
| P-n22-k2 | 4 | 1365,00 | 63,30 | 88,28 | 99,34 | 6,15 | 1,54 | 4693 | 0,97 | 0,00 | 1109 | 0,42 | 0,51 | 8595 | 3,32 | 0,40 | 0,16 |
| P-n45-k5 | 4 | 3079,00 | 66,26 | 81,55 | 98,83 | – | 2,14 | 579 | 0,73 | 0,45 | 5095 | 11,81 | 0,62 | 5265 | 12,28 | 0,20 | 0,24 |
| P-n76-k4 | 4 | 3545,00 | 66,91 | 89,99 | 97,95 | – | 3,81 | 5169 | 22,87 | 0,45 | 4444 | 32,26 | 0,65 | 1267 | 4,76 | 0,11 | 0,10 |
| P-n76-k5 | 4 | 3704,00 | 67,71 | 86,12 | 97,60 | – | 2,43 | 7976 | 32,75 | 0,65 | 2856 | 19,63 | 0,51 | 2997 | 14,12 | 0,11 | 0,10 |
| P-n101-k4 | 4 | 4038,00 | 67,46 | 91,58 | 98,66 | 6,54 | 1,46 | 1251 | 9,20 | 0,45 | 6315 | 87,53 | 0,52 | 643 | 2,36 | 0,08 | 0,05 |
| VRPNC1 | 4 | 3215,19 | 65,27 | 81,87 | 97,45 | – | 5,23 | 9732 | 18,27 | 0,62 | 2321 | 6,79 | 0,30 | 9274 | 27,19 | 0,17 | 0,25 |
| VRPNC3 | 4 | 4699,58 | 70,32 | 79,44 | 98,68 | 8,96 | 1,45 | 3878 | 28,80 | 0,75 | 1210 | 16,85 | 0,50 | 4638 | 54,18 | 0,09 | 0,09 |
| VRPNC4 | 4 | 5972,71 | 68,87 | 73,85 | 97,50 | – | 2,77 | 9693 | 158,95 | 0,77 | 2470 | 76,13 | 0,15 | 2659 | 49,02 | 0,06 | 0,09 |
| VRPNC5 | 4 | 7263,92 | 71,11 | 68,07 | 94,56 | – | 2,86 | 9896 | 293,19 | 0,13 | 6877 | 377,96 | 0,58 | 1861 | 34,85 | 0,04 | 0,13 |
| VRPNC11 | 4 | 5275,63 | 79,01 | 60,09 | 71,96 | – | 3,07 | 9787 | 90,87 | 0,81 | 5711 | 98,89 | 0,61 | 7216 | 91,25 | 0,07 | 0,12 |
| VRPNC12 | 4 | 4466,61 | 73,39 | 65,74 | 92,36 | – | 5,73 | 9508 | 74,90 | 1,06 | 6110 | 91,15 | 1,01 | 6641 | 73,02 | 0,09 | 0,15 |
| P-n21-k2 | 5 | 1899,00 | 55,56 | 88,99 | 99,68 | – | 1,00 | 59 | 0,02 | 0,42 | 7625 | 2,43 | 0,42 | 1050 | 0,36 | 0,52 | 0,27 |
| P-n22-k2 | 5 | 1897,00 | 56,93 | 89,14 | 99,58 | 6,38 | 1,58 | 78 | 0,03 | 0,47 | 3551 | 1,33 | 0,11 | 1332 | 0,50 | 0,50 | 0,19 |
| P-n76-k4 | 5 | 4880,00 | 60,75 | 90,04 | 98,14 | – | 3,09 | 6053 | 31,29 | 0,68 | 4941 | 44,62 | 0,59 | 2613 | 20,14 | 0,14 | 0,09 |
| P-n76-k5 | 5 | 5082,00 | 55,95 | 90,25 | 98,27 | – | 3,10 | 8697 | 40,26 | 0,12 | 6126 | 50,12 | 0,33 | 7246 | 58,97 | 0,14 | 0,13 |
| P-n101-k4 | 5 | 5525,00 | 51,92 | | 98,95 | – | 0,05 | 8451 | 75,13 | 0,05 | 6608 | 111,51 | 0,51 | 1119 | 12,67 | 0,10 | 0,06 |
| VRPNC1 | 5 | 4424,74 | 59,28 | 82,22 | 96,99 | – | – | – | – | – | – | – | – | * | * | 0,22 | 0,33 |
| VRPNC3 | 5 | 6431,66 | 64,22 | 79,54 | 98,75 | 9,10 | 1,44 | 9838 | 87,45 | 0,35 | 5375 | 91,09 | 0,13 | 2447 | 33,13 | 0,11 | 0,10 |
| VRPNC4 | 5 | 8141,82 | 63,16 | 74,08 | 98,06 | – | 2,49 | 9960 | 198,60 | 0,73 | 5376 | 211,94 | 0,76 | 3296 | 102,38 | 0,07 | 0,10 |
| VRPNC5 | 5 | 9949,03 | 63,16 | 74,08 | 98,06 | – | 2,49 | 9633 | 341,35 | 0,73 | 5377 | 342,52 | 0,39 | 5967 | 350,50 | 0,05 | 0,17 |
| VRPNC11 | 5 | 7075,20 | 73,65 | 61,19 | 79,05 | – | 3,60 | 9320 | 112,54 | 1,28 | 5378 | 183,94 | 1,51 | 8639 | 175,74 | 0,09 | 0,11 |
| VRPNC12 | 5 | 6699,49 | 61,17 | 70,26 | 96,63 | – | 3,55 | 9653 | 89,47 | 1,20 | 5379 | 81,04 | 0,15 | 7989 | 133,91 | 0,11 | 0,24 |
| P-n76-k4 | 6 | 6367,00 | 61,62 | 86,36 | 97,88 | – | 2,59 | 8645 | 49,36 | 0,57 | 4888 | 46,18 | 0,08 | 6212 | 59,69 | 0,17 | 0,09 |
| P-n76-k5 | 6 | 6593,00 | 57,06 | 87,05 | 98,35 | – | 1,91 | 7869 | 40,95 | 0,73 | 7167 | 69,86 | 0,77 | 6352 | 57,69 | 0,17 | 0,15 |
| P-n101-k4 | 6 | 7168,00 | 53,00 | | 99,14 | – | 0,14 | 7602 | 79,25 | 0,14 | 5658 | 112,07 | 0,40 | 3364 | 59,72 | 0,12 | 0,07 |
| VRPNC3 | 6 | 8321,82 | 59,56 | 79,85 | 98,85 | – | 2,90 | 9192 | 95,27 | 0,62 | 5380 | 28,69 | 0,27 | 9645 | 187,12 | 0,13 | 0,14 |
| VRPNC4 | 6 | 10576,20 | 58,34 | 74,15 | 98,19 | – | 2,42 | 9962 | 233,44 | 0,29 | 5381 | 447,69 | 0,21 | 7541 | 314,19 | 0,09 | 0,14 |
| VRPNC11 | 6 | 9238,82 | 67,68 | 60,57 | 83,96 | – | 4,65 | 9542 | 126,94 | 1,80 | 5382 | 238,40 | 0,60 | 7710 | 154,22 | 0,11 | 0,10 |
| P-n76-k4 | 7 | 7993,00 | 61,65 | 91,91 | 98,96 | 6,79 | 1,18 | 8500 | 56,89 | 0,48 | 5746 | 66,44 | 0,68 | 6803 | 74,38 | 0,20 | 0,11 |
| P-n76-k5 | 7 | 8292,00 | 56,92 | | 99,00 | – | 1,91 | 8002 | 48,38 | 0,18 | 2624 | 28,47 | 0,41 | 4868 | 49,05 | 0,20 | 0,19 |
| P-n101-k4 | 7 | 8965,00 | 53,17 | | 99,10 | – | 0,00 | 9528 | 111,77 | 0,00 | 7205 | 169,23 | 0,09 | 1435 | 19,67 | 0,14 | 0,08 |
| VRPNC11 | 7 | 11668,00 | 62,52 | 60,28 | 86,58 | – | 0,00 | 8344 | 132,52 | 3,11 | 5383 | 253,25 | – | – | – | 0,12 | 0,08 |

Table 3: $m$*VRP, LB1, LB2, CW, ILS, TS et GTSD pour le $m$-PTVP



## A-2.2: Résultats des approches polyédrales et de type génération de colonnes pour le $m$-PTVP

Légende: $m$ = nombre de périodes; UB* = optimum (prouvé); UB = meilleure borne supérieure trouvée par BCPA; H1 = heuristique duale basée sur les $q$-tournées; H2 = heuristique duale basée sur la génération de tournées élémentaires; $T$ = durée en secondes; Nbqt = nombre de $q$-tournées générées; Nbt = nombre de tournées élémentaires générées; BCRPA = algorithme de branchement et coupes pour la formulation agrégée basée sur

les arêtes; BCPA = algorithme de branchement et coupes pour la formulation basée sur les arêtes; CAP = contraintes de capacité; PEI = contraintes de peigne; MET = contraintes de multi-étoiles.

