POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Algorithmes de génération de colonnes avec plans coupants pour des problèmes de tournées de véhicules à deux échelons

TAYEB MHAMEDI
Département de mathématiques et de génie industriel

Thèse présentée en vue de l'obtention du diplôme de *Philosophiæ Doctor*
Mathématiques

Juillet 2025

© Tayeb Mhamedi, 2025.

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Cette thèse intitulée :

Algorithmes de génération de colonnes avec plans coupants pour des problèmes de tournées de véhicules à deux échelons

présentée par **Tayeb MHAMEDI**
en vue de l'obtention du diplôme de *Philosophiæ Doctor*
a été dûment acceptée par le jury d'examen constitué de :

Michel GENDREAU, président
Guy DESAULNIERS, membre et directeur de recherche
Marilène CHERKESLY, membre et codirectrice de recherche
Yossiri ADULYASAK, membre
Aurélien FROGER, membre externe

DÉDICACE

To my sister Fadoua,
you are a beacon of light in my life.
Love you to the moon and back.

To my mother Ihssan, my father Ali and my aunt Mariam,
having you as parents is truely priceless.
I love you.

To my father Ali, my best friend, my guy.
You are my source of inspiration.
I love you.

To my nephew Ayman,
I am extremely proud of you.
I love you.

REMERCIEMENTS

On a probablement tous déjà entendu l'expression "*It takes a village*" ou une forme équivalente de celle-ci. L'expression est souvent utilisée pour souligner le rôle important qu'une communauté joue dans l'éducation d'un enfant. Quand j'entends cette expression aujourd'hui, elle me fait penser à toutes les personnes qui m'ont aidé le long de mon parcours doctoral. Bien que cela soit difficile, je vais tenter de trouver les justes mots pour exprimer mon appréciation et ma profonde gratitude envers elles. *Allons y !*

J'aimerais tout d'abord remercier mes directeurs de recherche Guy Desaulniers et Marilène Cherkesly. Merci infiniment de m'avoir supporté le long de ma thèse. Merci d'avoir cru en moi, d'avoir toujours été disponibles, et de m'avoir tant appris. Je m'estime extrêmement chanceux d'avoir été dirigé par vous dans mes travaux de recherche. En particulier, j'aimerais remercier M. Desaulniers pour ses conseils, son écoute et sa confiance. Merci pour tout !

Un grand merci à François Lessard et Benoît Rochefort pour leur disponibilité et soutien technique. J'ai tellement appris avec vous. Merci beaucoup !

Je souhaiterais aussi exprimer ma gratitude aux membres de mon jury : Michel Gendreau, Yossiri Adulyasak et Aurélien Froger. Merci d'avoir accepté d'être sur mon jury de thèse.

J'ai eu l'occasion de croiser des personnes spéciales durant mon parcours au Gerad. Je suis chanceux aujourd'hui de les compter parmi mes amis les plus proches. Un merci spécial s'en va à Rachid et Adil. Merci d'avoir été là depuis le début. Merci pour les fous rires, les discussions, et de m'avoir épaulé durant les périodes difficiles. Merci également à Mouad et Bani. Vous êtes tous formidables !

À mes amis de longues date : Mehdi, Hamza, Adil, Touhami, Amine a.k.a. Mikey, Marouane, Saad, Mouad et Abdessamad. Merci les gars pour tous les bons moments.

À Laurence, Maryse et Anne Marie : merci d'avoir fait de l'année passée ma meilleure jusqu'à date à Montréal ! À Laurence en particulier, merci pour les barres de rires, de m'avoir appris de nouveaux mots rigolos, et de m'avoir introduit aux pattes d'ours ! Je vous apprécie toutes vraiment beaucoup !

Finalement, j'aimerais remercier les personnes qui me sont les plus chères : ma mère Ihssan, mon père Ali, ma soeur Fadoua, ma tante Mariam, mon neveu Ayman et mon beau-frère Phil. Merci de m'avoir toujours encouragé au fil des années et de m'avoir aidé à me relever après chaque trébuchement. Je vous aime tous très fort ! Merci également à mes tantes, oncles et cousines d'avoir toujours trouver les bons mots. Je vous aime tous.

RÉSUMÉ

L'étude de la distribution de marchandises en zones urbaines a suscité beaucoup d'intérêt ces dernières années au sein de la communauté de la recherche opérationnelle. En effet, une planification minutieuse de ces activités contribue à réduire les charges sociales et environnementales placées sur les centres-villes tout en permettant de réaliser des économies d'échelle. Un moyen efficace pour atteindre ces objectifs est de diviser les chaînes d'approvisionnement en deux échelons, ce qui permet d'utiliser de gros véhicules en dehors du centre-ville et de plus petits dans le centre-ville.

Cette thèse porte sur le développement d'approches de résolution exactes pour trois variantes du problème de tournées de véhicules à deux échelons : le problème de tournées de véhicules à deux échelons avec fenêtres de temps (2E-VRPTW) et deux de ses extensions, à savoir le problème de tournées de véhicules à deux échelons avec fenêtres de temps et commodités multiples (MC-2E-VRPTW) et le problème de tournées de véhicules à deux échelons avec fenêtres de temps, collectes et livraisons simultanées (2E-VRPTW-SPD). Dans ces problèmes, les véhicules de premier échelon assurent le transport des marchandises des dépôts centraux vers des dépôts intermédiaires, appelés satellites. Les véhicules de second échelon, plus petits que ceux de permier échelon et affectés à des routes commençant et terminant à un même satellite, desservent les clients à l'intérieur de leurs fenêtres de temps. En outre, les satellites n'ont pas de capacité d'entreposage et ne permettent pas la consolidation de fret, ce qui nécessite une synchronisation exacte des opérations (c'est-à-dire que les véhicules de premier et de second échelon doivent être présents simultanément aux satellites afin de permettre le transfert de marchandises). L'objectif est de déterminer un ensemble de routes de premier et de second échelon de moindres coûts, de sorte que chaque client soit desservi par un seul véhicule de second échelon pendant sa fenêtre de temps et que les contraintes liées à la synchronisation des opérations au niveau des satellites soient satisfaites.

Pour résoudre le 2E-VRPTW, nous développons un algorithme exact de génération de colonnes avec plans coupants (*branch-price-and-cut*, BPC) basé sur une formulation impliquant un grand nombre de routes de second échelon. Les sous-problèmes sont résolus à l'aide d'un algorithme d'étiquetage ad hoc qui génère des routes de second échelon, tout en respectant le schéma de synchronisation considéré aux satellites. Nous décrivons les ajustements non-triviaux à apporter à l'algorithme d'étiquetage afin de générer des variables associées à des routes de second échelon alimentées par des routes de premier échelon uniques. Pour accélérer le processus de résolution, nous proposons des inégalités duales, appelées inégalités de

transfert, qui exploitent des informations spécifiques au problème afin de couper une partie de l'espace dual et de réduire le nombre d'itérations de génération de colonnes. L'algorithme BPC développé est capable de résoudre à l'optimalité, en trois heures de temps de calcul, des instances comportant jusqu'à trois dépôts, cinq satellites et 100 clients. De plus, de nouvelles solutions optimales sont trouvées pour 41 instances précédemment non résolues parmi 180 instances testées.

Le MC-2E-VRPTW est une extension du 2E-VRPTW qui prend en compte un cas de figure rencontré en pratique où la demande de chaque client n'est disponible qu'à partir d'un dépôt central spécifique. Dans ce problème, les contraintes de synchronisation impliquent que la demande de chaque client doit être satisfaite par exactement un véhicule de premier échelon et un véhicule de second échelon. Cependant, il est possible qu'un véhicule de second échelon reçoive sa marchandise à partir de plusieurs véhicules de premier échelon. Pour résoudre ce problème, nous proposons un algorithme BPC adapté dans lequel la synchronisation des opérations est prise en compte au niveau des sous-problèmes. Pour résoudre ceux-ci, nous développons un algorithme d'étiquetage ad hoc générant des routes de second échelon tout en spécifiant la route de premier échelon ayant fourni la demande de chaque client visité. Nous proposons une règle de branchement adaptée au contexte de commodités multiples ainsi qu'une procédure de récupération assurant la faisabilité des solutions entières lorsque l'on considère les inégalités de transfert. Des instances comportant jusqu'à trois dépôts, cinq satellites et 100 dépôts sont résolues à l'optimalité en trois heures de temps de calcul.

Enfin, le 2E-VRPTW-SPD aborde l'aspect des collectes et livraisons simultanées dans un contexte de deux échelons. Dans ce problème, chaque client a une demande de livraison, de collecte, ou les deux. De plus, la demande totale livrée (collectée) par un véhicule de second échelon est fournie (collectée) par un seul véhicule de premier échelon. Nous proposons un algorithme exact de type BPC pour ce problème. Nous considérons un sous-problème par satellite que nous résolvons via un algorithme d'étiquetage spécialisé. Chaque variable générée est associée à une route de second échelon et à une paire de routes de premier échelon. Nous appliquons des inégalités valides connues afin de renforcer les bornes inférieures obtenues et utilisons des inégalités de transfert pour accélérer la convergence de la génération de colonnes. Des instances comportant jusqu'à deux dépôts, trois satellites et 100 clients sont résolues à l'optimalité en trois heures de temps de calcul.

ABSTRACT

The study of freight distribution in urban areas have garnered a lot of interest during recent years within the operations research community. In fact, careful planning for these activities contribute to lessening social and environmental burdens placed on city centers while enabling economies of scale. An efficient and common way for achieving these goals is to divide supply chains into two echelons, allowing to use large vehicles outside the city center and smaller ones in the city center.

This thesis addresses the development of exact solution approaches for three variants of the two-echelon vehicle routing problem : the two-echelon vehicle routing problem with time windows (2E-VRPTW) and two of its extensions, namely the multi-commodity two-echelon vehicle routing problem with time windows (MC-2E-VRPTW) and the two-echelon vehicle routing problem with time windows and simultaneous pickup and delivery (2E-VRPTW-SPD). In these problems, first-echelon vehicles transport goods from depots to intermediate depots, called satellites. Second-echelon vehicles, smaller than the first-echelon vehicles and assigned to routes starting and ending at a same satellite, service customers within their allocated time windows. Moreover, satellites have no storage capacity and do not allow for freight consolidation, which requires exact operation synchronization (i.e., first- and second-echelon vehicles need to be present simultaneously at the satellites to allow load transfer). The aim is to determine a set of least-cost and capacity-feasible first- and second-echelon routes such that each customer is serviced by a single second-echelon vehicle during its time window and that constraints related to operation synchronization at the satellites are satisfied.

To solve the 2E-VRPTW, we develop an exact branch-price-and-cut (BPC) algorithm based on a route-based formulation involving a large number of second-echelon routes. The column generation subproblems are solved using an ad hoc labeling algorithm that generates second-echelon routes, while abiding by the considered synchronization scheme. We describe non-trivial adjustments to the labeling algorithm in order to generate variables associated with second-echelon routes supplied by unique first-echelon routes. To speed up the solution process, we propose dual inequalities, called transfer inequalities, that leverage problem-specific information to cut part of the dual space and decrease the number of column generation iterations. The developed BPC algorithm is able to solve to optimality, within a three-hour computational time limit, instances with up to three depots, five satellites and 100 customers. It also computes new optimal solutions for 41 previously unsolved instances out of 180 tested instances from the literature.

The MC-2E-VRPTW is an extension of the 2E-VRPTW that considers a commomly encountered situation in practice where the demand of each customer is available solely at one specific depot. In this problem, synchronization constraints imply that each customer's demand is supplied by exactly one first-echelon vehicle and one second-echelon vehicle. However, the total demand of any given second-echelon route can be supplied by multiple first-echelon vehicles. To tackle this problem, we propose a tailored BPC algorithm where operation synchronization is considered at the subproblem level. To solve the column generation subproblems, we develop an ad hoc labeling algorithm that generates second-echelon routes while determining each visited customer's supplying-first-echelon route. The devised algorithm proposes a branching rule adapted to the multi-commodity context and a recovery procedure that ensures the feasibility of integer solutions when considering transfer inequalities. Instances with up to three depots, five satellites and 100 depots are solved to optimality within a three-hour computational time limit.

Finally, the 2E-VRPTW-SPD addresses the aspect of simultaneous pickup and delivery in a two-echelon setting. In this problem, each customer has a delivery demand, a pickup demand, or both. Moreover, the total demand delivered (picked up) by a second-echelon vehicle is supplied (collected) by a single first-echelon vehicle. We propose an exact BPC algorithm to solve the problem. We consider one subproblem per satellite that we solve using a specialized labeling algorithm. Each generated variable is associated with a second-echelon route and a pair of first-echelon routes. We use transfer inequalities to speed up column generation and apply known valid inequalities to improve the lower bounds. Instances with up to two depots, three satellites and 100 customers are solved to optimality within three hours of computational time.

TABLE DES MATIÈRES

DÉDICACE iii

REMERCIEMENTS iv

RÉSUMÉ v

ABSTRACT vii

TABLE DES MATIÈRES ix

LISTE DES TABLEAUX xii

LISTE DES FIGURES xiv

LISTE DES SIGLES ET ABRÉVIATIONS xv

LISTE DES ANNEXES xvii

CHAPITRE 1 INTRODUCTION 1

CHAPITRE 2 REVUE DE LITTÉRATURE 5
- 2.1 Problèmes de tournées de véhicules à deux échelons 5
- 2.2 Problèmes de tournées de véhicules à deux échelons avec fenêtres de temps . 6
- 2.3 Problèmes de tournées de véhicules à deux échelons avec multiples commodités 8
- 2.4 Problèmes de tournées de véhicules à deux échelons avec collectes et livraisons simultanées 9
- 2.5 Sommaire 9

CHAPITRE 3 ORGANISATION DE LA THÈSE 12

CHAPITRE 4 ARTICLE 1 : A BRANCH-PRICE-AND-CUT ALGORITHM FOR THE TWO-ECHELON VEHICLE ROUTING PROBLEM WITH TIME WINDOWS 15
- 4.1 Introduction 15
- 4.2 Mathematical formulation 18
- 4.3 Branch-price-and-cut algorithm 23
  - 4.3.1 Pricing problems and labeling algorithm 23
  - 4.3.2 Stabilization by dual inequalities 28

4.3.3 Valid inequalities 33
4.3.4 Branching 35
4.3.5 Alternative model and algorithm 36
4.4 Computational experiments 37
4.4.1 Characteristics of the instances 37
4.4.2 Computational results 38
4.4.3 Sensitivity analysis 41
4.4.4 Managerial insights 46
4.5 Conclusions 48

CHAPITRE 5 ARTICLE 2 : A BRANCH-PRICE-AND-CUT ALGORITHM FOR THE MULTI-COMMODITY TWO-ECHELON VEHICLE ROUTING PROBLEM WITH TIME WINDOWS 50
5.1 Introduction 50
5.2 Mathematical formulation 53
5.3 Branch-price-and-cut algorithm 56
5.3.1 Subproblems 57
5.3.2 Stabilization by dual inequalities 61
5.3.3 Valid inequalities 63
5.3.4 Branching 64
5.4 Computational experiments 66
5.4.1 MC-2E-VRPTW test instances 66
5.4.2 Main computational results 67
5.4.3 Sensitivity analysis 70
5.5 Conclusions 72

CHAPITRE 6 UN ALGORITHME EXACT POUR LE PROBLÈME DE TOURNÉES DE VÉHICULES À DEUX ÉCHELONS AVEC FENÊTRES DE TEMPS, COLLECTES ET LIVRAISONS SIMULTANÉES 74
6.1 Formulation mathématique 74
6.2 Algorithme BPC pour le 2E-VRPTW-SPD 77
6.2.1 Sous-problèmes 77
6.2.2 Stabilisation par inégalités duales 82
6.2.3 Inégalités valides 86
6.2.4 Branchement 88
6.3 Expériences numériques 90
6.3.1 Description des instances 91

6.3.2 Résultats numériques 91
6.3.3 Analyse de sensibilité 94
6.3.4 Conclusion 97

CHAPITRE 7 DISCUSSION GÉNÉRALE 98

CHAPITRE 8 CONCLUSION 101
8.1 Synthèse des travaux 101
8.2 Limitations et améliorations futures 102

RÉFÉRENCES 106

ANNEXES 111
A.1 Detailed computational results 111
A.2 Detailed computational results for the sensitivity analysis 118
A.3 Detailed results for managerial insights 130
B.1 Proof of Proposition 5.1 140
B.2 Detailed example of solution feasibility with transfer inequalities 141
B.3 Adjustments to the labeling algorithm 142
B.4 Detailed computational results 143
C.1 Résultats numériques détaillés 151

LISTE DES TABLEAUX

Tableau 2.1 Travaux proposant des algorithmes exacts pour des variantes du 2E-CVRP . . . 11
Table 4.1 Summarized computational results . . . 40
Table 4.2 Summarized statistics on the algorithm execution . . . 40
Table 4.3 Summarized solution statistics . . . 42
Table 4.4 Impact of omitting families of inequalities . . . 44
Table 4.5 Impact of omitting branching rules . . . 46
Table 4.6 Impact on costs of considering only back-and-forth first-echelon routes 47
Table 4.7 Impact on computational performance of considering only back-and-forth first-echelon routes . . . 48
Table 5.1 Number of instances per size and category . . . 67
Table 5.2 Summary of the computational results . . . 68
Table 5.3 Comparison with Dellaert et al. (2021) . . . 69
Table 5.4 Summarized statistics on the algorithm execution . . . 69
Table 5.5 Summarized optimal solution statistics . . . 70
Table 5.6 Impact of omitting algorithmic components . . . 72
Tableau 6.1 Récapitulatif des résultats numériques . . . 92
Tableau 6.2 Statistiques résumées sur l'exécution de l'algorithme . . . 94
Tableau 6.3 Statistiques moyennes sur les solutions optimales . . . 95
Tableau 6.4 Impact de l'omission de composantes algorithmiques . . . 96
Table A.1 Detailed computational results with $|N^C| = 30$ . . . 112
Table A.2 Detailed computational results with $|N^C| = 50$ . . . 114
Table A.3 Detailed computational results with $|N^C| = 100$ . . . 116
Table A.4 Detailed results when omitting DDOIs . . . 119
Table A.5 Detailed results when omitting RCIs . . . 120
Table A.6 Detailed results when omitting SRIs . . . 121
Table A.7 Detailed results when omitting VSIs . . . 122
Table A.8 Detailed results when adding a second-copy variable for first-echelon routes visiting more than one satellite . . . 123
Table A.9 Detailed results when omitting branching level 1 . . . 124
Table A.10 Detailed results when omitting branching level 2 . . . 125
Table A.11 Detailed results when omitting branching level 3 . . . 126
Table A.12 Detailed results when omitting branching levels 2 and 3 . . . 127

Table A.13 Detailed results when omitting branching levels 1, 2 and 3 128
Table A.14 Detailed results when prioritizing branching over levels 2 and 3 over branching level 1 129
Table A.15 Detailed results when considering only back-and-forth first-echelon routes for $|N^C| = 30$ 131
Table A.16 Detailed results when considering only back-and-forth first-echelon routes for $|N^C| = 50$ 134
Table A.17 Detailed results when considering only back-and-forth first-echelon routes for $|N^C| = 100$ 137
Table B.1 Detailed computational results for instances with $|N^C| = 30$ 145
Table B.2 Detailed computational results for instances with $|N^C| = 50$ 147
Table B.3 Detailed computational results for instances with $|N^C| = 100$ 149
Tableau C.1 Résultats numériques détaillés pour les instances avec $|N^C| = 30$ . . . 152
Tableau C.2 Résultats numériques détaillés pour les instances avec $|N^C| = 50$ . . . 154
Tableau C.3 Résultats numériques détaillés pour les instances avec $|N^C| = 100$ . . 156

LISTE DES FIGURES

Figure 4.1 Example of a dominated first-echelon route 20
Figure 4.2 Example of an infeasible integer solution 33
Figure 5.1 Example of a solution for an instance of the MC-2E-VRPTW 55
Figure 6.1 Exemple d'une solution entière non réalisable 84
Figure B.1 Example of an integer infeasible solution 143

LISTE DES SIGLES ET ABRÉVIATIONS

2E-2P	Two-Echelon Two-Path Formulation
2E-CVRP	Two-Echelon Capacitated Vehicle Routing Problem
2E-VRP	Two-Echelon Vehicle Routing Problem
2E-VRPGS	Two-Echelon Vehicle Routing Problem with Grouping constraints and Simultaneous pickup and delivery
2E-VRPTW-SPD	Two-Echelon Vehicle Routing Problem with Time Windows and Simultaneous Pickup and Delivery
2E-VRPTW	Two-Echelon Vehicle Routing Problem with Time Windows
2E-VRPSPD	Two-Echelon Vehicle Routing Problem with Simultaneous Pickup and Delivery
BB	Branch-and-Bound
BC	Branch-and-Cut
BFFER	Back-and-Forth First-Echelon Route
BP	Branch-and-Price
BPB	BP-Based
BPC	Branch-Price-and-Cut
CG	Column Generation
CVRP	Capacitated Vehicle Routing Problem
DDOI	Deep Dual-Optimal Inequality
DI	Dual Inequality
DOI	Dual-Optimal Inequality
DSSR	Decremental State-Space Relaxation
ESPPRC	Elementary Shortest Path Problem with Resource Constraints
GC	Génération de Colonnes
MC-2EDP	Multi-Commodity Two-Echelon Distribution Problem
MC-2E-VRPTW	Multi-Commodity Two-Echelon Vehicle Routing Problem with Time Windows
MP	Master Problem
PCL	Paire de Collecte-Livraison
PIB	Produit Intérieur Brut
PM	Problème Maître
PMR	Problème Maître Restreint
RCI	Rounded Capacity Inequality

REF	Resource Extension Function
RMP	Restricted Master Problem
SRI	Subset-Row Inequality
TI	Transfer Inequality
TPI	Two-Path Inequality
VRPTW	Vehicle Routing Problem with Time Windows
VSI	Visited Satellite Inequality

LISTE DES ANNEXES

Annexe A Detailed results for the 2E-VRPTW 111
Annexe B Supplement for the MC-2E-VRPTW 140
Annexe C Résultats détaillés pour le 2E-VRPTW-SPD 151

CHAPITRE 1 INTRODUCTION

La résolution des problématiques liées au transport et à la distribution de marchandises conditionne fortement le développement des économies. Selon *Trading Economics*, le produit intérieur brut (PIB) provenant du secteur de transport en 2024 s'élevait à 102,3 milliards de dollars, soit une contribution de 4,8% au PIB du Canada. Les impacts sociaux et environnementaux des flux de marchandises en milieu urbain ont créé le besoin de repenser les structures des systèmes de distribution (Taniguchi and Thompson, 2002). La logistique urbaine vise à réduire la pollution environnementale et sonore, alléger les congestions des routes en agglomérations et contribuer au développement socio-économique des villes (Savelsbergh and Van Woensel, 2016). Les études s'intéressant à la planification des opérations relatives au transport de marchandises en zones urbaines ont ainsi suscité beaucoup d'intérêt lors des dernières années. Dans le contexte de la logistique urbaine, les chaînes de distribution sont structurées en plusieurs échelons. Cette configuration permet l'utilisation de flottes de véhicules adaptées aux caractéristiques de chaque échelon de distribution contribuant ainsi à réduire la pollution et la congestion sur les artères routières des centres-villes et à réaliser des économies d'échelle.

Dans cette thèse, nous traitons trois variantes de problèmes de tournées de véhicules définis sur des chaînes de distribution à deux échelons : le problème de tournées de véhicules à deux échelons avec fenêtres de temps (*the two-echelon vehicle routing problem with time windows*, 2E-VRPTW) et deux extensions de ce dernier, à savoir le problème de tournées de véhicules à deux échelons avec fenêtres de temps et multiple commodités (*the multi-commidity two-echelon vehicle routing problem with time windows*, MC-2E-VRPTW) et le problème de tournées de véhicules à deux échelons avec fenêtres de temps, collectes et livraisons simultanées (*the two-echelon vehicle routing problem with time windows and simultaneous pickup and delivery*, 2E-VRPTW-SPD). Le 2E-VRPTW est une variante du problème de tournées de véhicules à deux échelons avec contraintes de capacité (*the two-echelon capacitated vehicle routing problem*, 2E-CVRP) (voir Guastaroba et al., 2016). Avant de définir chacune des variantes considérées, débutons par décrire le 2E-CVRP.

Le 2E-CVRP généralise le problème de tournées de véhicules avec capacité (*capacitated vehicle routing problem*, CVRP) en considérant un système de distribution défini sur deux échelons. Dans ce contexte, les demandes des clients proviennent d'un dépôt central et sont acheminées vers leurs destinations finales à travers des dépôts intermédiaires, appelés satellites. Deux flottes homogènes de véhicules, à capacités respectives limitées, sont disponibles

pour la distribution de marchandises sur les deux échelons. Les routes de premier échelon, commençant et se terminant au dépôt central, livrent une quantité de marchandises à un sous-ensemble de satellites. Les routes de second échelon commencent et se terminent au même satellite, et desservent un sous-ensemble de clients. Une route de premier échelon (respectivement, de second échelon) est réalisable si la quantité totale de marchandise transportée ne dépasse pas la capacité d'un véhicule de premier (respectivement, second) échelon, et si chaque satellite (respectivement, client) desservi est visité une seule fois. Les marchandises livrées au niveau de chaque satellite sont transférées à bord de véhicules de second échelon afin d'être livrées aux clients. Chaque unité de fret transférée entraîne un coût de manutention unitaire. La quantité totale de fret livrée à un satellite est égale à la somme des quantités transférées à bord des véhicules de second échelon qui y débutent leur route. De plus, afin d'éviter la congestion au niveau des satellites, des limites sur le nombre de véhicules de second échelon pouvant débuter leur route à partir de chaque satellite sont imposées. Le 2E-CVRP consiste à déterminer des routes réalisables de premier et de second échelon minimisant les coûts totaux de transport et de manutention encourus, de sorte que chaque client soit visité par une seule route de second échelon.

Le 2E-VRPTW se distingue du 2E-CVRP du fait que : i) plusieurs dépôts sont considérés, ii) des fenêtres de temps sont imposées aux clients, iii) les véhicules de premier et de second échelon disposent de capacité illimitée, iv) aucune limite n'est imposée sur le nombre de véhicules de second échelon pouvant débuter à partir d'un satellite, v) les transferts de marchandises aux satellites ne génèrent aucun coût de manutention, vi) les contraintes de synchronisation entre les véhicules des deux échelons imposent qu'un véhicule de second échelon ne peut être alimenté que par un seul véhicule de premier échelon. En effet, la synchronisation des opérations entre les deux échelons dans le 2E-VRPTW est exacte (*exact operation synchronization*, voir Drexl 2012). Cela signifie que, pour chaque transfert de fret entre un véhicule de premier et de deuxième échelon à un satellite donné, ceux-ci doivent être présents simultanément au satellite et effectuer leurs opérations de transferts de marchandises au même moment. Cette exigence s'explique par l'absence de capacité d'entreposage au niveau des satellites, excluant toute possibilité de consolidation de fret. Ainsi, chaque route de second échelon est approvisionnée entièrement par un seul véhicule de premier échelon, garantissant une traçabilité des flux de marchandises entre les deux niveaux du système de distribution.

Le MC-2E-VRPTW étend le 2E-VRPTW en prenant en compte la gestion de plusieurs commodités, un aspect souvent présent en pratique, où les demandes des clients sont non substituables et proviennent de dépôts différents. Ce problème inclut plusieurs dépôts, satellites et clients, où chaque client est affecté à un dépôt spécifique qui fournit sa demande, chaque demande étant considérée comme une commodité. La synchronisation entre les deux échelons

est exacte et impose que la demande de chaque client soit entièrement fournie par une route de premier échelon, débutant du dépôt associé à ce client, et par une route de second échelon. Un véhicule de second échelon peut cependant être approvisionné par plusieurs véhicules de premier échelon, notamment lorsque des clients de différents dépôts sont desservis.

Le 2E-VRPTW-SPD est une variante du 2E-VRPTW intégrant la gestion simultanée de livraisons et de collectes. Chaque client peut requérir une livraison, une collecte, ou les deux. Une telle configuration apparaît notamment dans des contextes industriels ou urbains, tels que la distribution de produits accompagnée de la reprise d'emballages consignés, la collecte de déchets ou les services de logistique inverse. Dans ce problème, les véhicules de premier échelon effectuent exclusivement soit des livraisons, soit des collectes. De plus, la synchronisation des opérations entre les véhicules des premier et second échelons est exacte. Cela veut dire que, pour chaque route de second échelon, la demande totale de livraison est fournie par un véhicule de premier échelon et la demande totale de collecte est transférée vers un véhicule de premier échelon.

Depuis deux décennies, les algorithmes de type *branch-price-and-cut* (BPC), imbriquant la génération de colonnes dans un algorithme de type branch-and-cut, sont reconnus comme les plus efficaces pour résoudre une vaste gamme de problèmes de tournées de véhicules (Costa et al., 2019). Toutefois, la complexité des problèmes à deux échelons pose des défis au développement de tels algorithmes auxquels nous nous attaquons dans cette thèse. Ainsi, pour chacune des variantes traitées, nous développons une méthode BPC qui permet de calculer des solutions optimales. À notre connaissance, pour ces problèmes, aucune méthode de résolution exacte basée sur la génération de colonnes ne détaille clairement comment les aspects de synchronisation exacte des opérations entre les véhicules des deux échelons peuvent être gérés en cours de résolution au niveau des sous-problèmes. L'intégration des aspects de synchronisation exacte dans les algorithmes de programmation dynamiques dédiés à la génération de routes de second échelon constitue une contribution méthodologique importante de cette thèse. Pour accélérer les algorithmes BPC développés, nous introduisons des inégalités duales, appelées inégalités de transfert, qui permettent de stabiliser les variables duales associées à certaines contraintes du problème maître. Bien que les algorithmes proposés pour ces trois problèmes soient similaires, leur conception requiert des spécificités non triviales, notamment, au niveau des algorithmes de programmation dynamique utilisés pour générer de nouvelles variables. De nombreuses expérimentations numériques effectuées sur des instances artificielles provenant de la littérature démontrent l'efficacité des méthodes proposées en comparaison avec l'état de l'art.

Le présent document est organisé comme suit. Le chapitre 2 permet de faire une brève revue

de littérature sur les variantes traitées des 2E-VRPs. Le chapitre 3 introduit l'organisation du cœur de la thèse, soit les chapitres 4, 5 et 6. Le chapitre 4 présente un article publié dans la revue *Transportation Science* où nous proposons un algorithme BPC pour le 2E-VRPTW. Le chapitre 5 est constitué d'un article soumis à la revue *INFORMS Journal On Computing* où nous étudions le MC-2E-VRPTW pour lequel nous développons un algorithme BPC. Le chapitre 6 traite le 2E-VRPTW-SPD. Finalement, une discussion générale et une conclusion sont présentées dans les chapitres 7 et 8, respectivement.

CHAPITRE 2 REVUE DE LITTÉRATURE

Les problèmes de tournées de véhicules à deux échelons se trouvent au cœur de nombreux travaux qui ont été synthétisés dans Cuda et al. (2015) et récemment dans Sluijk et al. (2023). Dans ce chapitre, nous présentons les meilleurs algorithmes exacts développés pour résoudre différentes variantes du 2E-CVRP : la variante avec fenêtres de temps (2E-VRPTW), ainsi que deux extensions de celle-ci intégrant respectivement des commodités multiples (MC-2E-VRPTW) et des collectes et livraisons simultanées (2E-VRPTW-SPD).

Ce chapitre s'organise comme suit. La section 2.1 présente les contributions majeures concernant le 2E-CVRP. La section 2.2 décrit les algorithmes conçus pour le 2E-VRPTW ou pour des variantes apparentées. Enfin, les sections 2.3 et 2.4 détaillent les algorithmes proposés pour la résolution des variantes considérant respectivement des commodités multiples et des collectes et livraisons simultanées.

2.1 Problèmes de tournées de véhicules à deux échelons

Au meilleur de nos connaissances, le 2E-CVRP a été introduit pour la première fois dans la littérature par Gonzalez-Feliu et al. (2008). Ce problème survient naturellement lorsque des marchandises doivent être distribuées dans les zones urbaines. Les auteurs proposent une formulation de flots à commodités qu'ils renforcent par l'introduction de deux familles d'inégalités valides inspirées de la littérature existante sur le CVRP. L'algorithme d'énumération implicite avec plans coupants (*branch-and-cut*, BC) développé réussit à résoudre à optimalité des instances comprenant un dépôt central, deux satellites et jusqu'à 32 clients.

D'autres auteurs ont proposé des algorithmes BC pour le 2E-CVRP. Perboli et al. (2011) renforcent la formulation proposée par Gonzalez-Feliu et al. (2008) en introduisant deux nouvelles familles d'inégalités valides. Des instances comprenant jusqu'à deux satellites et 32 clients sont résolues à optimalité. Il est à noter que Jepsen et al. (2013) ont montré que le modèle présenté par Perboli et al. (2011) n'est pas valide lorsque plus de trois satellites sont considérés. Jepsen et al. (2013) adressent la résolution du 2E-CVRP en proposant une formulation qui est une relaxation du problème. Les contraintes relâchées ainsi que les inégalités valides introduites sont gérées par l'algorithme BC (via un test de réalisabilité). La mise en place d'un branchement spécialisé permet de retrouver des solutions réalisables en écartant toute solution entière non réalisable. Des instances avec au plus cinq satellites et 50 clients sont résolues à l'optimalité.

Baldacci et al. (2013) conçoivent un algorithme exact pour le 2E-CVRP. Les auteurs décrivent une procédure calculant des bornes inférieures et supérieures pour le 2E-CVRP. Les bornes calculées sont utilisées pour générer les configurations de routes de premier niveau pouvant potentiellement faire partie d'une solution optimale du 2E-CVRP. En fixant une configuration, le problème est réduit à un nombre limité de problèmes de tournées de véhicules à multi-dépôts avec contraintes supplémentaires. Ils arrivent à résoudre à l'optimalité des instances contenant jusqu'à six satellites et 100 clients.

Santos et al. (2015) conçoivent le premier algorithme BPC pour le 2E-CVRP. Dans la formulation présentée, les routes de premier échelon sont énumérées et les variables associées ajoutées a priori. Les variables de second échelon sont ajoutées dynamiquement en résolvant un sous-problème par satellite. Pour accélérer la résolution des sous-problèmes, les contraintes d'élémentarité sont relâchées en considérant des routes (partielles) de second échelon respectant uniquement les contraintes de ressources (*q-routes*). L'élémentarité est imposée par le biais de décisions de branchement et de plans coupants. Des coupes issues de la littérature sur le CVRP et le problème du voyageur de commerce (*the traveling salesman problem*) sont introduites pour renforcer la formulation. Les résultats présentés sont similaires à ceux rapportés par Jepsen et al. (2013).

Marques et al. (2020) ont proposé un algorithme BPC amélioré pour le 2E-CVRP. Les auteurs présentent une nouvelle formulation qui garantit la conservation des flux de marchandises traversant les satellites à l'aide de contraintes de balancement (*balancing constraints*). Les auteurs introduisent une nouvelle famille d'inégalités valides (*satellite supply inequalities*), inspirées de coupes introduites par Belenguer et al. (2011) pour le problème de localisation-routage (*location routing problem*), et montrent, à travers des tests empiriques, qu'elles améliorent la qualité des bornes duales obtenues. L'algorithme BPC développé réussit à résoudre à l'optimalité des instances considérant jusqu'à 10 satellites et 200 clients.

2.2 Problèmes de tournées de véhicules à deux échelons avec fenêtres de temps

La littérature portant sur les extensions du 2E-CVRP avec fenêtres de temps est relativement limitée par rapport à celle sur le 2E-CVRP. La présence de contraintes temporelles, imposées par la synchronisation des opérations des véhicules aux satellites, rend la conception d'algorithmes exacts plus difficile. À notre connaissance, deux travaux se sont intéressés à des extensions du 2E-CVRP avec fenêtres de temps et une commodité : le 2E-VRPTW introduit par Dellaert et al. (2019) et le 2E-VRPTW avec consolidation de fret, considérant des routes simples ou multiples, étudié par Marques et al. (2022). Nous présentons ci-après les algorithmes exacts développés pour ces variantes.

Dellaert et al. (2019) se focalisent sur le problème de tournées de véhicules à deux échelons avec fenêtres de temps, plusieurs dépôts et synchronisation exacte (2E-VRPTW). Les auteurs proposent deux formulations étendues, qui diffèrent par le niveau d'agrégation considéré, et une formulation compacte résolue par un solveur commercial. La première formulation, appelée formulation à deux échelons et un chemin (*two-echelon one-path formulation*), est une formulation de partitionnement d'ensemble qui utilise un grand nombre de variables, chacune représentant un arbre-tournée (*tour-tree*) défini comme la combinaison d'une route de premier échelon et des routes de second échelon alimentées par celle-ci. Pour résoudre cette formulation, les auteurs développent un algorithme de branch-and-bound avec génération de colonnes (*branch-and-price*, BP) considérant un sous-problème par dépôt, modélisé comme un problème de plus court chemin élémentaires avec contraintes de ressources (*elementary shortest path problem with resource constraints*, ESPPRC) et résolu à l'aide d'un algorithme de programmation dynamique à étiquettes (*labeling algorithm*). La deuxième formulation, dite à deux échelons et deux chemins (*two-echelon two-path formulation*, 2E-2P), utilise deux groupes de variables, où chaque groupe est associé aux routes sur un échelon de routage. Les deux groupes de variables sont reliés par des contraintes de synchronisation de quantités (*load synchronization constraints*), qui assurent que la quantité totale de marchandise alimentée par chaque véhicule de premier échelon utilisé ne dépasse pas sa capacité. La formulation 2E-2P est résolue via un algorithme de type énumération implicite avec génération de colonnes (*branch-and-price-based algorithm*, BPB), qui opère comme suit. L'algorithme commence par générer des sous-ensembles de routes de premier échelon, appelés configurations, et les trie par ordre croissant d'une borne inférieure obtenue en considérant la configuration associée comme solution du premier échelon. Grâce à leurs bornes associées, les configurations sont utilisées pour créer des sous-arbres de branchement (*search sub-trees*), dans lesquels les routes de premier échelon appartenant à la configuration sélectionnée sont fixées. L'algorithme BPB commence par créer un sous-arbre, associé à la configuration de plus petite borne inférieure, et l'explore au moyen d'un algorithme BP qui : i) considère un sous-problème pour chaque route de premier échelon m appartenant à la configuration courante et chaque satellite visité par m, et ii) impose des décisions de branchement ciblant uniquement les variables (ou les entités) associées au second échelon. Chaque fois que la meilleure borne inférieure trouvée lors de l'exploration du sous-arbre actuel devient supérieure à la borne inférieure associée à la prochaine configuration non explorée, l'algorithme BPB crée un nouveau sous-arbre associé à cette dernière et l'explore de la même manière. Ce processus se poursuit jusqu'à ce que le sous-arbre actuel soit entièrement exploré et que la meilleure borne supérieure obtenue soit inférieure ou égale aux bornes inférieures des configurations non explorées. Cet algorithme, parmi ceux développés, est le plus performant et résout à l'optimalité des instances

considérant jusqu'à trois dépôts, cinq satellites et 100 clients.

Marques et al. (2022) développent un algorithme BPC pour le 2E-VRPTW avec consolidation de fret aux satellites. La formulation proposée est indicée par routes et comporte un nombre exponentiel de contraintes de précédence assurant le respect des synchronisations des quantités et des opérations (*load and operation synchronization*). L'algorithme conçu est adapté pour traiter les cas avec des routes simples (*single-trip*) ou multiples (*multi-trip*) sur le second échelon. Les auteurs présentent également une procédure de post-traitement pour vérifier si les solutions trouvées sont valides dans un contexte de synchronisation exacte, c'est-à-dire si une solution de même coût, ne nécessitant ni entreposage ni consolidation de fret, existe. L'algorithme BPC est efficace sur les instances de la littérature, produisant de nouvelles solutions optimales pour 9 et 54 instances considérant respectivement des routes simples et multiples sur le second échelon.

2.3 Problèmes de tournées de véhicules à deux échelons avec multiples commodités

Les variantes de problèmes de tournées de véhicules à commodités multiples ont largement été étudiées dans la littérature. Comme souligné par Gu et al. (2023), la prise en compte explicite de plusieurs commodités survient lorsqu'il existe plusieurs types de marchandises à transporter qui ne peuvent être agrégées en un seul type. L'article synthèse de Gu et al. (2023) présente une classification pour ces problèmes. La littérature sur les variantes du 2E-CVRP considérant explicitement des commodités multiples est relativement limitée. Nous avons identifié deux travaux considérant de telles variantes pour lesquelles sont développés des algorithmes exacts : le MC-2E-VRPTW étudié par Dellaert et al. (2021) et le problème de distribution à deux échelons et commodités multiples (*the multi-commodity two-echelon distribution problem*, MC-2EDP) introduit par Gu et al. (2022) et revisité par Petris et al. (2024). Dans ce qui suit, nous présentons les meilleurs algorithmes exacts conçus pour ces variantes.

Le meilleur algorithme exact développé par Dellaert et al. (2021) pour le MC-2E-VRPTW est une adaptation de l'algorithme BPB proposé pour le 2E-VRPTW et décrit plus haut (voir section 2.2). De manière similaire, cet algorithme résout une formulation à deux échelons et deux chemins pour le MC-2E-VRPTW. L'algorithme BPB est testé sur un ensemble restreint d'instances comprenant jusqu'à trois dépôts, cinq satellites et 100 clients, avec un temps limite de trois heures. La majorité des instances comportant 50 clients sont résolues dans cet intervalle, ainsi qu'une seule instance avec trois dépôts, cinq satellites et 100 clients.

Le MC-2EDP, introduit par Gu et al. (2022), est une variante du 2E-CVRP à plusieurs dépôts, dans laquelle : i) chaque route de premier échelon visite un satellite, et ii) un client peut avoir une demande pour plusieurs commodités ; cependant, chaque commodité doit être livrée lors d'une seule visite d'un véhicule de second échelon. Pour ce problème, Petris et al. (2024) proposent une formulation en recouvrement d'ensemble, comprenant un grand nombre de variables associées aux routes de second échelon, et conçoivent un algorithme BPC pour la résoudre. Les variables de second échelon sont générées dynamiquement via la résolution, pour chaque satellite, d'un sous-problème modélisé comme un ESPPRC et résolu à l'aide d'un algorithme d'étiquetage. Ils arrivent à résoudre à l'optimalité, dans un temps limite d'une heure, 439 des 736 instances introduites par Gu et al. (2022).

2.4 Problèmes de tournées de véhicules à deux échelons avec collectes et livraisons simultanées

Plusieurs travaux récents ont étudié des variantes du 2E-CVRP avec collectes et livraisons simultanées (*the two-echelon vehicle routing problem with simultaneous pickup and delivery*, 2E-VRPSPD). La majorité de ceux-ci ont proposé des approches heuristiques pour ces problèmes. Au meilleur de notre connaissance, l'article de Li et al. (2022) est le seul existant dans la littérature qui propose un algorithme exact pour résoudre une extension du 2E-VRPSPD qui considère des contraintes de regroupement pour les clients au second échelon. Ces contraintes imposent que les clients appartenant au même groupe soient tous desservis à partir d'un même satellite.

Pour résoudre cette variante, Li et al. (2022) développent un algorithme BPC. La formulation proposée implique des variables associées aux routes sur les deux échelons. Des contraintes de synchronisation de quantités assurent la conservation des flux de marchandises à travers les satellites pour les collectes et les livraisons. Les auteurs décrivent un algorithme d'étiquetage bidirectionnel, incluant un critère de dominance adapté, pour résoudre les sous-problèmes et proposent quatre familles d'inégalités valides pour améliorer la qualité des bornes obtenues. L'algorithme BPC développé résout des instances allant jusqu'à six satellites et 60 clients dans un temps limite de deux heures.

2.5 Sommaire

Le tableau 2.1 présente un récapitulatif des articles ayant étudié des variantes du 2E-CVRP et développé des algorithmes exacts pour leur résolution. Nous classifions les travaux selon les caractéristiques des variantes étudiées et des méthodes de résolution exactes développées pour

celles-ci. Les caractéristiques considérées sont : le nombre de dépôts (# Dépôts), et l'inclusion (indiquée par •) des aspects suivants : synchronisation exacte, capacité des satellites, coûts de manutention, commodités multiples (MC), fenêtres de temps (TW) pour les débuts de service chez les clients, et collectes et livraisons simultanées (CLS). Les entrées dans la colonne Algorithme font référence aux algorithmes de types BC, BPB et BPC et aux algorithmes exacts spécialisés (ES). Les trois dernières lignes de ce tableau indiquent les caractéristiques des problèmes que nous étudierons aux chapitres 4, 5 et 6.

Tableau 2.1 Travaux proposant des algorithmes exacts pour des variantes du 2E-CVRP

	# Dépôts	Synchronisation exacte	Capacité des satellites*	Coûts de manutention	MC	TW	CLS	Algorithme
Gonzalez-Feliu et al. (2008)	1		•	•				BC
Perboli et al. (2011)	1		•	•				BC
Jepsen et al. (2013)	1		•	•				BC
Baldacci et al. (2013)	1		•	•				ES
Santos et al. (2015)	1		•	•				BPC
Marques et al. (2020)	1		•	•				BPC
Dellaert et al. (2019)	≥ 1	•				•		BPB
Marques et al. (2022)	≥ 1					•		BPC
Dellaert et al. (2021)	≥ 1	•			•	•		BPB
Petris et al. (2024)	≥ 1				•			BPC
Li et al. (2022)	≥ 1		•†	•			•	BPC
Chapitre 4	≥ 1	•				•		BPC
Chapitre 5	≥ 1	•			•	•		BPC
Chapitre 6	≥ 1	•				•	•	BPC

* : Limite sur le nombre de véhicules de second échelon débutant à partir de chaque satellite.

† : Limites sur les quantités de marchandises livrées à partir de ou collectées à chaque satellite.

CHAPITRE 3 ORGANISATION DE LA THÈSE

Dans cette thèse, nous nous sommes intéressés à des problèmes qui surviennent en contexte de logistique urbaine, soit des problèmes de tournées de véhicules à deux échelons avec fenêtres de temps. Ces problèmes sont importants compte tenu de l'augmentation des livraisons urbaines, notamment, des colis directement chez les consommateurs. Afin de proposer des solutions pour ces problèmes, nous avons développé des algorithmes exacts de génération de colonnes avec plans coupants (BPC). Ces algorithmes sont parmi les meilleurs pour résoudre des problèmes de tournées de véhicules (voir le chapitre 2). Dans ce contexte, plusieurs composantes algorithmiques sont à étudier. D'abord, lors du développement d'algorithmes de BPC, il existe plusieurs alternatives pour traiter la génération ainsi que la synchronisation des routes de premier et de second échelon. Pour la génération des routes (sans considérer la synchronisation), voici quelques possibilités : (i) générer (ou énumérer) des routes complètes comportant une route de premier échelon ainsi que les routes de deuxième échelon qui sont alimentées par cette dernière, ou (ii) générer (ou énumérer) des routes de premier échelon et générer (ou énumérer) de façon distincte des routes de deuxième échelon. Dans la première alternative, la synchronisation est imbriquée dans la construction des routes complètes, alors que dans la seconde alternative, il faut également imposer la synchronisation entre les routes de premier et de second échelon. En génération de colonnes, il est alors possible d'imposer la synchronisation dans le problème maître ou (non exclusif) dans les sous-problèmes. Dans cette thèse, les algorithmes développés consisteront à énumérer les routes de premier échelon compte tenu de leur nombre limité, générer les routes de deuxième échelon via des algorithmes d'étiquetage, et imposer la synchronisation des opérations directement dans les sous-problèmes en génération de colonnes. Les contraintes liées à la synchronisation des quantités *load synchronization*) sont considérées dans le problème maître. Par ailleurs, nous implanterons également plusieurs composantes algorithmiques novatrices telles les inégalités duales, des processus de recouvrement requis pour établir la faisabilité des solutions entières ainsi que des règles de branchement adaptées aux variantes considérées. Ces composantes sont basées sur des constats mathématiques et peuvent uniquement être utilisés dans certains contextes, et permettent une réelle amélioration de la performance des algorithmes développés.

Dans le chapitre 4, nous étudierons le 2E-VRPTW. Pour ce problème, nous nous sommes basés sur les travaux de Dellaert et al. (2019), plus particulièrement sur la formulation à deux chemins et deux échelons (2E-2P) qu'ils ont proposée pour ce problème. Ces auteurs ont introduit le 2E-VRPTW et ont développé des formulations étendues basées sur l'énumération des routes. Dans la formulation 2E-2P, la définition des variables associées aux routes de

second échelon encapsule l'aspect de synchronisation des opérations exacte entre les véhicules de premier et de second échelon. Le lien entre ce groupe de variables et celui associé aux routes de premier échelon est établi via des contraintes de synchronisation de quantités. La difficulté majeure découlant de la considération de cette formulation réside dans la gestion, au niveau des sous-problèmes, de l'information duale associée aux contraintes de synchronisation des quantités lors de la génération de routes de second échelon. Dans cette optique, nous avons ainsi développé un nouvel algorithme exact de type BPC, incluant un algorithme d'étiquetage ad hoc qui permet de générer des routes de second échelon satisfaisant les aspects temporels imposés par la synchronisation des opérations des véhicules aux satellites. La formulation considérée permet d'exploiter des relations existantes entre les routes de premier échelon et d'introduire ainsi des inégalités duales, appelées inégalités de transfert, permettant d'accélérer la convergence de la génération de colonnes.

Dans le chapitre 5, nous étendons les idées développées au chapitre précédent à une variante avec commodités multiples (MC-2E-VRPTW). Cette variante, introduite par Dellaert et al. (2021), enrichit le problème précédent en considérant la spécificité des demandes clients, soit que la demande de chaque client n'est disponible qu'à partir d'un dépôt spécifique. Les contraintes liées à la synchronisation des opérations imposent que la demande de chaque client soit satisfaite à partir d'une route de premier échelon et une route de second échelon. Toutefois, et contrairement au 2E-VRPTW, une route de second échelon peut recevoir de la marchandise de plusieurs véhicules de premier échelon. Dans ce contexte, il faut être en mesure de spécifier, pour chaque route de second échelon et pour chaque client visité le long de celle-ci, la route de premier échelon ayant fourni la demande du client au satellite de départ. Pour y parvenir, nous introduisons la notion de patron de réapprovisionnement (*supply pattern*) associé à une route de second échelon. Nous adaptons la formulation 2E-2P pour ce problème en considérant des variables de second échelon où chacune est associée à une route de second échelon et un patron de réapprovisionnement. L'algorithme BPC développé incorpore les contraintes de synchronisation des opérations au niveau des sous-problèmes. À cette fin, l'algorithme d'étiquetage est ajusté de sorte à pouvoir générer simultanément des routes de second échelon et des patrons de réapprovisionnement associés. Nous proposons une règle de branchement qui tire profit du contexte à commodités multiples et développons une procédure de récupération permettant d'établir la faisabilité des solutions entières en présence des inégalités de transfert.

Dans le chapitre 6, nous introduisons une nouvelle variante du 2E-VRPTW lorsque les clients ont des demandes simultanées de collecte et de livraison, le 2E-VRPTW-SPD. Au meilleur de nos connaissances, il s'agit de la première fois où cette version du problème est étudiée. Nous proposons ainsi une description formelle et mathématique du 2E-VRPTW-SPD. Dans

cette variante, la synchronisation des opérations aux satellites impose que chaque véhicule de second échelon reçoive sa demande totale de livraison d'un véhicule de premier échelon, et fournisse sa demande totale de collecte à un véhicule de premier échelon. Nous développons un algorithme exact de type BPC pour cette variante formulée avec une adaptation de la formulation 2E-2P, où chaque variable de second échelon est associée à une route sur cet échelon et à une paire de routes de premier échelon. Nous décrivons les ajustements spécifiques à apporter à l'algorithme d'étiquetage afin de permettre la génération de routes de second échelon adhérant aux contraintes liées à la synchronisation des opérations. Nous présentons les résultats obtenus en testant l'algorithme développé sur des instances de la littérature et effectuons des analyses de sensibilité sur des composantes de notre algorithme afin de déterminer leur impact sur la performance globale de l'algorithme.

Une discussion générale sur les travaux présentés dans ces trois chapitres est ensuite présentée au chapitre 7.

CHAPITRE 4 ARTICLE 1 : A BRANCH-PRICE-AND-CUT ALGORITHM FOR THE TWO-ECHELON VEHICLE ROUTING PROBLEM WITH TIME WINDOWS

Recopié avec permission, T. Mhamedi, H. Andersson, M. Cherkesly, G. Desaulniers, (2022), A Branch-Price-and-Cut Algorithm for the Two-Echelon Vehicle Routing Problem with Time Windows. *Transportation Science,* 56(1) :245–264. Publié en ligne dans *Articles in Advance,* le 6 Décembre 2021.

4.1 Introduction

City logistics and challenges related to transportation and distribution in urban areas have received a huge interest lately (Crainic et al., 2009; Savelsbergh and Van Woensel, 2016). Due to the growth of e-commerce and new trends in delivery (e.g., same-day delivery, sustainable delivery), there are more freight movements inside urban areas. This increases pollution, noise, and traffic for people living in those areas. Therefore, many authors have studied distribution systems adapted to city logistics which aim at reducing the negative impacts while sustaining the economy. One common way is to design the distribution system into echelons, most often with two echelons (Crainic et al., 2004, 2009; Savelsbergh and Van Woensel, 2016).

In two-echelon distribution systems, freight transported from a distribution center (outside the city) to a customer in the city is transported by a high-capacity vehicle to another distribution center or a satellite on the outskirts of the city (the first echelon), and then from this distribution center to the destination using a smaller vehicle (the second echelon). With this setup, the fleet can be adapted to the characteristics of each echelon and the overall performance can be improved. In particular, smaller vehicles are operated in city centers to increase safety and reduce the negative impacts. These vehicles are also more flexible and can travel more easily in densely populated areas and in smaller streets.

In this paper, we consider the two-echelon vehicle routing problem with time windows (2E-VRPTW) introduced by Dellaert et al. (2019). First-echelon routes handle the transportation of freight from depots to satellites while second-echelon routes, departing from satellites, ensure that goods are being delivered to customers within their specified time windows. Considering the structure of the network (e.g., number of satellites), each satellite can be visited by more than one first-echelon route, but each second-echelon route must be supplied by exactly one first-echelon route to facilitate freight transfer at the satellites. The 2E-

VRPTW aims to find sets of first- and second-echelon routes such that total routing costs are minimized and each customer is visited during its allowed time window by exactly one second-echelon route. The total routing costs comprise fixed vehicle costs for first- and second-echelon vehicles as well as distance-related transportation costs.

Two-echelon routing problems have gained increased attention in the recent years as can be seen in the reviews by Gonzalez-Feliu (2011), Cuda et al. (2015) and Guastaroba et al. (2016). Gonzalez-Feliu (2011) performed a meta-narrative analysis to unify the main concepts, issues and methods within two-echelon distribution systems. Cuda et al. (2015) focused on three classes of two-echelon problems; the two-echelon vehicle routing problem, the two-echelon location routing problem, and the truck and trailer routing problem, as well as a detailed description of the solution methods used. Finally, Guastaroba et al. (2016) took a broader view and analyze the impact of intermediate facilities in freight transportation. Guastaroba et al. (2016) advocated a global look at the problems and emphasized the studies of more general service network design problems as well as international and intercontinental settings with multiple consolidation centers along the same route. Cuda et al. (2015) pointed to richer versions of the problem as open research areas, especially the introduction of time considerations and satellite synchronization which we explore in this paper.

To the best of our knowledge, the two-echelon capacitated vehicle routing problem (2E-CVRP) was first studied in Gonzalez-Feliu et al. (2008). Compared with the 2E-VRPTW, the 2E-VRP lacks the time windows and does not require each second-echelon route to be served by exactly one first-echelon route but imposes a restriction on the total number of second-echelon routes starting from each satellite. Moreover, it includes, along with transportation costs, handling costs into the objective. The authors proposed a commodity-flow formulation and a branch-and-cut algorithm that was able to solve instances with up to 32 customers and two satellites to optimality. These results were later improved by Perboli et al. (2011) where the formulation was strengthened by the addition of new valid inequalities. Although instances with two satellites were solved to optimality, Jepsen et al. (2013) later showed that the model proposed by Perboli et al. (2011) was not valid for instances with three satellites or more. Jepsen et al. (2013) also proposed a new mathematical model that is a relaxation of the 2E-CVRP. Constraints are relaxed and added dynamically together with valid inequalities in a branch-and-cut algorithm. The algorithm was able to solve instances with up to 50 customers and 5 satellites to optimality. Baldacci et al. (2013) formulated the problem using routes and described a bounding procedure for deriving valid lower and upper bounds. The bounds allow for efficient generation of all possible configurations of first-echelon routes that can be part of an optimal solution. By fixing a configuration, the problem decomposes into a small number of multi-depot capacitated vehicle routing problem with

side constraints. The exact algorithm was able to solve instances with up to 100 customers and 6 satellites. Computational results showed that the proposed method outperformed the algorithms proposed by Jepsen et al. (2013) and Perboli et al. (2011) in terms of quality of dual bounds, size and number of instances solved, and computing time.

Branch-price-and-cut (BPC) algorithms have also been proposed for the 2E-CVRP. The first algorithm was developed by Santos et al. (2015). First-echelon routes are enumerated a priori while second-echelon routes are priced through resource constrained shortest path problems. Results similar to those of Jepsen et al. (2013) were reported by the authors. Recently, Marques et al. (2020) proposed an improved BPC algorithm for the 2E-CVRP. They introduced a new formulation that guarantees satellite flow conservation through balancing constraints rather than defining product flows and outperformed the results obtained by the exact method of Baldacci et al. (2013) as instances with up to 200 customers and 10 satellites were solved to optimality.

There are also several heuristics developed for the 2E-CVRP. Perboli et al. (2011) proposed a matheuristic along their exact method, while Crainic et al. (2008, 2011) presented cluster-based heuristics. Large neighborhood searches were developed by Hemmelmayr et al. (2012) and Breunig et al. (2016), and most recently a route generation and recombination heuristic was devised by Amarouche et al. (2018).

The 2E-VRPTW has not received much attention even though time windows are a common extension to many routing problems. In this problem, the customers have time windows which causes a precedence relation between the first-echelon routes and the second-echelon routes at the satellites. To our knowledge, the only study of this problem is by Dellaert et al. (2019). The authors proposed two path-based formulations for the problem. The main difference between the formulations is the level of aggregation. In their one-path formulation (2E-1P), tour-trees are used. A tour-tree consists of a first-echelon route and one or several second-echelon routes starting from each of the satellites visited by the first-echelon route and adhering to the time restriction at the satellites imposed by the first-echelon route. The formulation is that of a pure set-partitioning problem, only stating that each customer must be visited exactly once. The two-path formulation (2E-2P) uses separate first- and second-echelon routes and links these through capacity constraints. The solution method for 2E-2P first generates all non-dominated first-echelon routes, where dominance is performed only with regards to the depot associated with the routes. Feasible first-echelon solutions are obtained by combining a given number of these routes. Next, the method enumerates first-echelon solutions and sorts them according to a lower bound obtained by solving a relaxation of the problem with the given first-echelon solution fixed. Solving the problem with a fixed

first-echelon solution is then repeated until the lower bound of the first-echelon solution is worse than the objective value of the best found solution. The computational study clearly showed the superiority of 2E-2P over 2E-1P. The algorithm based on 2E-2P succeeded in solving instances with up to 100 customers and 5 satellites to optimality.