### A-2.2.1: Génération de colonnes et Branchement et Coupes pour le VRP = 1-PTVP

| | $m$ | UB* | H1 | | | H1+H2 | | | BCRPA | | | | | BCPA | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | $\frac{LB}{UB^*}$ % | $T$ | Nbqt | $\frac{LB}{UB^*}$ % | $T$ | Nbt | $\frac{LB}{UB^*}$ % | $T$ | CAP | PEI | MET | $\frac{LB}{UB^*}$ % | $T$ | CAP | PEI | MET |
| A-n32-k5 | 1 | 784,00 | 96,54 | 0,81 | 477 | 98,33 | 1,36 | 3804 | 100,00 | 1,61 | 476 | 1 | 22 | 100,00 | 1,59 | 476 | 1 | 22 |
| A-n33-k5 | 1 | 661,00 | 97,50 | 0,48 | 405 | 99,40 | 0,83 | 2118 | 100,00 | 1,77 | 378 | 5 | 16 | 100,00 | 1,80 | 378 | 5 | 16 |
| A-n33-k6 | 1 | 742,00 | 94,77 | 0,48 | 389 | 98,76 | 1,31 | 5033 | 100,00 | 8,11 | 487 | 4 | 225 | 100,00 | 8,16 | 487 | 4 | 225 |
| A-n34-k5 | 1 | 778,00 | 93,74 | 0,56 | 381 | 95,40 | 1,45 | 5034 | 100,00 | 7,49 | 494 | 31 | 74 | 100,00 | 7,55 | 494 | 31 | 74 |
| A-n36-k5 | 1 | 799,00 | 96,62 | 0,75 | 491 | 98,18 | 1,62 | 5035 | 100,00 | 19,19 | 732 | 3 | 292 | 100,00 | 19,45 | 732 | 3 | 292 |
| A-n37-k5 | 1 | 669,00 | 96,50 | 12,90 | 1596 | 98,53 | 14,59 | 5036 | 100,00 | 3,14 | 512 | 6 | 42 | 100,00 | 3,19 | 512 | 6 | 42 |
| A-n37-k6 | 1 | 949,00 | 95,45 | 1,19 | 397 | 97,53 | 2,86 | 5036 | 100,00 | 330,25 | 1592 | 167 | 2220 | 100,00 | 332,92 | 1592 | 167 | 2220 |
| A-n38-k5 | 1 | 730,00 | 91,67 | 1,08 | 317 | 95,73 | 3,32 | 5038 | 100,00 | 23,08 | 819 | 43 | 180 | 100,00 | 23,36 | 819 | 43 | 180 |
| A-n39-k5 | 1 | 822,00 | 96,52 | 2,08 | 599 | 97,35 | 4,37 | 5038 | 100,00 | 37,58 | 1577 | 41 | 655 | 100,00 | 38,17 | 1577 | 41 | 655 |
| A-n39-k6 | 1 | 831,00 | 95,28 | 13,90 | 7007 | 97,60 | 15,54 | 5038 | 100,00 | 45,27 | 792 | 81 | 488 | 100,00 | 45,70 | 792 | 81 | 488 |
| A-n44-k7 | 1 | 937,00 | 97,56 | 2,81 | 625 | 98,86 | 4,54 | 5043 | 100,00 | 315,19 | 2159 | 147 | 2948 | 100,00 | 318,44 | 2159 | 147 | 2948 |
| A-n45-k6 | 1 | 944,00 | 94,41 | 2,03 | 427 | 97,80 | 4,29 | 5044 | 100,00 | 182,38 | 1846 | 92 | 1453 | 100,00 | 185,53 | 1846 | 92 | 1453 |
| A-n45-k7 | 1 | 1146,00 | 97,39 | 3,18 | 691 | 98,17 | 5,07 | 5044 | 99,33 | 10800,63 | 4648 | 333 | 20662 | 99,36 | 10800,42 | 5310 | 314 | 16924 |
| A-n46-k7 | 1 | 914,00 | 98,85 | 3,79 | 756 | 99,63 | 5,29 | 1728 | 99,95 | 1,66 | 1660 | 1 | 0 | 99,95 | 2,02 | 1422 | 1 | 0 |
| A-n48-k7 | 1 | 1073,00 | 96,78 | 2,51 | 658 | 97,90 | 5,59 | 5047 | 100,00 | 247,16 | 2397 | 47 | 1299 | 100,00 | 292,89 | 2184 | 64 | 1408 |
| A-n53-k7 | 1 | 1010,00 | 96,92 | 12,42 | 1281 | 98,40 | 17,68 | 5057 | 100,00 | 271,01 | 2759 | 83 | 773 | 100,00 | 265,77 | 2613 | 63 | 788 |
| A-n54-k7 | 1 | 1167,00 | 95,44 | 4,02 | 804 | 97,64 | 22,65 | 5053 | 100,00 | 7235,63 | 4740 | 411 | 8566 | 99,46 | 10801,64 | 5779 | 571 | 9516 |
| A-n55-k9 | 1 | 1073,00 | 95,66 | 3,03 | 686 | 98,63 | 6,69 | 5065 | 100,00 | 489,16 | 3356 | 71 | 1275 | 100,00 | 500,86 | 3356 | 71 | 1275 |
| A-n60-k9 | 1 | 1354,00 | 96,59 | 4,35 | 990 | 98,06 | 221,23 | 5059 | 98,60 | 10800,66 | 7269 | 490 | 7404 | 98,59 | 10801,95 | 7265 | 487 | 7339 |
| A-n61-k9 | 1 | 1034,00 | 96,18 | 4,31 | 739 | 97,55 | 8,74 | 5061 | 98,93 | 10804,69 | 8145 | 799 | 11725 | 98,92 | 10800,39 | 8130 | 791 | 11667 |
| A-n62-k8 | 1 | 1290,00 | 94,35 | 8,36 | 1370 | 97,22 | 179,62 | 5264 | 98,78 | 10802,05 | 7437 | 273 | 11753 | 98,79 | 10802,03 | 7457 | 276 | 11877 |
| A-n63-k9 | 1 | 1616,00 | 96,81 | 2,78 | 977 | 98,30 | 61,65 | 5063 | 98,50 | 10803,53 | 6766 | 538 | 12348 | 98,50 | 10804,19 | 6766 | 538 | 12343 |
| A-n63-k10 | 1 | 1315,00 | 96,39 | 5,85 | 1095 | 97,74 | 14,04 | 5062 | 97,16 | 10808,31 | 6927 | 229 | 10276 | 97,17 | 10817,78 | 6930 | 229 | 10312 |
| A-n64-k9 | 1 | 1402,00 | 96,87 | 21,14 | 1566 | 98,40 | 61,67 | 5063 | 96,95 | 10802,14 | 7968 | 428 | 12170 | 96,94 | 10804,30 | 7950 | 426 | 12044 |
| A-n65-k9 | 1 | 1174,00 | 96,46 | 31,47 | 13483 | 97,92 | 35,59 | 5064 | 100,00 | 2688,06 | 4635 | 505 | 3143 | 100,00 | 2704,19 | 4635 | 505 | 3143 |
| A-n69-k9 | 1 | 1159,00 | 96,35 | 7,72 | 1193 | 97,98 | 20,50 | 5070 | 97,35 | 10804,30 | 8763 | 706 | 6444 | 97,36 | 10806,61 | 8820 | 718 | 6602 |
| A-n80-k10 | 1 | 1763,00 | 96,54 | 37,35 | 2405 | 97,97 | 138,64 | 5082 | 96,88 | 11047,86 | 10177 | 102 | 7767 | 96,89 | 10869,98 | 10424 | 151 | 8697 |
| B-n31-k5 | 1 | 672,00 | 87,06 | 0,53 | 360 | 98,48 | 12,62 | 5135 | 100,00 | 2,59 | 209 | 0 | 113 | 100,00 | 2,55 | 209 | 0 | 113 |
| B-n34-k5 | 1 | 788,00 | 89,05 | 4,26 | 1851 | 96,01 | 8,16 | 5037 | 100,00 | 33,36 | 400 | 29 | 283 | 100,00 | 33,56 | 400 | 29 | 283 |
| B-n35-k5 | 1 | 955,00 | 88,98 | 0,87 | 554 | 90,80 | 1,79 | 5034 | 100,00 | 0,97 | 58 | 0 | 0 | 100,00 | 0,97 | 58 | 0 | 0 |
| B-n38-k6 | 1 | 805,00 | 87,45 | 12,06 | 1557 | 92,56 | 15,21 | 5052 | 100,00 | 3,38 | 534 | 0 | 23 | 100,00 | 3,42 | 534 | 0 | 23 |
| B-n39-k5 | 1 | 549,00 | 89,68 | 14,77 | 7047 | 96,60 | 20,48 | 5038 | 100,00 | 1,83 | 144 | 0 | 12 | 100,00 | 1,81 | 144 | 0 | 12 |
| B-n41-k6 | 1 | 829,00 | 88,61 | 1,47 | 428 | 96,16 | 24,29 | 5156 | 100,00 | 8,83 | 943 | 2 | 121 | 100,00 | 8,83 | 943 | 2 | 121 |
| B-n43-k6 | 1 | 742,00 | 89,73 | 17,27 | 8750 | 95,10 | 88,11 | 5266 | 100,00 | 44,23 | 1345 | 15 | 531 | 100,00 | 44,88 | 1345 | 15 | 531 |
| B-n44-k7 | 1 | 909,00 | 91,77 | 3,09 | 697 | 91,77 | 100,64 | 5043 | 100,00 | 1,55 | 886 | 0 | 0 | 100,00 | 2,03 | 936 | 0 | 0 |
| B-n45-k5 | 1 | 751,00 | 86,18 | 20,19 | 7978 | 86,18 | 99,12 | 5044 | 100,00 | 18,03 | 929 | 13 | 74 | 100,00 | 18,31 | 929 | 13 | 74 |
| B-n45-k6 | 1 | 678,00 | 87,28 | 17,96 | 10159 | 96,52 | 67,75 | 5177 | 100,00 | 60,89 | 1195 | 8 | 763 | 100,00 | 61,81 | 1195 | 8 | 763 |
| B-n50-k7 | 1 | 741,00 | 89,61 | 21,67 | 10047 | 93,26 | 25,10 | 5051 | 100,00 | 2,20 | 791 | 0 | 6 | 100,00 | 2,00 | 383 | 0 | 59 |
| B-n50-k8 | 1 | 1312,00 | 92,82 | 5,59 | 1301 | 92,82 | 152,01 | 5049 | 100,00 | 8102,84 | 6879 | 231 | 16406 | 100,00 | 8218,13 | 6879 | 231 | 16406 |
| B-n51-k7 | 1 | 1020,00 | 89,87 | 3,10 | 658 | 92,97 | 8,25 | 5151 | 99,80 | 1,38 | 349 | 0 | 0 | 99,80 | 1,00 | 299 | 0 | 0 |
| B-n52-k7 | 1 | 747,00 | 86,54 | 3,65 | 809 | 86,54 | 83,41 | 5051 | 100,00 | 4,01 | 650 | 1 | 0 | 100,00 | 2,06 | 204 | 0 | 8 |
| B-n56-k7 | 1 | 707,00 | 85,89 | 5,12 | 1067 | 85,89 | 242,35 | 5055 | 100,00 | 2,11 | 730 | 0 | 7 | 100,00 | 1,56 | 622 | 0 | 4 |

| | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| B-n57-k7 | 1 | 1140,00 | 92,74 | 6,05 | 1402 | 95,57 | 346,06 | 5214 | 100,00 | 1,63 | 527 | 0 | 0 | 100,00 | 2,05 | 903 | 0 | 0 |
| B-n57-k9 | 1 | 1598,00 | 94,09 | 31,78 | 9341 | 94,09 | 120,09 | 5056 | 100,00 | 778,36 | 4456 | 12 | 3809 | 100,00 | 800,53 | 4456 | 12 | 3809 |
| B-n63-k10 | 1 | 1496,00 | 94,84 | 4,76 | 1051 | 97,47 | 126,85 | 5162 | 100,00 | 290,81 | 3074 | 30 | 1742 | 100,00 | 296,63 | 3074 | 30 | 1742 |
| B-n64-k9 | 1 | 861,00 | 89,34 | 5,18 | 922 | 93,35 | 109,93 | 5167 | 100,00 | 29,28 | 1498 | 18 | 132 | 100,00 | 29,78 | 1498 | 18 | 132 |
| B-n66-k9 | 1 | 1316,00 | 92,99 | 5,84 | 1172 | 92,99 | 155,81 | 5065 | 99,91 | 10800,31 | 7348 | 487 | 10286 | 99,90 | 10800,86 | 7348 | 487 | 10286 |
| B-n67-k10 | 1 | 1032,00 | 95,63 | 4,27 | 1017 | 96,85 | 27,99 | 5068 | 100,00 | 625,80 | 5724 | 43 | 1307 | 100,00 | 642,09 | 5724 | 43 | 1307 |
| B-n68-k9 | 1 | 1275,00 | 91,37 | 37,53 | 14489 | 91,37 | 752,77 | 5067 | 99,01 | 10801,50 | 10391 | 262 | 9910 | 99,00 | 10801,88 | 10236 | 254 | 9723 |
| B-n78-k10 | 1 | 1221,00 | 92,00 | 6,82 | 1533 | 92,00 | 246,78 | 5077 | 99,48 | 10800,64 | 8923 | 269 | 8388 | 99,48 | 10800,88 | 8887 | 258 | 8198 |
| P-n19-k2 | 1 | 212,00 | 95,14 | 0,52 | 184 | 96,32 | 0,89 | 259 | 100,00 | 1,03 | 54 | 10 | 0 | 100,00 | 1,02 | 54 | 10 | 0 |
| P-n20-k2 | 1 | 216,00 | 96,88 | 0,52 | 180 | 98,14 | 0,81 | 246 | 100,00 | 1,34 | 71 | 9 | 8 | 100,00 | 1,34 | 71 | 9 | 8 |
| P-n21-k2 | 1 | 211,00 | 98,25 | 0,55 | 176 | 99,78 | 0,80 | 96 | 100,00 | 0,52 | 75 | 0 | 0 | 100,00 | 0,52 | 75 | 0 | 0 |
| P-n22-k2 | 1 | 216,00 | 98,16 | 0,66 | 182 | 99,61 | 0,95 | 234 | 100,00 | 0,91 | 69 | 5 | 7 | 100,00 | 0,91 | 69 | 5 | 7 |
| P-n22-k8 | 1 | 603,00 | 98,31 | 127,83 | 1156 | 98,68 | 132,24 | 71 | 100,00 | 1,22 | 229 | 0 | 2 | 100,00 | 1,20 | 229 | 0 | 2 |
| P-n23-k8 | 1 | 529,00 | 98,18 | 0,12 | 206 | 98,34 | 0,28 | 126 | 100,00 | 19,33 | 486 | 0 | 822 | 100,00 | 19,50 | 486 | 0 | 822 |
| P-n40-k5 | 1 | 458,00 | 98,38 | 1,95 | 457 | 99,17 | 3,03 | 2380 | 100,00 | 7,89 | 364 | 63 | 114 | 100,00 | 7,66 | 364 | 63 | 114 |
| P-n45-k5 | 1 | 510,00 | 97,65 | 28,42 | 9397 | 98,70 | 30,23 | 5044 | 100,00 | 46,63 | 1141 | 81 | 186 | 100,00 | 47,33 | 1141 | 81 | 186 |
| P-n50-k10 | 1 | 696,00 | 98,24 | 2,23 | 515 | 98,73 | 3,84 | 4659 | 98,33 | 10805,83 | 4354 | 352 | 23213 | 98,31 | 10801,67 | 4316 | 348 | 22976 |
| P-n50-k7 | 1 | 554,00 | 98,61 | 3,96 | 611 | 98,96 | 5,79 | 4983 | 100,00 | 371,91 | 2335 | 200 | 2841 | 100,00 | 361,55 | 2016 | 201 | 2871 |
| P-n50-k8 | 1 | 631,00 | 96,37 | 1,55 | 518 | 96,89 | 3,28 | 5049 | 97,41 | 10832,91 | 5214 | 829 | 23325 | 97,41 | 10802,86 | 5167 | 828 | 23193 |
| P-n51-k10 | 1 | 741,00 | 98,23 | 1,62 | 561 | 98,75 | 3,40 | 5050 | 98,32 | 10800,61 | 5170 | 410 | 17036 | 98,30 | 10805,94 | 5154 | 407 | 16931 |
| P-n55-k10 | 1 | 694,00 | 98,20 | 5,98 | 887 | 98,51 | 7,78 | 5054 | 97,56 | 10801,22 | 4480 | 349 | 25802 | 97,56 | 10804,13 | 4480 | 349 | 25798 |
| P-n55-k15 | 1 | 945,00 | 98,00 | 1,25 | 550 | 98,47 | 2,11 | 5054 | 97,05 | 10802,47 | 4083 | 33 | 30173 | 97,11 | 10803,92 | 5071 | 25 | 31379 |
| P-n55-k7 | 1 | 568,00 | 98,16 | 41,59 | 12892 | 98,77 | 43,76 | 5054 | 100,00 | 3821,31 | 3411 | 746 | 10825 | 100,00 | 5308,78 | 3200 | 898 | 11192 |
| P-n55-k8 | 1 | 576,00 | 98,23 | 9,55 | 973 | 98,72 | 11,65 | 5054 | 100,00 | 1373,70 | 3213 | 429 | 5159 | 100,00 | 1172,63 | 2596 | 432 | 4753 |
| P-n60-k10 | 1 | 744,00 | 98,64 | 4,74 | 814 | 99,08 | 6,86 | 5059 | 97,72 | 10809,61 | 4970 | 411 | 18457 | 97,72 | 10812,38 | 4967 | 410 | 18391 |
| P-n60-k15 | 1 | 968,00 | 98,81 | 6,79 | 7901 | 99,14 | 7,61 | 3110 | 94,39 | 101,45 | 3587 | 3 | 22572 | 94,38 | 129,80 | 3220 | 2 | 23915 |
| P-n65-k10 | 1 | 792,00 | 98,58 | 12,51 | 1211 | 99,02 | 14,93 | 5064 | 97,62 | 10802,28 | 6116 | 533 | 18956 | 97,61 | 10808,22 | 6100 | 518 | 18834 |
| P-n70-k10 | 1 | 827,00 | 97,65 | 11,36 | 1205 | 97,96 | 14,46 | 5069 | 96,32 | 10860,59 | 6804 | 479 | 19304 | 96,32 | 10806,56 | 6795 | 479 | 19256 |
| P-n76-k4 | 1 | 593,00 | 93,56 | 20,75 | 799 | 100,00 | 121,92 | 5075 | 100,00 | 88,95 | 2417 | 165 | 138 | 100,00 | 90,95 | 2417 | 165 | 138 |
| P-n76-k5 | 1 | 627,00 | 96,97 | 159,20 | 2417 | 98,22 | 204,30 | 5089 | 99,66 | 10800,45 | 8137 | 2786 | 4463 | 99,66 | 10800,86 | 8137 | 2788 | 4463 |
| P-n101-k4 | 1 | 681,00 | 97,52 | 462,37 | 6441 | 98,90 | 8,04 | 5051 | 100,00 | 49,39 | 1729 | 109 | 3 | 100,00 | 50,22 | 1729 | 109 | 3 |
| VRPNC1 | 1 | 524,61 | 96,92 | 4,23 | 612 | 97,74 | 68,45 | 5075 | 99,60 | 17,17 | 1155 | 80 | 8 | 99,60 | 17,45 | 1155 | 80 | 8 |
| VRPNC2 | 1 | 835,26 | 97,06 | 61,14 | 11861 | 96,94 | 230,98 | 5122 | 96,47 | 10806,01 | 8563 | 282 | 16087 | 96,44 | 11119,17 | 8468 | 265 | 15695 |
| VRPNC3 | 1 | 826,14 | 94,78 | 185,31 | 33594 | 97,96 | 503,65 | 5261 | 98,69 | 9642,36 | 7306 | 1149 | 6243 | 98,69 | 9783,66 | 7306 | 1149 | 6243 |
| VRPNC4 | 1 | 1028,42 | 96,55 | 392,00 | 38531 | 97,47 | 792,77 | 105 | 92,26 | 125,70 | 9429 | 0 | 0 | 92,26 | 126,92 | 9429 | 0 | 0 |
| VRPNC5 | 1 | 1291,29 | 97,47 | 703,66 | 37898 | 94,83 | 361,46 | 5120 | 82,96 | 10,05 | 70 | 0 | 0 | 82,96 | 11,15 | 70 | 0 | 0 |
| VRPNC11 | 1 | 1042,11 | 94,83 | 277,45 | 8339 | 99,79 | 119,37 | 5128 | 91,85 | 41,75 | 10960 | 0 | 0 | 91,67 | 54,81 | 10305 | 0 | 0 |
| VRPNC12 | 1 | 819,56 | 97,25 | 60,83 | 34280 | 99,29 | 649,76 | 5100 | 100,00 | 51,59 | 9118 | 3 | 37 | 100,00 | 40,97 | 9118 | 3 | 37 |