In this paper, we propose a new BPC algorithm for the 2E-VRPTW based on the 2E-2P formulation of Dellaert et al. (2019). The main contributions of the paper are:

- A new pricing problem modeling where the generation of second-echelon routes is combined with determining the associated first-echelon route;
- A set of deep dual optimal inequalities to help stabilize the overall column generation process; and
- A specialized branching strategy to handle the use of variables associated with copies of a first-echelon route visiting a single satellite.

The remaining of the paper is organized as follows. Section 4.2 presents the mathematical formulation of the problem. In Section 4.3, the BPC algorithm is outlined together with a stabilization scheme based on dual inequalities and three families of valid inequalities taken from the literature. A thorough computational study is presented in Section 4.4, ending with managerial insights. Finally, Section 4.5 concludes the paper.

4.2 Mathematical formulation

The 2E-VRPTW can be formulated on a directed graph $G = (N, A)$, where N is the set of nodes and A the set of arcs. Node set $N = N^D \cup N^S \cup N^C$ is composed of disjoint sets of depots (N^D), satellites (N^S), and customers (N^C). Each customer $i \in N^C$ has a demand $q_i > 0$. The service at a customer must be conducted during a time window $[\underline{w}_i, \overline{w}_i]$, where $\underline{w}_i$ and $\overline{w}_i$ are the earliest and latest times at which service can start, respectively. If a vehicle arrives before $\underline{w}_i$, then it must wait until $\underline{w}_i$ at no additional cost to start its service. Note that no demands, nor time windows are defined for the depots and satellites, and vehicles leave at time zero from each depot. In addition, each node is associated with a service time $\tau_u, u \in N$, such that $\tau_d = 0, \forall d \in N^D$, $\tau_s > 0$, $\forall s \in N^S$, and $\tau_i > 0$, $\forall i \in N^C$. The set of arcs is defined as $A = A^1 \cup A^2$, where A^1 and A^2 describe the sets of arcs for the first- and second-echelon vehicles, respectively. The arc set $A^1 = \{(d, s) \cup (s, d) : d \in N^D, s \in N^S\} \cup \{(s_1, s_2) : s_1, s_2 \in N^S, s_1 \neq s_2\}$ contains an arc (in both directions) between each pair of depot and satellite, and between each pair of satellites. In set A^2, there is an arc (in both directions) between each pair of satellite and customer, and each pair of customers, that is, $A^2 = \{(s, i) \cup (i, s) : s \in N^S, i \in N^C\} \cup \{(i_1, i_2) : i_1, i_2 \in N^C, i_1 \neq i_2\}$. Each arc $(u, v) \in A$

is associated with a transportation cost c_{uv} and a travel time t_{uv} which includes the service time at node u.

A homogeneous fleet of first-echelon vehicles, as well as a homogeneous fleet of second-echelon vehicles are available. Using a first-echelon vehicle or a second-echelon vehicle induces a fixed cost $f_1 \geq 0$ or $f_2 \geq 0$, respectively. In addition, a capacity Q^1 and Q^2, with $0 < Q^2 < Q^1$, is imposed on first- and second-echelon vehicles, respectively. In practice, in city logistics, first-echelon vehicles have a high capacity, while second-echelon vehicles have a smaller capacity in order to reduce the negative effects on congestion, safety and environment in the city, i.e., the second-echelon. Note that second-echelon arcs $(i_1, i_2) \in A^2$ between pairs of customers $i_1, i_2 \in N^C$ are only generated if they can respect time windows and the capacity constraint, i.e., if $\underline{w}_{i_1} + t_{i_1 i_2} \leq \overline{w}_{i_2}$ and $q_{i_1} + q_{i_2} \leq Q^2$. A first-echelon route is said to be feasible if it starts and ends at a given depot $d \in N^D$, visits a non-empty subset of satellites $s \in N^S$, and respects the capacity of a first-echelon vehicle. A second-echelon route is said to be feasible if it starts and ends at a given satellite $s \in N^S$, visits a non-empty subset of customers $i \in N^C$, and respects the capacity of a second-echelon vehicle and the time windows of the visited customers.

The 2E-VRPTW aims to find a set of least-cost first- and second-echelon feasible routes such that each customer is visited by exactly one second-echelon vehicle and the load of each second-echelon route is supplied by exactly one first-echelon vehicle. The proposed definition of the 2E-VRPTW is the same as that of Dellaert et al. (2019) except that we impose an additional constraint. In theory, multiple vehicles can be assigned to the same first-echelon route. In practice, this seldom occurs for the routes visiting more than one satellite. In fact, it did not occur in the preliminary computational experiments that we conducted. Therefore, we assume that multiple vehicles can only be assigned to first-echelon routes visiting a single satellite. To further support this assumption, we present in Section 4.4.3.4 computational results that were obtained when this assumption is relaxed.

Let $\mathcal{M}$ denote the set of non-dominated feasible first-echelon routes (the dominance relation is defined below). A first-echelon route $m \in \mathcal{M}$ is an ordered set of nodes $(d, s_1, s_2, \ldots, s_k, d)$, starting and ending at a given depot $d \in N^D$ and visiting a sequence of distinct satellites $s_1, s_2, \ldots, s_k \in N^S$, $s_j \neq s_{j'}$ for $j, j' \in \{1, \ldots, k\}$, $j \neq j'$. Let c_m be its total cost, comprising the fixed vehicle cost f_1 and the total transportation costs. Furthermore, let $S(m) = \{s_1, s_2, ..., s_k\}$ represent the set of satellites visited in route m. Every first-echelon route $m \in \mathcal{M}$ starts at time zero and arrives at satellite $s \in S(m)$ at time t_s^m. To be able to service customer $i \in N^C$ at the beginning of its time window $\underline{w}_i$ directly from a satellite $s \in N^S$, the latest time at which service can begin at satellite s, i.e., a first-echelon route

can arrive at that satellite, is given by $L_i^s = \underline{w}_i - t_{si}$. With this notation, we say that a first-echelon route m_1 dominates another first-echelon route m_2 if

$$c_{m_1} \leq c_{m_2}, \tag{4.2.1}$$

$$S(m_1) = S(m_2), \tag{4.2.2}$$

$$\nexists i \in N^C \text{ such that } t_s^{m_2} \leq L_i^s \text{ and } t_s^{m_1} > L_i^s, \quad \forall s \in S(m_1). \tag{4.2.3}$$

Condition (4.2.1) stipulates that the cost of route m_1 must not exceed that of m_2. Condition (4.2.2) imposes that routes m_1 and m_2 visit the same subset of satellites (in no particular order). Condition (4.2.3) ensures that arriving earlier at satellite s using route m_2 does not allow to service a customer earlier than with route m_1 from the same satellite. Together, the latter two conditions ensure that every second-echelon route that can be supplied by m_2 can also be supplied by m_1.

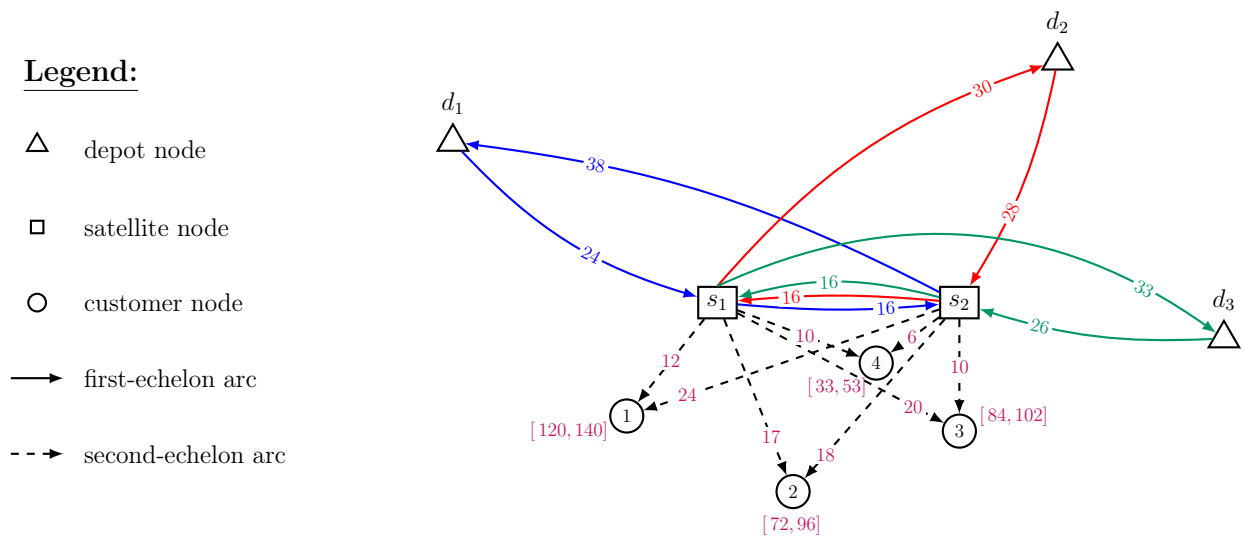

Figure 4.1 Example of a dominated first-echelon route

An example of the dominance occurring between first-echelon routes is presented in Figure 4.1, where quantities indicated on arcs represent both their costs and travel times while time windows are given next to the customer nodes. There are three first-echelon routes $m_1 = (d_2, s_2, s_1, d_2)$ (in red), $m_2 = (d_1, s_1, s_2, d_1)$ (in blue), and $m_3 = (d_3, s_2, s_1, d_3)$ (in green). In this example, m_2 is dominated by m_1. Indeed, conditions (4.2.2) and (4.2.1) are easily verified as both first-echelon routes visit the same subset of satellites and $c_{m_1} = 74 \leq c_{m_2} = 78$. Checking whether condition (4.2.3) holds for satellite s_2 is straightforward as using m_1 instead of m_2 allows for an earlier arrival time at satellite s_2, i.e. $t_{s_2}^{m_1} = 28 \leq t_{s_2}^{m_2} = 40$.

Condition (4.2.3) for s_1 is also verified. In fact, although m_2 arrives at satellite s_1 later than m_1 ($t_{s_1}^{m_1} = 44 > t_{s_1}^{m_2} = 24$), it is not possible to start service earlier at customers 1, 2, and 3 because $L_1^{s_1} = 108$, $L_2^{s_1} = 55$ and $L_3^{s_1} = 64$. For customer 4, when using m_1 or m_2, service can only begin from either satellites after the start of its time window ($L_4^{s_1} = 23$, $L_4^{s_2} = 27$, and $t_{s_1}^{m_1} > L_4^{s_1}$, $t_{s_1}^{m_2} > L_4^{s_1}$, $t_{s_2}^{m_1} > L_4^{s_2}$, and $t_{s_2}^{m_2} > L_4^{s_2}$). First-echelon routes m_1 and m_3 are non-dominated. Verifying that m_3 does not dominate m_1 is straightforward as condition (4.2.1) does not hold ($c_{m_3} = 75 > c_{m_1} = 74$). When checking whether m_1 dominates m_3, both (4.2.1) and (4.2.2) are satisfied. However, condition (4.2.3) is violated for satellite s_2. Indeed, customer 4 can only be reached before the start of its time window from s_2 when m_3 is used because $t_{s_2}^{m_3} = 26 \leq L_4^{s_2} = 27$ and $t_{s_2}^{m_1} = 28 > L_4^{s_2} = 27$.

Let us also introduce the concept of back-and-forth routes, that is, first-echelon routes visiting exactly one satellite. The set $\mathcal{M}$ contains $\left\lceil \frac{\sum_{i \in N^C} q_i}{Q^1} \right\rceil$ copies for each non-dominated back-and-forth route and one copy for each non-dominated first-echelon route visiting at least two satellites. For a given satellite $s \in N^S$, let $\mathcal{M}_s^1$ denote its set of back-and-forth route copies. For notational purposes, we assume that this set is ordered and we denote its last element by $\bar{m}_s^1$.

Let $\mathcal{L}$ be the set of feasible second-echelon routes, $\mathcal{L}_m \subseteq \mathcal{L}$ the subset of feasible second-echelon routes supplied by first-echelon route $m \in \mathcal{M}$, and $\mathcal{L}_{ms} \subseteq \mathcal{L}_m$ the subset of feasible second-echelon routes associated with satellite $s \in S(m)$ that are supplied by m. These sets are linked by the relations $\mathcal{L} = \bigcup_{m \in \mathcal{M}} \mathcal{L}_m$ and $\mathcal{L}_m = \bigcup_{s \in S(m)} \mathcal{L}_{ms}$ for all $m \in \mathcal{M}$. A second-echelon route $l \in \mathcal{L}$ is an ordered set of nodes $(s, i_1, i_2, ..., i_k, s)$, starting and ending at a given origin satellite $s \in N^S$, and visiting a sequence of customers $i_1, i_2, ..., i_k \in N^C$, $i_j \neq i_{j'}$ for $j, j' \in \{1, \ldots, k\}$, $j \neq j'$. Let c_l be its total cost, which is equal to the sum of the fixed vehicle cost f_2 and the total transportation costs. For each customer $i \in N^C$, second-echelon route l is associated with a binary parameter a_{il} equal to 1 if customer i is served by route l, and 0 otherwise. Denote by $Q_l = \sum_{i \in N^C} a_{il} q_i$ the total load transported in route l that must come from a single first-echelon route.

The 2E-VRPTW can then be modeled using the following route-based formulation. Let x_m and y_{ml} be binary variables equal to 1 if and only if first-echelon route $m \in \mathcal{M}$ and second-echelon route $l \in \mathcal{L}_m$ are used in the solution, respectively. The binary linear programming

model is:

$$\min \quad \sum_{m \in \mathcal{M}} c_m x_m + \sum_{m \in \mathcal{M}} \sum_{l \in \mathcal{L}_m} c_l y_{ml} \tag{4.2.4}$$

$$\text{s.t.} \quad \sum_{m \in \mathcal{M}} \sum_{l \in \mathcal{L}_m} a_{il} y_{ml} = 1, \qquad \forall i \in N^C, \tag{4.2.5}$$

$$\sum_{l \in \mathcal{L}_m} Q_l y_{ml} \leq Q^1 x_m, \qquad \forall m \in \mathcal{M}, \tag{4.2.6}$$

$$x_m \in \{0, 1\}, \qquad \forall m \in \mathcal{M}, \tag{4.2.7}$$

$$y_{ml} \in \{0, 1\}, \qquad \forall m \in \mathcal{M}, l \in \mathcal{L}_m. \tag{4.2.8}$$

The objective function (4.2.4) minimizes the total cost. Constraints (4.2.5) ensure that each customer is served by exactly one second-echelon vehicle. Constraints (4.2.6) synchronize the loads of first- and second-echelon vehicles, i.e., the total load transported on all second-echelon routes supplied by a given first-echelon vehicle must respect its capacity. Constraints (4.2.7) and (4.2.8) define the variable domains.

Model (4.2.4)–(4.2.8) is the same as the two-echelon, two-path formulation of Dellaert et al. (2019) except that it considers multiple variable copies for the first-echelon back-and-forth routes. To clarify how the assignment of multiple vehicles to the same first-echelon route is handled, Dellaert (2020) mentioned in a private communication that they consider multiple binary variable copies for certain instances (in category `Cb`; see Section 4.4.1) and integer variables for others (categories `Ca`, `Cc`, and `Cd`) for which they can prove that it yields a valid model. In general, it is not easy to prove its validity and, consequently, we have not opted to use such a model directly. Nevertheless, the alternative solution method described in Section 4.3.5 exploits a model with integer first-echelon route variables.

Given that the set $\mathcal{M}$ considers multiple copies for each first-echelon route visiting exactly one satellite, additional symmetry breaking constraints can be added to force the usage of the last route copies in sets $\mathcal{M}_s^1$, that is,

$$x_m \leq x_{m^+}, \qquad \forall s \in N^S, m \in \mathcal{M}_s^1 \setminus \{\bar{m}_s^1\}, \tag{4.2.9}$$

where m^+ indicates the route copy immediately after route m in $\mathcal{M}_s^1$.

In practice, the number of satellites is rather limited. Therefore, the number of first-echelon route variables in formulation (4.2.4)–(4.2.9) is not very large and they can be enumerated a priori. On the other hand, the number of second-echelon route variables is, typically, very large. To address this difficulty, we propose in the next section a BPC algorithm that

generates only second-echelon route variables as needed.

4.3 Branch-price-and-cut algorithm

BPC is the leading exact solution methodology for many VRPs Costa et al. (2019), especially when applied to a set-partitioning (or set-covering) formulation involving a huge number of variables. A BPC algorithm is a branch-and-bound algorithm in which column generation is used to compute lower bounds by solving linear relaxations, hereafter called master problems (MPs), and additional cuts may be added to strengthen these MPs. Column generation iterates between solving a restricted master problem (RMP) and one or several pricing problems. For the 2E-VRPTW, the RMP corresponds to the linear relaxation of formulation (4.2.4)–(4.2.9) defined over a subset of the second-echelon route variables $\mathcal{L}' \subseteq \mathcal{L}$. Solving it yields a pair of primal and dual solutions. Using this dual solution, the pricing problems, which are cast as a variant of the elementary shortest path problems with resource constraints (ESPPRC, see Section 4.3.1), search for negative reduced cost second-echelon route variables (columns) to add to the RMP before starting a new iteration. If none can be found, column generation stops with an optimal solution to the MP that provides a valid lower bound. To reduce the number of column generation iterations, deep dual-optimal inequalities (Section 4.3.2) are added to the MP. Additional valid inequalities (Section 4.3.3) may be generated to strengthen the lower bound before adding branching decisions (Section 4.3.4) if required.

Dellaert et al. (2019) highlighted the complexity of solving the 2E-VRPTW with BPC, i.e., handling dual variables of the load synchronization constraints in the pricing problem especially when having separate pricing problems for first- and second-echelon routes. In this paper, we propose a BPC algorithm where, like in Santos et al. (2015), all first-echelon routes are enumerated a priori because their number is not very large, and where the pricing problems only generate second-echelon routes. To further address the concerns raised by Dellaert et al. (2019), we solve one pricing problem per satellite and determine its associated first-echelon route through an *ad hoc* labeling algorithm. In this section, we explain in detail the different elements of our BPC algorithm and highlight an alternative model and algorithm (Section 4.3.5) that did not provide any significant speed-up for the tested instances, but could be useful for future work.

4.3.1 Pricing problems and labeling algorithm

The aim of the pricing problems is to find a negative reduced-cost feasible second-echelon route supplied by exactly one first-echelon route, starting and ending at the same satellite,

respecting the capacity of the second-echelon vehicle and the visited customer's time windows, or to prove that no such route exists. The reduced cost $\bar{c}_{ml}$ of variable y_{ml} associated with a second-echelon route $l \in \mathcal{L}_m$ supplied by a first-echelon route $m \in \mathcal{M}$ is computed as

$$\bar{c}_{ml} = c_l - \sum_{i \in N^C} a_{il}(\sigma_i + q_i \pi_m), \tag{4.3.1}$$

where σ_i, $i \in N^C$, and π_m, $m \in \mathcal{M}$, (with $\pi_m \leq 0$) are the dual variables associated with constraints (4.2.5) and (4.2.6), respectively. The pricing problems are solved using a dynamic programming labeling algorithm. Given the expression of the reduced cost, this suggests that there should be one pricing problem per pair of first-echelon route and satellite in the 2E-VRPTW. In what follows, we propose one pricing problem per satellite which selects the associated first-echelon route, thus reducing the number of pricing problems in our algorithm and yielding a more efficient one.

The pricing problem associated with satellite $s \in N^S$, denoted PP^s, can be modeled as a variant of the ESPPRC (see Irnich and Desaulniers, 2005) and defined on a graph $G_s = (N_s, A_s)$, where N_s and A_s are its specific sets of nodes and arcs, respectively. The node set $N_s = N^C \cup \{s\} \cup \{s'\}$ is comprised of the customer nodes, satellite s and a duplicate of satellite s denoted s', which represents the origin and destination copies of the satellite, respectively. The duplicate satellite s' is associated with a null demand, i.e., $q_{s'} = 0$. The arc set $A_s = \{(s,i) \in A^2 : i \in N^C\} \cup \{(i_1, i_2) \in A^2 : i_1, i_2 \in N^C\} \cup \{(i, s') : i \in N^C\}$ contains arcs between the origin satellite s and customers, between customers, and between customers and the destination satellite s'. Each arc $(i, s') \in A_s$ is associated with a transportation cost $c_{is'} = c_{si}$ and a travel time $t_{is'} = t_{si}$. Moreover, the cost c_{si} of an arc $(s,i) \in A_s, s \in N^S, i \in N^C$ exiting the satellite is modified to include the transportation cost as well as the fixed vehicle cost f_2, while the cost $c_{is'}$ only includes the transportation cost.

In the labeling algorithm, a label E is a vector of information that represents a partial path $R(E) = (s, i_1, i_2, ..., \eta(E))$ starting from satellite s, supplied by a specific first-echelon route, and ending at a given node $\eta(E)$. The components of label E are:

- $\eta(E)$: the last visited node of the partial path;
- $\bar{c}(E)$: the cumulated reduced cost of the partial path;
- $t(E)$: the earliest service start time at node $\eta(E)$;
- $q(E)$: the total delivered quantity on the partial path;
- $\mathcal{U}(E)$: the set of unreachable nodes with respect to elementarity conditions, capacity and time window constraints;
- $m(E)$: the first-echelon route supplying the partial path.

The labeling algorithm starts with multiple initial labels associated with node s, namely, exactly one label E^m is created for each first-echelon route $m \in \mathcal{M}$ that visits satellite s, i.e., $s \in S(m)$. Given the definition of $\mathcal{M}$, which has multiple copies of first-echelon routes visiting exactly one satellite, there will be one label associated with each route in $\mathcal{M}_s^1$. The components of E^m are given by

$$\begin{aligned}
&\eta(E^m) = s,\\
&\bar{c}(E^m) = 0,\\
&t(E^m) = t_s^m,\\
&q(E^m) = 0,\\
&\mathcal{U}(E^m) = \{i \in N^C : t_s^m + t_{si} > \overline{w}_i\},\\
&m(E^m) = m.
\end{aligned}$$

Note that the start of the service time at node s, the set of unreachable nodes, and the value of the associated dual variable π_m (needed to compute the reduced cost) depend on the first-echelon route m associated with label E^m.

These initial labels are propagated through network G_s using extension functions to create new partial paths, eventually reaching destination node s'. For each extension of a label E along an arc $(\eta(E), j) \in A_s$ such that $j \notin \mathcal{U}(E)$, which by definition includes unreachable nodes with respect to elementarity conditions, time windows and capacity constraints, a new label E' is created and the components of the new label are computed as follows:

$$\eta(E') = j, \tag{4.3.2}$$

$$\bar{c}(E') = \bar{c}(E) + c_{\eta(E),j} - \sigma_j - q_j \pi_{m(E)}, \tag{4.3.3}$$

$$t(E') = \max\{\underline{w}_j, t(E) + t_{\eta(E),j}\}, \tag{4.3.4}$$

$$q(E') = q(E) + q_j, \tag{4.3.5}$$

$$\mathcal{U}(E') = \mathcal{U}(E) \cup \{j\} \cup \{i \in N^C : t(E') + t_{ji} > \overline{w}_i \text{ or } q(E') + q_i > Q^2\}, \tag{4.3.6}$$

$$m(E') = m(E), \tag{4.3.7}$$

where $\underline{w}_{s'} = 0$ and $\sigma_{s'} = 0$. Equation (4.3.2) sets the resident node of E' to node j. The cumulated reduced cost along the partial path $R(E') = (s, ..., i, j)$ is computed with (4.3.3) and comprises the cumulated reduced cost from the initial label E, the transportation cost along arc $(\eta(E), j)$, the dual variable associated with constraint (4.2.5) for node j, and the dual variable associated with (4.2.6) for the specific first-echelon route supplying that route multiplied by the demand at node j. The earliest service start time at node j and the total

load along path $R(E')$ are computed with equations (4.3.4) and (4.3.5), respectively. The set of unreachable nodes is updated with equation (4.3.6) by including node j (unreachable due to elementarity conditions) as well as all nodes that are unreachable with respect to time windows and capacity constraints. Finally, equation (4.3.7) states that the first-echelon route supplying path $R(E')$ is the same as the one supplying the initial path $R(E)$. This information is necessary to compute the reduced cost in subsequent label extensions.

To avoid enumerating all feasible paths, a dominance rule is applied to discard unpromising labels. This rule specifies that a label E_1 dominates a label E_2 if

$$\eta(E_1) = \eta(E_2), \tag{4.3.8}$$

$$\bar{c}(E_1) \leq \bar{c}(E_2) + \min\{0, (Q^2 - q(E_2))(\pi_{m(E_1)} - \pi_{m(E_2)})\}, \tag{4.3.9}$$

$$t(E_1) \leq t(E_2), \tag{4.3.10}$$

$$q(E_1) \leq q(E_2), \tag{4.3.11}$$

$$\mathcal{U}(E_1) \subseteq \mathcal{U}(E_2). \tag{4.3.12}$$

Condition (4.3.8) ensures that the labels represent partial paths ending at the same node. Condition (4.3.9) allows the dominance of labels supplied by different first-echelon routes by imposing that the reduced cost of the first path is at most the reduced cost of the second plus the minimum (negative) difference between the dual contributions of constraints (4.2.6) that can occur in an extension of both labels. Conditions (4.3.10) and (4.3.11) state that it is better to have a partial path for which neither the time, nor the total quantity are greater. Condition (4.3.12) imposes that a label can dominate another one only if its set of unreachable nodes is included in the set of unreachable nodes of the dominated label. Any dominated path can be discarded unless two paths dominate each other (i.e., when all relations (4.3.8)–(4.3.12) are satisfied at equality). In this case, one of the two labels must be kept.

Proposition 4.1. *Conditions* (4.3.8)–(4.3.12) *constitute a valid dominance rule, i.e., at least one optimal path can be found by the labeling algorithm.*

Proof 4.1. To prove this statement, we show that every feasible extension of $R(E_2)$ is also a feasible extension of $R(E_1)$, and that the reduced cost of the path obtained by extending E_1 is no greater than that obtained by extending E_2.

Let ω be a feasible extension of $R(E_2)$ and denote by $R'_1 = (R(E_1), \omega)$ and $R'_2 = (R(E_2), \omega)$ the paths obtained by extending $R(E_1)$ and $R(E_2)$ with ω, respectively. Let E'_1 and E'_2 be the labels representing these two paths, respectively. Because R'_2 is feasible and the extension functions (4.3.10) and (4.3.11) are non-decreasing, R'_1 is also feasible with respect to the time

windows and the vehicle capacity. Furthermore, because ω is a feasible extension for $R(E_2)$ and condition (4.3.12) holds, R'_1 is elementary and, thus, feasible.

Next, let us show that $\bar{c}(E'_1) \leq \bar{c}(E'_2)$. For convenience, denote by $c(\omega)$ the total cost of the arcs traversed along ω, $\sigma(\omega)$ the sum of the dual variables of constraints (4.2.5) associated with the customers visited along ω (except at node $\eta(E_2)$), and $q(\omega)$ the total demand of these customers. The reduced cost of paths R'_1 and R'_2 are equal to $\bar{c}(E'_1) = \bar{c}(E_1) + c(\omega) - \sigma(\omega) - q(\omega)\pi_{m(E_1)}$ and $\bar{c}(E'_2) = \bar{c}(E_2) + c(\omega) - \sigma(\omega) - q(\omega)\pi_{m(E_2)}$.

If $\pi_{m(E_1)} \geq \pi_{m(E_2)}$, condition (4.3.9) writes as $\bar{c}(E_1) \leq \bar{c}(E_2)$, which implies, together with $\pi_{m(E_1)} \geq \pi_{m(E_2)}$, that $\bar{c}(E'_1) \leq \bar{c}(E'_2)$. On the other hand, if $\pi_{m(E_1)} < \pi_{m(E_2)}$, condition (4.3.9) is equivalent to $\bar{c}(E_1) \leq \bar{c}(E_2) + (Q^2 - q(E_2))(\pi_{m(E_1)} - \pi_{m(E_2)})$. Given that $q(\omega) \leq Q^2 - q(E_2)$, we get that

$$
\begin{aligned}
& \bar{c}(E_1) \leq \bar{c}(E_2) + (Q^2 - q(E_2))(\pi_{m(E_1)} - \pi_{m(E_2)}) \\
\Longrightarrow\ & \bar{c}(E_1) \leq \bar{c}(E_2) + q(\omega)(\pi_{m(E_1)} - \pi_{m(E_2)}) \\
\Longrightarrow\ & \bar{c}(E_1) - q(\omega)\pi_{m(E_1)} \leq \bar{c}(E_2) - q(\omega)\pi_{m(E_2)} \\
\Longrightarrow\ & \bar{c}(E_1) + c(\omega) - \sigma(\omega) - q(\omega)\pi_{m(E_1)} \leq \bar{c}(E_2) + c(\omega) - \sigma(\omega) - q(\omega)\pi_{m(E_2)} \\
\Longrightarrow\ & \bar{c}(E'_1) \leq \bar{c}(E'_2). \ \square
\end{aligned}
$$

In the BPC literature for vehicle routing (Costa et al., 2019), several acceleration techniques have been proposed to speed up the pricing problem solution process. Given the ESPPRC variant considered and the fact that the routes generated in our tests do not contain a large number of customers, we only apply decremental state-space relaxation (DSSR) and ng-route relaxation. Here, we briefly describe these strategies and refer the interested readers to Costa et al. (2019) and the references therein, for further details.