Table 4: H1, H1+H2, BCRPA et BCPA pour le 1-PTVP = VRP



### A-2.2.2: Branchement et Coupes pour le $m$-PVP = $m$-PTVP avec $\lambda = 1$

| | $m$ | UB* | BCRPA | | | | | BCPA | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | $\frac{LB}{UB^*}$% | $T$ | CAP | PEI | MET | $\frac{LB}{UB^*}$% | $T$ | CAP | PEI | MET |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A280 | 2 | 6317,00 | 100,00 | 2866,61 | 1 | 0 | 0 | 99,96 | 187,14 | 2294 | 1564 | 0 |
| Ch130 | 2 | 14854,00 | 100,00 | 84,27 | 14 | 0 | 0 | 99,95 | 3707,26 | 1108 | 7602 | 0 |
| D198 | 2 | 35404,00 | 100,00 | 238,61 | 57 | 0 | 0 | 100,00 | 1162,31 | 1914 | 3120 | 0 |
| Eil101 | 2 | 1453,00 | 100,00 | 2,19 | 0 | 0 | 0 | 100,00 | 2575,70 | 2124 | 8088 | 0 |
| Eil51 | 2 | 982,00 | 100,00 | 0,36 | 0 | 0 | 0 | 100,00 | 4,27 | 174 | 142 | 0 |
| Eil76 | 2 | 1257,00 | 100,00 | 5,13 | 0 | 0 | 0 | 100,00 | 93,98 | 744 | 908 | 0 |
| KroA150 | 2 | 64249,00 | 100,00 | 70,80 | 22 | 0 | 0 | 100,00 | 251,73 | 1044 | 1416 | 0 |
| KroA200 | 2 | 70871,00 | 100,00 | 18,98 | 34 | 0 | 0 | 100,00 | 299,64 | 1114 | 842 | 0 |
| KroB100 | 2 | 53600,00 | 99,99 | 34,88 | 10 | 0 | 0 | 99,99 | 296,22 | 444 | 1004 | 0 |
| KroC100 | 2 | 51157,00 | 99,89 | 12,59 | 16 | 0 | 0 | 100,00 | 353,55 | 696 | 1686 | 0 |
| KroD100 | 2 | 51838,00 | 100,00 | 9,63 | 15 | 0 | 0 | 100,00 | 70,58 | 464 | 800 | 0 |
| KroE100 | 2 | 54183,00 | 100,00 | 1,98 | 0 | 0 | 0 | 100,00 | 8,00 | 210 | 14 | 0 |
| Lin105 | 2 | 36361,00 | 99,98 | 16,25 | 20 | 0 | 0 | 100,00 | 46,59 | 604 | 626 | 0 |
| Pr107 | 2 | 98381,00 | 100,00 | 2,44 | 31 | 0 | 0 | 100,00 | 34,17 | 330 | 86 | 0 |
| Pr226 | 2 | 183537,00 | 100,00 | 169,17 | 130 | 0 | 0 | 100,00 | 190,80 | 1332 | 204 | 0 |
| Pr264 | 2 | 122941,00 | 100,00 | 240,42 | 64 | 0 | 0 | 100,00 | 1620,77 | 2800 | 1098 | 0 |
| Pr76 | 2 | 266395,00 | 100,00 | 2,03 | 3 | 0 | 0 | 100,00 | 59,78 | 606 | 906 | 0 |
| Rat99 | 2 | 2928,00 | 100,00 | 11,84 | 5 | 0 | 0 | 100,00 | 432,95 | 1292 | 2692 | 0 |
| St70 | 2 | 1619,00 | 99,88 | 0,92 | 4 | 0 | 0 | 100,00 | 47,30 | 312 | 630 | 0 |
| Eil101 | 3 | 2468,00 | 100,00 | 2,61 | 0 | 0 | 0 | 100,00 | 130,70 | 3882 | 627 | 0 |
| Eil51 | 3 | 1737,00 | 100,00 | 2,73 | 0 | 0 | 0 | 100,00 | 746,83 | 2958 | 6969 | 360 |
| Eil76 | 3 | 2141,00 | 100,00 | 2,66 | 0 | 0 | 0 | 100,00 | 148,52 | 3573 | 2721 | 0 |
| KroB100 | 3 | 96691,00 | 100,00 | 10,75 | 2 | 0 | 0 | 100,00 | 850,38 | 3711 | 4809 | 0 |
| KroC100 | 3 | 92775,00 | 100,00 | 34,30 | 0 | 0 | 0 | 100,00 | 1276,16 | 3684 | 6876 | 0 |
| KroD100 | 3 | 91200,00 | 99,99 | 2,34 | 0 | 0 | 0 | 100,00 | 132,22 | 2940 | 1872 | 0 |
| KroE100 | 3 | 96406,00 | 100,00 | 80,45 | 0 | 0 | 0 | 99,97 | 2080,89 | 7980 | 13452 | 0 |
| Pr107 | 3 | 169298,00 | 100,00 | 2,92 | 20 | 0 | 0 | 100,00 | 96,31 | 2679 | 963 | 0 |
| Pr76 | 3 | 485719,00 | 99,99 | 4,92 | 0 | 0 | 0 | 100,00 | 257,80 | 2958 | 3780 | 0 |
| Eil51 | 4 | 2622,00 | 100,00 | 0,56 | 0 | 0 | 0 | 100,00 | 2453,38 | 6200 | 12440 | 628 |
| KroB100 | 4 | 147224,00 | 100,00 | 14,86 | 0 | 0 | 0 | 100,00 | 10800,70 | 17856 | 5584 | 0 |
| KroD100 | 4 | 139652,00 | 100,00 | 2,92 | 0 | 0 | 0 | 100,00 | 10800,52 | 11608 | 6292 | 0 |
| Pr107 | 4 | 263332,00 | 100,00 | 3,06 | 1 | 0 | 0 | 100,00 | 10800,84 | 11404 | 5092 | 0 |
| Pr76 | 4 | 738418,00 | 100,00 | 8,27 | 0 | 0 | 0 | 100,00 | 10801,80 | 10760 | 19416 | 312 |
| Rat99 | 4 | 7760,00 | 100,00 | 15,38 | 0 | 0 | 0 | 99,99 | 10801,28 | 11336 | 5148 | 8 |
| St70 | 4 | 4340,00 | 100,00 | 1,09 | 0 | 0 | 0 | 100,00 | 286,64 | 4312 | 1016 | 0 |

Table 5: QT, QT+VT, BCRPA et BCPA pour le $m$-PVP



### A-2.2.3: Génération de colonnes et Branchement et Coupes pour le $m$-PTVP

| | $m$ | $UB^M$ | UB | H1 | | | H1+H2 | | | BCRPA | | | | | BCPA | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | $\frac{LB}{UB}$ % | $T$ | Nbqt | $\frac{LB}{UB}$ % | $T$ | Nbt | $\frac{LB}{UB}$ % | $T$ | CAP | PEI | MET | $\frac{LB}{UB}$ % | $T$ | CAP | PEI | MET |
| A-n32-k5 | 2 | 1806,00 | 1806,00 * | 98,27 | 4,18 | 1784 | 98,96 | 6,77 | 757 | 98,01 | 0,28 | 22 | 0 | 0 | 100,00 | 2,30 | 184 | 0 | 0 |
| A-n33-k5 | 2 | 1501,00 | 1501,00 * | 98,18 | 2,70 | 1598 | 98,67 | 4,76 | 835 | 94,67 | 0,22 | 60 | 0 | 0 | 100,00 | 50,77 | 724 | 90 | 258 |
| A-n33-k6 | 2 | 1653,00 | 1647,00 * | 98,19 | 1,42 | 1007 | 98,97 | 3,28 | 630 | 99,21 | 2,61 | 43 | 0 | 31 | 100,00 | 2648,48 | 3032 | 872 | 7784 |

| | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A-n34-k5 | 2 | 1690,00 | 1689,00 * | 97,49 | 3,34 | 2135 | 98,49 | 6,29 | 688 | 99,17 | 3,78 | 66 | 0 | 164 | 100,00 | 991,47 | 2232 | 626 | 2460 |
| A-n36-k5 | 2 | 1833,00 | 1818,00 | 97,19 | 2,90 | 1944 | 98,30 | 25,71 | 764 | 99,45 | 1,88 | 27 | 0 | 184 | 99,74 | 10800,20 | 3184 | 3480 | 8036 |
| A-n37-k5 | 2 | 1583,00 | 1574,00 * | 98,30 | 3,40 | 2359 | 99,11 | 68,05 | 5424 | 99,36 | 0,86 | 38 | 0 | 0 | 100,00 | 484,55 | 2456 | 530 | 2260 |
| A-n37-k6 | 2 | 2040,00 | 2040,00 | 96,27 | 1,59 | 1311 | 97,90 | 9,56 | 730 | 97,79 | 4,28 | 142 | 0 | 66 | 97,91 | 10800,09 | 5244 | 2788 | 15846 |
| A-n38-k5 | 2 | 1655,00 | 1655,00 | 94,72 | 2,93 | 1894 | 96,28 | 33,65 | 929 | 97,58 | 2,13 | 95 | 0 | 11 | 98,47 | 10801,88 | 5964 | 4076 | 11182 |
| A-n39-k5 | 2 | 1809,00 | 1809,00 | 96,63 | 4,88 | 2228 | 97,44 | 32,32 | 795 | 98,07 | 6,70 | 246 | 0 | 288 | 98,21 | 10800,14 | 6908 | 2506 | 17862 |
| A-n39-k6 | 2 | 1929,00 | 1915,00 | 97,66 | 4,77 | 1760 | 98,27 | 22,47 | 783 | 98,96 | 5,47 | 122 | 0 | 23 | 99,18 | 10800,05 | 6742 | 3814 | 11728 |
| A-n44-k7 | 2 | 2065,00 | 2065,00 | 97,10 | 4,12 | 2401 | 97,67 | 14,71 | 606 | 97,72 | 288,72 | 207 | 0 | 1342 | 97,26 | 10803,91 | 4450 | 1234 | 26828 |
| A-n45-k6 | 2 | 2101,00 | 2101,00 | 94,04 | 4,66 | 1891 | 96,00 | 44,13 | 799 | 95,81 | 50,95 | 203 | 0 | 796 | 95,98 | 10800,50 | 4688 | 998 | 15640 |
| A-n45-k7 | 2 | 2453,00 | 2453,00 | 95,59 | 3,09 | 2120 | 97,51 | 36,55 | 883 | 97,47 | 181,97 | 480 | 0 | 1129 | 97,21 | 10800,76 | 9416 | 536 | 22612 |
| A-n46-k7 | 2 | 2100,00 | 2100,00 | 96,74 | 3,95 | 1754 | 97,64 | 39,03 | 1003 | 98,43 | 336,84 | 215 | 0 | 1144 | 97,71 | 10807,42 | 5312 | 784 | 17682 |
| A-n48-k7 | 2 | 2346,00 | 2346,00 | 97,28 | 4,54 | 2764 | 97,93 | 34,29 | 927 | 98,38 | 158,50 | 311 | 0 | 594 | 98,10 | 10800,66 | 5706 | 1652 | 16662 |
| A-n53-k7 | 2 | 2241,00 | 2241,00 | 95,67 | 10,02 | 2708 | 96,71 | 81,85 | 851 | 97,41 | 6,58 | 408 | 0 | 128 | 97,61 | 10800,86 | 11002 | 1328 | 12130 |
| A-n54-k7 | 2 | 2516,00 | 2516,00 | 95,00 | 5,98 | 2806 | 96,50 | 88,05 | 1250 | 96,03 | 448,39 | 1329 | 0 | 1511 | 95,60 | 10802,25 | 11008 | 722 | 12316 |
| A-n55-k9 | 2 | 2366,00 | 2366,00 | 95,74 | 7,40 | 2696 | 96,92 | 42,46 | 936 | 97,72 | 7,77 | 592 | 0 | 50 | 97,87 | 10800,16 | 10004 | 1274 | 16332 |
| A-n60-k9 | 2 | 2895,00 | 2895,00 | 95,63 | 10,97 | 3272 | 96,68 | 141,74 | 903 | 96,30 | 223,44 | 997 | 0 | 1068 | 96,04 | 10801,03 | 13608 | 274 | 11486 |
| A-n61-k9 | 2 | 2311,00 | 2311,00 | 93,63 | 5,46 | 2925 | 94,78 | 52,28 | 839 | 95,15 | 34,52 | 945 | 0 | 166 | 95,00 | 10803,76 | 15876 | 1102 | 9652 |
| A-n62-k8 | 2 | 2791,00 | 2791,00 | 95,10 | 19,59 | 3517 | 96,45 | 161,17 | 997 | 96,51 | 10823,28 | 1044 | 0 | 8059 | 96,08 | 10803,02 | 13440 | 316 | 9240 |
| A-n63-k9 | 2 | 3454,00 | 3454,00 | 95,52 | 6,58 | 3553 | 96,74 | 100,12 | 5231 | 96,64 | 10805,88 | 817 | 0 | 5341 | 96,44 | 3601,64 | 9996 | 194 | 4026 |
| A-n63-k10 | 2 | 2844,00 | 2844,00 | 95,67 | 8,27 | 3633 | 97,24 | 58,39 | 841 | 96,45 | 2829,59 | 856 | 0 | 2387 | 96,17 | 10805,34 | 13744 | 216 | 11568 |
| A-n64-k9 | 2 | 3040,00 | 3040,00 | 95,35 | 14,93 | 3743 | 96,37 | 86,91 | 1060 | 96,51 | 1285,59 | 784 | 0 | 2033 | 96,11 | 10802,30 | 12674 | 472 | 8972 |
| A-n65-k9 | 2 | 2611,00 | 2611,00 | 95,97 | 10,08 | 3626 | 96,66 | 61,56 | 1075 | 96,32 | 90,11 | 893 | 0 | 257 | 96,13 | 10801,14 | 11284 | 968 | 9742 |
| A-n69-k9 | 2 | 2602,00 | 2602,00 | 96,17 | 15,41 | 4662 | 96,72 | 112,96 | 1121 | 96,43 | 26,20 | 1027 | 0 | 217 | 96,36 | 10810,38 | 11204 | 1018 | 7386 |
| A-n80-k10 | 2 | 3829,00 | 3829,00 | 96,24 | 19,73 | 6888 | 97,59 | 181,51 | 5079 | 96,48 | 3615,20 | 1418 | 0 | 5552 | 96,30 | 10855,08 | 15590 | 336 | 3576 |
| B-n31-k5 | 2 | 1387,00 | 1387,00 * | 89,15 | 9,00 | 4767 | 98,43 | 39,67 | 967 | 100,00 | 1,72 | 61 | 0 | 68 | 100,00 | 25,95 | 1028 | 12 | 554 |
| B-n34-k5 | 2 | 1746,00 | 1745,00 * | 96,13 | 3,45 | 1967 | 97,75 | 84,72 | 884 | 99,71 | 1,08 | 87 | 0 | 26 | 100,00 | 192,86 | 1688 | 208 | 1066 |
| B-n35-k5 | 2 | 2020,00 | 2020,00 | 89,67 | 3,54 | 1971 | 91,67 | 107,95 | 1013 | 99,80 | 0,42 | 16 | 0 | 0 | 99,80 | 3,05 | 676 | 12 | 0 |
| B-n38-k6 | 2 | 1688,00 | 1687,00 * | 90,23 | 6,80 | 2638 | 93,38 | 66,92 | 5578 | 99,53 | 0,59 | 105 | 0 | 34 | 100,00 | 22,81 | 868 | 24 | 232 |
| B-n39-k5 | 2 | 1190,00 | 1190,00 * | 88,06 | 2,56 | 2122 | 96,12 | 686,06 | 1142 | 99,92 | 2,16 | 72 | 0 | 26 | 100,00 | 26,95 | 446 | 4 | 542 |
| B-n41-k6 | 2 | 1772,00 | 1772,00 | 91,03 | 9,05 | 2305 | 94,34 | 49,44 | 940 | 97,80 | 1,66 | 181 | 0 | 2 | 98,05 | 10800,31 | 8694 | 1044 | 30276 |
| B-n43-k6 | 2 | 1609,00 | 1590,00 * | 93,17 | 3,32 | 2197 | 96,02 | 62,93 | 1061 | 99,31 | 6,17 | 116 | 0 | 78 | 100,00 | 8165,00 | 10266 | 572 | 15362 |
| B-n44-k7 | 2 | 1913,00 | 1901,00 | 87,60 | 3,64 | 2642 | – | – | – | 99,53 | 14,67 | 318 | 0 | 168 | 99,71 | 10800,92 | 5248 | 628 | 17286 |
| B-n45-k5 | 2 | 1601,00 | 1601,00 | 87,21 | 5,59 | 2626 | – | – | – | 98,63 | 1,83 | 77 | 0 | 45 | 99,03 | 10800,41 | 6534 | 3380 | 9424 |
| B-n45-k6 | 2 | 1531,00 | 1531,00 | 91,76 | 7,13 | 3441 | 93,32 | 75,02 | 966 | 96,80 | 4,58 | 164 | 0 | 106 | 96,84 | 10801,03 | 7590 | 1104 | 25324 |
| B-n50-k7 | 2 | 1572,00 | 1571,00 * | 89,91 | 10,36 | 4467 | 92,55 | 61,53 | 5534 | 100,00 | 1,05 | 85 | 0 | 7 | 100,00 | 11,48 | 828 | 6 | 178 |
| B-n50-k8 | 2 | 2730,00 | 2730,00 | 93,88 | 5,62 | 2116 | 96,30 | 255,87 | 1323 | 97,14 | 189,24 | 779 | 0 | 934 | 97,19 | 10806,27 | 12516 | 222 | 19896 |
| B-n51-k7 | 2 | 2140,00 | 2136,00 * | 91,05 | 5,88 | 3661 | 93,66 | 35,61 | 819 | 99,77 | 0,63 | 45 | 0 | 0 | 100,00 | 111,56 | 2224 | 144 | 552 |
| B-n52-k7 | 2 | 1602,00 | 1594,00 * | 89,02 | 19,41 | 6396 | – | – | – | 99,69 | 1,58 | 113 | 0 | 4 | 100,00 | 27,61 | 1396 | 32 | 156 |
| B-n56-k7 | 2 | 1506,00 | 1502,00 | 87,06 | 25,10 | 7084 | – | – | – | 99,33 | 4,58 | 172 | 0 | 85 | 99,48 | 10800,31 | 8048 | 832 | 13876 |
| B-n57-k7 | 2 | 2382,00 | 2382,00 | 92,58 | 14,01 | 4422 | 95,37 | 1151,04 | 5619 | 98,49 | 8,36 | 423 | 0 | 67 | 98,99 | 10800,39 | 15638 | 1086 | 14184 |
| B-n57-k9 | 2 | 3314,00 | 3314,00 | 94,93 | 7,43 | 3817 | 97,86 | 72,21 | 5625 | 99,37 | 1713,03 | 793 | 0 | 3467 | 99,15 | 10833,30 | 7878 | 154 | 22156 |
| B-n63-k10 | 2 | 3194,00 | 3194,00 | 93,89 | 12,61 | 4713 | 95,49 | 164,58 | 902 | 96,81 | 14,39 | 424 | 0 | 383 | 97,12 | 10801,45 | 10540 | 428 | 13964 |
| B-n64-k9 | 2 | 1868,00 | 1868,00 | 86,40 | 9,95 | 4656 | 91,42 | 210,26 | 5657 | 97,16 | 4,00 | 228 | 0 | 28 | 97,32 | 10801,14 | 16784 | 1222 | 12214 |
| B-n66-k9 | 2 | 2833,00 | 2833,00 | 93,36 | 19,00 | 6879 | 94,49 | 509,23 | 5495 | 96,96 | 170,23 | 728 | 0 | 438 | 97,09 | 10815,55 | 13626 | 1428 | 12022 |
| B-n67-k10 | 2 | 2222,00 | 2222,00 | 94,48 | 16,71 | 5265 | 95,90 | 76,07 | 5517 | 98,47 | 56,30 | 522 | 0 | 231 | 98,54 | 10800,98 | 11288 | 482 | 16432 |
| B-n68-k9 | 2 | 2663,00 | 2663,00 | 91,29 | 29,20 | 6610 | – | – | – | 97,56 | 42,97 | 1315 | 0 | 371 | 97,52 | 10801,16 | 14722 | 384 | 11134 |
| B-n78-k10 | 2 | 2613,00 | 2613,00 | 92,25 | 17,60 | 7036 | 94,97 | 527,13 | 1182 | 97,86 | 838,47 | 1734 | 0 | 1155 | 97,67 | 5714,02 | 20912 | 154 | 2030 |
| P-n19-k2 | 2 | 501,00 | 500,00 * | 96,84 | 2,40 | 887 | 97,81 | 4,32 | 704 | 97,60 | 0,19 | 3 | 0 | 0 | 100,00 | 12,27 | 316 | 56 | 94 |
| P-n20-k2 | 2 | 511,00 | 511,00 * | 98,03 | 2,31 | 916 | 98,46 | 3,65 | 631 | 98,83 | 0,17 | 0 | 0 | 57 | 100,00 | 40,44 | 372 | 298 | 212 |
| P-n21-k2 | 2 | 511,00 | 507,00 * | 98,85 | 2,20 | 903 | 99,43 | 4,09 | 844 | 99,21 | 0,16 | 0 | 0 | 76 | 100,00 | 3,06 | 230 | 16 | 24 |
| P-n22-k2 | 2 | 512,00 | 511,00 * | 98,25 | 2,23 | 869 | 99,06 | 4,35 | 667 | 99,41 | 0,34 | 6 | 0 | 85 | 100,00 | 12,48 | 306 | 136 | 32 |
| P-n40-k5 | 2 | 1072,00 | 1065,00 * | 99,09 | 8,53 | 2317 | 99,26 | 14,57 | 1061 | 99,72 | 2,84 | 9 | 0 | 15 | 100,00 | 1745,08 | 3182 | 1894 | 3330 |

| | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P-n45-k5 | 2 | 1170,00 | 1148,00 * | 99,30 | 6,13 | 2332 | 99,65 | 32,00 | 1026 | 99,83 | 1,98 | 3 | 0 | 50 | 100,00 | 903,36 | 3428 | 866 | 1440 |
| P-n50-k10 | 2 | 1549,00 | 1549,00 | 95,98 | 3,90 | 2975 | 96,79 | 9,42 | 779 | 96,97 | 0,80 | 4 | 0 | 0 | 97,07 | 10800,25 | 12750 | 1296 | 18110 |
| P-n50-k7 | 2 | 1257,00 | 1257,00 | 98,57 | 9,95 | 3458 | 98,69 | 15,09 | 863 | 99,12 | 1,70 | 24 | 0 | 49 | 99,22 | 10801,03 | 7730 | 2314 | 14758 |
| P-n50-k8 | 2 | 1424,00 | 1424,00 | 92,95 | 5,23 | 2489 | 93,32 | 24,35 | 543 | 93,61 | 7,91 | 65 | 0 | 88 | 93,54 | 10809,19 | 9242 | 1948 | 17498 |
| P-n51-k10 | 2 | 1678,00 | 1678,00 | 94,91 | 3,90 | 2629 | 95,59 | 23,10 | 591 | 96,01 | 4,69 | 64 | 0 | 9 | 96,02 | 10801,83 | 14520 | 918 | 14880 |
| P-n55-k10 | 2 | 1543,00 | 1543,00 | 97,92 | 9,49 | 3045 | 97,99 | 15,77 | 872 | 98,31 | 16,38 | 7 | 0 | 108 | 98,17 | 10801,59 | 10782 | 848 | 19660 |
| P-n55-k7 | 2 | 1286,00 | 1286,00 | 98,42 | 24,91 | 5033 | 98,47 | 42,06 | 1093 | 98,83 | 6,03 | 14 | 0 | 43 | 98,83 | 10801,03 | 12114 | 2108 | 12568 |
| P-n55-k8 | 2 | 1294,00 | 1294,00 | 97,83 | 10,81 | 3915 | 98,01 | 32,14 | 846 | 98,22 | 1,45 | 32 | 0 | 120 | 98,22 | 10806,70 | 11334 | 2330 | 16264 |
| P-n60-k10 | 2 | 1659,00 | 1659,00 | 96,25 | 8,21 | 2615 | 96,32 | 30,72 | 988 | 96,38 | 2,48 | 26 | 0 | 111 | 96,35 | 10800,69 | 11378 | 1210 | 16736 |
| P-n65-k10 | 2 | 1761,00 | 1761,00 | 96,63 | 7,68 | 3432 | 97,03 | 37,50 | 833 | 97,10 | 1,66 | 10 | 0 | 58 | 97,10 | 10800,75 | 13180 | 456 | 13680 |
| P-n70-k10 | 2 | 1826,00 | 1826,00 | 95,03 | 12,89 | 3403 | 95,30 | 56,93 | 833 | 95,29 | 781,23 | 175 | 0 | 2026 | 95,03 | 10801,00 | 12906 | 654 | 10060 |
| P-n76-k4 | 2 | 1385,00 | 1385,00 | 96,86 | 281,93 | 29806 | 97,39 | 2043,89 | 5585 | 97,62 | 1,20 | 35 | 0 | 0 | 97,62 | 10808,42 | 13006 | 1808 | 4194 |
| P-n76-k5 | 2 | 1440,00 | 1440,00 | 97,20 | 76,44 | 12130 | 97,48 | 368,01 | 6858 | 97,64 | 10,78 | 114 | 0 | 1939 | 97,57 | 10801,36 | 13836 | 1424 | 4128 |
| P-n101-k4 | 2 | 1581,00 | 1581,00 | 97,21 | 634,75 | 47948 | – | – | – | 98,48 | 18,17 | 9 | 0 | 1130 | 98,45 | 1371,63 | 10542 | 1474 | 456 |
| VRPNC1 | 2 | 1198,38 | 1198,38 | 97,40 | 7,94 | 3837 | – | – | – | 98,02 | 5,88 | 20 | 0 | 356 | 98,06 | 10803,34 | 5348 | 3864 | 8962 |
| VRPNC2 | 2 | 1846,10 | 1846,10 | 95,50 | 25,90 | 5461 | – | – | – | 95,68 | 2305,33 | 190 | 0 | 4220 | 95,36 | 10806,03 | 11822 | 356 | 6348 |
| VRPNC3 | 2 | 1871,83 | 1871,83 | 96,85 | 218,26 | 24960 | – | – | – | 97,11 | 39,88 | 228 | 0 | 794 | 96,93 | 101,70 | 14918 | 0 | 0 |
| VRPNC4 | 2 | 2381,38 | 2381,38 | | 446,12 | 21642 | – | – | – | 95,11 | 13,05 | 989 | 0 | 0 | 94,68 | 51,22 | 9406 | 0 | 0 |
| VRPNC5 | 2 | 2947,26 | 2947,26 | 88,98 | 157,45 | 29129 | – | – | – | 88,72 | 90,80 | 7118 | 0 | 0 | 88,97 | 12,13 | 208 | 0 | 0 |
| VRPNC11 | 2 | 2318,86 | 2318,86 | 93,70 | 200,34 | 33970 | – | – | – | 92,53 | 189,95 | 4052 | 0 | 5965 | 84,68 | 32,02 | 12970 | 0 | 0 |
| VRPNC12 | 2 | 1821,71 | 1810,58 | 97,04 | 86,80 | 16003 | – | – | – | 99,90 | 57,05 | 493 | 0 | 886 | 99,99 | 10146,61 | 16056 | 662 | 7608 |
| A-n32-k5 | 3 | 3368,00 | 3340,00 * | 99,29 | 10,36 | 7120 | 99,48 | 23,09 | 1742 | 100,00 | 0,33 | 0 | 0 | 0 | 100,00 | 43,91 | 2523 | 54 | 132 |
| A-n33-k5 | 3 | 2684,00 | 2669,00 * | 98,93 | 5,46 | 4198 | 99,34 | 10,53 | 1112 | 100,00 | 1,03 | 0 | 0 | 0 | 100,00 | 82,70 | 5634 | 99 | 318 |
| A-n34-k5 | 3 | 3010,00 | 2987,00 | 99,31 | 7,66 | 5870 | 99,44 | 19,64 | 1815 | 99,63 | 0,19 | 0 | 0 | 0 | 99,63 | 10802,39 | 26844 | 5415 | 13908 |
| A-n36-k5 | 3 | 3258,00 | 3229,00 * | 99,69 | 27,67 | 16712 | 99,78 | 50,70 | 2126 | 100,00 | 0,28 | 2 | 0 | 0 | 100,00 | 3669,36 | 12486 | 2649 | 3525 |
| A-n37-k5 | 3 | 2774,00 | 2751,00 * | 98,74 | 38,94 | 22393 | 99,34 | 73,48 | 3722 | 100,00 | 0,22 | 0 | 0 | 0 | 100,00 | 108,95 | 6180 | 159 | 453 |
| A-n37-k6 | 3 | 3754,00 | 3754,00 | 96,57 | 8,38 | 5800 | 96,49 | 30,61 | 1833 | 96,67 | 0,67 | 0 | 0 | 0 | 96,67 | 10800,48 | 31590 | 2208 | 15714 |
| A-n38-k5 | 3 | 2900,00 | 2851,00 * | 99,33 | 7,52 | 5183 | 99,46 | 33,45 | 1247 | 100,00 | 0,31 | 3 | 0 | 0 | 100,00 | 7815,30 | 21426 | 2031 | 6045 |
| A-n39-k5 | 3 | 3133,00 | 3133,00 | 96,90 | 5,54 | 3069 | 97,26 | 49,58 | 1863 | 97,67 | 0,69 | 29 | 0 | 0 | 97,67 | 10801,19 | 20664 | 2895 | 12852 |
| A-n39-k6 | 3 | 3468,00 | 3417,00 * | 99,48 | 5,58 | 6405 | 99,85 | 30,23 | 2531 | 99,97 | 0,25 | 0 | 0 | 0 | 100,00 | 3123,80 | 16308 | 1515 | 5436 |
| A-n44-k7 | 3 | 3558,00 | 3558,00 | 97,36 | 33,24 | 12625 | 97,55 | 67,30 | 1451 | 98,00 | 1,94 | 6 | 0 | 0 | 98,00 | 10801,19 | 25020 | 1869 | 13041 |
| A-n45-k6 | 3 | 3856,00 | 3856,00 | 91,88 | 8,22 | 4848 | 92,19 | 169,37 | 6020 | 92,92 | 1,49 | 10 | 0 | 16 | 92,91 | 10800,80 | 26115 | 3033 | 7008 |
| A-n45-k7 | 3 | 4138,00 | 4138,00 | 97,49 | 13,78 | 7295 | 97,65 | 46,72 | 1623 | 98,43 | 0,41 | 23 | 0 | 0 | 98,43 | 10814,52 | 25629 | 1416 | 14871 |
| A-n46-k7 | 3 | 3703,00 | 3686,00 * | 98,71 | 17,93 | 12099 | 98,77 | 47,44 | 2418 | 99,54 | 0,41 | 1 | 0 | 0 | 100,00 | 6161,06 | 28449 | 570 | 6765 |
| A-n48-k7 | 3 | 4158,00 | 4158,00 | 97,57 | 34,37 | 12339 | 97,80 | 78,36 | 2404 | 98,68 | 0,45 | 2 | 0 | 0 | 98,68 | 10801,88 | 36015 | 894 | 10083 |
| A-n53-k7 | 3 | 3908,00 | 3908,00 | 96,31 | 9,94 | 7039 | 96,61 | 76,55 | 2699 | 97,67 | 0,63 | 14 | 0 | 0 | 97,67 | 10801,48 | 26751 | 1407 | 6255 |
| A-n54-k7 | 3 | 4168,00 | 4168,00 | 95,78 | 19,45 | 7625 | 96,51 | 91,00 | 2553 | 96,31 | 2,17 | 158 | 0 | 1 | 96,37 | 10801,23 | 21720 | 1503 | 8181 |
| A-n55-k9 | 3 | 4177,00 | 4177,00 | 96,96 | 40,37 | 15595 | 97,00 | 79,98 | 2998 | 97,80 | 6,23 | 0 | 0 | 0 | 97,80 | 10812,09 | 27819 | 1491 | 14205 |
| A-n60-k9 | 3 | 4834,00 | 4834,00 | 97,40 | 17,94 | 9951 | 97,69 | 70,73 | 6672 | 97,93 | 9,73 | 42 | 0 | 10 | 97,91 | 10801,49 | 23367 | 924 | 10926 |
| A-n61-k9 | 3 | 4035,00 | 4035,00 | 93,65 | 25,01 | 10145 | 93,71 | 72,14 | 1604 | 94,03 | 0,84 | 10 | 0 | 0 | 94,03 | 10962,05 | 23172 | 447 | 4278 |
| A-n62-k8 | 3 | 4769,00 | 4769,00 | 96,32 | 60,58 | 15229 | 96,59 | 509,51 | 2049 | 97,50 | 6,17 | 33 | 0 | 4 | 97,48 | 10809,86 | 26946 | 693 | 6393 |
| A-n63-k9 | 3 | 5888,00 | 5888,00 | 94,67 | 16,63 | 8665 | 96,01 | 110,71 | 2190 | 96,47 | 1,59 | 9 | 0 | 3 | 96,47 | 11051,50 | 23028 | 966 | 4752 |
| A-n63-k10 | 3 | 4967,00 | 4967,00 | 97,06 | 54,01 | 16113 | 97,09 | 114,52 | 3726 | 97,12 | 5,44 | 6 | 0 | 0 | 97,10 | 10873,22 | 27999 | 495 | 6753 |
| A-n64-k9 | 3 | 5271,00 | 5271,00 | 95,79 | 20,01 | 9428 | 97,39 | 497,75 | 2184 | 97,70 | 1,50 | 0 | 0 | 0 | 97,70 | 10805,99 | 32328 | 1077 | 4599 |
| A-n65-k9 | 3 | 4420,00 | 4420,00 | 93,67 | 7,53 | 7858 | 93,67 | 7,53 | 5064 | 96,72 | 4,17 | 18 | 0 | 0 | 96,71 | 10804,66 | 24537 | 633 | 7683 |
| A-n69-k9 | 3 | 4457,00 | 4457,00 | 96,64 | 73,73 | 20836 | 96,64 | 191,34 | 3959 | 96,84 | 9,70 | 28 | 0 | 27 | 96,84 | 10806,91 | 24972 | 576 | 4515 |
| A-n80-k10 | 3 | 6454,00 | 6454,00 | 97,23 | 46,83 | 17062 | 97,58 | 541,90 | 7548 | 97,83 | 22,34 | 12 | 0 | 30 | 97,75 | 35,39 | 21657 | 0 | 0 |
| B-n31-k5 | 3 | 2607,00 | 2603,00 * | 97,50 | 10,33 | 11671 | 99,31 | 26,77 | 1512 | 100,00 | 1,63 | 0 | 0 | 0 | 100,00 | 17,70 | 2265 | 24 | 93 |
| B-n34-k5 | 3 | 2903,00 | 2892,00 | 98,36 | 9,97 | 3431 | 99,11 | 462,20 | 959 | 99,93 | 0,34 | 4 | 0 | 0 | 99,93 | 3602,91 | 13602 | 1140 | 6129 |
| B-n35-k5 | 3 | 3222,00 | 3205,00 * | 93,19 | 17,35 | 7601 | 93,50 | 63,17 | 1695 | 100,00 | 0,34 | 10 | 0 | 0 | 100,00 | 33,61 | 2946,00 | 45 | 66 |
| B-n38-k6 | 3 | 2860,00 | 2860,00 | 96,92 | 24,15 | 11279 | 97,24 | 57,53 | 1702 | 99,62 | 0,25 | 4 | 0 | 0 | 99,62 | 82,70 | 6423,00 | 66 | 108 |
| B-n39-k5 | 3 | 2076,00 | 2018,00 * | 94,80 | 6,23 | 7362 | – | – | – | 100,00 | 0,33 | 9 | 0 | 0 | 100,00 | 41,00 | 2133,00 | 99 | 282 |

| | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| B-n41-k6 | 3 | 3015,00 | 3015,00 | 94,62 | 7,50 | 6423 | 95,11 | 34,88 | 1090 | 93,43 | 0,36 | 6 | 0 | 0 | 97,48 | 10801,58 | 26307,00 | 1521 | 16125 |
| B-n43-k6 | 3 | 2733,00 | 2733,00 | 96,46 | 18,19 | 10900 | 96,95 | 349,82 | 2082 | 98,17 | 0,34 | 14 | 0 | 0 | 98,17 | 10802,06 | 27120,00 | 3012 | 14634 |
| B-n44-k7 | 3 | 3217,00 | 3173,00 * | 93,94 | 29,08 | 12221 | 94,18 | 199,06 | 2309 | 99,91 | 0,41 | 9 | 0 | 0 | 100,00 | 4266,67 | 12396,00 | 672 | 5790 |
| B-n45-k5 | 3 | 2666,00 | 2666,00 | 86,69 | 7,99 | 6756 | 91,65 | 726,72 | 1485 | 96,70 | 5,31 | 38 | 0 | 12 | 96,70 | 10827,42 | 24303,00 | 3672 | 10578 |
| B-n45-k6 | 3 | 2585,00 | 2585,00 | 92,52 | 14,24 | 7498 | 93,95 | 88,98 | 1042 | 96,71 | 37,16 | 40 | 0 | 69 | 96,60 | 10800,42 | 20679,00 | 1515 | 13038 |
| B-n50-k7 | 3 | 2616,00 | 2606,00 * | 95,55 | 38,47 | 15435 | – | – | – | 100,00 | 0,50 | 16 | 0 | 0 | 100,00 | 9,77 | 1005 | 24 | 0 |
| B-n50-k8 | 3 | 4488,00 | 4488,00 | 95,25 | 13,17 | 7317 | 95,81 | 75,01 | 2213 | 96,26 | 7,66 | 40 | 0 | 29 | 96,17 | 10874,78 | 26922 | 606 | 9753 |
| B-n51-k7 | 3 | 3542,00 | 3542,00 | 91,59 | 13,25 | 9929 | 94,73 | 77,44 | 1739 | 97,80 | 6,34 | 34 | 0 | 4 | 97,57 | 10820,67 | 29091 | 1929 | 10341 |
| B-n52-k7 | 3 | 2709,00 | 2676,00 * | 93,96 | 34,94 | 18315 | – | – | – | 100,00 | 0,55 | 7 | 0 | 0 | 100,00 | 2656,70 | 16500 | 627 | 5391 |
| B-n56-k7 | 3 | 2427,00 | 2427,00 | 75,98 | 6,90 | 9893 | – | – | – | 98,89 | 2,91 | 41 | 0 | 10 | 98,81 | 10801,63 | 19365 | 1839 | 15267 |
| B-n57-k7 | 3 | 4127,00 | 4127,00 | 89,22 | 28,24 | 9769 | – | – | – | 93,12 | 1,00 | 37 | 0 | 0 | 93,12 | 10801,17 | 20454 | 903 | 3189 |
| B-n57-k9 | 3 | 5490,00 | 5490,00 | 97,04 | 30,75 | 15360 | 97,59 | 85,69 | 6870 | 99,21 | 1,25 | 18 | 0 | 0 | 99,21 | 10801,11 | 27942 | 609 | 13494 |
| B-n63-k10 | 3 | 5553,00 | 5553,00 | 95,93 | 32,14 | 21238 | 96,72 | 86,77 | 6859 | 98,60 | 0,88 | 8 | 0 | 0 | 98,60 | 10801,58 | 27357 | 459 | 10440 |
| B-n64-k9 | 3 | 3286,00 | 3286,00 | 87,60 | 18,44 | 10480 | 88,64 | 109,29 | 2206 | 92,94 | 1,14 | 81 | 0 | 0 | 92,94 | 10841,09 | 24036 | 1461 | 3312 |
| B-n66-k9 | 3 | 4696,00 | 4696,00 | 94,37 | 59,19 | 16390 | – | – | – | 96,85 | 4,39 | 59 | 0 | 13 | 96,83 | 10804,05 | 20907 | 882 | 10935 |
| B-n67-k10 | 3 | 3693,00 | 3693,00 | 94,71 | 34,21 | 18003 | 95,74 | 224,22 | 6669 | 97,62 | 0,86 | 25 | 0 | 0 | 97,62 | 10810,28 | 31197 | 345 | 6783 |
| B-n68-k9 | 3 | 4336,00 | 4336,00 | 91,84 | 52,34 | 13328 | – | – | – | 95,57 | 4,91 | 99 | 0 | 5 | 95,55 | 10832,01 | 22896 | 363 | 6912 |
| B-n78-k10 | 3 | 4234,00 | 4234,00 | 91,84 | 44,60 | 22487 | – | – | – | 96,79 | 34,42 | 208 | 0 | 151 | 96,74 | 435,66 | 23187 | 0 | 0 |
| P-n19-k2 | 3 | 882,00 | 881,00 | 98,90 | 4,88 | 2131 | 99,21 | 7,63 | 876 | 99,77 | 0,42 | 2 | 0 | 0 | 99,77 | 64,72 | 1668,00 | 537 | 282 |
| P-n20-k2 | 3 | 897,00 | 897,00 | 98,83 | 2,57 | 1997 | 99,02 | 6,18 | 799 | 99,55 | 0,09 | 0 | 0 | 0 | 99,55 | 24,41 | 1293,00 | 210 | 168 |
| P-n21-k2 | 3 | 898,00 | 898,00 * | 99,28 | 17,99 | 9371 | 99,53 | 23,09 | 935 | 100,00 | 0,09 | 0 | 0 | 0 | 100,00 | 23,56 | 963,00 | 126 | 30 |
| P-n22-k2 | 3 | 906,00 | 901,00 * | 99,46 | 21,58 | 9831 | 99,70 | 38,80 | 1636 | 100,00 | 0,11 | 0 | 0 | 0 | 100,00 | 51,58 | 2283,00 | 564 | 231 |
| P-n40-k5 | 3 | 1869,00 | 1853,00 * | 99,62 | 40,62 | 20944 | 99,61 | 64,96 | 4967 | 100,00 | 0,77 | 0 | 0 | 0 | 100,00 | 4990,00 | 24339,00 | 1719 | 5229 |
| P-n45-k5 | 3 | 2063,00 | 2063,00 | 97,69 | 45,63 | 17345 | 98,49 | 142,40 | 4000 | 98,74 | 1,33 | 0 | 0 | 0 | 98,74 | 10802,44 | 29526,00 | 3339 | 9240 |
| P-n50-k7 | 3 | 2203,00 | 2203,00 | 98,72 | 47,81 | 22189 | 98,66 | 72,93 | 4744 | 98,82 | 3,67 | 0 | 0 | 0 | 98,82 | 10801,67 | 34779,00 | 936 | 12621 |
| P-n55-k7 | 3 | 2244,00 | 2244,00 | 98,98 | 58,59 | 29113 | 99,06 | 98,55 | 6707 | 99,15 | 0,52 | 0 | 0 | 0 | 99,15 | 10803,98 | 40782,00 | 1023 | 5367 |
| P-n55-k8 | 3 | 2259,00 | 2233,00 | 99,42 | 62,98 | 26078 | 99,57 | 115,44 | 7219 | 99,64 | 0,53 | 0 | 0 | 0 | 99,64 | 10803,83 | 27909,00 | 1815 | 9390 |
| P-n65-k10 | 3 | 3029,00 | 3029,00 | 98,13 | 40,05 | 21142 | 98,19 | 95,85 | 5828 | 98,25 | 7,80 | 0 | 0 | 0 | 98,25 | 10801,61 | 31971,00 | 711 | 9015 |
| P-n70-k10 | 3 | 3159,00 | 3159,00 | 95,39 | 45,93 | 23824 | 95,57 | 112,34 | 6784 | 95,82 | 8,67 | 0 | 0 | 0 | 95,79 | 7781,20 | 24987,00 | 252 | 1572 |
| P-n76-k4 | 3 | 2374,00 | 2374,00 | 96,79 | 317,68 | 37965 | – | – | – | 97,64 | 2,41 | 0 | 0 | 0 | 97,60 | 163,42 | 15435,00 | 0 | 0 |
| P-n76-k5 | 3 | 2464,00 | 2464,00 | 97,39 | 200,66 | 30351 | – | – | – | 97,93 | 7,42 | 0 | 0 | 0 | 97,93 | 10802,11 | 15972,00 | 981 | 1284 |
| P-n101-k4 | 3 | 2702,00 | 2702,00 | 86,45 | 298,51 | 21842 | – | – | – | 98,37 | 7,97 | 0 | 0 | 0 | 98,37 | 29,81 | 7470,00 | 0 | 0 |
| VRPNC1 | 3 | 2127,77 | 2127,77 | 97,76 | 24,09 | 17000 | 98,06 | 96,13 | 14991 | 98,20 | 0,44 | 0 | 0 | 0 | 98,20 | 10822,69 | 30045 | 3435 | 7023 |
| VRPNC2 | 3 | 3184,81 | 3184,81 | 95,94 | 82,03 | 30063 | 95,94 | 156,49 | 11324 | 95,96 | 1,25 | 0 | 0 | 0 | 95,96 | 787,83 | 23136 | 15 | 543 |
| VRPNC3 | 3 | 3174,74 | 3174,74 | 97,67 | 211,79 | 76955 | – | – | – | 97,97 | 31,56 | 2 | 0 | 2 | 97,95 | 48,31 | 13797 | 0 | 0 |
| VRPNC4 | 3 | 4023,86 | 4023,86 | 95,52 | 294,95 | 117531 | – | – | – | 96,89 | 45,72 | 0 | 0 | 0 | 96,89 | 101,05 | 23466 | 0 | 0 |
| VRPNC5 | 3 | 4913,56 | 4913,56 | 90,27 | 453,37 | 136916 | – | – | – | 93,37 | 1066,67 | 62 | 0 | 1322 | 93,22 | 10,05 | 126 | 0 | 0 |
| VRPNC11 | 3 | 3706,39 | 3706,39 | 88,58 | 235,08 | 99085 | – | – | – | 92,78 | 10802,61 | 1138 | 0 | 4193 | 87,89 | 11,00 | 315 | 0 | 0 |
| VRPNC12 | 3 | 3000,82 | 3000,82 | 95,23 | 38,69 | 36969 | – | – | – | 99,92 | 441,38 | 96 | 0 | 609 | 99,51 | 1550,59 | 33834 | 36 | 207 |
| P-n19-k2 | 4 | 1330,00 | 1321,00 * | 98,23 | 3,14 | 3921 | 99,60 | 17,30 | 6241 | 100,00 | 0,16 | 1 | 0 | 0 | 100,00 | 244,52 | 4324 | 820 | 464 |
| P-n20-k2 | 4 | 1360,00 | 1352,00 * | 96,55 | 4,31 | 4893 | 99,36 | 57,97 | 7096 | 100,00 | 0,11 | 0 | 0 | 0 | 100,00 | 112,09 | 4792 | 656 | 500 |
| P-n21-k2 | 4 | 1366,00 | 1362,00 * | 99,19 | 10,34 | 7777 | 99,88 | 38,19 | 7586 | 100,00 | 0,09 | 0 | 0 | 0 | 100,00 | 106,69 | 4420 | 704 | 296 |
| P-n22-k2 | 4 | 1365,00 | 1356,00 * | 98,97 | 34,93 | 15353 | 99,86 | 77,98 | 9405 | 100,00 | 0,11 | 0 | 0 | 0 | 100,00 | 138,38 | 5988 | 1028 | 368 |
| P-n45-k5 | 4 | 3079,00 | 3079,00 | 96,98 | 88,41 | 42539 | 98,64 | 221,72 | 44636 | 98,83 | 0,39 | 0 | 0 | 0 | 98,83 | 10804,99 | 31420 | 1144 | 0 |
| P-n76-k4 | 4 | 3545,00 | 3545,00 | 94,96 | 383,65 | 118859 | – | – | – | 98,00 | 25,08 | 0 | 0 | 0 | 97,95 | 2184,31 | 23420 | 0 | 0 |
| P-n76-k5 | 4 | 3704,00 | 3704,00 | 94,09 | 212,85 | 112465 | – | – | – | 97,60 | 11,08 | 0 | 0 | 0 | 97,60 | 225,75 | 17872 | 0 | 0 |
| P-n101-k4 | 4 | 4038,00 | 4038,00 | 96,20 | 1299,96 | 148457 | – | – | – | 98,66 | 15,00 | 0 | 0 | 0 | 98,66 | 154,59 | 11896 | 0 | 0 |
| VRPNC1 | 4 | 3215,19 | 3215,19 | 91,59 | 76,99 | 45305 | – | – | – | 97,45 | 0,53 | 0 | 0 | 0 | 97,45 | 10831,97 | 26444 | 1544 | 3748 |
| VRPNC3 | 4 | 4699,58 | 4699,58 | 96,74 | 227,09 | 131051 | – | – | – | 98,68 | 6,20 | 0 | 0 | 0 | 98,68 | 80,06 | 17696 | 0 | 0 |
| VRPNC4 | 4 | 5972,71 | 5972,71 | 93,53 | 329,55 | 205712 | – | – | – | 97,50 | 62,19 | 0 | 0 | 0 | – | – | – | – | – |
| VRPNC5 | 4 | 7263,92 | 7263,92 | 87,02 | 508,50 | 258766 | – | – | – | 94,57 | 481,53 | 0 | 0 | 5 | 94,57 | 1108,20 | 2988 | 0 | 0 |
| VRPNC11 | 4 | 5275,63 | 5275,63 | 84,56 | 250,80 | 166092 | – | – | – | 92,86 | 10811,13 | 255 | 0 | 2149 | 90,34 | 30,59 | 8192 | 0 | 0 |

| | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| VRPNC12 | 4 | 4466,61 | 4466,61 | 91,07 | 39,39 | 66730 | – | – | – | 97,75 | 8,13 | 50 | 0 | 17 | 97,63 | 924,48 | 25004 | 0 | 0 |
| P-n21-k2 | 5 | 1899,00 | 1893,00 * | 99,08 | 34,74 | 19379 | 99,76 | 151,51 | 9668 | 100,00 | 0,13 | 0 | 0 | 0 | 100,00 | 445,50 | 11925 | 1310 | 290 |
| P-n22-k2 | 5 | 1897,00 | 1894,00 | 98,56 | 23,20 | 16592 | 99,67 | 141,35 | 10543 | 99,74 | 0,13 | 0 | 0 | 0 | 99,74 | 3247,52 | 19560 | 3880 | 3055 |
| P-n76-k4 | 5 | 4880,00 | 4880,00 | 93,23 | 157,86 | 52742 | – | – | – | 98,16 | 27,63 | 0 | 0 | 0 | 98,15 | 743,28 | 19670 | 0 | 0 |
| P-n76-k5 | 5 | 5082,00 | 5082,00 | 94,72 | 283,59 | 146489 | – | – | – | 97,99 | 9,94 | 0 | 0 | 0 | 97,99 | 170,00 | 24350 | 0 | 0 |
| P-n101-k4 | 5 | 5525,00 | 5525,00 | 93,73 | 651,37 | 104304 | – | – | – | 98,95 | 49,00 | 0 | 0 | 0 | 98,94 | 269,56 | 13935 | 0 | 0 |
| VRPNC1 | 5 | 4424,74 | 4424,74 | 90,46 | 90,95 | 67857 | – | – | – | 97,00 | 1,86 | 0 | 0 | 0 | 97,00 | 9934,73 | 38420 | 0 | 0 |
| VRPNC3 | 5 | 6431,66 | 6431,66 | 96,58 | 254,22 | 187251 | – | – | – | 98,75 | 17,83 | 0 | 0 | 0 | 98,75 | 33,27 | 21135 | 0 | 0 |
| VRPNC4 | 5 | 8141,82 | 8141,82 | 92,93 | 330,10 | 279350 | – | – | – | 98,06 | 123,44 | 0 | 0 | 0 | 98,06 | 163,02 | 9280 | 0 | 0 |
| VRPNC5 | 5 | 9949,03 | 9949,03 | 86,79 | 520,73 | 367056 | – | – | – | 94,93 | 293,99 | 0 | 0 | 0 | 94,93 | 180,16 | 68500 | 0 | 0 |
| VRPNC11 | 5 | 7075,20 | 7075,20 | 82,99 | 266,23 | 228337 | – | – | – | 93,03 | 1944,24 | 105 | 0 | 350 | 90,81 | 99,23 | 39175 | 0 | 0 |
| VRPNC12 | 5 | 6699,49 | 6699,49 | 81,43 | 47,52 | 111000 | – | – | – | 96,65 | 14,58 | 2 | 0 | 0 | 96,63 | 40,31 | 15210 | 0 | 0 |
| P-n76-k4 | 6 | 6367,00 | 6367,00 | 95,60 | 610,43 | 187029 | – | – | – | 98,15 | 7,42 | 0 | 0 | 0 | 98,15 | 16,72 | 13698 | 0 | 0 |
| P-n76-k5 | 6 | 6593,00 | 6593,00 | 95,86 | 277,06 | 188486 | – | – | – | 98,35 | 1,89 | 0 | 0 | 0 | 98,35 | 146,80 | 25602 | 0 | 0 |
| P-n101-k4 | 6 | 7168,00 | 7168,00 | 97,57 | 2338,08 | 237208 | – | – | – | 99,14 | 39,91 | 0 | 0 | 0 | 99,14 | 127,58 | 15558 | 0 | 0 |
| VRPNC3 | 6 | 8321,82 | 8321,82 | 95,06 | 484,31 | 226856 | – | – | – | 98,85 | 3,98 | 0 | 0 | 0 | 98,85 | 45,28 | 17226 | 0 | 0 |
| VRPNC4 | 6 | 10576,20 | 10576,20 | 94,75 | 471,44 | 348755 | – | – | – | 98,19 | 96,36 | 0 | 0 | 0 | 98,19 | 123,55 | 41460 | 0 | 0 |
| VRPNC11 | 6 | 9238,82 | 9238,82 | 78,30 | 281,43 | 298815 | – | – | – | 91,72 | 3601,91 | 108 | 0 | 350 | – | – | – | – | – |
| P-n76-k4 | 7 | 7993,00 | 7993,00 | 94,66 | 676,47 | 223741 | – | – | – | 98,16 | 16,86 | 0 | 0 | 0 | 98,16 | 76,02 | 15526 | 0 | 0 |
| P-n76-k5 | 7 | 8292,00 | 8292,00 | 94,58 | 370,64 | 225707 | – | – | – | 98,09 | 17,05 | 0 | 0 | 0 | 98,09 | 30,92 | 14042 | 0 | 0 |
| P-n101-k4 | 7 | 8965,00 | 8965,00 | 97,07 | 289,21 | 47352 | – | – | – | 99,10 | 4,38 | 0 | 0 | 0 | 99,10 | 45,20 | 10115 | 0 | 0 |
| VRPNC11 | 7 | 11668,00 | 11668,00 | 73,66 | 296,20 | 362791 | – | – | – | 90,30 | 69,45 | 32 | 0 | 19 | – | – | – | – | – |