DSSR (Boland et al., 2006; Righini and Salani, 2008) consists of first solving the pricing problem without considering the unreachable customer resources, highly increasing the possibility of dominating labels. If elementary routes with a negative reduced cost are found, they are added to the RMP before starting a new column generation iteration. Otherwise, if at least one negative reduced cost route with a cycle is found, a customer resource is added to forbid this cycle and the labeling algorithm is invoked again. This process is repeated until finding at least one negative reduced cost elementary route or no negative reduced cost routes at all. As suggested by Contardo et al. (2015), DSSR can be initialized at each iteration from the final subset of unreachable customer resources generated in the last column generation iteration.

The ng-route relaxation (Baldacci et al., 2011) also aims at increasing the number of dominated labels by resetting the value of some customer resources to zero when certain conditions are met, introducing the possibility of generating routes with cycles (and, thus, relaxing the MP). With each customer $i \in N^C$ is associated a neighborhood $\mathcal{N}_i$ that contains a subset of its closest customers. When a partial route reaches a customer i, its customer resource is set to one. When extending this route through a sequence of customers, this resource is reset to zero when it visits a customer $j \in N^C$ such that $i \notin \mathcal{N}_j$ and i is still reachable. When the size of the neighborhoods is well adjusted, the allowed cycles are not attractive and most generated routes are elementary, yielded a strong relaxation. Following some preliminary experiments, we have chosen to use neighborhoods containing 15 customers.

4.3.2 Stabilization by dual inequalities

Column generation may suffer from convergence issues stemming mainly from degeneracy of the MP and the instability of its associated dual variables as their values may vary greatly from one column generation iteration to the next before converging to an optimal dual solution (Lübbecke and Desrosiers, 2005). Several works in the literature have addressed the stabilization of dual variables. Stabilization techniques can be classified into trust-region-based methods (Du Merle et al., 1999; Oukil et al., 2007), smoothing methods (Pessoa et al., 2018), interior-point methods (Rousseau et al., 2007; Gondzio et al., 2013), and dual-inequality-based methods (Ben Amor et al., 2006; Gschwind and Irnich, 2016; Yarkony et al., 2020). In our work, we focus on the latter technique as prior knowledge on the set of optimal dual solutions is used to derive problem-specific dual inequalities.

This section is divided in three parts. First, we introduce in Section 4.3.2.1 the proposed dual inequalities that we call the *transfer inequalities* (TIs). Second, we specify the subset of TIs that are generated in our BPC algorithm (Section 4.3.2.2). Finally, we discuss the feasibility of the integer solutions when some TIs are binding (Section 4.3.2.3).

4.3.2.1 Definition of the transfer inequalities

The dual inequalities developed in this section are linear inequalities involving only the dual variables π_m, $m \in \mathcal{M}$, associated with constraints (4.2.6). Before formally introducing them, we provide the following related definitions.

Consider the dual of a feasible and bounded linear program, where μ denotes the vector of its r dual variables. Let D^* be its set of optimal solutions. A dual inequality $\Lambda^\top \mu \leq \lambda$ (with $\Lambda \in \mathbb{R}^r$ and $\lambda \in \mathbb{R}$) is a *dual-optimal inequality* if it is satisfied by all solutions

in D^*, i.e., if $D^* \subseteq \{\mu \in \mathbb{R}^r \ : \ \Lambda^\top \mu \leq \lambda\}$. Let J be an index set. If a set of dual inequalities $\Lambda_j^\top \mu \leq \lambda_j$, $j \in J$, is satisfied by at least one optimal solution in D^*, i.e., if $D^* \cap \{\mu \in \mathbb{R}^r \ : \ \Lambda_j^\top \mu \leq \lambda_j, \forall j \in J\} \neq \emptyset$, then these dual inequalities are said to be *deep dual-optimal inequalities* (DDOIs). The dual inequalities that we introduce below form a set of DDOIs.

The proposed DDOIs come from the following intuition. Recall that the dual value π_m of the constraint (4.2.6) associated with a first-echelon route $m \in \mathcal{M}$ indicates the rate of change of the optimal value when its right-hand side increases. Consider two distinct first-echelon routes $m_1, m_2 \in \mathcal{M}$ such that m_2 can supply at least all second-echelon routes that can be supplied by m_1, i.e., $\mathcal{L}_{m_1} \subseteq \mathcal{L}_{m_2}$ which implies that $S(m_1) \subseteq S(m_2)$. Allocating extra capacity to route m_2, i.e., increasing the right-hand side of its associated constraint (4.2.6), to supply additional second-echelon routes should be at least as advantageous as allocating it to m_1 because of its additional flexibility. Thus, we can suspect that $\pi_{m_2} \leq \pi_{m_1}$ in at least one optimal dual solution. The next proposition introduces the DDOIs.

Proposition 4.2. *Let $\mathcal{K} = \{(m_1, m_2) \in \mathcal{M} \times \mathcal{M} \ : \ m_1 \neq m_2, \ \mathcal{L}_{m_1} \subseteq \mathcal{L}_{m_2}\}$. The dual inequalities*

$$\pi_{m_1} \geq \pi_{m_2}, \qquad \forall (m_1, m_2) \in \mathcal{K}, \tag{4.3.13}$$

called the TIs, form a set of DDOIs for the linear relaxation of (4.2.4)–(4.2.9).

Proof 4.2. The linear relaxation of (4.2.4)–(4.2.9), i.e., the MP, writes as:

$$\min_{x,y} \quad \sum_{m \in \mathcal{M}} c_m x_m + \sum_{m \in \mathcal{M}} \sum_{l \in \mathcal{L}_m} c_l y_{ml} \tag{4.3.14}$$

$$\text{s.t.} \quad \sum_{m \in \mathcal{M}} \sum_{l \in \mathcal{L}_m} a_{il} y_{ml} = 1, \qquad \forall i \in N^C, \tag{4.3.15}$$

$$\sum_{l \in \mathcal{L}_m} Q_l y_{ml} - Q^1 x_m \leq 0, \qquad \forall m \in \mathcal{M}, \tag{4.3.16}$$

$$x_m - x_{m^+} \leq 0, \qquad \forall s \in N^S, m \in \mathcal{M}_s^1 \setminus \{\bar{m}_s^1\}, \tag{4.3.17}$$

$$x_m \leq 1, \qquad \forall m \in \mathcal{M} \tag{4.3.18}$$

$$x_m \geq 0, \qquad \forall m \in \mathcal{M}, \tag{4.3.19}$$

$$y_{ml} \geq 0, \qquad \forall m \in \mathcal{M}, l \in \mathcal{L}_m, \tag{4.3.20}$$

where the constraints $y_{ml} \leq 1$, $m \in \mathcal{M}$, $l \in \mathcal{L}_m$, are not considered because they are redundant with (4.3.15) and (4.3.20). Let σ_i ($i \in N^C$), π_m ($m \in \mathcal{M}$), γ_{sm} ($s \in N^S$, $m \in \mathcal{M}_s^1 \setminus \{\bar{m}_s^1\}$), and β_m ($m \in \mathcal{M}$), be the dual variables associated with the constraints (4.3.15)–(4.3.18), respectively. To prove that the TIs are valid DDOIs, we show that, if the

MP has an optimal dual solution, then there always exists one such that the TIs (4.3.13) hold.

Let us assume that the MP has an optimal dual solution which we denote $(\sigma^*, \pi^*, \gamma^*, \beta^*)$ in vectorial form. If there exists a pair of distinct first-echelon routes $(m_1, m_2) \in \mathcal{K}$ such that $\pi^*_{m_1} < \pi^*_{m_2}$, then the solution $(\hat{\sigma}, \hat{\pi}, \hat{\gamma}, \hat{\beta})$ given by

$$\begin{cases} \hat{\sigma} & = & \sigma^* & \\ \hat{\pi}_m & = & \pi^*_m, & \forall m \in \mathcal{M} \setminus \{m_1\} \\ \hat{\pi}_{m_1} & = & \pi^*_{m_2} & \\ \hat{\gamma} & = & \gamma^* & \\ \hat{\beta} & = & \beta^*, & \end{cases}$$

is a dual optimal solution that satisfies the TI (4.3.13) for the pair (m_1, m_2). To prove this, it suffices to show that solution $(\hat{\sigma}, \hat{\pi}, \hat{\gamma}, \hat{\beta})$ is dual feasible because its cost is equal to that of $(\sigma^*, \pi^*, \gamma^*, \beta^*)$. This solution is dual feasible if and only if the reduced costs of all variables x_m, $m \in \mathcal{M}$, and y_{ml}, $m \in \mathcal{M}$, $l \in \mathcal{L}_m$, are non-negative. Let $\bar{c}^x_m(\sigma, \pi, \gamma, \beta)$ (resp., $\bar{c}^y_{ml}(\sigma, \pi, \gamma, \beta)$) be the reduced cost of variable x_m (resp., y_{ml}) with respect to a dual solution $(\sigma, \pi, \gamma, \beta)$. Because the dual solutions $(\sigma^*, \pi^*, \gamma^*, \beta^*)$ and $(\hat{\sigma}, \hat{\pi}, \hat{\gamma}, \hat{\beta})$ differ only by the value of π_{m_1}, we have that

$$\bar{c}^x_m(\hat{\sigma}, \hat{\pi}, \hat{\gamma}, \hat{\beta}) = \bar{c}^x_m(\sigma^*, \pi^*, \gamma^*, \beta^*) \geq 0, \qquad \forall m \in \mathcal{M} \setminus \{m_1\} \tag{4.3.21}$$

$$\bar{c}^y_{ml}(\hat{\sigma}, \hat{\pi}, \hat{\gamma}, \hat{\beta}) = \bar{c}^y_{ml}(\sigma^*, \pi^*, \gamma^*, \beta^*) \geq 0, \qquad \forall m \in \mathcal{M} \setminus \{m_1\}, l \in \mathcal{L}_m. \tag{4.3.22}$$

Furthermore, we find that

$$\bar{c}^x_{m_1}(\hat{\sigma}, \hat{\pi}, \hat{\gamma}, \hat{\beta}) = \bar{c}^x_{m_1}(\sigma^*, \pi^*, \gamma^*, \beta^*) - Q^1(\pi^*_{m_1} - \pi^*_{m_2}) \geq 0 \tag{4.3.23}$$

because $\bar{c}^x_{m_1}(\sigma^*, \pi^*, \gamma^*, \beta^*) \geq 0$ and $\pi^*_{m_1} < \pi^*_{m_2}$, and that

$$\bar{c}^y_{m_1 l}(\hat{\sigma}, \hat{\pi}, \hat{\gamma}, \hat{\beta}) = \bar{c}^y_{m_2 l}(\sigma^*, \pi^*, \gamma^*, \beta^*) \geq 0, \qquad \forall l \in \mathcal{L}_{m_1}, \tag{4.3.24}$$

where the existence of variable $y_{m_2 l}$ is guaranteed by the condition $\mathcal{L}_{m_1} \subseteq \mathcal{L}_{m_2}$ used to define $\mathcal{K}$. Consequently, $(\hat{\sigma}, \hat{\pi}, \hat{\gamma}, \hat{\beta})$ is a dual optimal solution.

Applying this transformation recursively on every pair of first-echelon routes $(m_1, m_2) \in \mathcal{K}$ for which TI (4.3.13) is violated yields a dual solution that satisfies all TIs. Note that the recursive procedure is finite because i) the dual variables π_m, $m \in \mathcal{M}$, only take values in

the set $\{\pi^*_m\}_{m\in\mathcal{M}}$ during this procedure, ii) one dual value increases at every iteration, and iii) all dual values π^*_m, $m \in \mathcal{M}$, are upper bounded by 0. □

Adding the TIs (4.3.13) to the dual of the MP is done by adding new non-negative variables $u_{m_1m_2}$, $(m_1, m_2) \in \mathcal{K}$, to the MP. These variables, called the *transfer variables*, have no cost and only appear in the constraints (4.3.16) which become:

$$\sum_{l\in\mathcal{L}_m} Q_l y_{ml} + \sum_{(m_1,m)\in\mathcal{K}} u_{m_1m} - \sum_{(m,m_2)\in\mathcal{K}} u_{mm_2} \leq Q^1 x_m, \qquad \forall m \in \mathcal{M}. \tag{4.3.25}$$

Observe that the presence of a non-negative variable $u_{m_1m_2}$ in the MP allows for a possible load transfer, at no additional cost, from first-echelon route m_1 to route m_2. To illustrate this, consider that $\mathcal{K}$ contains a single pair (m_1, m_2), i.e., $\mathcal{K} = \{(m_1, m_2)\}$. Then, the constraints (4.3.25) for m_1 and m_2 write as:

$$\sum_{l\in\mathcal{L}_{m_1}} Q_l y_{m_1l} - u_{m_1m_2} \leq Q^1 x_{m_1} \tag{4.3.26}$$

$$\sum_{l\in\mathcal{L}_{m_2}} Q_l y_{m_2l} + u_{m_1m_2} \leq Q^1 x_{m_2}. \tag{4.3.27}$$

Hence, it might be possible to have a solution where $\sum_{l\in\mathcal{L}_{m_1}} Q_l y_{m_1l} = u_{m_1m_2} = \zeta$ and $x_{m_1} = 0$, where $\zeta > 0$ is a positive load value. This corresponds to transferring the load $\sum_{l\in\mathcal{L}_{m_1}} Q_l y_{m_1l}$ assigned to route m_1 to route m_2 and induced the name of the TIs. For the MP, such a complete or partial transfer is acceptable because $\mathcal{L}_{m_1} \subseteq \mathcal{L}_{m_2}$ when $(m_1, m_2) \in \mathcal{K}$, i.e., the second-echelon routes associated with m_1 that are part of this solution could also be generated as associated with m_2. The generation of these extra columns is, however, not necessary when the TIs are considered.

4.3.2.2 Generated transfer inequalities

To avoid burdening the MP by inserting too many transfer variables, we restrain the set $\mathcal{K}$ to a subset of first-echelon route pairs. In fact, when designing our algorithm, we observed that violated TIs were frequently induced by certain pairs of routes $(m_1, m_3) \in \mathcal{K}$ of the form $m_k = (d, s^k_1, \ldots, s^k_{j_k}, d)$, $k = 1, 3$, where d is a common depot, s^k_j, $j = 1, \ldots, j_k$, represent distinct satellites, $j_3 > j_1$, and $s^1_j = s^3_j$ for all $j \in \{1, \ldots, s^1_{j_1}\}$. A typical example is given by $m_1 = (d, s_1, d)$ and $m_3 = (d, s_1, s_2, s_3, d)$ which entails the TI $\pi_{m_1} \geq \pi_{m_3}$. Furthermore, considering also route $m_2 = (d, s_1, s_2, d)$, we can identify redundant TIs. Indeed, given the structure of the routes m_1, m_2 and m_3, we can deduce that $\mathcal{L}_{m_1} \subseteq \mathcal{L}_{m_2}$ and $\mathcal{L}_{m_2} \subseteq \mathcal{L}_{m_3}$, yielding the additional TIs $\pi_{m_1} \geq \pi_{m_2}$ and $\pi_{m_2} \geq \pi_{m_3}$, that, together, dominate $\pi_{m_1} \geq \pi_{m_3}$.

Based on these observations, we have decided to only generate the TIs for the pairs of first-echelon routes $(m_1, m_2) \in \mathcal{M} \times \mathcal{M}$ such that: 1) $\mathcal{L}_{m_1} \subseteq \mathcal{L}_{m_2}$; 2) m_1 and m_2 are associated with the same depot; 3) all satellites in $S(m_1)$ are visited in the same order by both routes; and 4) m_2 visits a single satellite that is not visited by m_1, i.e. $|S(m_2) \setminus S(m_1)| = 1$. In the following, we keep using $\mathcal{K}$ to denote this subset of route pairs.

4.3.2.3 Integer solution feasibility

To accelerate column generation, the TIs (4.3.13) can be added to the MP in all branch-and-bound nodes of the search tree. In fact, it is easy to prove that the valid inequalities and branching decisions described in Sections 4.3.3 and 4.3.4, respectively, do not invalidate the TIs. Nevertheless, including the TIs yields a relaxation as considering constraints (4.3.25) instead of constraints (4.2.6) enlarge the solution space (Figure 4.2 illustrates such an example). Consequently, there is no guarantee that an integer solution to model (4.2.4)–(4.2.5), (4.3.25), (4.2.7)–(4.2.9) is feasible for model (4.2.4)–(4.2.9). On the one hand, it is easy to prove that it is feasible when all transfer variables are equal to 0, but it may not be the case when at least one transfer variable takes a positive value as illustrated by the following example.

Consider a 2E-VRPTW instance with $|N^C| = 22$, $|N^S| = 3$, $|N^D| = 1$, $Q^1 = 200$ and $Q^2 = 50$. Figure 4.2 presents an integer solution for this instance. There are three first-echelon routes m_1, m_2, and m_3 for which $x_{m_1} = 0$ and $x_{m_2} = x_{m_3} = 1$. We assume that $\mathcal{K} = \{(m_1, m_2), (m_1, m_3)\}$, allowing the possibility to transfer load supplied by route m_1 to routes m_2 and m_3. The solution also includes nine second-echelon routes l_j, $j = 1, \ldots, 9$. The first three with $Q_{l_1} = Q_{l_2} = Q_{l_3} = 40$ are assigned to route m_1, i.e., $y_{m_1 l_1} = y_{m_1 l_2} = y_{m_1 l_3} = 1$. The next three with $Q_{l_4} = Q_{l_5} = 50$ and $Q_{l_6} = 40$ are assigned to route m_2, i.e., $y_{m_2 l_4} = y_{m_2 l_6} = y_{m_2 l_6} = 1$. The last three with $Q_{l_7} = 50$ and $Q_{l_8} = Q_{l_9} = 40$ are assigned to route m_3, i.e., $y_{m_3 l_7} = y_{m_3 l_8} = y_{m_3 l_9} = 1$. Given that $x_{m_1} = 0$, the total load $Q_{l_1} + Q_{l_2} + Q_{l_3} = 120$ of the routes assigned to m_1 is transferred to routes m_2 and m_3 through the transfer variables $u_{m_1 m_2}$ and $u_{m_1 m_3}$. It is easy to determine that $u_{m_1 m_2} \in [50, 60]$ and $u_{m_1 m_3} \in [60, 70]$ to ensure the full transfer ($u_{m_1 m_2} + u_{m_1 m_3} \geq 120$) while taking into account the load already assigned to m_2 and m_3 and the vehicle capacity. These intervals indicate that, because of the load synchronization constraints, it is not possible to reassign the second-echelon routes l_1 to l_3 to the first-echelon routes m_2 and m_3 without rearranging them. Consequently, we can conclude that this integer solution is infeasible.

To take this possibility into account, the proposed BPC algorithm includes the following features:

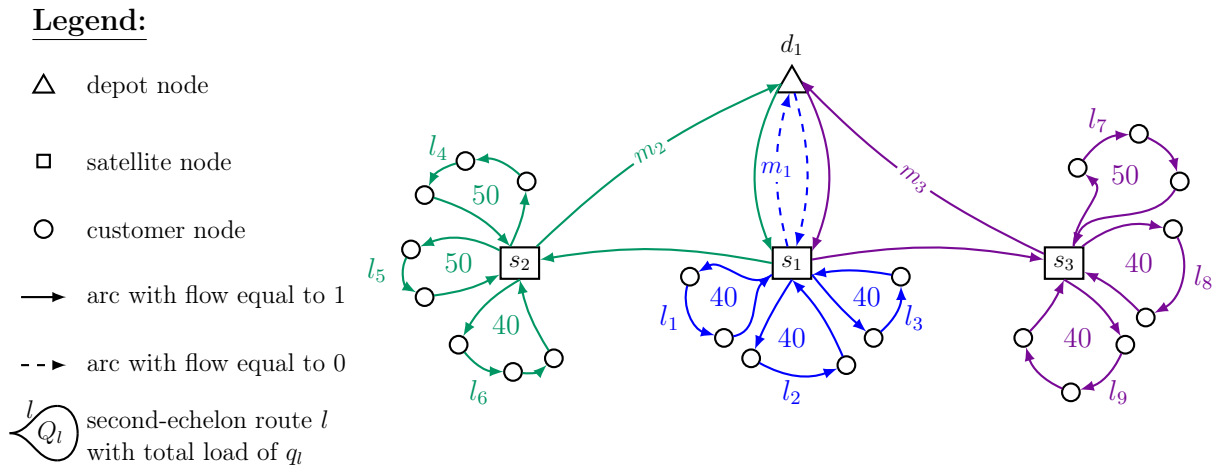

Figure 4.2 Example of an infeasible integer solution

1. An integer solution is deemed feasible only when all transfer variables are equal to 0.
2. When column generation stops with an integer solution in which some transfer variables take a positive value, these transfer variables are fixed to 0 before continuing the column generation process at the current branch-and-bound node.

4.3.3 Valid inequalities

Three families of valid inequalities are used in our BPC algorithm: *rounded capacity inequalities* (RCIs), *subset-row inequalities* (SRIs), and *visited satellite inequalities* (VSIs). In this section, we formulate the inequalities in each family and explain how they are separated.

Introduced by Laporte and Nobert (1983), the RCIs have been shown to be very effective for VRPs (Naddef and Rinaldi, 2002; Semet et al., 2014; Poggi and Uchoa, 2014) and have been used for variants with multiple echelons (e.g., Santos et al., 2015). For a given subset of customers $\mathcal{C} \subseteq N^C$, a lower bound on the number of second-echelon vehicles required to deliver the total demand of the customers in $\mathcal{C}$ can be computed as $\xi(\mathcal{C}) = \left\lceil \frac{\sum_{i \in \mathcal{C}} q_i}{Q^2} \right\rceil$. In addition, let b^l_{uv} be a binary parameter equal to 1 if arc $(u, v) \in A^2$ is used in second-echelon route $l \in \mathcal{L}$, and 0 otherwise, and let $\delta(\mathcal{C})$ denote the set of arcs exiting the set $\mathcal{C}$, i.e., $\delta(\mathcal{C}) = \{(u, v) \in A^2 | u \in \mathcal{C}, v \notin \mathcal{C}\}$. The RCIs can then be expressed as:

$$\sum_{m \in \mathcal{M}} \sum_{l \in \mathcal{L}_m} \sum_{(u,v) \in \delta(\mathcal{C})} b^l_{uv} y_{ml} \geq \xi(\mathcal{C}), \qquad \forall \mathcal{C} \subseteq N^C. \tag{4.3.28}$$

Because separating these inequalities is NP-hard (Naddef and Rinaldi, 2002), we heuristically

separate them using the procedure of Lysgaard et al. (2004). In the pricing problems, the dual variable of a generated RCI defined for a customer subset $\mathcal{C}$ must be subtracted from $\bar{c}(E)$ in the extension function (4.3.3) of a label E if the arc $(\eta(E), j) \in \delta(\mathcal{C})$.

We also add a priori the following single RCI defined on the first-echelon routes:

$$\sum_{m \in \mathcal{M}} x_m \geq \left\lceil \frac{\sum_{i \in N^C} q_i}{Q^1} \right\rceil. \tag{4.3.29}$$

SRIs are rank-one Chvátal-Gomory inequalities defined over a subset of set-partitioning constraints (4.2.5) and were introduced by Jepsen et al. (2008) for the VRPTW. As commonly suggested by several other authors, we consider only a subset of them, namely, those that are defined for subsets of three customers. These SRIs can be expressed as:

$$\sum_{m \in \mathcal{M}} \sum_{l \in \mathcal{L}_m} \left\lfloor \frac{\sum_{i \in \mathcal{C}} a_{il}}{2} \right\rfloor y_{ml} \leq 1, \qquad \forall \mathcal{C} \subseteq N^C,\ |\mathcal{C}| = 3. \tag{4.3.30}$$

They are separated through enumeration of the subset of three customers that appear pairwise in a least one route of the current linear relaxation solution. While they have been shown to be very effective for different VRP variants, their application requires additional resources in the pricing problems to handle their dual variables, which increases the pricing problem complexity (see, e.g., Desaulniers et al., 2011). Indeed, when executing the labeling algorithm, the dual of an SRI associated with a subset $\mathcal{C}$ needs to be subtracted from the reduced cost of a path only when a second customer in $\mathcal{C}$ is visited. Consequently, a resource for each SRI is defined to count the number of customers visited in its associated set $\mathcal{C}$. Moreover, the dominance rule needs to be modified to foresee that the duals of some SRIs might be subtracted in a feasible extension of the dominating label but not in the dominated label (for details, see Jepsen et al., 2008; Costa et al., 2019).

Introduced by Marques et al. (2020) for the 2E-CVRP, the VSIs impose that a customer $i \in N^C$ cannot be visited by a second-echelon route departing from a satellite $s \in N^S$ if that satellite is not visited by at least one first-echelon route. These inequalities can be formulated as:

$$\sum_{m \in \mathcal{M} | s \in S(m)} x_m \geq \sum_{m \in \mathcal{M} | s \in S(m)} \sum_{l \in \mathcal{L}_{ms}} a_{il} y_{ml},\ \forall i \in N^C,\ s \in N^S. \tag{4.3.31}$$

They are separated by enumeration. In the pricing problem PP^s, $s \in N^S$, the dual variable of a generated VSI defined for a customer $i \in N^C$ and a satellite s must be added to $\bar{c}(E)$ in the extension function (4.3.3) of a label E if the arc $(\eta(E), j)$ is such that $j = i$.

When searching for violated inequalities, we look first for VSIs. If no violated VSIs are found, then we search for violated RCIs. Finally, if no violated RCIs are identified, we look for violated SRIs. Several rounds of cuts can be added per branch-and-bound node. However, given that the SRIs can significantly increase the difficulty of solving the pricing problems, a maximum of 10 SRIs (the most violated) can be added in each round and this type of inequalities is not separated at nodes with a depth larger than 10 in the search tree.

4.3.4 Branching

To derive integer solutions, we have developed *ad hoc* branching rules for the 2E-VRPTW which are compatible with the pricing problems. These branching rules are inspired by existing branching rules for the VRPTW, that is, branching on the number of vehicles and branching on the arcs, and are implemented in a hierarchical fashion.

All branching rules create two child nodes in the search tree. Some of them are handled by adding constraints to the MP, others by fixing some variables either in the MP or in the pricing problems. When branching constraints involving second-level route variables are added to the MP, their associated dual variables must be handled by the pricing problems. For the sake of conciseness, we do not present such details and refer to Desaulniers et al. (2011) for further information. In the following, we explain the proposed branching rules, including the hierarchy between them which was determined based on preliminary computational results.

First, if the total flow on a back-and-forth first-echelon route (i.e., the sum of the flows over its copies) is fractional, we branch on this flow. The lower bound is enforced by fixing to 1 the value of some route-copy variables in the MP. Similarly, the upper bound is imposed by fixing to 0 the value of some route-copy variables. Moreover, in the latter case, when solving the pricing problem associated with the satellite visited by this route, the initial labels corresponding to the fixed route copies are not created in the labeling algorithm.

Second, three additional branching rules (with the same priority) can be imposed: on the number of first-echelon vehicles, on the number of second-echelon vehicles, and on the number of first-echelon vehicles per satellite. All corresponding branching decisions are enforced by adding constraints in the MP.

Third, when no candidates were found for the above rules, the following two other branching rules on the number of vehicles can be applied: on the number of first-echelon vehicles per depot and on the number of second-echelon vehicles per satellite. These two rules have the same priority and their decisions are also imposed through constraints in the MP.

Fourth, branching can be performed, at a lesser priority, on the individual use of a given

first-echelon route m. These branching decisions are enforced by fixing the bounds of the corresponding first-echelon variable x_m.

Finally, when all previous branching rules cannot be applied, we can branch on an arc $(i_1, i_2) \in A^2, i_1, i_2 \in N^C$, that links two customer nodes and bears a fractional total flow (between 0 and 1). On one branch, this flow is set to 0 by removing this arc (i_1, i_2) from all sets $A_s, \forall s \in N^S$ and removing all columns that use this arc (i_1, i_2). On the other branch, this flow is set to 1 by deleting all arcs $(i_1, i_3), i_3 \neq i_2$ and $(i_4, i_2), i_4 \neq i_1$ from all sets $A_s, \forall s \in N^S$ and deleting all columns that use such arcs. Alternatively, we can also branch on an arc $(s, i) \in A_s$ or $(i, s') \in A_s$ linking a satellite node s or s', representing a satellite $s \in N^S$, and a customer node $i \in N^C$. The branching decisions are imposed similarly to branching on the arcs between two customers, except that no arcs adjacent to a satellite node are removed on the branch fixing the flow to 0.

For the total flow on a back-and-forth first-echelon route, the variable with the fractional part closest to 0.5 is selected for branching. For all other branching rules, the variable with the fractional part farthest from 0.5 is selected for branching. Ties are broken arbitrarily for each branching decision. In addition, the search tree is explored using a best-first strategy.