Table 6: H1, H1+H2, BCRPA et BCPA pour le $m$-PTVP

# Bibliographie


[1] N. Altay and W. G. Green III. Or/ms research in disaster operations management. *European Journal of Operational Research*, 175(1) :475–493, 2006.

[2] D.L. Applegate, V. Chvátal, Cook W.J., and R.E Bixby. *The Traveling Salesman Problem*. Princeton University Press, 2006.

[3] J.R. Araque. Lots of combs of different sizes for vehicle routing. Technical Report OR 232-90, Center for Operations Research and Econometrics, Catholic University of Louvain, Belgium, August 1990.

[4] J.R. Araque, L.A. Hall, and T.L. Magnanti. Capacitated trees, capacitated routing and associated polyhedra. Discussion paper, Center for Operations Research and Econometrics, Catholic University of Louvain, Belgium, August 1990.

[5] Aaron Archer and David P. Williamson. Faster approximation algorithms for the minimum latency problem. In *SODA '03 : Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 88–96. Society for Industrial and Applied Mathematics, 2003.

[6] C. Archetti, M. G. Speranza, and A. Hertz. A tabu search algorithm for the split delivery vehicle routing problem. *Transportation Science*, 40(1) :64–73, 2006.

[7] E.M. Arkin, R. Hassin, and A. Levin. Approximations for minimum and min-max vehicle routing problems. *J. Algorithms*, 59(1) :1–18, 2006.

[8] P. Augerat. *Approche polyédrale du problème de tournées de véhicules*. PhD thesis, Institut National Polytechnique de Grenoble, France, 1995.

[9] P. Augerat, J.M. Belenguer, E. Benavent, A. Corberan, D. Naddef, and G. Rinaldi. Computational results with a branch-and-cut code for the capacitated vehicle routing problem. Research report rr949-m, ARTEMIS-IMAG, France, 1995.

[10] R. Baldacci, N. Christofides, and 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.

[11] R. Baldacci, N., and A. Mingozzi. A unified exact method for solving different classes of vehicle routing problems. *Mathematical Programming*, 120(2) :347–380, 2009.

[12] M. Balinski and R. Quandt. On an integer program for a delivery problem. *Operations Research*, 12 :300–3004, 1964.

[13] G. Barbarosoglu, L. Özdamar, and A. Cevik. An interactive approach for hierarchical analysis of helicopter logistics in disaster relief operations. *European Journal of Operational Research*, 140(1) :118–133, 2002.

[14] R. Bellman. Dynamic programming and stochastic control processes. *Information and Control*, 1(3) :228–239, 1958.

[15] C. Berge. *Théorie des graphes et ses applications*. Dunod, 1958.

[16] L. Bertacco, L. Brunetta, and M. Fischetti. The linear ordering problem with cumulative costs. *European Journal of Operational Research*, 189(3) :1345–1357, 2008.

[17] L. Bianco, A. Mingozzi, and S. Ricciardelli. The traveling salesman problem with cumulative costs. *Networks*, 23(2) :81–91, 1993.

[18] J. Blazewicz, R. Burkard, and G. Finke, G. and. Woeginger. Vehicle scheduling in two-cycle flexible manufacturing systems. *Mathematics and Computer Modeling*, 20 :19–31, 1994.

[19] A. Blum, P. Chalasani, D. Coppersmith, B. Pulleyblank, P. Raghavan, and M. Sudan. The minimum latency problem. In *Proceedings of the twenty-sixth annual ACM Symposium on Theory of Computing (STOC '94)*, pages 163–171, 1994.

[20] A. M. Campbell, D. Vandenbussche, and W. Hermann. Routing for relief efforts. *Transportation Science*, 42(2) :127–145, 2008.

[21] M. S. Chang, S. R. Chen, and C. F. Hsueh. Real-time vehicle routing problem with time windows and simultaneous delivery/pickup demands. *Journal of the Eastern Asia Society for Transportation Studies*, 5 :2273–2286, 2003.

[22] K. Chaudhuri, B. Godfrey, S. Rao, and K. Talwar. Paths, trees, and minimum latency tours. In *FOCS '03 : Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*, page 36, Washington, DC, USA, 2003. IEEE Computer Society.

[23] N. Christofides, A. Mingozzi, and P. Toth. The vehicle routing problem. In N. Christofides, A. Mingozzi, P. Toth, and L. Sandi, editors, *Combinatorial Optimization*, pages 315–338. Wiley, Wichester, 1979.

[24] N. Christofides, A. Mingozzi, and P. Toth. Exact algorithms for the vehicle routing problem based on spanning tree and shortest path relaxation. *Mathematical Programming*, 20 :255–282, 1981.

[25] V. Chvátal. Edmonds polytopes and weakly hamiltonian graphs. *Mathematical Programming*, 5 :29–40, 1973.

[26] G. Clarke and J. Wright. Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12 :568–581, 1964.

[27] W. Cook. Concorde tsp solver. `http ://www.tsp.gatech.edu/concorde/index.html`, 1995.

[28] R. Cordone and R. Wolfler Calvo. Note on time window constraints in routing problems. Internal report 96.005, Dipartimento di Electronica e Informazione, Politechnico di Milano, Milan, Italy, 1996.

[29] G. Cornuejols and F. Harche. Polyhedral study of the capacitated vehicle routing problem. *Mathematical Programming*, 60 :21–52, 1993.

[30] P.I.. Cowling and R. Keuthen. Embedded local search approaches for routing optimization. *Computers and Operations Research*, 32 :465–490, 2005.

[31] G. Dantzig, R. Fulkerson, and S. Johnson. Solution of a large scale traveling salesman problem. *Journal of the Operations Research Society of America*, 2 :393–410, 1954.

[32] M. J. D. De brey and A. Volgenant. Well-solved cases of the 2-peripatetic salesman problem. *Optimization : A Journal of Mathematical Programming and Operations Research*, 39(3) :275–293, 1997.

[33] J.B.J.M De Kort. Lower bounds for symmetric k-peripatetic salesman problems. *Optimization*, 22 :113–122, 1991.

[34] J.B.J.M. De Kort. Bounds for the symmetric 2-peripatetic salesman problem. *Optimization : A Journal of Mathematical Programming and Operations Research*, 23(4) :357–367, 1992.

[35] J.B.J.M. De Kort. A branch and bound algorithm for symmetric 2-peripatetic salesman problems. *European Journal of Operational Research*, 70 :229–243, 1994.

[36] M. Desrochers, J. Desrosiers, and M. Solomon. A new optimization algorithm for the vehicle routing problem with time windows. *Operations Reseach*, 40(2) :342–354, 1992.

[37] J.J. Dongarra. Performance of various computers using standard linear equations software. Technical report cs-89-85, University of Tennessee, March 2009.

[38] M. Dorigo, V. Maniezzo, and A. Colorni. The ant system : Optimiation by a colony of cooperating agents. *IEEE Transactions on Systems, Man and Cybernetics-Part B*, 26(1) :29–41, 1996.

[39] O. Du Merle, D. Villeneuve, J. Desrosiers, and P. Hansen. Stabilized column generation. discrete mathematics. *ORSA Journal on Computing*, 194 :229–237, 1999.

[40] E. Duchenne, G. Laporte, and F. Semet. Branch and cut algorithms for the undirected $m$-peripatetic salesman problem. *European Journal of Operational Research*, 162 :700–712, 2005.

[41] E. Duchenne, G. Laporte, and F. Semet. The undirected $m$-peripatetic salesman problem : Polyhedral results and new algorithms. *Operations Research*, 55(5) :949–965, 2007.

[42] E. Duchenne, G. Laporte, and F. Semet. Heuristiques pour le problème du vendeur $m$-péripatétique. *RAIRO - Operations Research*, 43(1) :13–26, 2009.

[43] R. Eberhart and J. Kennedy. A new optimizer using particle swarm theory. In *Micro Machine and Human Science, 1995. MHS '95., Proceedings of the Sixth International Symposium on*, pages 39–43, 1995.

[44] J. Edmonds. Maximum matching and a polyhedron with 0-1 vertices. *J. Res. Nat. Bur. Standards*, 69B :125–130, 1965.

[45] J. Edmonds. Paths, trees, and flowers. *Canadian Journal of Mathematics*, 17 :449–467, 1965.

[46] R. Fahrion and M. Wrede. On a principle of chain-exchange for vehicle-routeing problems(1-vrp). *Journal of the Operational Research Society*, 41(9) :821–827, 1990.

[47] A. Federgruen and P. Zipkin. A combined vehicle routing and inventory allocation problem. *Operations Research*, 32(5) :1019–1037, 1984.

[48] D. Feillet, P. Dejax, and M. Gendreau. Traveling salesman problems with profits. *Transportation Science*, 39(2) :188–205, 2005.

[49] T. Feo, M. Resende, and S. Smith. A greedy randomized adaptive search procedure for maximum independent set. *Operations Research*, 42 :860–878, 1994.

[50] M. Fischetti, G. Laporte, and S. Martello. The delivery man problem and cumulative matroids. *Operations Research*, 6 :1055–1064, 1993.

[51] M. Fischetti, P. Toth, and D. Vigo. A branch-and-bound algorithm for the capacitated vehicle routing problem on directed graphs. *Operations Research*, 42(5) :846–859, 1994.

[52] R. Fukasawa, H. Longo, J. Lysgaard, M. Poggi de Aragão, M. Reis, E. Uchoa, and R.F. Werneck. Robust branch-and-cut-and-price for the capacitated vehicle routing problem. *Mathematical Programming Series A*, 106 :491–511, 2006.

[53] M. R. Garey and D. S. Johnson. *Computers and Intractability : A Guide to the Theory of NP-Completeness (Series of Books in the Mathematical Sciences)*. W. H. Freeman & Co Ltd, 1979.

[54] B. E. Gillett and L. R. Miller. A heuristic algorithm for the vehicle dispatch problem. *Operations Research*, 22 :340–349, 1974.

[55] F. Glover and M. Laguna. Tabu search. In C. R. Reeves, editor, *Modern heuristic techniques for combinatorial problems*, pages 70–150. Blackwell, Oxford, U.K., 1993.

[56] M. Goemans and J. Kleinberg. An improved approximation ratio for the minimum latency problem. In *SODA '96 : Proceedings of the seventh annual ACM-SIAM symposium on Discrete algorithms*, pages 152–158. Society for Industrial and Applied Mathematics, 1996.

[57] R. Gomory. Outline of an algorithm for integer solutions to linear programs. *Bulletin of the American Mathematical Society*, 64 :275–278, 1958.

[58] Q. Gong and R. Batta. Allocation and reallocation of ambulances to casualty clusters in a disaster relief operation. *IIE Transactions*, 39(1) :27–39, 2007.

[59] L. Gouveia. A result on projection for the vehicle routing problem. *European Journal of Operational Research*, 85 :610–624, 1995.

[60] M. Grötschel and M. W. Padberg. On the symmetric travelling salesman problem i : inequalities. *Mathematical Programming*, 16 :281–302, 1979.

[61] M. Held and R. M. Karp. The traveling salesman problem and the minimum spanning trees. *Operations Research*, 18(6) :1138–1162, 1970.

[62] K. Helsgaun. Lkh tsp heuristic. `http ://www.akira.ruc.dk/ keld/research/LKH/`, 1998.

[63] J. H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.