4.3.5 Alternative model and algorithm

To determine the best BPC algorithm, several ideas have been investigated. In particular, we tested one that aimed at reducing the symmetry induced by the multiple copies of the non-dominated back-and-forth first-echelon routes appearing in formulation (4.2.4)–(4.2.9). Like in Dellaert et al. (2019), we can associate each non-dominated back-and-forth first-echelon route with a single copy and impose integrality requirements on these variables. However, this model is, in general, only a relaxation as some of its feasible solutions may be infeasible for the 2E-VRPTW. Indeed, when a back-and-forth route variable takes a value greater than 1, there is no guarantee that all second-echelon routes associated with it can be supplied by a single vehicle assigned to this first-echelon route. Nevertheless, it is possible to check if this constraint is satisfied for the second-echelon routes associated with a back-and-forth route to which multiple vehicles are assigned by solving a small-sized bin-packing problem. Therefore, our BPC algorithm can be changed as follows to solve this alternative model.

When an integer solution is found, the subset of back-and-forth first-echelon routes m such that $x_m \geq 2$, denoted $\mathcal{M}^2$, is computed. If $\mathcal{M}^2 = \emptyset$, this solution is feasible. Otherwise, for each route $m \in \mathcal{M}^2$, a bin-packing problem that tries to fit the total demand of each second-echelon route assigned to m (without splitting it) into x_m bins of capacity Q^1 is solved. If this is possible for each route m, the current solution is deemed feasible. Otherwise, it is

infeasible and a combinatorial cut is added to the MP to forbid the infeasible synchronization between these first- and second-echelon routes from happening in later stages of the solution process. To properly handle the dual variables from these cuts in the pricing problems, one pricing problem per pair of satellite and back-and-forth route needs to be considered, which doubles the number of pricing problems to solve at each iteration.

This alternative model/algorithm did not outperform the proposed BPC algorithm described in the previous sections for our test instances. Nevertheless, we believe that it could provide some computational gain for other instances.

4.4 Computational experiments

In this section, we first describe the characteristics of the 2E-VRPTW instances used to test our BPC algorithm. Second, we report summarized computational results and compare our algorithm with the current state-of-the-art (Dellaert et al., 2019). Third, we conduct a sensitivity analysis on different components of our algorithm. Fourth, we derive managerial insights by considering only back-and-forth first-echelon routes.

All tests were performed on a Linux computer equipped with an Intel(R) Core(TM) i7-8700 processor (3.2 GHz). Our BPC algorithm uses the GENCOL library and calls the simplex algorithm of CPLEX 12.6 to solve the RMPs. In addition, we have imposed a maximal time limit of three hours (10,800 seconds).

4.4.1 Characteristics of the instances

Our algorithm was tested on the benchmark instances for the 2E-VRPTW proposed by Dellaert et al. (2019) which are asymmetric due to the inclusion of time windows. For all instances, the capacity of the first-echelon and second-echelon vehicles are $Q^1 = 200$ and $Q^2 = 50$, respectively. In addition, the fixed vehicle cost for first- and second-echelon vehicles are $f_1 = 50$ and $f_2 = 25$, respectively. The service time at each satellite and customer location is set to $\tau_i = 10, \forall i \in N^S \cup N^C$. The number of customers ranges from 15 to 100, i.e., $|N^C| = \{15, 30, 50, 100\}$. For each value of $|N^C|$, three combinations for values of $|N^D|$ and $|N^S|$ are used, that is 1) $|N^D| = 2$ and $|N^S| = 3$, 2) $|N^D| = 3$ and $|N^S| = 5$, and 3) $|N^D| = 6$ and $|N^S| = 4$. For each combination of values of $|N^D|, |N^S|$ and $|N^C|$, there are 20 instances divided in four categories (five instances per category), i.e., categories `Ca`, `Cb`, `Cc`, and `Cd`. Each category has specific characteristics for the time windows and customer demands. The time windows are set as follows: $20 \leq \underline{w}_i \leq 260$ and $\underline{w}_i \leq \overline{w}_i \leq \underline{w}_i + 20$, $\forall i \in N^C$, for categories `Ca` and `Cb`; $60 \leq \underline{w}_i \leq 360$ and $\underline{w}_i \leq \overline{w}_i \leq \underline{w}_i + 90$, $\forall i \in N^C$,

for category `Cc`; and $60 \leq \underline{w}_i \leq 360$ and $\underline{w}_i \leq \overline{w}_i \leq \underline{w}_i + 20$, $\forall i \in N^C$, for category `Cd`. The demands are set as follows: $q_i \in \{10, 20\}, \forall i \in N^C$, for categories `Ca`, `Cc`, and `Cd`; and $5 \leq q_i \leq 25$ and $q_i \in \mathbb{N}, \forall i \in N^C$, for category `Cb`. The name of each instance contains its category, the instance number as well as the number of depots, satellites and customers. For example, instance `Ca1-2,3,30` is the first instance of category `Ca` with $|N^D| = 2$, $|N^S| = 3$, and $|N^C| = 30$.

4.4.2 Computational results

In this section, we present summarized results obtained with the proposed BPC algorithm. We first compare its computational performance to that of the algorithm of Dellaert et al. (2019). We then provide statistics on the execution of our algorithm, including the number of cuts generated and the number of branching nodes explored. Finally, we analyze the characteristics of the solutions. Detailed results are reported in Appendix A.1.

4.4.2.1 Computational time statistics and comparison with Dellaert et al. (2019)

In this section, we compare our BPC algorithm with the current state-of-the-art algorithm of Dellaert et al. (2019). Table 4.1 presents summarized computational results that have been collected by group of instances. For conciseness reasons, it does not report the results with $|N^C| = 15$ as all instances are solved in less than one minute (generally less than one second) by both algorithms. The first three columns in Table 4.1 indicate the instance group according to the values of $|N^D|$, $|N^S|$, and $|N^C|$. Then, we give the average total computational time (T) in seconds and the number of instances solved to optimality (*# Opt*) by the algorithm of Dellaert et al. (2019) within the 3-hour time limit. The average is computed over the instances solved to optimality and the times are those reported by Dellaert et al. (2019). Next, for our BPC algorithm, we provide the same two statistics as well as the number of new optimal solutions found (*# New*). Given that our computational experiments were run on a computer that is more powerful than that used by Dellaert et al. (2019), it may be risky to establish a comparison based on the reported computational times. However, as our results show, the comparison is quite obvious. To see this, we specify in the last column the following average biased minimal speed-up factor (*BMSU Factor*), where the biased comes for the fact that we used a faster computer. This factor is equal to the ratio T^1/T^2, where T^2 is the average time reported for our BPC algorithm and T^1 is the minimal average time required by the algorithm of Dellaert et al. (2019) (on the computer they used) to solve the instances solved to optimality by our algorithm. This minimal average time is obtained by setting the minimal time for instances that are not solved by their algorithm to 10,800

seconds. For example, for the instance group with $|N^D| = 2$, $|N^S| = 3$, and $|N^C| = 50$, given that six instances are not solved to optimality, $T^1 = (14 \cdot 733.1 + 6 \cdot 10800)/20 = 3753.1$ seconds and the average BMSU factor is equal to $3753.1/208.2 = 18.0$.

The results show that our BPC algorithm can solve all instances solved by Dellaert et al. (2019) within the prescribed time limit. In fact, it allowed to find 41 new optimal solutions, leaving only 17 open instances out of the 180 tested instances. The number of instances solved to optimality by both algorithms, especially for the 100-customer instances, indicate the superiority of our BPC algorithm. While the computers used favor our algorithm, the average BMSU factors are much larger than the speed-up factor that can be provided by our computer over the one used by Dellaert et al. (2019). Observe that the BMSU factors are smaller for the instances with $|N^C| = 100$ because the average computational time increases while the 10,800-second time limit, which is used as a lower bound on the computational time for the unsolved instances, does not change. We believe that these results clearly highlight that our BPC algorithm supersedes the algorithm of Dellaert et al. (2019).

Table 4.1 indicates that, as the number of customers increases, the computational time rapidly increases and that the number of instances solved to optimality within the prescribed computational time decreases significantly. On the other hand, the impact of the numbers of satellites and depots seems to be limited though the configuration $|N^D| = 3$ and $|N^S| = 5$ tends to require the largest average computational time, at least for the 30- and 100-customer instances. This can be explained by the fact that, compared to the other two configurations, this configuration induces the largest number of non-dominated first-echelon routes as reported in the next section.

4.4.2.2 Algorithm statistics

To better analyze the performance of our BPC algorithm, we disclose some statistics that were collected during its execution. These statistics are reported in Table 4.2 by group of instances. Again, the instance groups are identified by the information provided in the first three columns. Then, for instances solved to optimality, we report the average number of branch-and-bound nodes explored (*# Nodes*), the average number of non-dominated first-echelon routes considered ($|\mathcal{M}|$), the average number of DDOIs (4.3.13) added to the MP at the start of the algorithm, as well as the total numbers of RCIs (4.3.28), SRIs (4.3.30), and VSIs (4.3.31) generated throughout the solution process.

These results again show that, as the number of customers increases, the problem becomes harder to solve because the algorithm requires more branch-and-bound nodes, RCIs and SRIs. Note that the number of SRIs is not very large but this is due to a low average

Table 4.1 Summarized computational results

$\|N^D\|$	$\|N^S\|$	$\|N^C\|$	Dellaert et al. † T (s)	Dellaert et al. † # Opt	This paper ‡ T (s)	This paper ‡ # Opt	This paper ‡ # New	BMSU Factor
2	3	30	61.7	20	6.6	20	0	9.3
3	5	30	68.3	20	11.6	20	0	5.9
6	4	30	88.6	20	5.0	20	0	17.7
2	3	50	733.1	14	208.2	20	6	18.0
3	5	50	1,270.0	19	216.4	20	1	8.1
6	4	50	1,526.9	18	71.5	20	2	34.3
2	3	100	2,947.1	6	2,258.1	17	11	3.6
3	5	100	2,711.6	3	2,631.8	11	8	3.3
6	4	100	3,181.6	2	2,188.7	15	13	4.5

† Tests ran on a computer equipped with an Intel Core i7-4770 processor.
‡ Tests ran on a computer equipped with an Intel Core i7-8700 processor.

number of customers visited per route as the results of the next section reveal. The results in Table 4.2 also indicate that the number of DDOIs increases moderately with the number of customers. In fact, the number of DDOIs depends on the number of non-dominated first-echelon routes. The latter mostly depends on the numbers of satellites and depots but also, to a lower degree, on the number of customers because condition (4.2.3) becomes harder to fulfill with a larger number of customers. Finally, the number of VSIs tends to reduce with the number of customers because these inequalities are satisfied as soon as their associated satellite is visited by at least one first-echelon vehicle, which is a condition easily met when there is a large number of customers to service close to this satellite.

Table 4.2 Summarized statistics on the algorithm execution

$\|N^D\|$	$\|N^S\|$	$\|N^C\|$	# Nodes	$\|\mathcal{M}\|$	# DDOIs	# RCIs	# SRIs	# VSIs
2	3	30	24.2	21.8	19.0	29.0	11.4	16.3
3	5	30	15.7	368.0	342.1	29.7	12.0	37.5
6	4	30	20.1	100.8	74.8	30.7	10.3	32.9
2	3	50	244.0	27.7	29.5	58.6	19.4	20.9
3	5	50	111.8	406.0	372.5	77.8	33.5	47.4
6	4	50	83.9	107.9	90.5	62.0	23.0	40.0
2	3	100	2,324.8	38.8	52.1	183.5	55.2	6.4
3	5	100	539.7	468.1	479.3	196.5	65.5	35.6
6	4	100	1,523.3	126.5	135.8	171.4	62.0	10.9

4.4.2.3 Solution statistics

Table 4.3 provides averaged statistics on the optimal solutions obtained by the BPC algorithm, where the averages are computed by group of instances identified in the first three columns of this table. For each instance group, we report the average number of depots ($|N^D|^*$), number of satellites ($|N^S|^*$), number of first-echelon routes ($|\mathcal{M}|^*$), percentage of first-echelon routes with only one satellite ($|\mathcal{M}^1|^*$), and number of second-echelon routes ($|\mathcal{L}|^*$) that are used in the optimal solutions.

First, observe that the number of satellites used increases with the number of customers. This is not surprising because, when there are more customers, their spatial distribution tends to better cover the whole region, increasing the usefulness of opening more satellites. Similarly, it can be expected that a larger number of depots might be used to supply a larger number of satellites. However, as the average number of satellites used does not increase much, this trend is not striking, especially for the case $|N^D| = 6$ and $|N^S| = 4$ which shows a decrease in the number of open depots when $|N^C|$ varies from 50 to 100.

Second, the results show that both numbers of first-echelon and second-echelon routes increase proportionally to the number of customers, which can be explained by the fact that the capacities of the vehicles are constant while the total demand increases proportionally to the number of customers. Interestingly, most first-echelon routes used in the optimal solutions are back-and-forth routes, and every optimal solution contains at most one first-echelon route with more than one satellite (never more than two satellites). First-echelon routes with more than one satellite are used more often when $|N^D| = 3$ and $|N^S| = 5$, and for instances with $|N^C| = \{30, 50\}$, which suggests that increasing the number of satellites yields more first-echelon routes visiting more than one satellite. This may be explained by a closer proximity between certain pairs of satellites when there are a larger number of satellites. Finally, our solutions show that second-echelon vehicles visit on average between three and four customers each, while no specific behavior is noted according to the instance category.

4.4.3 Sensitivity analysis

In this section, we perform a sensitivity analysis on some components of our BPC algorithm to assess their impact on its computational performance. To do so, we have selected a subset of 40 instances with $|N^C| = \{50, 100\}$ that are not too easy but not too difficult to solve with our BPC algorithm. For $|N^C| = 50$, we have selected the 20 instances with the largest computational time, while for $|N^C| = 100$, we have selected the 20 instances that were solved under 3,600 seconds and with the largest computational time. Therefore, our subset

Table 4.3 Summarized solution statistics

$\lvert N^D\rvert$	$\lvert N^S\rvert$	$\lvert N^C\rvert$	$\lvert N^D\rvert^*$	$\lvert N^S\rvert^*$	$\lvert\mathcal{M}\rvert^*$	$\lvert\mathcal{M}^1\rvert^*$ (%)	$\lvert\mathcal{L}\rvert^*$
2	3	30	1.8	2.4	2.9	93.3	9.4
3	5	30	2.8	3.8	3.0	72.5	9.5
6	4	30	3.0	3.3	3.0	90.0	9.5
2	3	50	2.0	3.0	4.0	100.0	15.4
3	5	50	2.7	4.6	4.0	85.0	15.3
6	4	50	3.9	3.9	4.1	100.0	15.5
2	3	100	1.9	3.0	7.9	100.0	30.2
3	5	100	3.0	5.0	8.0	100.0	30.3
6	4	100	3.2	4.0	7.7	100.0	30.0

of instances comprises: 20 instances with $|N^C| = 50$, including one instance with $|N^D| = 2$ and $|N^S| = 3$, 15 instances with $|N^D| = 3$ and $|N^S| = 5$, and four instances with $|N^D| = 6$ and $|N^S| = 4$; and 20 instances with $|N^C| = 100$, including four instances with $|N^D| = 2$ and $|N^S| = 3$, eight instances with $|N^D| = 3$ and $|N^S| = 5$, and eight instances with $|N^D| = 6$ and $|N^S| = 4$.

Below, we discuss the impact of using one pricing problem per pair of first-echelon route and visited satellite and of omitting each type of inequalities (DDOIs, RCIs, SRIs, and VSIs) on the computational performance of our algorithm. We also analyze the computational performance of each branching rule by omitting specific sets of branching rules and changing their priority level. Furthermore, we assess the impact on the computational time of considering a second-copy variable for each first-echelon route with more than one satellite. Unless specified otherwise, detailed results are presented in Appendix A.2.

4.4.3.1 Disaggregating the pricing problems

Given that the reduced cost (4.3.1) of a second-level route variable y_{ml} depends on the dual variable π_m associated with first-echelon route m, an intuitive column generation algorithm would consider one pricing problem per pair of first-echelon route and visited satellite. This naïve approach would simplify the labeling algorithm but would substantially increase the number of pricing problems to consider. We have tested this approach and the results obtained show that more than half of the test instances cannot be solved to optimality within the three-hour time limit, and that the proposed BPC algorithm with one pricing problem per satellite is much faster. For conciseness reasons and because they would not be bring more insights, we do not report these results (here or in Appendix A.2).

4.4.3.2 Omitting inequality families

To understand the impact of each family of inequalities, we present in this section the computational results obtained when we remove from the proposed BPC algorithm one family of inequalities at a time, namely, the DDOIs (4.3.13), the RCIs (4.3.28), the SRIs (4.3.30), and the VSIs (4.3.31). Table 4.4 reports average results for the 20 instances with $|N^C| = 50$ and for the 20 instances with $|N^C| = 100$ separately. Each row in this table corresponds to a different version of the algorithm that omits the family of inequalities mentioned in the first column. For each algorithm version and group of 20 instances, we report the number of instances solved to optimality within the 3-hour time limit (*# Opt*), the average computational time (T) in seconds for solving these instances, and the average computational time difference in percentage between the full BPC algorithm and the algorithm without the selected inequality family (ΔT). This difference is computed as $(T - T^*)/T^*$, where T^* is the total computational time required by the complete algorithm.

From this table, we deduce that, to achieve the performance of our BPC algorithm, the SRIs seem to be the most critical inequalities. The DDOIs and the RCIs also play an important role. As for the VSIs, they have less impact (but still significant) for the 50-customer instances, and practically no impact for the 100-customer instances. Therefore, all these components are required to maintain the algorithm's computational performance. Let us further comment on each inequality family separately.

First, when omitting DDOIs, our results show an average computational time increase of 231.6% and 138.7% for the solved instances with 50 and 100 customers, respectively. Note that the impact is not necessarily less important for the 100-customer instances because less instances can be solved to optimality. Looking at the detailed results reported in Appendix A.2, we can observe that a larger increase is recorded for larger values of $|N^S|$, i.e., when the number of non-dominated first-echelon routes is large. In particular, out of the eight test instances with $|N^D| = 3$, $|N^S| = 5$, and $|N^C| = 100$, five cannot be solved within the prescribed time limit when omitting the DDOIs.

Second, the RCIs also have an important impact on the total computational time. In fact, when they are not applied in our BPC algorithm, six test instances cannot be solved within the time limit: one for each combination of $(|N^D|, |N^S|, |N^C|) \in \{(2, 3, 50), (2, 3, 100), (6, 4, 100)\}$ and three with $|N^D| = 3$, $|N^S| = 5$, and $|N^C| = 100$. Again here, the impact of not considering the RCIs might seem less important for the 100-customer instances ($\Delta T = 132.5\%$ compared to $\Delta T = 197.2\%$ for the 50-customer instances), but less instances were solved to optimality within the time limit.

Table 4.4 Impact of omitting families of inequalities

Family	Instances with $\lvert N^C \rvert = 50$			Instances with $\lvert N^C \rvert = 100$		
	# Opt	T (s)	ΔT (%)	# Opt	T (s)	ΔT (%)
DDOIs	20	1,210.3	231.6	15	3,013.6	138.7
RCIs	19	777.5	197.2	15	3,421.9	132.5
SRIs	19	921.6	267.4	6	6,474.5	578.7
VSIs	20	431.3	44.2	20	1,933.0	1.1

Third, without the SRIs, 15 instances cannot be solved within the prescribed time limit. The impact of applying these inequalities is clearly larger for the instances with 100 customers. In fact, 14 of these instances cannot be solved within the 3-hour time limit and the average computational time increases by 578.7% for the others.

Finally, for the VSIs, the results are mitigated. They have their highest positive impact for the instances with 5 satellites. On the other hand, they deteriorate the computational times for a large number of instances with 3 or 4 satellites. Nevertheless, we have decided to keep these inequalities in the proposed BPC algorithm as, for some instances with large computational time, they help to significantly decrease the total computational time.

4.4.3.3 Changing the branching scheme

In our proposed BPC algorithm, we apply different branching rules in a hierarchical fashion (see Section 4.3.4). In the first priority level (level 1), there is only one branching rule: branching on the total flow of back-and-forth first-echelon route. The second priority level (level 2) contains three branching rules: branching on the number of first-echelon vehicles, on the number of second-echelon vehicles, and on the number of first-echelon vehicles per satellite. The third priority level (level 3) is comprised of two branching rules: branching on the number of first-echelon vehicles per depot and on the number of second-echelon vehicles per satellite. Finally, at the fourth and fifth levels (levels 4 and 5), branching on the first-echelon route variables and on the second-echelon arcs can be invoked when there are no previous candidate decisions. Note that the branching decisions in levels 4 and 5 are necessary to ensure the validity of the branching process but all the others can be omitted.

To understand the impact of this branching scheme on the computational performance of our algorithm, we have conducted additional experiments using six algorithm variants that have been obtained by: 1) omitting level 1; 2) omitting level 2; 3) omitting level 3; 4) omitting levels 2 and 3; 5) omitting levels 1, 2, and 3; and 6) prioritizing levels 2 and 3 over level 1.

Table 4.5 reports the average results obtained and is organized similarly to Table 4.4.

First, our results show that when omitting different sets of branching rules, the total computational time increases by at least 17.3% and 40.1% for instances with 50 and 100 customers, respectively. For all the tested instances, omitting levels 1, 2, and 3 has the highest increase on the total computational time (i.e., 851.9% and 585.2% for instances with 50 and 100 customers) and is the combination that solves the least instances to optimality (i.e., 23 instances out of a total of 40). Looking at each individual level, omitting level 2 has the highest increase on the total computational time independently of the number of customers. For instances with 50 customers, level 3 seems to be the most critical to solve instances to optimality, while level 2 is the most critical for instances with 100 customers.

Second, the results obtained when omitting the different sets of branching rules suggest that levels 2 and 3 has a higher impact on decreasing the total computational time and solving instances to optimality. Therefore, we have conducted an additional analysis (prioritizing levels 2 and 3 over level 1) applying the branching rules in the following order: level 2, level 3, and level 1. With this new order, all instances are solved to optimality. For instances with 50 customers, the computational time decreases by an average of 3.2%, while it increases by 30.8% for instances with 100 customers. These results are mitigated and given that instances with 100 customers tend to be harder to solve, we have decided to apply the branching rules in the proposed initial order.

4.4.3.4 Adding a second copy for first-echelon routes visiting more than one satellite

Given that first-echelon routes visiting more than one satellite are rarely used in optimal solutions, we made the assumption that such routes cannot be used more than once. This assumption favors faster computational times. In this section, we verify this assertion by analyzing the impact of adding second-copy variables for the first-echelon routes with more than one satellite. Note that upon adding this second copy, we also impose symmetry breaking constraints similar to constraints (4.2.9).

The results obtained by our BPC algorithm after adding these second-copy first-echelon route variables show an average computational time increase of 87.3% for the 20 test instances with 50 customers and of 107.5% for those with 100 customers. As expected, the increase turns out to be more important for the instances involving the largest number of first-echelon routes visiting more than one satellite, i.e., for the instances with $|N^D| = 3$ and $|N^S| = 5$. In particular, the average computational time increase reaches 151% for the eight instances with $|N^C| = 100$.

Table 4.5 Impact of omitting branching rules

Branching configuration	Instances with $\lvert N^C \rvert = 50$ # Opt	T (s)	ΔT (%)	Instances with $\lvert N^C \rvert = 100$ # Opt	T (s)	ΔT (%)
Omit level 1	20	527.5	17.3	20	2,694.0	40.1
Omit level 2	20	1,429.0	674.2	10	4,426.6	244.6
Omit level 3	17	542.3	105.1	13	2,886.5	144.7
Omit levels 2 and 3	17	1,333.3	663.1	6	3,841.0	257.2
Omit levels 1, 2 and 3	17	1,682.4	851.9	6	6,793.2	585.2
Prioritize levels 2 and 3 over 1	20	405.2	–3.2	20	2,217.0	30.8

This computational time increase can be explained by several factors, namely, a larger number of first-echelon route variables in the MP, a larger number of symmetry-breaking constraints in the MP, and a larger number of labels generated by the labeling algorithm due a larger number of initial labels. Nevertheless, we observed that considering these first-echelon route copies did not change the optimal solutions for our test instances, i.e., none of these routes are used twice in an optimal solution. Consequently, we believe that, instead of adding these second-copy variables a priori, generating them as needed (together with their associated constraints) could yield a BPC algorithm that would require similar computational times than those obtained for the case with one copy for these first-echelon routes.

4.4.4 Managerial insights

In this section, we derive managerial insights related to the usage of first-echelon routes visiting more than one satellite. In fact, for our set of instances, the solution statistics presented in Table 4.3 show that many optimal solutions require only back-and-forth first-echelon routes. Therefore, we analyze how forbidding the usage of first-echelon routes visiting more than one satellite impacts the optimal solution costs and the computational performance.

Table 4.6 summarizes the impact of this interdiction on the optimal solution costs for the 37 instances (25 with 30 customers and 12 with 50 customers) that initially used first-echelon routes with more than one satellite. For a given instance, this impact is computed as the relative difference Δz^* between the optimal values z_1^* and z^* obtained without and with first-echelon routes visiting more than one satellite, respectively, i.e., $\Delta z^* = (z_1^* - z^*)/z^*$. For each instance group identified by the first three columns, we provide the number of instances in this group (# *Inst*). Then, for each group, we present the average impact (*Avg*), the maximal impact (*Max*), as well as the numbers of instances with an impact of less than 0.5% and than 1.0% (# $\leq$ 0.5% and # $\leq$ 1.0%), respectively. Detailed results can be found in

Appendix A.3.

These results show an average increase of less than 1.0%, while at least 50% and 70% of the instances have a maximal increase of 0.5% and 1.0%, respectively. In addition, the maximal increase over all instances is of 1.8%. Note that, while most optimal solutions for these instances use a similar number of depots, the number of satellites used usually decreases by one or two satellites when considering only back-and-forth first-echelon routes. Consequently, we observe that the average operational costs do not increase much (especially, if we consider the full set of instances) but that satellite fixed costs could also be saved in some cases.

Table 4.6 Impact on costs of considering only back-and-forth first-echelon routes

				$\Delta\ z^*$			
$\lvert N^D\rvert$	$\lvert N^S\rvert$	$\lvert N^C\rvert$	# Inst	Avg (%)	Max (%)	$\# \leq 0.5\%$	$\# \leq 1.0\%$
2	3	30	4	1.0	1.5	1	2
3	5	30	15	0.6	1.8	8	12
6	4	30	6	0.5	1.1	3	5
3	5	50	12	0.7	1.8	7	8

Considering only back-and-forth first-echelon routes seems an interesting option to simplify the operations while not increasing the operational costs too much. Let us now investigate the impact of this hypothesis on the performance of the BPC algorithm which should benefit from a large reduction of the number of first-echelon routes. For this experiment, all instances are considered and detailed results can again be found in Appendix A.3. Here, we report in Table 4.7 average results that are computed over the instances solved within the 3-hour time limit and which are grouped by their numbers of depots, satellites, and customers. For each group, the table indicates the number of instances in the group (*# Opt*), the average computational time (T) in seconds, and the average impact on the total computational time (ΔT) computed as $(T - T^*)/T^*$, where T^* is the total computational time obtained when considering all first-echelon routes.

These results show that the problem is easier to solve, i.e., more instances are solved to optimality within the prescribed computational time limit as the computational time decreases, when considering only back-and-forth first-echelon routes. In fact, the number of labels generated in the pricing algorithm decreases significantly when the number of first-echelon routes also decreases. As expected, the average computational time gain is larger for groups of instances with a larger number of satellites, often reaching more than 70%.

For our set of instances, considering only back-and-forth first-echelon routes do not increase

Table 4.7 Impact on computational performance of considering only back-and-forth first-echelon routes

		$\lvert N^C\rvert = 30$			$\lvert N^C\rvert = 50$			$\lvert N^C\rvert = 100$		
$\lvert N^D\rvert$	$\lvert N^S\rvert$	# Opt	T (s)	ΔT (%)	# Opt	T (s)	ΔT (%)	# Opt	T (s)	ΔT (%)
2	3	20	5.8	–14.4	20	269.8	–10.1	18	1,375.6	–25.7
3	5	20	3.1	–78.3	20	55.0	–72.7	19	747.3	–63.9
6	4	20	1.7	–72.3	20	41.6	–42.3	15	1,097.4	–39.8

too much the operational costs and significantly reduces the computational burden of finding operational solutions. In practice, considering only back-and-forth first-echelon routes also simplify the operations. Therefore, based on these results and for instances with similar characteristics, one can assume that a decision-maker might decide to rely only on back-and-forth first-echelon routes. For instances with different characteristics, e.g., a higher number of satellites with clustered customers close to these satellites, additional testing would have to be conducted to understand the impact of using only back-and-forth first-echelon routes. For such instances, we can expect that the increase in the operational costs when considering only back-and-forth first-echelon routes might be more important.

4.5 Conclusions

In this paper, we model the 2E-VRPTW using multiple copies of back-and-forth first-echelon routes and propose a BPC algorithm to solve the problem. In this algorithm, the first-echelon routes are enumerated a priori while the second-echelon routes are generated using column generation pricing problems defined by satellite and solved using a labeling algorithm which keeps track of the first-echelon route associated with each enumerated second-echelon route. Thus, this labeling algorithm includes a dominance rule that allows to compare second-echelon routes starting from the same satellite, but supplied by different first-echelon routes. Our BPC algorithm also relies on new DDOIs that help reducing the number of column generation iterations, and applies three families of valid inequalities, namely, the RCIs, SRIs and VSIs, to improve the lower bounds. Finally, *ad hoc* branching rules are derived for the problem.