[64] C.-H. Hsueh, H.-K. Chen, and H.-W. Chou. Dynamic vehicle routing for relief logistics in natural disasters. In T. Caric and H. Gold, editors, *Vehicle Routing Problem*, pages 71–84. IN-TECH, Croatia, 2008. ISBN 978-953-7619-09-1.

[65] Fritz Institute. logistics and the effective delivery of humanitarian relief. Technical report, Fritz Institure, 2005.

[66] A. Itai and M. Rodeh. Finding a minimum circuit in a graph. *SIAM Journal of Computing*, 7(4) :413–423, 1978.

[67] B. Kallehauge, J. Larsen, and O. B. G. Madsen. Lagrangean duality applied to the vehicle routing problem with time windows. *Computers and Operations Research*, 33 :1464–1487, 2006.

[68] I. Kara, B. K. Yetis, and K. Yetis. Energy minimizing vehicle routing problem. In A. Dress and B. Zhu, editors, *Combinatorial Optimization and Applications*, pages 62–71. Springer, Berlin, Heidelberg, 2007.

[69] C. P. Keller. Algorithms to solve the orienteering problem : A comparison. *European Journal of Operational Research*, 41(2) :224–231, 1989.

[70] G. Kindervater and M. Savelsbergh. Vehicle routing : Handling edge exchanges. In J. Lenstra and E. Aarts, editors, *Local Search in Combinatorial Optimization*, pages 337–360. Wiley, Chichester, 1997.

[71] S. C. D. Kirkpatrick, J. Gelatt, and M. Vecchi. Optimization by simulated annealing. *Science*, 220(4598) :671–680, 1983.

[72] J. Krarup. The peripatetic salesman and some related unsolved problems. In B. Roy, editor, *Combinatorial Programming Methods and Applications*, pages 173–178. Reidel, Dordrecht, 1975.

[73] J. B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, 7(1) :48–50, 1956.

[74] S. P. Ladany and A. Mehrez. Optimal routing of a single vehicle with loading and unloading constraints. *Transportation Planning and Technology*, 8(4) :301–306, 1984.

[75] A. H. Land and A. G. Doig. An automatic method of solving discrete programming problems. *Econometrica*, 28(3) :497–520, 1960.

[76] G. Laporte and Y. Norbert. A branch and bound algorithm for the capacitated vehicle routing problem. *Operations Research Spektrum*, 5 :77–85, 1983.

[77] E. K. Lauritzen. Emergency construction waste management. *Safety Science*, 30(1-2) :45–53, 1998.

[78] E. Lawler, J. Lenstra, A. R. Kan, and D. Shmoys. *The traveling Salesman Problem.* John Wiley & sons, 1985.

[79] A.N. Letchford, R.W. Eglese, and J. Lysgaard. Multistars, partial multistars and the capacitated vehicle routing problem. *Mathematical Programming*, 94 :21–40, 2002.

[80] A.N. Letchford, J. Lysgaard, and R.W. Eglese. A branch-and-cut algorithm for the capacitated open vehicle routing problem. *Journal of the Operational Research Society*, 58 :1642–1651, 2007.

[81] C.M.R.R. Lima, M.C. Goldbarg, and E.F.G. Goldbarg. A memetic algorithm for the heterogeneous fleet vehicle routing problem. *Electronic Notes in Discrete Mathematics*, 18 :171–176, 2004.

[82] S. Lin and B. Kernighan. An effective heuristical algorithm for the traveling salesman problem. *Operations Research*, 21 :498–516, 1973.

[83] L. Lindner-Dutton, R. Batta, and M. H. Karwan. Equitable sequencing of a given set of hazardous materials shipments. *Transportation Science*, 25(2) :124–137, 1991.

[84] J. Lysgaard. A package of separation routines for the capacitated vehicle routing problem. `http ://www.hha.dk/ lys/`, 2003.

[85] R. Martí, M. Laguna, and F. Glover. Principles of scatter search. *European Journal of Operational Research*, 169(2) :359–372, 2006.

[86] T. McCormick, M.R. Rao, and G. Rinaldi. Easy and difficult objective functions for max cut. *Mathematical Programming*, 94 :459–466, 2003.

[87] C.E. Miller, A.W. Tucker, and R.A. Zemlin. Integer programming formulations and traveling salesman problem. *Journal of the ACM*, 7 :326–329, 1960.

[88] D.L. Miller and J. Wright. A matching based exact algorithm for capacitated vehicle routing problems. *ORSA Journal on Computing*, 7 :298–320, 1995.

[89] M. Minoux. *Programmation mathématique : théorie et algorithmes*. Lavoisier, 2ème edition, 2008.

[90] N. Mladenović and P. Hansen. Variable neighbourhood decomposition search. *Computers and Operations Research*, 24 :1097–1100, 1997.

[91] A. S. Mohaymany, Hosseini, and H. M. M. Habibi. Obtaining the emergency transportation network for rescue and relief activities in large cities based on the life loss mitigation criteria. *Technical Council on Lifeline Earthquake Engineering Monograph*, 25 :231–240, 2003.

[92] P. Moscato. On evolution, search, optimization, genetic algorithms, martial arts : towards memetic algorithms. Caltech concurrent computation program c3p report 826, California Institute of Technology, 1989.

[93] M. Mourgaya and F. Vanderbeck. Problème de tournées de véhicules multipériodiques : classification et heuristique pour la planification tactique. *RAIRO - Operations Research*, 40(2) :169–194, 2006.

[94] R.J. Mullen, D. Monekosso, S. Barman, and P. Remagnino. A review of ant algorithms. *Expert Systems with Applications*, 36(6) :9608–9617, 2009.

[95] D. Naddef and S. Thienel. Efficient separation routines for the symmetric traveling salesman problem I : general tools and comb separation. *Mathematical Programming*, 92 :237–255, 2002.

[96] D. Naddef and S. Thienel. Efficient separation routines for the symmetric traveling salesman problem II : separating multi handle inequalities. *Mathematical Programming*, 92 :257–283, 2002.

[97] S.U. Ngueveu, C. Prins, and R. Wolfler Calvo. The $m$-peripatetic vehicle routing problem. In *Proceedings of EURO XXII*, page 209, Prague, République Tchèque, 2007. 22nd European Conference on Operations Research 2007.

[98] S.U. Ngueveu, C. Prins, and R. Wolfler Calvo. The $m$-peripatetic vehicle routing problem. In *Book of Abstracts*, page 93, Porto, Portugal, 2007. OPTIMIZATION 2007.

[99] S.U. Ngueveu, C. Prins, and R. Wolfler Calvo. Bornes supérieures et inférieures pour le problème de tournées de véhicules $m$-péripatétiques. In *Actes de la 7ème conférence internationale de Modélisation et Simulation*, volume 3, pages 1617–1625, Paris, France, 2008. Modélisation, Optimisation et Simulation des Systèmes : Communication, Coopération et Coordination.

[100] S.U. Ngueveu, C. Prins, and R. Wolfler Calvo. A hybrid tabu search for the $m$-peripatetic vehicle routing problem ($m$-pvrp). In *Proceedings of Matheuristics 2008 Conference*, Bertinoro, Italy, 2008. Second International Workshop on Model Based Metaheuristics.

[101] S.U. Ngueveu, C. Prins, and R. Wolfler Calvo. A memetic algorithm for the cumulative capacitated vehicle routing problem. In *Proceedings of EUME08 Conference*, Troyes, France, 2008. The European chapter on metaheuristics - EURO Working Group. http ://webhost.ua.ac.be/eume/workshops/eume08/papers/papers.zip.

[102] S.U. Ngueveu, C. Prins, and R. Wolfler Calvo. An effective evolutionary algorithm for the cumulative capacitated vehicle routing problem. In M. Giacobini *et al.*, editor, *EvoWorkshops 2009, LNCS 5484*, pages 778–787. Springer, 2009.

[103] S.U. Ngueveu, C. Prins, and R. Wolfler Calvo. An effective memetic algorithm for the cumulative capacitated vehicle routing problem. *Computers and Operations Research*, In Press, Corrected Proof, 2009.

[104] S.U. Ngueveu, C. Prins, and R. Wolfler Calvo. Le problème de tournées de véhicules $m$-péripatétiques : Résumé des travaux. In *Livre des résumés ROADEF09*, pages 358–359, Nancy, France, 2009. Société Française de Recherche Opérationnelle et d'Aide à la Décision.

[105] S.U. Ngueveu, C. Prins, and R. Wolfler Calvo. A hybrid tabu search for the $m$-peripatetic vehicle routing problem. In T. Stutzle and S. Voss, editors, *Matheuristics : Hybridizing Metaheuristics and Mathematical Programming*, Annals of Information Systems Series, 10, pages 253–264. Springer, 2010.

[106] S.U. Ngueveu, C. Prins, and R. Wolfler Calvo. Solving the $m$-peripatetic vehicle routing problem. *4OR*, en cours de révision.

[107] I. Or. *Traveling salesman-type combinatorial optimization problems and their relation to the logistics of regional blood banking*. PhD thesis, Northwestern University, Evanston, IL, 1976.

[108] L. Özdamar, E. Ekinci, and B. Küçükyazici. Emergency logistics planning in natural disasters. *Annals of Operations Research*, 129 :217–45, 2004.

[109] M. Padberg and G. Rinaldi. A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM Review*, 33 :60–100, 1991.

[110] G. Pesant, M. Gendreau, J.-Y. Potvin, and J.-M. Rousseau. On the flexibility of constraint programming models : From single to multiple time windows for the traveling salesman problem. *European Journal of Operational Research*, 117 :253–263, 1999.

[111] J.C. Picard and H.D. Ratliff. Minimum cuts and related problems. *Networks*, 5 :357–370, 1975.

[112] C. Prins. A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers and Operations Research*, 31 :1985–2002, 2004.

[113] C. Rego. A subpath ejection method for the vehicle routing problem. *Management Science*, 44(10) :1447–1459, 1998.

[114] M. Reimann, K. Doerner, and R.F. Hartl. D-ants : Savings based ants divide and conquer the vehicle routing problem. *Computers and Operations Research*, 31(4) :563–591, 2004.

[115] G. Reinelt. `http ://www.iwr.uniheidelberg.de/groups/comopt/software/TSPLIB95/`.

[116] J. Renaud, F.F. Boctor, and J. Ouenniche. A heuristic for the pickup and delivery traveling salesman problem. *Computers and Operations Research*, 27(9) :905–916, 2000.

[117] J. Renaud, G. Laporte, and F.F. Boctor. A tabu search heuristic for the multi-depot vehicle routing problem. *Computers and Operations Research*, 23(3) :229–235, 1996.

[118] J. Roskind and R. E. Tarjan. A note on finding minimum-cost edge disjoint spanning trees. *Mathematics of Operations Research*, 10 :701–708, 1985.

[119] S. Sahni and T. Gonzalez. P-complete problems and approximate solutions. In *15th Annual Symposium on Switching and Automata Theory*, pages 28–32. IEEE, Oct. 1974.

[120] A. Salehipour, K. Sörensen, P. Goos, and O. Bräysy. An efficient GRASP+VND metaheuristic for the traveling repairman problem. Working paper, University of Antwerp, Faculty of Applied Economics, May 2008.

[121] J. Sarubbi, H. Luna, and G. Miranda. Minimum latency problem as a shortest path problem with side constraints. In *XIV Latin Ibero-American Congress on Operations Research (CLAIO)*, 2008.

[122] M. W. P. Savelsbergh and M. Sol. The general pickup and delivery problem. *Transportation Science*, 29(1) :17–29, 1995.

[123] D. Simchi-Levi and O. Berman. Minimizing the total flow time of n jobs on a network. *IIE Transactions*, 23(3) :236–244, 1991.

[124] E.D. Taillard. Parallel iterative search methods for vehicle routing problems. Technical Report G-2002-15, Les Cahiers du GERAD, Canada, 2002.

[125] P. Toth and D. Vigo. Models, relaxations and exact approaches for the capacitated vehicle routing problem. *Discrete Applied Mathematics*, 123(1-3) :487–512, 2002.

[126] P. Toth and D. Vigo. *The vehicle routing problem*. SIAM, Philadelphia, 2002.

[127] P. Toth and D. Vigo. The granular tabu search (and its application to the vehicle routing problem). *INFORMS Journal on Computing*, 15(4) :333–346, 2003.

[128] C. Tsai, C. Tsai, and C. Tseng. A new hybrid heuristic approach for solving large traveling salesman problem. *InformationSciences*, 166 :67–81, 2004.

[129] G. Ulusoy. The fleet size and mix problem for capacitated arc routing. *European Journal of Operational Research*, 22 :329–337, 1985.

[130] LN. Van Wassenhove. Humanitarian aid logistics : supply chain management in high gear. *Journal of the Operational Research Society*, 57 :475–489, 2006.

[131] E. Von Boventer. The relationship between transportation costs and location rent in transportation problems. *Journal of Regional Science*, 3 :27–40, 1961.

[132] R. Wolfler Calvo and R. Cordone. A heuristic approach to the overnight security service problem. *Computers and Operations Research*, 30 :1269–1287, 2003.

[133] B.Y. Wu, Z.-N. Huang, and F.-J. Zhan. Exact algorithms for the minimum latency problem. *Information Processing Letters*, 92(6) :303–309, 2004.

[134] W. Yi and L. Özdamar. A dynamic logistics coordination model for evacuation and support in disaster response activities. *European Journal of Operational Research*, 179(3) :1177–1193, 2007.

[135] R. Zamani and S.K. Lau. Embedding learning capability in lagrangean relaxation an application to the travelling salesman problem. *European Journal of Operational Research*, accepted, to be published. `http ://10.1016/j.ejor.2009.02.008`.

[136] Q. Zhu, L. Qian, Y. Li, and S. Zhu. An improved particle swam optimization algorithm for vehicle routing problem with time windows. In *IEEE Congress on Evolutionary Computation(CEC)*, pages 1386–1390, 2006.