We have conducted extensive computational experiments on existing benchmark instances which show that our BPC algorithm outperforms the current state-of-the-art algorithm for the problem. We report an average biased minimal speed-up factor between 3.3 and 34.3, according to the instance group, and our algorithm solves to optimality 41 previously unsolved instances within the 10,800-second time limit leaving 17 open instances. We also analyze the

impact of different components of our BPC algorithm on its performance, and highlight the importance of each family of valid inequalities as well as the proposed DDOIs. In addition, considering only one copy of first-echelon routes visiting more than one satellite seems like a valid assumption, as when considering a second copy there is no impact on the optimal solution values. Finally, considering that resorting to only back-and-forth first-echelon routes is simpler in practice, we show that this results in a maximal increase in total costs of 2.0% on the set of tested instances. Therefore, for instances with similar characteristics (number of depots, number of satellites, and distances between depots and satellites as well as between satellites and customers), it would be reasonable to assume that the network might be designed with only back-and-forth first-echelon routes. For instances with different characteristics, more testing would have to be conducted to assess the performance on the solution quality of using only back-and-forth first-echelon routes.

As possible extensions to this work, we envision that some algorithmic findings of this paper (in particular, the DDOIs and the handling of the first-echelon routes in the pricing algorithm) can be used to develop efficient BPC algorithms for variants of the 2E-VRPTW involving load synchronization constraints between the first- and second-echelon routes. Also, it might be interesting to design an efficient technique, possibly a machine learning one, to identify a priori which first-echelon routes should be considered in the model.

CHAPITRE 5 ARTICLE 2 : A BRANCH-PRICE-AND-CUT ALGORITHM FOR THE MULTI-COMMODITY TWO-ECHELON VEHICLE ROUTING PROBLEM WITH TIME WINDOWS

T. Mhamedi, M. Cherkesly, G. Desaulniers, (2024), A Branch-Price-and-Cut Algorithm for the Multi-Commodity Two-Echelon Vehicle Routing Problem with Time Windows. Soumis à *INFORMS Journal on Computing* le 24 octobre 2024, révisions majeures demandées le 20 février 2025.

5.1 Introduction

In this paper, we propose a branch-price-and-cut algorithm for the multi-commodity two-echelon vehicle routing problem with time windows (MC-2E-VRPTW). This problem extends the two-echelon vehicle routing problem (2E-VRP, see Cuda et al., 2015; Guastaroba et al., 2016; Sluijk et al., 2023, for surveys on the 2E-VRP) to consider real-life characteristics that arise in city logistics, i.e., time windows and multiple commodities. Considering multiple commodities is of major importance when optimizing distribution processes on two-echelon network structures involving multiple origins with, e.g., customer-specific (non-substitutable) demands. The MC-2E-VRPTW involves depots, satellites and customers. Each customer is assigned to one depot that supplies its demand, which is referred to as a commodity. Therefore, there are as many commodities as there are customers. In addition, the service at each customer must start within a predefined time window. The first echelon corresponds to delivering goods from the depots to the satellites, and a set of identical first-echelon vehicles with limited capacity is available to conduct the first-echelon routes. Each first-echelon route starts and ends at the same depot, and visits a subset of satellites. We define two types of first-echelon routes: i) routes visiting only one satellite, referred to as back-and-forth routes, and ii) routes visiting more than one satellite. As often seen in practice, we assume that back-and-forth routes can be used more than once, whereas routes visiting more than one satellite cannot. The second echelon corresponds to delivering goods from the satellites to the customers, and a set of identical second-echelon vehicles with limited capacity is available to operate second-echelon routes. The merchandise is transferred from first- to second-echelon vehicles at the satellites, and transfer times at the satellites are assumed to be constant, i.e., they do not depend on the quantity transferred. The synchronization between first- and second-echelon vehicles is *exact* (see Drexl, 2012, for a taxonomy), implying that each demand must be supplied by exactly one first-echelon route and one second-echelon route,

but that a second-echelon route can be supplied by more than one first-echelon route. The MC-2E-VRPTW consists in determining a set of least-cost feasible first- and second-echelon routes such that each customer is visited by exactly one second-echelon route within its time window, and that its load is supplied by its assigned depot.

To the best of our knowledge, only the paper of Dellaert et al. (2021) studies the MC-2E-VRPTW, proposing several formulations and solution algorithms. Their best algorithm is a branch-and-price-based algorithm that solves their two-path formulation. It first generates subsets of first-echelon routes (called configurations) and sorts them in increasing order of a lower bound obtained by selecting the corresponding configuration as the first-echelon solution. These configurations with their lower bounds can then be used to define multiple root nodes in a branch-and-price algorithm that does not need to branch to achieve integrality at the first echelon. The algorithm starts with a single root node associated with the least-lower-bound configuration and proceeds as a standard branch-and-price algorithm, branching only to achieve integrality at the second echelon. When exploring the search tree, new root nodes (configurations) are added whenever the current lower bound in the tree exceeds the lower bounds associated with unexplored configurations. The algorithm stops when the current tree is fully explored and the best upper bound is less than or equal to the lower bounds of the unexplored configurations. Dellaert et al. (2021) report results on a small set of instances with up to 3 depots, 5 satellites, and 100 customers.

On the other hand, many heuristics (e.g., Breunig et al., 2016; Crainic et al., 2008, 2011; Dumez et al., 2023; Jia et al., 2023) and exact algorithms (e.g., Baldacci et al., 2013; Perboli et al., 2011) have been proposed for the classical 2E-VRP and its variants. For the exact algorithms, some of the best results are obtained by branch-and-price algorithms (e.g., Marques et al., 2020; Santos et al., 2015). In addition, many variants with time windows or multiple commodities have been studied. The two most relevant variants with time windows are the multi-depot 2E-VRP with time windows (2E-VRPTW, see Dellaert et al., 2019), and the 2E-VRPTW with freight consolidation at the satellites (see Marques et al., 2022). The 2E-VRPTW considers exact synchronization (Drexl, 2012; Soares et al., 2024) between the first- and second-echelon vehicles, i.e., no freight consolidation occurs at the satellites and each second-echelon vehicle is supplied from a single first-echelon vehicle. In the 2E-VRPTW with freight consolidation, the load aboard a second-echelon vehicle can be supplied by more than one first-echelon vehicle. The literature on the 2E-VRP with multiple commodities is limited, but has received more attention lately. In this literature, the commodities are defined either in a *many-to-many* or a *one-to-one* context, as highlighted in Gu et al. (2023). The two most relevant variants with multiple commodities are the multi-commodity 2E-VRP with satellite synchronization and the multi-commodity two-echelon distribution problem (MC-2EDP). In

the first variant introduced by Jia et al. (2023), a commodity refers to an origin-destination pair in a one-to-one setting. Customers request two commodities, and the total demand of a customer must be delivered by a single second-echelon vehicle. In the MC-2EDP studied by Gu et al. (2022), commodities are sent from depots to satellites using back-and-forth routes before being transferred to second-echelon vehicles. A customer can request more than one commodity and each commodity needs to be supplied in a single visit. In the following, we provide a brief overview of the state-of-the-art exact algorithms for the most relevant variants of the 2E-VRP with time windows or multiple commodities, with an emphasis on branch-and-price algorithms.

To solve the 2E-VRPTW, Dellaert et al. (2019) propose the first branch-and-price-based algorithm, which is similar to the algorithm of Dellaert et al. (2021) described above and can solve to optimality instances with up to 3 depots, 5 satellites and 100 customers within a 3-hour time limit. Their results are outperformed by Mhamedi et al. (2022) who develop a branch-price-and-cut (BPC) algorithm involving dual inequalities and in which the labeling algorithm used to solve the pricing subproblems generates second-echelon routes and specifies their load-supplying first-echelon route. Marques et al. (2022) introduce a branch-and-price algorithm for the 2E-VRPTW with freight storage and consolidation at the satellites. They formulate the problem with a route-based formulation containing an exponential number of precedence constraints to enforce synchronization between first- and second-echelon vehicles. Their algorithm is adapted to consider two cases: the second-echelon routes contain a single trip each or they can involve multiple trips, possibly starting and ending at different satellites. The authors also present a post-processing procedure to verify whether feasible solutions can be transformed into same-cost ones requiring no storage or freight consolidation. Their algorithm performs well on instances from the literature, yielding new optimal solutions for 9 and 54 instances of the multi- and single-trip cases, respectively. To our knowledge, Petris et al. (2024) propose the only exact algorithm for the MC-2EDP, which is formulated as a set-covering model with an exponential number of variables, one for each second-echelon route. To solve it, the authors devise a BPC algorithm where second-echelon variables are generated dynamically by solving one subproblem per satellite. Each subproblem is modeled as an elementary shortest path problem with resource constraints (ESPPRC) and solved by a label setting algorithm. To tighthen the lower bounds, capacity inequalities and two families of cuts exploiting the multicommodity aspect of the problem are considered. Within a 1-hour time limit, their algorithm solves to optimality 439 of the 736 instances created by Gu et al. (2022).

In this paper, we address the MC-2E-VRPTW and develop an exact BPC algorithm that outperforms the algorithm of Dellaert et al. (2021). Our contributions are fourfold. First, we

propose a BPC algorithm (inspired from that of Mhamedi et al., 2022) that does not rely on an a priori enumeration of first-echelon route subsets as in Dellaert et al. (2021). This algorithm generates dynamically the second-echelon routes using a labeling algorithm that determines which first-echelon route supplies each customer visited in the second-echelon routes. Second, we devise a recovery procedure that is applied to ensure integer solution feasibility in the presence of dual inequalities. Third, to tackle instances with a large number of first-echelon routes, we introduce a branching rule adapted to the multi-commodity context. Finally, we extend the benchmark dataset of Dellaert et al. (2021) by creating 140 new MC-2E-VRPTW instances based on the 2E-VRPTW instances of Dellaert et al. (2019) and test our algorithm on a total of 180 test instances with 30, 50 and 100 customers.

The rest of this paper is organized as follows. Section 5.2 presents notation and formulates the MC-2E-VRPTW as a set-partitioning problem with side constraints. Section 5.3 describes the proposed BPC algorithm. Computational results are reported in Section 5.4 to assess this algorithm and the importance of different algorithmic components. Conclusions are drawn in Section 5.5.

5.2 Mathematical formulation

The MC-2E-VRPTW is defined on a directed graph $G = (N, A)$, where N is its vertex set and A its arc set. Set $N = N^D \cup N^S \cup N^C$ comprises the sets of depots (N^D), satellites (N^S), and customers (N^C). For each depot $d \in N^D$, let $N^C_d \subseteq N^C$ be its set of assigned customers. Therefore, $N^C_{d_1} \cap N^C_{d_2} = \emptyset, \forall d_1, d_2 \in N^D, d_1 \neq d_2$. Each customer $i \in N^C$ has a demand q_i that needs to be supplied from its assigned depot $d_i \in N^D$. Service at customer i requires a time τ_i and can only start during time window $[\underline{w}_i, \overline{w}_i]$, where $\underline{w}_i$ and $\overline{w}_i$ are the earliest and latest start times. Each satellite $s \in N^S$ has a predefined transfer time τ_s representing the time to unload the merchandise from first-echelon vehicles and load the second-echelon vehicles. Arc set $A = A_1 \cup A_2$ comprises the subsets of arcs that can be traversed by first-echelon vehicles (A_1) and by second-echelon vehicles (A_2). Set A_1 contains three arc types: *i*) arcs from a depot to a satellite, *ii*) arcs from a satellite to a depot, and *iii*) arcs between satellites, that is, $A_1 = \{(d, s) : d \in N^D, s \in N^S\} \cup \{(s, d) : s \in N^S, d \in N^D\} \cup \{(s_1, s_2) : s_1, s_2 \in N^S, s_1 \neq s_2\}$. Similarly, set A_2 contains three arc types: i) arcs from a satellite to a customer, ii) arcs from a customer to a satellite, and iii) arcs between customers, that is, $A_2 = \{(s, i) : s \in N^S, i \in N^C\} \cup \{(i, s) : i \in N^C, s \in N^S\} \cup \{(i_1, i_2) : i_1, i_2 \in N^C, i_1 \neq i_2\}$. Each arc $(i, j) \in A$ is associated with a routing cost c_{ij} and a travel time t_{ij} which includes the service time at vertex i (i.e., 0 if $i \in N^D$ and τ_i if $i \in N^S \cup N^C$).

The set of first-echelon routes is denoted by $\mathcal{M}$, and each first-echelon route is performed

by a first-echelon vehicle with a capacity Q_1 and a fixed cost f_1. Each first-echelon route $m = (i_0,\ i_1, \ldots,\ i_h,\ i_{h+1})$ starts and ends at the same depot $i_0 = i_{h+1} = d_m \in N^D$ and visits a subset of satellites $i_1, \ldots, i_h \in N^S$, such that $i_j \neq i_k, \forall j, k \in \{1, \ldots, h\},\ j \neq k$. Its total load cannot exceed vehicle capacity Q_1. For each pair of depot $d \in N^D$ and satellite $s \in N^S$, we create $\lceil \sum_{i \in N_d^C} q_i / Q_1 \rceil$ copies of the back-and-forth first-echelon route (d, s, d) and denote by $\mathcal{M}_{ds}^1$ the set of these copies that are included in $\mathcal{M}$. For each pair of first-echelon route $m \in \mathcal{M}$ and satellite $s \in N^S$, we define the binary parameter a_{sm} which is equal to 1 if route m visits satellite s and 0 otherwise. Each route m incurs a total cost c_m that includes a first-echelon vehicle fixed cost and its routing cost, i.e., $c_m = f_1 + \sum_{k=0}^{h} c_{i_k, i_{k+1}}$. We assume that every first-echelon route departs at time 0 from depot i_0 and we denote by t_m^s its arrival time at a visited satellite s. Finally, we denote by $\mathcal{M}_d$ the subset of first-echelon routes associated with depot d and by $\mathcal{M}_{ds} \subseteq \mathcal{M}_d$ the subset of those routes visiting satellite s.

The set of second-echelon routes is denoted by $\mathcal{L}$ and the second-echelon vehicles have a capacity $Q_2 < Q_1$ and a fixed cost f_2. A second-echelon route $l = (i_0,\ i_1, \ldots,\ i_h,\ i_{h+1})$ starts and ends at a satellite $i_0 = i_{n+1} = s_l \in N^S$ and visits a subset of customers $i_1, \ldots, i_h \in N^C$, such that $i_j \neq i_k$ for all $j, k \in \{1, \ldots, h\},\ j \neq k$. Its total load cannot exceed vehicle capacity Q_2. For every pair of second-echelon route $l \in \mathcal{L}$ and customer $i \in N^C$, we define the binary parameter a_{il} which is equal to 1 if route l visits customer i and 0 otherwise. In addition, the total cost c_l of a route l is composed of its fixed cost and its routing cost, i.e., $c_l = f_2 + \sum_{k=0}^{h} c_{i_k, i_{k+1}}$. Each route l is associated with a departure time t_l^{start} from satellite s_l that guarantees servicing customers $i_1, \ldots, i_h$ within their respective time windows. For each second-echelon route l, we denote by $\mathcal{M}_l \subseteq \{m \in \mathcal{M} : a_{s_l m} = 1 \text{ and } t_m^s \leq t_l^{start} - \tau_{s_l}\}$ the set of first-echelon routes that can supply it before its departure time and by $\mathcal{P}_l$ its set of supply patterns. A supply pattern $p \in \mathcal{P}_l$ specifies, for each visited customer, the first-echelon route in $\mathcal{M}_l$ that supplies its demand and is expressed as $p = \{(m_1, i_1), \ldots, (m_h, i_h)\}$, where $m_k \in \mathcal{M}_l \cap \mathcal{M}_{d_{i_k}}$, $\forall k \in \{1, \ldots, h\}$. Finally, let Q_{mp}^l be the total demand supplied by first-echelon route m when assigning supply pattern p to route l, i.e., $Q_{mp}^l = \sum_{\{i \in N^C:\ a_{il} = 1\ \wedge\ (m,i) \in p\}} q_i$.

To better understand the concept of a supply pattern, Figure 5.1 presents a solution for a MC-2E-VRPTW instance with two depots ($|N^D| = 2$), three satellites ($|N^S| = 3$), and 12 customers ($|N^C| = 12$). Customers 1 to 7 are assigned to depot d_1, whereas customers 8 to 12 are assigned to depot d_2, i.e., $N_{d_1}^C = \{1, 2, \ldots, 7\}$ and $N_{d_2}^C = \{8, 9, \ldots, 12\}$. There are five feasible first-echelon routes $(m_1, m_2, \ldots, m_5)$, but only the first three are used in this solution, which also includes four second-echelon routes l_1, l_2, l_3, and l_4. As an example, the supply pattern set for l_1 is $\mathcal{P}_{l_1} = \{(m_1, 1), (m_1, 4)\} \cup \{(m_1, 1), (m_2, 4)\} \cup \{(m_2, 1), (m_1, 4)\} \cup \{(m_2, 1), (m_2, 4)\}$, assuming that m_1 and m_2 arrive at satellite s_3 early enough to supply route l_1. Finally, in this solution, routes l_1 to l_4 are associated with sup-

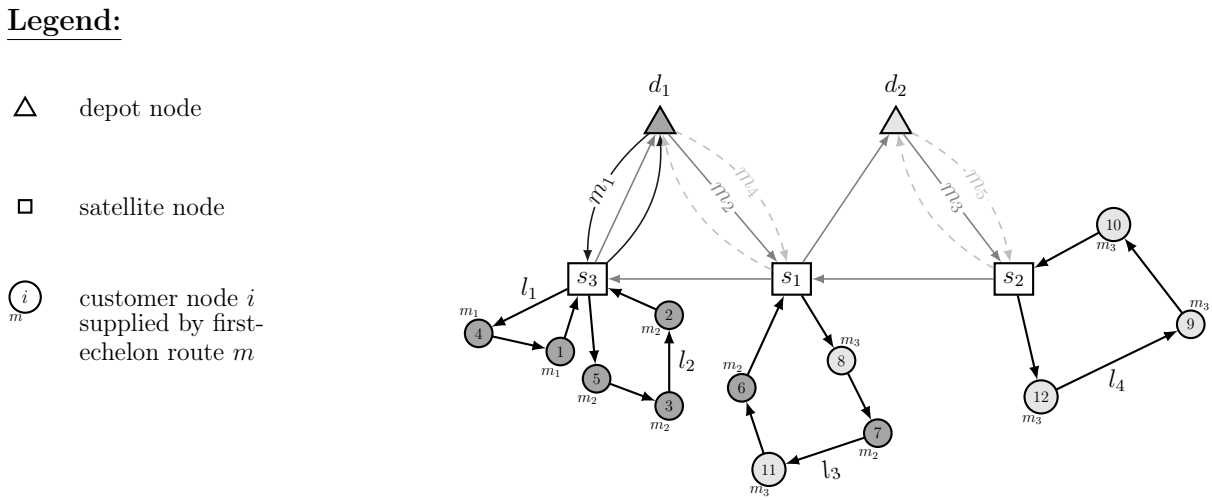

Figure 5.1 Example of a solution for an instance of the MC-2E-VRPTW

ply patterns $\{(m_1, 1), (m_1, 4)\}$, $\{(m_2, 2), (m_2, 3), (m_2, 5)\}$, $\{(m_2, 6), (m_2, 7), (m_3, 8), (m_3, 11)\}$, and $\{(m_3, 9), (m_3, 10), (m_3, 12)\}$, respectively.

The MC-2E-VRPTW can then be formulated using two sets of binary variables. Let x_m be equal to 1 if first-echelon route $m \in \mathcal{M}$ is used in the solution, and 0 otherwise. For each pair of second-echelon route $l \in \mathcal{L}$ and supply pattern $p \in \mathcal{P}_l$, let y_{lp} be equal to 1 if route l is used in the solution and supplied using pattern p, and 0 otherwise. The problem can then be formulated as the following integer linear programming model:

$$\min \quad \sum_{m \in \mathcal{M}} c_m x_m + \sum_{l \in \mathcal{L}} \sum_{p \in \mathcal{P}_l} c_l y_{lp} \tag{5.2.1}$$

$$\text{s.t.} \quad \sum_{l \in \mathcal{L}} \sum_{p \in \mathcal{P}_l} a_{il} y_{lp} = 1, \qquad \forall i \in N^C, \tag{5.2.2}$$

$$\sum_{l \in \mathcal{L}} \sum_{p \in \mathcal{P}_l} Q^l_{mp} y_{lp} \leq Q_1 x_m, \qquad \forall m \in \mathcal{M}, \tag{5.2.3}$$

$$x_m \in \{0, 1\}, \qquad \forall m \in \mathcal{M}, \tag{5.2.4}$$

$$y_{lp} \in \{0, 1\}, \qquad \forall l \in \mathcal{L}, p \in \mathcal{P}_l. \tag{5.2.5}$$

Objective function (5.2.1) minimizes the total costs (sum of fixed vehicle and routing costs) incurred at both echelons. Set-partitioning constraints (5.2.2) ensure that each customer is serviced by a single second-echelon route. The synchronization constraints (5.2.3) between the first- and second-echelon routes ensure that the first-echelon routes involved in the supply pattern assigned to each selected second-echelon route are also selected in the solution. They

also guarantee that the capacity of each first-echelon route is respected. Finally, (5.2.4) and (5.2.5) define the domain of the variables.

To model (5.2.1)–(5.2.5), we add symmetry-breaking constraints to distinguish the back-and-forth route copies in set $\mathcal{M}^1_{ds}$ for each pair of depot $d \in N^D$ and satellite $s \in N^S$. Without loss of generality, let us assume that each set $\mathcal{M}^1_{ds}$ is ordered, and let $\bar{m}^1_{ds}$ denote the last element of set $\mathcal{M}^1_{ds}$. The symmetry-breaking constraints express as follows:

$$x_m \leq x_{m^+}, \qquad \forall s \in N^S, d \in N^D, m \in \mathcal{M}^1_{ds} \setminus \{\bar{m}^1_{ds}\}, \tag{5.2.6}$$

where m^+ denotes the immediate successor of back-and-forth copy m in $\mathcal{M}^1_{ds}$.

5.3 Branch-price-and-cut algorithm

Model (5.2.1)–(5.2.6) usually contains a large number of variables. In the MC-2E-VRPTW, the cardinality of the first-echelon route set $\mathcal{M}$ is usually quite limited, but that of the second-echelon route set $\mathcal{L}$ can be very large. To avoid enumerating all second-echelon route variables, we resort to a branch-price-and-cut (BPC) algorithm, where the first-echelon route variables are all enumerated a priori. A BPC algorithm is a branch-and-cut algorithm where the linear relaxations are solved by column generation (CG, see, e.g., Costa et al., 2019). CG is an iterative algorithm that solves at each iteration a restricted master problem (RMP) and one or several (pricing) subproblems. In our case, the RMP corresponds to the linear relaxation of model (5.2.1)–(5.2.6) with a limited subset of second-echelon route variables, i.e., with index subsets $\mathcal{L}' \subseteq \mathcal{L}$ and $\mathcal{P}'_l \subseteq \mathcal{P}_l$. Given an optimal dual solution to the current RMP, the subproblems aim at finding negative reduced cost y_{lp} variables if some exist. In our case, there is one subproblem per satellite that can be cast as an ESPPRC (ESPPRC, Irnich and Desaulniers, 2005). When negative reduced cost variables (columns) are found, they are added to the RMP before starting a new iteration. Otherwise, CG stops as the current RMP solution is also optimal for the complete linear relaxation. If this solution is not integer, branching and cutting is applied, re-optimizing each linear relaxation by CG.

In the following, we describe our BPC algorithm. First, we define the CG subproblems and develop a labeling algorithm to solve them. Second, we propose dual inequalities to stabilize the CG process and describe how the BPC algorithm handles them. Finally, we present the valid inequalities and branching rules used.

5.3.1 Subproblems

There is one subproblem per satellite $s \in N^S$, denoted SP_s and used to generate second-echelon variables y_{lp} associated with routes starting from satellite s. This subproblem is an ESPPRC that determines $i)$ the customers visited in a second-echelon route and $ii)$ the first-echelon routes supplying these customers while satisfying the customer-to-depot assignments, the second-echelon vehicle capacity, and the customer time windows. Given $(\sigma_i)_{i \in N^C}$ and $(\pi_m)_{m \in \mathcal{M}}$ the dual variables associated with constraints (5.2.2)–(5.2.3), respectively, the reduced cost of a variable y_{lp}, $l \in \mathcal{L}$ and $p \in \mathcal{P}_l$, is computed as

$$\bar{c}_{lp} = c_l - \sum_{i \in N^C} a_{il}\sigma_i - \sum_{m \in \mathcal{M}} Q_l^{mp}\pi_m. \tag{5.3.1}$$

Subproblem SP_s aims at finding a pair of second-echelon route l departing from s and supply pattern p with a minimum reduced cost $\bar{c}_{lp}$.

For a given second-echelon route l starting from satellite s, there might exist a large number of feasible supply patterns $p \in \mathcal{P}_l$. The feasibility of a pattern depends on the route starting time t_l^{start}. Given that this time is a priori unknown when solving SP_s, we propose to consider that $t_l^{start} = t + \tau_s$, where t belongs to the set T_s of the first-echelon route visiting times to satellite s (sorted in ascending order). For all starting times $t_l^{start} = t + \tau_s$, $t \in T_s$, we can then find the supply pattern $p \in \mathcal{P}_l$ that yields the least reduced cost $\bar{c}_{lp}$ among the supply patterns that are feasible if route l starts at time t_l^{start}. For $t \in T_s$, let $\mathcal{M}_s^t = \cup_{d \in N^D}\{m \in \mathcal{M}_d : t_s^m \leq t\}$ be the set of first-echelon routes visiting satellite s no later than t and $D_s^t = \{d \in N^D : \mathcal{M}_s^t \cap \mathcal{M}_d \neq \emptyset\}$ the set of depots associated with those routes. Furthermore, for $d \in D_s^t$, we denote by $m(d,t)$ a first-echelon route $m \in \mathcal{M}_s^t \cap \mathcal{M}_d$ with the largest dual value π_m, i.e., $m(d,t) \in \arg\max_{m \in \mathcal{M}_s^t \cap \mathcal{M}_d} \pi_m$. With this notation, the reduced cost defined in (5.3.1) can be rewritten as

$$\bar{c}_{lp} = c_l - \sum_{i \in N^C} a_{il}\sigma_i - \sum_{i \in N^C} a_{il}q_i\pi_{m(d_i,t)} \quad \text{if } t_l^{start} = t + \tau_s. \tag{5.3.2}$$

The subproblems are solved by the labeling algorithm (Irnich and Desaulniers, 2005) described in Subsection 5.3.1.1. Two acceleration strategies, summarized in Subsection 5.3.1.2, are applied to speed up the solution process.

5.3.1.1 Labeling algorithm.

Each subproblem SP_s, $s \in N^S$, consists in finding a least reduced cost path on a directed graph $G_s = (N_s, A_s)$. Vertex set $N_s = N^C \cup \{n_s^{src}, n_s^{sk}\}$ comprises the set of customers as well as a source and a sink vertex, which represent copies of satellite s. We set $q_s^{src} = q_s^{sk} = 0$. Arc set $A_s = \{(n_s^{src}, i) : i \in N^C\} \cup \{(i, n_s^{sk}) : i \in N^C\} \cup \{(i_1, i_2 : i_1, i_2 \in N^C, i_2 \neq i_2)\}$. Each arc $(i, j) \in A_s$ is associated with the time t_{ij} of the corresponding arc in A_2 and with a modified cost $\bar{c}_{ij}$ defined as:

$$\bar{c}_{ij} = \begin{cases} c_{ij} + f_2 - \sigma_j, & \text{if } i = n_s^{src} \\ c_{ij} - \sigma_j, & \text{if } i, j \in N^C \\ c_{ij}, & \text{if } j = n_s^{sk}. \end{cases}$$

In a standard labeling algorithm, a label is a multi-dimensional vector that represents a partial path starting at the source node n_s^{src} and ending at a vertex $i \in N_s$. Starting from an initial label representing the path containing only vertex n_s^{src}, this algorithm recursively extends it towards the other vertices in N_s using resource extension functions (REFs) to generate longer partial paths until reaching the sink vertex n_s^{sk}. Infeasible paths are discarded as the search progresses. To avoid enumerating all feasible source-to-sink paths, a dominance rule comparing label pairs is applied. Below, we define the labels, describe how the algorithm is initialized with multiple labels (instead of a single one), and present the REFs and the dominance rule.

Label definition. A label represents a combination of a partial path l starting from n_s^{src} at a given time $t + \tau_s$, $t \in T_s$, and an associated supply pattern p which can be retrieved from the list of visited customers and time t. Such a label $E_{lt} = (T_{lt}^{rdc}, T_{lt}^{time}, T_{lt}^{load}, T_{lt}^{start}, U_{lt})$ associated with a path ending at vertex $i \in N_s$ holds the following information:

T_{lt}^{rdc}: the reduced cost of this path;

T_{lt}^{time}: the earliest time for starting service at vertex i;

T_{lt}^{load}: the total delivered demand;

T_{lt}^{start}: time t corresponding to the latest time a first-echelon route can visit satellite s to supply this partial route;

U_{lt}: the set of unreachable customers from label E_{lt}. A customer h is said to be unreachable from E_{lt} if $d_h \notin D_s^{T_{lt}^{start}}$, or h has already been visited in this path or cannot be due to time or load constraints (i.e., $T_{lt}^{time} + t_{ih} > \overline{w}_h$ or $T_{lt}^{load} + q_h > Q_2$).

Initialization. Recall from the discussion above that only $|T_s|$ starting times are relevant, i.e., the times $t + \tau_s$ for $t \in T_s$. Given one of these times, it becomes easy to identify the best supply pattern for a partial path. Therefore, our labeling algorithm starts with $|T_s|$ initial labels at source vertex n_s^{src}, namely, one for each $t \in T_s$. Denoting by 0 the partial path containing only the source vertex n_s^{src}, an initial label E_{0t}, $t \in T_s$, is set to $E_{0t} = (0, t, 0, t, \{h \in N^C : d_h \notin D_s^t \text{ or } t + t_{n_s^{src}h} > \overline{w}_h\})$.

Resource extension functions. To generate new labels, the following REFs are applied. Let $E_{lt} = (T_{lt}^{rdc}, T_{lt}^{time}, T_{lt}^{load}, T_{lt}^{start}, U_{lt})$ be a label associated with a partial path l ending at vertex $i \in N_s \setminus \{n_s^{sk}\}$ and a path starting time $t \in T_s$. Extending E_{lt} along an arc $(i, j) \in A_s$ such that $j \notin U_{lt}$ yields a label $E_{l't}$ whose components are computed as follows:

$$T_{l't}^{rdc} = T_{lt}^{rdc} + \bar{c}_{ij} - \begin{cases} q_j \cdot \pi_{m(d_j, T_{lt}^{start})} & \text{if } j \neq n_s^{sk}, \\ 0 & \text{otherwise} \end{cases}, \tag{5.3.3}$$

$$T_{l't}^{time} = \max\{\underline{w}_j, T_{lt}^{time} + t_{ij}\}, \tag{5.3.4}$$

$$T_{l't}^{load} = T_{lt}^{load} + q_j, \tag{5.3.5}$$

$$T_{l't}^{start} = T_{lt}^{start}, \tag{5.3.6}$$

$$U_{l't} = U_{lt} \cup \{j\} \cup \{h \in \bigcup_{d \in D_s^{T_{lt}^{start}}} N_d^C : T_{l't}^{time} + t_{jh} > \overline{w}_h \vee T_{l't}^{load} + q_h > Q_2\}. \tag{5.3.7}$$

Label $E_{l't}$ is always deemed feasible because $j \notin U_{lt}$. Observe that, if $j \in N^C$, the first-echelon route supplying j is route $m(d_j, T_{lt}^{start})$ as identified in the first case of (5.3.3).

Dominance rule. To discard non-promising labels, we apply the following dominance rule.

Definition 5.1. *A label $E_{lt} = (T_{lt}^{rdc}, T_{lt}^{time}, T_{lt}^{load}, T_{lt}^{start}, U_{lt})$ dominates a label $E_{l't'} = (T_{l't'}^{rdc}, T_{l't'}^{time}, T_{l't'}^{load}, T_{l't'}^{start}, U_{l't'})$ if both paths l and l' end at the same vertex and the following conditions hold:*

$$T_{lt}^{rdc} \leq T_{l't'}^{rdc} + (Q_2 - T_{l't'}^{load}) \cdot \Delta_s(T_{lt}^{start}, T_{l't'}^{start}) \tag{5.3.8}$$

$$T_{lt}^{time} \leq T_{l't'}^{time} \tag{5.3.9}$$

$$T_{lt}^{load} \leq T_{l't'}^{load} \tag{5.3.10}$$

$$U_{lt} \subseteq U_{l't'}, \tag{5.3.11}$$

where

$$\Delta_s(t_1, t_2) = \min\{0, \min_{d \in D_s^{t_1} \cap D_s^{t_2}} \pi_{m(d,t_1)} - \pi_{m(d,t_2)}\} \text{ for } t_1, t_2 \in T_s. \tag{5.3.12}$$

Function $\Delta_s(T_{lt}^{start}, T_{l't'}^{start})$ specifies the minimum (negative) difference between the dual values of the synchronization constraints (5.2.3) that could be activated in a feasible extension of the labels and, as such, is used in (5.3.8) to anticipate any decrease of the reduced cost in an extension of label $E_{l't'}$ that would not occur in the same extension of label E_{lt} due to a difference in their associated starting times. Note that, in definition (5.3.12), the computation may exclude depots that supply customers visited in a feasible extension of $E_{l't'}$, i.e., depots in $D_s^{T_{l't'}^{start}} \setminus D_s^{T_{lt}^{start}}$. If this is the case, then $U_{lt} \not\subseteq U_{l't'}$ and condition (5.3.11) would prevent label E_{lt} to dominate label $E_{l't'}$.

Proposition 5.1. *Conditions* (5.3.8)-(5.3.11) *constitute a valid dominance rule for the considered ESPPRC variant.*

Proof 5.1. See Appendix B.1.

5.3.1.2 Acceleration strategies.

To speed up the resolution of the subproblems, we make use of the *ng-path relaxation* and *decremental state-space relaxation* (DSSR), two well-known ESPPRC relaxations that we briefly describe in what follows.

- The *ng*-path relaxation, introduced by Baldacci et al. (2011), relies on defining for each vertex $i \in N$, a neighborhood $\mathcal{N}_i \subseteq N^C$ that contains its κ closest customers, where κ is a predefined parameter. A path is allowed to contain a cycle $(i_1, i_2, \ldots, i_h)$ with $i_1 = i_h$ if there exists $k \in \{2, \ldots, h-1\}$ such that $i_1 \notin \mathcal{N}_{i_k}$. For that matter, the REF (5.3.7) is adjusted as follows:

$$\begin{aligned} U_{l't} = (U_{lt} \cap \mathcal{N}_j) \cup \{j\} \cup \{h \in N^C : d_h \notin D_s^{T_{l't}^{start}}\} \\ \cup \{h \in \bigcup_{d \in D_s^{T_{lt}^{start}}} N_d^C : T_{l't}^{time} + t_{jh} > \overline{w}_h \vee T_{l't}^{load} + q_h > Q_2\}. \end{aligned} \tag{5.3.13}$$

- The DSSR (Boland et al., 2006; Righini and Salani, 2008) attempts to solve an ESPPRC by first dropping the elementarity constraints on the customer vertices. If at least one elementary path with a negative reduced cost is found or no paths (elementary or not) with a negative reduced cost are found, the process is stopped. Otherwise, the labeling algorithm is restarted with elementarity requirements enforced on the customers visited more than once in the optimal non-elementary path found.

5.3.2 Stabilization by dual inequalities

To stabilize the column generation process, we propose to apply dual inequalities (see, e.g., Ben Amor et al., 2006; Gschwind and Irnich, 2016). Let $\mu \in \mathbb{R}^n$ be the vector of dual variables associated with a feasible bounded linear program and D^* the set of its dual optimal solutions. A dual inequality $\Lambda^\top \mu \leq \lambda$, with $\Lambda \in \mathbb{R}^n$ and $\lambda \in \mathbb{R}$, is a *dual-optimal inequality* (DOI) if $D^* \subseteq \{\mu \in \mathbb{R}^n : \Lambda^\top \mu \leq \lambda\}$. Furthermore, a set of dual inequalities $A\mu \leq a$, with $A \in \mathbb{R}^{m \times n}$ and $a \in \mathbb{R}^n$, forms a set of *deep-DOIs* (DDOIs) if $D^* \cap \{\mu \in \mathbb{R}^n : A\mu \leq a\} \neq \emptyset$. In the following, we first describe the DDOIs that we use before discussing integer solution feasibility and presenting a feasibility recovery procedure.

5.3.2.1 Transfer inequalities.

We consider the dual inequalities, called the *transfer inequalities* (TIs), that were introduced by Mhamedi et al. (2022) for the multi-depot 2E-VRPTW. For the MC-2E-VRPTW, the TIs are:

$$\pi_m \geq \pi_{m'}, \qquad \forall (m, m') \in \mathcal{K}, \tag{5.3.14}$$

where $\mathcal{K} = \bigcup_{d \in N^D} \left\{ (m_1, m_2) \in {\mathcal{M}_d}^2 \ : \ m_1 \neq m_2, \ S(m_1) \subseteq S(m_2) \wedge t_s^{m_2} \leq t_s^{m_1}, \forall s \in S(m_1) \right\}$ is the set of first-echelon route pairs (m_1, m_2) such that any customer-satellite assignment feasible for m_1 is also feasible for m_2. These TIs are DDOIs for the linear relaxation of model (5.2.1)–(5.2.6).

To add these TIs, we introduce a non-negative primal variable $u_{m_1 m_2}$ with a zero cost for every route pair $(m_1, m_2) \in \mathcal{K}$ and rewrite constraints (5.2.3) as follows:

$$\sum_{l \in \mathcal{L}} \sum_{p \in \mathcal{P}_l} Q_l^{mp} y_{lp} - \sum_{m' \in \mathcal{M}_{d(m)} : (m, m') \in \mathcal{K}} u_{m,m'} \ + \sum_{m' \in \mathcal{M}_{d(m)} : (m', m) \in \mathcal{K}} u_{m',m} \leq Q_1 x_m, \ \forall m \in \mathcal{M}. \tag{5.3.15}$$

These new variables are called *transfer variables* because they allow to transfer demand from one first-echelon route to another as discussed below. In the rest of this paper, we denote by F and $\tilde{F}$ the formulations (5.2.1)–(5.2.6) and (5.2.1)–(5.2.2),(5.3.15),(5.2.4)–(5.2.6), respectively.

5.3.2.2 Solution feasibility and recovery procedure.

Considering constraints (5.3.15) instead of constraints (5.2.3) yields a relaxation, i.e., $\tilde{F}$ is a relaxation of F. As a result, an integer solution $(\tilde{x}, \tilde{y}, \tilde{u})$ for $\tilde{F}$ is not feasible for F whenever

the value $\tilde{u}_{mm'}$ of a transfer variable is positive and, moreover, it might not be possible to convert this solution into a feasible one with the same cost by transferring demands between the first-echelon routes according to the positive-valued transfer variables. See Appendix B.2 for a detailed example.

To convert an integer solution $(\tilde{x}, \tilde{y}, \tilde{u})$ with a positive-valued transfer variable into a feasible integer solution for F with the same cost, we apply the following recovery procedure that consists in solving some small-sized bin packing problems. Let $\tilde{\mathcal{M}}$ and $\tilde{\mathcal{L}}$ be the subsets of first- and second-echelon routes used in solution $(\tilde{x}, \tilde{y}, \tilde{u})$, respectively. For each route $l \in \tilde{\mathcal{L}}$, let $\tilde{p}_l \in \mathcal{P}_l$ denote the (unique) supply pattern assigned to l, i.e., $\tilde{y}_{l\tilde{p}_l} = 1$. Also, for each customer i visited along l, let $\tilde{m}_i \in \mathcal{M}_{d_i}$ be the first-echelon route assigned to i in supply pattern $\tilde{p}_l$, and let $\tilde{\mathcal{M}}_i = \{m \in \tilde{\mathcal{M}} \cap \mathcal{M}_{d_i} : m = \tilde{m}_i \text{ or } (\tilde{m}_i, m) \in \mathcal{K}\}$ be the set of active first-echelon routes that could have been used to supply the demand of customer i. Let $\tilde{\mathcal{K}} = \{(m, m') \in \mathcal{K} : \tilde{u}_{mm'} > 0\}$ denote the subset of route pairs in $\mathcal{K}$ associated with positive-valued transfer variables set. Finally, let $\tilde{\mathcal{D}} = \{d \in N^D : |\tilde{\mathcal{K}} \cap (\mathcal{M}_d \times \mathcal{M}_d)| \geq 1\}$ be the subset of depots associated with at least one pair in $\tilde{\mathcal{K}}$.

For each depot $d \in \tilde{\mathcal{D}}$, we consider a feasibility bin packing problem FP_d that consists in assigning the demand of each customer associated with d to the first-echelon routes in $\tilde{\mathcal{M}} \cap \mathcal{M}_d$, while preserving the selected second-echelon routes in $\tilde{\mathcal{L}}$. This problem involves a binary variable λ^i_m for each customer $i \in N^C_d$ and first-echelon route in $m \in \tilde{\mathcal{M}}_i$ that takes value 1 if customer i is assigned to first-echelon route m, 0 otherwise. It is formulated as:

$$\min \quad 0 \tag{5.3.16}$$

$$\text{s.t.} \quad \sum_{m \in \tilde{\mathcal{M}}_i} \lambda^i_m = 1, \qquad \forall i \in N^C_d \tag{5.3.17}$$

$$\sum_{i \in N^C_d : m \in \tilde{\mathcal{M}}_i} q_i \lambda^i_m \leq Q_1, \qquad \forall m \in \tilde{\mathcal{M}} \cap \mathcal{M}_d \tag{5.3.18}$$

$$\lambda^i_m \in \{0, 1\}, \qquad \forall i \in N^C_d, m \in \tilde{\mathcal{M}}_i. \tag{5.3.19}$$

Constraints (5.3.17) ensure that each customer associated with depot d is assigned to a selected first-echelon route that can supply its second-echelon route on time. Constraints (5.3.18) guarantee that the capacity of each first-echelon route is not exceeded. Model (5.3.16)–(5.3.19) can be solved using a commercial mixed-integer programming solver.

If a feasible solution can be found for each problem FP_d, $d \in \tilde{\mathcal{D}}$, then solution $(\tilde{x}, \tilde{y}, \tilde{u})$ can be converted into a same-cost feasible solution for F, preserving the same first-echelon and second-echelon routes but rearranging the supply patterns according to the FP_d solutions.

Otherwise, for all depots $d \in \tilde{D}$ for which FP_d turned out to be infeasible, we fix to zero the transfer variables $u_{mm'}$, $(m, m') \in \tilde{\mathcal{K}} \cap (\mathcal{M}_d \times \mathcal{M}_d)$, before restarting column generation.

5.3.3 Valid inequalities

To strengthen the lower bounds, we consider three families of valid inequalities: (lifted) *visited satellite inequalities* (VSIs), *rounded capacity inequalities* (RCIs), and *subset-row inequalities* (SRIs). When looking for violated inequalities, we apply the following priority order: VSIs first, RCIs second, and SRIs last. As soon as violated cuts are found for a family, we do not search for violated cuts of the subsequent families for the same fractional solution. Valid inequalities can be added in multiple rounds within the same branch-and-bound node. Moreover, because the subproblems can become much more difficult to solve in the presence of SRIs, we only look for violated SRIs in branching nodes of depth less than or equal to a predefined parameter value, set to 10 for our tests. In what follows, we briefly present each type of inequalities, explain how they are separated and how the column generation/labeling algorithm should be modified if need be.

Visited satellite inequalities. The VSIs, initially introduced by Marques et al. (2020) for the 2E-CVRP, ensure that, whenever a customer i is serviced from a satellite s, at least one first-echelon route visiting s and originating from d_i is used. In our BPC algorithm, we consider a lifted version of these inequalities:

$$\sum_{m \in \mathcal{M}_{d_i s} \,:\, t_s^m \leq \overline{w}_i - t_{si}} x_m \geq \sum_{l \in \mathcal{L}_s} \sum_{p \in \mathcal{P}_l} a_{il} y_{lp}, \qquad \forall s \in N^S,\ i \in N^C, \tag{5.3.20}$$

where $\mathcal{L}_s$ denotes the subset of second-echelon routes departing from satellite s. The left-hand side of (5.3.20) disregards first-echelon routes that would lead to an arrival at customer i after its time window in the best case. The VSIs are separated by enumeration. They are robust in the sense that they do not alter the structure of the subproblems. In fact, the dual value of a VSI associated with a customer i and a satellite s is added to the modified cost of customer's i outgoing arcs in set A_s.

Rounded capacity inequalities. The RCIs (Laporte and Nobert, 1983) have been extensively used for different VRP variants, including those with two routing echelons (Marques et al., 2020; Santos et al., 2015; Mhamedi et al., 2022). For a subset of customers $\mathcal{C} \subseteq N^C$ such that $|\mathcal{C}| \geq 2$, let $\xi(\mathcal{C}) = \left\lceil \frac{\sum_{i \in \mathcal{C}} q_i}{Q_2} \right\rceil$ be a lower bound on the number of second-echelon

vehicles required to service the demand of customers in $\mathcal{C}$. The RCIs are defined as:

$$\sum_{l \in \mathcal{L}} \sum_{p \in \mathcal{P}_l} \sum_{(i,j) \in \delta(\mathcal{C})} b_{ij}^l y_{lp} \geq \xi(\mathcal{C}), \qquad \forall \mathcal{C} \subseteq N^C \text{ such that } |\mathcal{C}| \geq 2, \tag{5.3.21}$$

where b_{ij}^l is a binary parameter indicating whether or not second-echelon route l traverses arc (i, j) and $\delta(\mathcal{C})$ is the set of arcs leaving $\mathcal{C}$, i.e., $\delta(\mathcal{C}) = \{(i, j) \in A_2 : i \in \mathcal{C}, j \notin \mathcal{C}\}$. The RCIs are separated using the heuristic proposed by Lysgaard et al. (2004). They are also robust and the dual value of a RCI associated with a customer subset $\mathcal{C}$ is subtracted from the modified cost of all arcs in $\delta(\mathcal{C})$.

We also add a priori one RCI per depot d defined on its first-echelon routes:

$$\sum_{m \in \mathcal{M}_d} x_m \geq \left\lceil \frac{\sum_{i \in N_d^C} q_i}{Q_1} \right\rceil, \qquad \forall d \in N^D. \tag{5.3.22}$$

Subset-row inequalities. The SRIs, introduced by Jepsen et al. (2008) for the VRPTW, are valid inequalities for the set packing polytope and, thus, for the set partitioning polytope also. Similar to several previous works considering SRIs, we restrict ourselves to customer subsets $\mathcal{C} \subset N^C$ of cardinality three because their separation can be done by straighforward enumeration. The SRIs are expressed as:

$$\sum_{l \in \mathcal{L}} \sum_{p \in \mathcal{P}_l} \left\lfloor \frac{\sum_{i \in \mathcal{C}} a_{il}}{2} \right\rfloor y_{lp} \leq 1, \qquad \forall \mathcal{C} \subset N^C \text{ such that } |\mathcal{C}| = 3. \tag{5.3.23}$$

They are non-robust as each active SRI (associated with a negative dual price) requires an additional binary resource in the label definition. Furthermore, to take into account these additional label components, the dominance rule presented in Section 5.3.1.1 must be modified. For details, see Jepsen et al. (2008) or Costa et al. (2019).

5.3.4 Branching

The branch-and-bound tree is explored using a best-first search strategy and a hierarchical branching scheme. For each branching rule, the selected variable/entity has the fractional value closest to 0.5. Furthermore, for branching decisions involving modifications at the subproblem level, we remove from the MP, all columns that do not align with the imposed decisions. Let $(\hat{x}, \hat{y}, \hat{u})$ be a fractional solution at a given node. In what follows, we describe the considered branching rules in decreasing order of priority. The rules and their order have been selected based on preliminary computational experiments.

Depot-satellite supply. The first branching level decides on whether or not a depot d delivers freight to a satellite s. The selected depot-satellite pair $(\bar{d}, \bar{s})$ is defined by its $\max_{i \in N_d^C} \sum_{l \in \mathcal{L}_s, p \in \mathcal{P}_l} a_{il} \hat{y}_{lp}$ value. On one branch, we require depot $\bar{d}$ to use $\bar{s}$ as a supply satellite by adding to the MP a constraint forcing to select at least one route $m \in \mathcal{M}_{ds}$. On the other branch, we forbid depot d from using satellite s by setting $x_m = 0$, $\forall m \in \mathcal{M}_{ds}$.

Back-and-forth total flow. The second level of decisions targets the total flow on a back-and-forth route whenever it is fractional. To impose upper and lower bounds on the total back-and-forth flow, we set some of the route variables associated with copies of the selected back-and-forth route to zero (last copies) and one (first copies), respectively.

Vehicle usage. At a third level, we branch on different entities computing a number of vehicles used, by adding constraints to the MP. These entities are grouped in two priority sub-levels. The first sub-level considers branching on the total number of vehicles at the: i) first-echelon ($\sum_{m \in \mathcal{M}} \hat{x}_m$), ii) first-echelon visiting a satellite s ($\sum_{m \in \mathcal{M}_s} \hat{x}_m$), iii) first-echelon departing from depot d ($\sum_{m \in \mathcal{M}_d} \hat{x}_m$), iv) second-echelon ($\sum_{l \in \mathcal{L}, p \in \mathcal{P}_l} \hat{y}_{lp}$). At the second sub-level, we consider branching on the total number of vehicles at the: i) first-echelon departing from a depot d and visiting a satellite s ($\sum_{m \in \mathcal{M}_d \cap \mathcal{M}_s} \hat{x}_m$), ii) second-echelon departing from a satellite s ($\sum_{l \in \mathcal{L}_s, p \in \mathcal{P}_l} \hat{y}_{lp}$).

First-echelon route usage. The fourth level of branching imposes or forbids the use of a given first-echelon route by fixing bounds on the selected variable x_m.

Customer-satellite assignment. The fifth level decides whether to forbid or impose servicing a customer i from a satellite s. The selected customer-satellite pair (i, s) is defined by its $\sum_{l \in \mathcal{L}_s} \sum_{p \in \mathcal{P}_l} a_{il} \hat{y}_{lp}$ value. To forbid and impose servicing customer i from satellite s, we remove customer node i from node set N_s and from node sets $N_{s'}$ ($s' \neq s$), respectively.

Arc flow. The sixth level imposes an integer flow on a second-echelon arc $(i, j) \in A_2$ which is selected according to its $\sum_{l \in \mathcal{L}} \sum_{p \in \mathcal{P}_l} b_{ij}^l \hat{y}_{lp}$ value. Fixing to zero the flow on this arc is done by removing (i, j) from all arc sets A_s, $\forall s \in N^S$. Fixing it to one is achieved by removing arcs (i, k), $k \neq j$, and (h, j), $h \neq i$, from all arc sets A_s, $\forall s \in N^S$.

Customer-satellite-route assignment. Integer flows on first- and second-echelon arcs do not guarantee integrality requirements (5.2.5) on second-echelon variables. In fact, there might exist fractional y_{lp} variables associated with the same second-echelon route but with

different supply patterns, supplying at least one customer i with different first-echelon routes. This rule selects a triplet $(i, s, m) \in N^C \times N^S \times \mathcal{M}$ according to its $\sum_{l \in \mathcal{L}_s} \sum_{p \in \mathcal{P}_l : (m,i) \in p} a_{il} \hat{y}_{lp}$ value. On one branch, customer i must be serviced from satellite s by first-echelon route m. On the other branch, customer i must be serviced either through a satellite $s' \neq s$ or through satellite s as long as a first-echelon route $m' \neq m$ supplies the demand of customer i. To impose these branching decisions, the labeling algorithm needs to be modified as detailed in Appendix B.3.

5.4 Computational experiments

This section reports the results obtained by the proposed BPC algorithm described in Section 5.3. This algorithm was implemented in C/C++ using the GENCOL library (version 4.5). The RMPs are solved using CPLEX 22.1. All tests were performed on a Linux computer equipped with an Intel Core i7-8700 processor clocked at 3.2 GHz and 66 GB of RAM. For all tests, we impose a time limit of three hours.

This section is organized as follows. Section 5.4.1 describes the test instances. Section 5.4.2 summarizes the main computational results and compares the performance of our algorithm with that of Dellaert et al. (2021). Finally, Section 5.4.3 presents a sensitivity analysis on some of the main components of our BPC algorithm to highlight their contribution to the overall algorithm's performance.

5.4.1 MC-2E-VRPTW test instances

To evaluate our BPC algorithm, we consider the instances used by Dellaert et al. (2021), which are derived from the 2E-VRPTW benchmark instances of Dellaert et al. (2019). Let us start by describing the set of these 240 2E-VRPTW instances, which is composed of 12 groups of 20 instances. Each group is characterized by a pair of numbers of depots and satellites $(|N^D|, |N^S|) \in \{(2,3), (3,5), (4,6)\}$ and a number of customers $|N^C| \in \{15, 30, 50, 100\}$. Each group is equally divided into four categories: `Ca`, `Cb`, `Cc`, and `Cd`, that differ by the time window length and the demand distribution. Across all instances, the first- and second-echelon vehicle capacities are $Q_1 = 200$ and $Q_2 = 50$, whereas their fixed costs are set to $f_1 = 50$ and $f_2 = 25$. The service time at every customer and satellite is set to 10.

To create MC-2E-VRPTW instances, Dellaert et al. (2021) introduce commodities by randomly assigning each customer to a depot. They only created 100 MC-2E-VRPTW instances: 60 instances with 15 customers (that we do not consider because they are too easy to solve, often in less than one second), 20 with 30 customers, 10 with 50 customers, and 10 with 100

customers. To complement these instances, we also generated commodities randomly for the other 140 2E-VRPTW instances. For each size and category, Table 5.1 presents the number of instances considered in our tests and their origin.

Table 5.1 Number of instances per size and category

Size		Category											
		$\lvert N^C\rvert = 30$				$\lvert N^C\rvert = 50$				$\lvert N^C\rvert = 100$			
$\lvert N^D\rvert$	$\lvert N^S\rvert$	Ca	Cb	Cc	Cd	Ca	Cb	Cc	Cd	Ca	Cb	Cc	Cd
2	3	5†	5†	5†	5†	5†	5	5	5	5†	5	5	5
3	5	5	5	5	5	5†	5	5	5	5†	5	5	5
6	4	5	5	5	5	5	5	5	5	5	5	5	5

† From Dellaert et al. (2021)

5.4.2 Main computational results

This section presents computational results obtained with the proposed BPC algorithm. First, we report summarized computational results and compare the computational performance of our algorithm with that of the state-of-the-art algorithm of Dellaert et al. (2021). Second, we provide various algorithmic statistics to better understand the solution process. Finally, we list some of the features of the optimal solutions found.

5.4.2.1 Summarized results and comparison with Dellaert et al. (2021)

Table 5.2 presents the average results obtained by our BPC algorithm. For each group of 20 instances defined by its size ($|N^D|, |N^S|, |N^C|$), it indicates the average computational time in seconds for the instances solved to optimality (T) and the number of instances solved to optimality within the 3-hour time limit (# Opt). Detailed results can be found in Appendix B.4.

The results in Table 5.2 show that our BPC algorithm can solve 148 of the 180 instances within the time limit. Its overall performance is obviously impacted negatively by an increase in the number of customers: the average computational times increase rapidly and we can only solve 28 of the 60 instances with 100 customers. Moreover, instance groups with three depots and five satellites require the largest average computational times. The complexity of these instances largely stems from a larger number of first-echelon routes, which complicates the resolution of both the RMPs and the CG subproblems.

In Table 5.3, we compare our BPC algorithm with the best algorithm of Dellaert et al. (2021) (that based on their MC2E-2P model) on the 30-, 50-, and 100-customer instances they con-

Table 5.2 Summary of the computational results

		$\lvert N^C\rvert = 30$		$\lvert N^C\rvert = 50$		$\lvert N^C\rvert = 100$	
$\lvert N^D\rvert$	$\lvert N^S\rvert$	T (s)	# Opt	T (s)	# Opt	T (s)	# Opt
2	3	1.1	20	38.2	20	3247.0	13
3	5	46.1	20	331.1	20	7914.8	3
6	4	18.1	20	126.3	20	4703.7	12

sidered. For each instance group and each algorithm, we report the average computational time in seconds (T) computed over the instances solved to optimality, and the number of instances solved to optimality within the 3-hour time limit (# Opt). For the BPC algorithm, the number of new optimal solutions found (# New) is also provided. To make a fair comparison, we need to account for the bias induced by using a faster computer for our experiments than that used by Dellaert et al. (2021) for theirs. For that matter, the last column specifies the biased-minimal-speed-up (*BMSU*) factor, computed as T^D/T^{BPC}, where T^{BPC} and T^D are the average time of our BPC algorithm and the minimal average time of their algorithm (i.e., setting to 10800 seconds the time for each instance that is solved by our algorithm but not theirs), respectively.

Table 5.3 shows that we can solve all the 30 instances solved by Dellaert et al. (2021) within the time limit. Furthermore, we can also find 7 new solutions, leaving only 3 instances open out of the 40 used for this comparison. Despite using a faster computer for our tests, the average BMSU factors, being more prominent than the speed-up factor provided by using our computer, clearly highlight the fact that our algorithm outperforms the current state-of-the-art algorithm for the MC-2E-VRPTW. The speed-up is, however, less important for the instances with 3 depots and 5 satellites, due to a larger number of first-echelon routes.

5.4.2.2 Statistics on algorithm execution

We now present statistics retrieved during the execution of our BPC algorithm. Table 5.4 reports collected statistics for each group of instances identified by their size in the first three columns. The average statistics are computed over instances solved to optimality. For each group, we report the average numbers of branch-and-bound nodes (# Nodes), enumerated first-echelon routes ($|\mathcal{M}|$), and TIs considered in the model (# TIs). We also specify, for each family of valid inequalities, the number of cuts generated during the solution process (# RCIs, # SRIs, # VSIs).

Table 5.4 asserts previous observations relating the difficulty of solving an instance with

Table 5.3 Comparison with Dellaert et al. (2021)

$\lvert N^D\rvert$	$\lvert N^S\rvert$	$\lvert N^C\rvert$	Dellaert et al. (2021) † T (s)	# Opt	Our BPC algorithm ‡ T (s)	# Opt	# New	BMSU factor
2	3	30	79.6	20/20	1.1	20/20	0	75.1
2	3	50	93.8	4/5	3.5	5/5	1	642.2
3	5	50	2794.2	5/5	424.5	5/5	0	6.6
2	3	100	–	0/5	1536.0	5/5	5	7.0
3	5	100	2919.0	1/5	9867.8	2/5	1	1.4
		Total		30/40		37/40		

† Tests ran on a computer equipped with an Intel Core i7-4770 processor
‡ Tests ran on a computer equipped with an Intel Core i7-8700 processor

the numbers of customers and satellites it involves. For each customer count, the 3-depot, 5-satellite instances are consistently the most challenging to solve compared to their counterparts with depot-satellite pairs $(|N^D|, |N^S|) \in \{(2,3),(6,4)\}$ and the same number of customers $|N^C|$. The increased computational difficulty stems from the significant number of first-echelon routes, requiring many more branching nodes, SRIs, and RCIs. Table 5.4 also shows that the number of TIs is positively correlated with the number of first-echelon routes: an increase of $|\mathcal{M}|$ yields more route pairs in set $\mathcal{K}$. Finally, the VSIs are extensively used across the board. Their usage arises primarily from second-echelon routes requiring more than two first-echelon vehicles to supply their load.

Table 5.4 Summarized statistics on the algorithm execution

$\lvert N^D\rvert$	$\lvert N^S\rvert$	$\lvert N^C\rvert$	# Nodes	$\lvert\mathcal{M}\rvert$	# TIs	# RCIs	# SRIs	# VSIs
2	3	30	15.0	29.9	26.7	32.7	6.9	84.2
3	5	30	65.4	486.7	461.9	65.1	36.9	141.2
6	4	30	51.7	180.0	147.4	57.1	28.7	112.9
2	3	50	139.4	35.2	35.5	85.9	38.3	142.4
3	5	50	366.3	685.7	688.6	228.2	111.4	235.9
6	4	50	125.2	227.3	198.8	164.7	71.6	186.2
2	3	100	2100.8	47.7	59.1	206.2	64.4	278.9
3	5	100	3254.0	965.7	1042.7	476.0	222.7	447.3
6	4	100	1824.8	318.9	321.3	644.1	447.8	369.9

5.4.2.3 Optimal solution statistics

We now summarize some characteristics of the optimal solutions found. For each instance group, Table 5.5 presents the average numbers of satellites ($|N^S|^*$), first-echelon routes ($|\mathcal{M}|^*$), and second-echelon routes ($|\mathcal{L}|^*$) used, as well as the percentage of the first-echelon routes that are back-and-forth ones ($|\mathcal{M}^1|^*$). These results show that increasing the number of customers drives a higher satellite usage rate. In fact, this average rate increases from 70 to 96% as $|N^C|$ increases from 30 to 100. Obviously, the number of first- and second-echelon routes used increases with the customer count. Also, the usage of first-echelon vehicles tends to grow with the number of depots and satellites, especially because every depot must supply the demand of a subset of customers. Finally, the average percentages of first-echelon back-and-forth routes are quite high (between 50 and 83% per group) but much less than those reported for the 2E-VRPTW by Mhamedi et al. (2022) which vary between 72 and 100% per group. Hence, more first-echelon routes visiting multiple satellites are required for the multi-commodity case because the customers in the second-echelon routes must often be supplied by multiple first-echelon routes coming from different depots.

Table 5.5 Summarized optimal solution statistics

$\|N^D\|$	$\|N^S\|$	$\|N^C\|$	$\|N^S\|^*$	$\|\mathcal{M}\|^*$	$\|\mathcal{M}^1\|^*$ (%)	$\|\mathcal{L}\|^*$
2	3	30	2.0	3.4	79.2	9.5
3	5	30	3.3	3.3	50.0	9.6
6	4	30	3.0	6.0	82.5	9.7
2	3	50	3.0	4.4	75.5	15.4
3	5	50	4.3	5.2	52.3	15.4
6	4	50	3.5	6.3	72.9	15.7
2	3	100	3.0	8.3	66.7	30.5
3	5	100	4.3	9.3	75.6	30.0
6	4	100	4.0	10.6	64.1	30.1

5.4.3 Sensitivity analysis

In this section, we conduct a sensitivity analysis to evaluate how some specific components of the BPC algorithm impact its overall computational performance. To this end, we selected a subset of 36 instances with $|N^C| \in \{50, 100\}$ that were challenging but solved to optimality in our previous tests. For the 50-customer instances, we considered all 20 instances with $(|N^D|, |N^S|) = (3, 5)$. For the 100-customer instances, we selected 13 instances with $(|N^D|, |N^S|) = (2, 3)$ and 3 instances with $(|N^D|, |N^S|) = (3, 5)$. For this analysis, we fo-

cus on the following algorithmic components: the depot-satellite supply branching rule (see Section 5.3.4), the symmetry-breaking constraints (5.2.6), the VSIs (5.3.20), the TIs (see Section 5.3.2.1), and the recovery procedure (see Section 5.3.2.2). Other components such as the RCIs, the SRIs, and the other branching rules have been left out of this analysis because their impact has been well evaluated in previous works (RCIs and SRIs) or is relatively minor but essential to obtain the best results on all instances (the other branching rules). To assess the impact of each of the first four components, we have conducted separate computational tests, each time removing only the selected component from the algorithm. For the recovery procedure, we ran additional tests considering a modified procedure that fixes to zero all positive-valued transfer variables (if any) without solving the bin packing feasibility problems before restarting column generation. Table 5.6 presents the computed average results.

For each component and customer count, we report the number of instances solved to optimality within the time limit (# Opt), the average computational time in seconds (T), and the minimum computational time variation in percentage (ΔT) between the complete BPC algorithm and the BPC algorithm with the selected component removed or modified. We compute ΔT as $(T - \bar{T})/\bar{T}$, where $\bar{T}$ is the average computational time of the full BPC algorithm and T that of its corresponding modified counterpart or three hours if it did not solve the instance within the time limit.

First, when removing the depot-satellite supply branching, 5 of the 36 instances cannot be solved anymore within the time limit, showing the effectiveness of using this branching rule. Furthermore, for the 50-customer instances, the average computational time increases by a minimum of 1111.8%, showing the high effectiveness of this rule when the numbers of depots and satellites (3 and 5 for all these instances) yield a large number of first-echelon routes. For the instances with 100 customers, the average computational time slightly decreases, possibly because 13 of these 16 instances do not involve a large number of first-echelon routes. On the other hand, for the other 3 instances, removing the depot-satellite supply branching increases the average time by at least 20.1%.

Second, when omitting the symmetry-breaking constraints, we can only solve 17 50-customer instances with a minimum average time increase of 677.2% and none of the 100-customer instances. Because the number of back-and-forth route copies increases with the number of customers, considering symmetry-breaking constraints is crucial when solving instances with $|N^C| = 100$.

Third, the VSIs impact positively the solution process, especially for instances with 3 depots and 5 satellites where these cuts greatly help improving the lower bounds. When discarded, we can only solve 14 of the 20 50-customer instances while increasing the computational time

Table 5.6 Impact of omitting algorithmic components

Omitted component	Instances with $\lvert N^C\rvert = 50$ # Opt	T (s)	ΔT (%)	Instances with $\lvert N^C\rvert = 100$ # Opt	T (s)	ΔT (%)
Depot-satellite supply branching	17	4012.1	1111.8	14	3812.8	-7.5
Symmetry-breaking constraints	17	2573.3	677.2	0	–	162.0
Visited satellite inequalities	14	3696.4	1016.5	12	4381.7	6.3
Transfer inequalities	14	5833.6	1662.0	10	5859.9	42.2
Recovery procedure	20	444.4	34.2	14	4247.3	3.0

by a factor of at least 10. However, the impact of the VSIs is not as strong for instances with $(|N^D|, |N^S|) = (2, 3)$: the modified algorithm solves 11 of the 13 100-customer instances and yields a small minimum time increase of 5%.

Fourth, the absence of TIs significantly impacts the total times across the board. The total number of instances solved diminishes by 12 and the average times increase substantially, especially for the instances with a large number of first-echelon routes.

Finally, considering the full recovery procedure also has a positive impact on the overall solution process, but less than the above algorithmic components. With the simplified procedure, two instances with 100 customers cannot be solved within the time limit and the average computational time increases moderately (by 34.2%) for the instances with 50 customers. This time increase is due to the fact that there is a large number of first-echelon routes for the 50-customer instances (all with 3 depots and 5 satellites) and, thus, a large number of TIs. Consequently, the recovery procedure is invoked more often during the solution process as more integer solutions with active transfer variables are encountered.

Overall, this sensitivity analysis shows that all these algorithmic components play an important role to achieve the good results that were presented in Section 5.4.2.1. Some of them (depot-satellite supply branching, VSIs, recovery procedure) are, however, less useful when the number of first-echelon routes is small.

5.5 Conclusions

In this paper, we have developed a BPC algorithm for the MC-2E-VRPTW, which incorporates an ad hoc labeling algorithm for generating second-echelon routes and their associated supply patterns. We use dual inequalities to speed up CG convergence and propose a recovery procedure to enforce integer solution feasibility. Our BPC algorithm also includes three families of valid inequalities to improve the lower bounds and a branching scheme featuring

a problem-specific branching rule.

We extended the existing instances of Dellaert et al. (2021) by creating 140 new instances and showed through extensive computational experiments on 180 benchmark instances that our algorithm outperforms the best existing exact algorithm for the problem. It solves to optimality 148 instances within the 3-hour time limit, including 7 previously unsolved instances from Dellaert et al. (2021). We also showed the crucial role of some components of our BPC algorithm to achieve good computational performance. Three of these components proved essential when solving instances with a large number of first-echelon routes.

Nevertheless, solving instances with a large number of depots and satellites remains a challenge. A potential future research avenue could be the development of cuts to further strengthen the lower bounds. Another interesting one would be to extend the problem definition by considering load-dependent transfer times at the satellites.

CHAPITRE 6 UN ALGORITHME EXACT POUR LE PROBLÈME DE TOURNÉES DE VÉHICULES À DEUX ÉCHELONS AVEC FENÊTRES DE TEMPS, COLLECTES ET LIVRAISONS SIMULTANÉES

Dans ce chapitre, nous abordons le 2E-VRPTW-SPD, pour lequel nous développons un algorithme BPC. Dans ce problème, les véhicules de premier échelon assurent le transport de marchandises entre les dépôts et les satellites, tandis que les véhicules de second échelon desservent les clients à partir des satellites en respectant leurs fenêtres de temps. Chaque client a une demande de livraison, de collecte ou les deux. Le 2E-VRPTW-SPD peut s'énoncer comme le problème visant à déterminer les routes de premier et de second échelon les moins coûteuses, de sorte que chaque client soit desservi par une seule route de second échelon à l'intérieur de sa fenêtre de temps et que la synchronisation exacte des opérations soit respectée.

Ce chapitre s'organise comme suit. Dans la section 6.1, nous introduisons la notation utilisée et présentons une formulation mathématique de partitionnement d'ensemble avec contraintes supplémentaires pour notre problème. L'algorithme développé est ensuite décrit en détail à la section 6.2. Enfin, nous présentons, à la section 6.3, les résultats expérimentaux obtenus.

6.1 Formulation mathématique

Le 2E-VRPTW-SPD est formulé sur un graphe $G = (N, A)$, où N correspond à l'ensemble des nœuds et A à l'ensemble des arcs. L'ensemble $N = N^D \cup N^S \cup N^C$ est composé d'ensembles disjoints de dépôts (N^D), satellites (N^S) et clients (N^C). Chaque client $i \in N^C$ est associé à une demande de livraison $q_i^D (\geq 0)$, de collecte $q_i^P (\geq 0)$, ou aux deux. Le service chez un client i donné débute à l'intérieur d'une fenêtre de temps $[\underline{w}_i, \overline{w}_i]$, dure τ_i unités de temps et couvre ses demandes en livrant q_i^D puis en collectant q_i^P. Chaque satellite $s \in N^S$ est associé à un temps de service τ_s correspondant au temps nécessaire pour transférer de la marchandise entre des véhicules associés à différents échelons de routage.

L'ensemble $A = A^1 \cup A^2$ contient les arcs décrivant les déplacements des véhicules sur le premier (A^1) et le second échelon (A^2) . Ainsi, l'ensemble A^1 comprend les arcs entre les paires de i) satellites, ii) dépôts et satellites, et iii) satellites et dépôts, soit $A^1 = \{(d, s) \cup (s, d) : d \in N^D, s \in N^S\} \cup \{(s_1, s_2) : s_1, s_2 \in N^S, s_1 \neq s_2\}$. L'ensemble A^2 inclut les arcs entre les paires de i) clients, ii) clients et satellites, et iii) satellites et clients, soit $A^2 = \{(s, i) \cup (i, s) : s \in N^S, i \in N^C\} \cup \{(i_1, i_2) : i_1, i_2 \in N^C, i_1 \neq i_2\}$. Chaque arc $(u, v) \in A$ est associé à un coût

de transport c_{uv} et un temps de trajet t_{uv} incluant le temps de service au nœud u (c.-à-d. , 0 si $u \in N^D$ et τ_u autrement).

Une flotte homogène de véhicules pour le premier et second échelon est disponible. Le coût fixe d'utilisation d'un véhicule de premier (second) échelon est f_1 (f_2). Les capacités des véhicules de premier et second échelon sont égales à Q^1 et Q^2, respectivement, avec $Q^2 < Q^1$.

Soit $\mathcal{M}$ l'ensemble des routes de premier échelon. Une route de premier échelon $m = (i_0,\ i_1, \ldots,\ i_h,\ i_{h+1})$ commence et se termine à un même dépôt $d_m = i_0 = i_{h+1} \in N^D$, et visite les satellites $i_1, \ldots,\ i_h$, tel que $i_j \neq i_k, \forall j, k \in \{1, \ldots, h\},\ j \neq k$. Le coût de la route m, noté c_m, inclut un coût fixe d'utilisation de véhicule de premier échelon ainsi que les coûts totaux de transport encourus, c.-à-d. , $c_m = f_1 + \sum_{k=0}^{h} c_{i_k, i_{k+1}}$. Une route de premier échelon peut être utilisée pour effectuer des livraisons de marchandises uniquement ou des collectes de marchandises uniquement. Dans les deux cas, la charge totale transportée ne dépasse pas la capacité Q^1. Nous supposons que les routes de livraison peuvent débuter aussitôt que l'on souhaite, soit au temps 0, et que les routes de collecte peuvent terminer aussi tard que l'on veut, soit au temps H, où H désigne la fin de l'horizon de planification. Pour la route m et un satellite s visité par m, on note t_m^s et $\bar{t}_m^s$ les temps de visite au satellite s associés respectivement aux heures de départ la plus tôt et la plus tard à partir du dépôt d_m. Finalement, pour chaque route aller-retour de premier échelon (*back-and-forth first-echelon routes*, BFFERs) $m = (d, s, d)$, $s \in N^S$ et $d \in N^D$, on crée $\lceil \max\{\sum_{i \in N^C} q_i^D, \sum_{i \in N^C} q_i^P\}/Q^1 \rceil$ copies et on note $\mathcal{M}_{ds}^1 \subset \mathcal{M}$ le sous-ensemble définit par ces copies.

Soit $\mathcal{L}$ l'ensemble des routes de second échelon. Une route de second échelon $l = (i_0, i_1, \ldots, i_h, i_{h+1})$ commence et se termine au satellite $s_l = i_0 = i_{h+1} \in N^S$ et visite les clients $i_1, \ldots, i_h$ tel que $i_j \neq i_k, \forall j, k \in \{1, \ldots, h\},\ j \neq k$. Les coûts des routes de second échelon incluent les coûts de transport encourus et un coût fixe d'utilisation de véhicules sur ce dernier. Le coût c_l de la route l s'écrit donc $c_l = f_2 + \sum_{k=0}^{h} c_{i_k, i_{k+1}}$. Pour chaque client $i \in N^C$, soit a_{il} un paramètre binaire prenant la valeur 1 si le client i est visité par la route l, 0 autrement. Les demandes totales livrées et collectées respectent la capacité de véhicule Q^2, soit $\sum_{i \in N^C} a_{il} q_i^D \leq Q^2$ et $\sum_{i \in N^C} a_{il} q_i^P \leq Q^2$. De plus, les marchandises collectées et celles restant à livrer respectent, le long de la route l, la capacité Q^2, c.-à-d. , $\sum_{j=1}^{k} q_j^P + \sum_{j=k+1}^{h} q_j^D \leq Q^2$, pour tout $k \in \{1, \ldots, h\}$.

La synchronisation des opérations considérée entre les véhicules des premier et second échelons au niveau des satellites est *exacte* (*exact operation synchronization*, voir Drexl 2012). Pour le 2E-VRPTW-SPD, cela veut dire que, la demande totale de livraison (respectivement, collecte) associée à chaque route de second échelon est livrée (respectivement, collectée) par un seul véhicule de premier échelon. Pour chaque route $l \in \mathcal{L}$, soit $\mathcal{O}_l \subseteq \mathcal{M}^2$ le sous-

ensemble des *paires de collecte-livraison* (PCLs) associées à l. Une PCL $o \in \mathcal{O}_l$ associée à une route $l \in \mathcal{L}$ spécifie la route de premier échelon m_o^D livrant de la marchandise à l et la route de premier échelon m_o^P collectant de la marchandise de l. De plus, pour chaque route $m \in \mathcal{M}$, notons $Q_{mo}^{D,l}$ et $Q_{mo}^{P,l}$ les quantités de marchandises livrées et collectées par m lorsque la PCL associée à la route l est $o \in \mathcal{O}_l$, c.-à-d. , $Q_{m,o}^{D,l} = \sum_{\{i \in N^C : a_{il}=1 \wedge m=m_o^D\}} q_i^D$ et $Q_{m,o}^{P,l} = \sum_{\{i \in N^C : a_{il}=1 \wedge m=m_o^P\}} q_i^P$.

Soit x_m^D (respectivement, x_m^P) une variable binaire prenant la valeur 1 si la route $m \in \mathcal{M}$ est utilisée pour de la livraison (respectivement, collecte) de marchandises dans la solution, 0 autrement. Soit y_{lo} une variable binaire égale à 1 si la route $l \in \mathcal{L}$, associée à la PCL $o \in \mathcal{O}_l$, est choisie dans la solution, 0 autrement. Le 2E-VRPTW-SPD est décrit par le programme linéaire en nombres entiers suivant :

$$\min \quad \sum_{m \in \mathcal{M}} c_m (x_m^D + x_m^P) + \sum_{l \in \mathcal{L}} \sum_{o \in \mathcal{O}_l} c_l y_{lo} \tag{6.1.1}$$

$$\text{s.c.} \quad \sum_{l \in \mathcal{L}} \sum_{o \in \mathcal{O}_l} a_{il} y_{lo} = 1, \qquad \forall i \in N^C, \tag{6.1.2}$$

$$\sum_{l \in \mathcal{L}} \sum_{o \in \mathcal{O}_l} Q_{mo}^{D,l} y_{lo} \leq Q^1 x_m^D, \qquad \forall m \in \mathcal{M}, \tag{6.1.3}$$

$$\sum_{l \in \mathcal{L}} \sum_{o \in \mathcal{O}_l} Q_{mo}^{P,l} y_{lo} \leq Q^1 x_m^P, \qquad \forall m \in \mathcal{M}, \tag{6.1.4}$$

$$x_m^D, x_m^P \in \{0,1\}, \qquad \forall m \in \mathcal{M}, \tag{6.1.5}$$

$$y_{lo} \in \{0,1\}, \qquad \forall l \in \mathcal{L}, o \in \mathcal{O}_l. \tag{6.1.6}$$

La fonction objectif (6.1.1) minimise les coûts totaux de transport encourus aux deux échelons de routage. Les contraintes de partitionnement (6.1.2) imposent que chaque client soit visité par un véhicule de second échelon. Les contraintes (6.1.3) et (6.1.4) assurent la synchronisation entre les véhicules de premier et second échelon lors des transferts de fret associés aux livraisons et collectes, respectivement. Les contraintes (6.1.5) et (6.1.6) définissent les domaines des variables.

Étant donné que l'ensemble $\mathcal{M}$ considère des copies pour les BFFERs, nous ajoutons au modèle des contraintes de bris de symétrie données par (6.1.7).

$$\begin{aligned} x_m^D &\leq x_{m^+}^D, \\ x_m^P &\leq x_{m^+}^P, \end{aligned} \qquad \forall d \in N^D, s \in N^S, m \in \mathcal{M}_{ds}^1 \setminus \{\bar{m}_s^1\} \tag{6.1.7}$$

où m^+ fait référence à la copie BFFER associée au satellite s et dépôt d placée immédiatement

après la route m dans l'ensemble $\mathcal{M}^1_{ds}$ dont $\bar{m}^1_{ds}$ est le dernier élément.

Le nombre de satellites n'étant pas élevé en pratique, il est possible d'énumérer l'ensemble des routes de premier échelon a priori. Cependant, la même approche ne peut pas être appliquée aux routes de second échelon car leur nombre est généralement très grand. Pour contourner cette difficulté, nous proposons un algorithme de type BPC où les variables de second échelon sont générées dynamiquement lors de la résolution.

6.2 Algorithme BPC pour le 2E-VRPTW-SPD

Un algorithme BPC est un algorithme d'énumération implicite avec plans coupants où la résolution des relaxations linéaires se fait par génération de colonnes (GC). L'algorithme de GC alterne entre la résolution d'un problème maître restreint (PMR) et un ou plusieurs sous-problèmes. Dans le contexte présent, le PMR correspond à la relaxation linéaire de (6.1.1)-(6.1.7) définie sur un sous-ensemble de variables de second échelon, soit $\mathcal{L}' \subseteq \mathcal{L}$ et $\mathcal{O}_{l'} \subseteq \mathcal{O}_l$. Dans un effort de stabiliser les variables duales le long du processus de résolution, nous utilisons des inégalités duales spécifiques au problème.

6.2.1 Sous-problèmes

Dans ce qui suit, nous considérons un sous-problème SP_s par satellite $s \in N^S$. Chaque sous-problème SP_s a pour objectif d'identifier des variables y_{lo} à coût réduit négatif associées à des paires composées d'une route de second échelon $l \in \mathcal{L}$ débutant en s et d'une PCL $o \in \mathcal{O}_l$, ou de prouver qu'il n'en existe aucune.

En notant $(\sigma_i)_{i \in N^C}$, $(\pi_m)_{m \in \mathcal{M}}$ et $(\delta_m)_{m \in \mathcal{M}}$ les variables duales associées respectivement aux contraintes (6.1.2), (6.1.3) et (6.1.4), le coût réduit d'une variable y_{lo} associée à une route de second échelon $l \in \mathcal{L}$ et une PCL $o \in \mathcal{O}_l$ s'écrit comme suit :

$$\bar{c}_{lo} = c_l - \sum_{i \in N^C} a_{il}\sigma_i - \sum_{m \in \mathcal{M}} (Q^{D,l}_{mo}\pi_m - Q^{P,l}_{mo}\delta_m) \tag{6.2.1}$$

Pour chaque route $l \in \mathcal{L}$, il peut exister un grand nombre de PCLs dans l'ensemble $\mathcal{O}_l$. Les PCLs d'intérêt peuvent cependant être prédéterminées en fonction de l'heure de début t^{start}_l de la route l. Bien que cette dernière soit inconnue a priori, elle peut être restreinte à un nombre limité d'heures étant donné que l'attente à toutes les localisations est permise sans coûts additionnels.

En définissant T_s comme l'ensemble des heures de visite au plus tôt des routes de premier échelon au satellite s, nous pouvons considérer des heures de départ potentielles égales à

$t_l^{start} = t + \tau_s$, où $t \in T_s$. De cette manière, il est possible de déterminer, pour la route l, la PCL $o \in \mathcal{O}_l$ produisant un coût réduit $\bar{c}_{lo}$ moindre.

Pour une heure t donnée, notons $m^D(t)$ (respectivement, $m^P(t)$) la route de premier échelon pouvant visiter s avant (respectivement, après) l'heure t et qui est associée à la contrainte (6.1.3) (respectivement, (6.1.4)) dont la variable duale associée prend la valeur la plus élevée, c.-à-d. , $m^D(t) = \arg\max_{\{m \in \mathcal{M}_s \,:\, t_m^s \,\leq\, t\}} \pi_m$ et $m^P(t) = \arg\max_{\{m \in \mathcal{M}_s \,:\, \bar{t}_m^s \,\geq\, t\}} \delta_m$, où $\mathcal{M}_s$ est le sous-ensemble des routes de $\mathcal{M}$ visitant s. Le coût réduit $\bar{c}_{lo}$, donné en (6.2.1), peut désormais être réécrit comme suit :

$$\bar{c}_{lo} = c_l - \sum_{i \in N^C} a_{il}(\sigma_i + q_i^D \pi_{m^D(t_l^{start} - \tau_s)} + q_i^P \delta_{m^P(t_l^{end})}), \tag{6.2.2}$$

où t_l^{end} fait référence à l'heure de fin de la route l.

Chaque sous-problème est un problème du plus court chemin élémentaire avec contraintes de ressources (Irnich and Desaulniers, 2005) qui peut se résoudre par le biais d'un algorithme de programmation dynamique avec étiquettes que nous décrivons dans la prochaine section.

6.2.1.1 Algorithme d'étiquetage

Pour chaque satellite $s \in N^S$, soit $G_s = (N_s, A_s)$ le graphe orienté sur lequel le problème du plus court chemin élémentaire avec contraintes de ressource associé à SP_s est résolu. L'ensemble des nœuds $N_s = N^C \cup \{n_s^{src}, n_s^{sk}\}$ comprend l'ensemble des clients ainsi qu'un nœud source n_s^{src} et un nœud puits n_s^{sk}, représentant des copies du satellite s. Pour uniformiser les notations, nous associons n_s^{src} et n_s^{sk} à des demandes nulles, c.-à-d. , $q_{n_s^{src}}^D = q_{n_s^{sk}}^D = q_{n_s^{src}}^P = q_{n_s^{sk}}^P = 0$. L'ensemble des arcs A_s est défini comme $A_s = \cup_{i \in N^C}\{(n_s^{src}, i), (i, n_s^{sk})\} \cup \{(i_1, i_2) : i_1, i_2 \in N^C, i_1 \neq i_2)\}$. La durée de déplacement sur chaque arc $(i, j) \in A_s$ est égale à t_{ij}. Les coûts (réduits) $\bar{c}_{ij}$ associés aux arcs $(i, j) \in A_s$ sont égaux à :

$$\bar{c}_{ij} = \begin{cases} c_{ij} + f_2 - \sigma_j, & \text{if } i = n_s^{src}, \\ c_{ij} - \sigma_j, & \text{if } i, j \in N^C, \\ c_{ij}, & \text{if } j = n_s^{sk}. \end{cases}$$

Ces coûts d'arcs ne tiennent pas compte des variables duales π_m et δ_m qui seront prises en compte dans les fonctions de prolongation décrites ci-bas. Dans ce qui suit, nous définissons les composantes des étiquettes, décrivons l'initialisation de l'algorithme, les fonctions de prolongation et les règles de dominances considérées.

Définition des étiquettes Dans le contexte présent, une étiquette E_{lt} représente la combinaison d'une route partielle l sur le graphe G_s débutant à l'heure $t+\tau_s$, $t \in T_s$, et d'une PCL associée pouvant être déduite à partir des heures t et T_{lt}^{arr} (définie ci-dessous). L'étiquette E_{lt}, résidant au nœud $i \in N_s$, retient les informations suivantes :

T_{lt}^{rdc} : coût réduit accumulé le long du chemin partiel ;

T_{lt}^{time} : heure de début de service au nœud i ;

T_{lt}^{pick} : quantité totale de marchandise collectée le long du chemin partiel ;

T_{lt}^{max} : charge maximale transportée le long du chemin partiel ;

T_{lt}^{start} : heure la plus tardive à laquelle une route de premier échelon peut visiter s pour fournir la marchandise à livrer ;

T_{lt}^{arr} : heure associée à un retour au satellite s directement après avoir terminé le service au nœud i ;

$\mathcal{U}_{lt}$: ensemble des clients non-atteignables à partir de l'étiquette E_{lt} dû au non-respect des contraintes d'élémentarité, capacité ou fenêtres de temps.

Initialisation Comme mentionné précédemment, les heures de début d'intérêt auxquels nous nous restreignons sont les heures $t + \tau_s$ où $t \in T_s$. Ainsi, l'algorithme d'étiquetage est initialisé au moyen de plusieurs étiquettes, plus précisément une étiquette E_{0t} pour chaque heure $t \in T_s$, où 0 fait référence au chemin partiel contenant uniquement n_s^{src}. Les composantes d'une étiquette E_{0t}, associée à l'heure $t \in T_s$, sont les suivantes :

$$T_{0t}^{rdc} = 0, \tag{6.2.3}$$

$$T_{0t}^{time} = t, \tag{6.2.4}$$

$$T_{0t}^{pick} = 0, \tag{6.2.5}$$

$$T_{0t}^{max} = 0, \tag{6.2.6}$$

$$T_{0t}^{start} = t, \tag{6.2.7}$$

$$T_{0t}^{arr} = 0, \tag{6.2.8}$$

$$\mathcal{U}_{0t} = \{h \in N^C : t + t_{n_s^{src}h} > \overline{w}_h\}. \tag{6.2.9}$$

Fonctions de prolongation Les composantes de l'étiquette $E_{l't}$, produite en prolongeant une étiquette E_{lt} le long d'un arc $(i, j) \in A_s$, sont calculées à l'aide des fonctions de prolon-

gation suivantes :

$$T^{rdc}_{l't} = T^{rdc}_{lt} + \bar{c}_{ij} - \begin{cases} q^D_j \cdot \pi_{m^D(T^{start}_{lt})} - T^{pick}_{lt} \cdot \delta_{m^P(T^{arr}_{lt})} + T^{pick}_{l't} \cdot \delta_{m^P(T^{arr}_{l't})} & \text{si } j \neq n^{sk}_s, \\ 0 & \text{autrement} \end{cases}, \tag{6.2.10}$$

$$T^{time}_{l't} = \max\{\underline{w}_j, T^{time}_{lt} + t_{ij}\}, \tag{6.2.11}$$

$$T^{pick}_{l't} = T^{pick}_{lt} + q^P_j, \tag{6.2.12}$$

$$T^{max}_{l't} = \max\{T^{pick}_{lt} + q^P_j \ , \ T^{max}_{lt} + q^D_j\}, \tag{6.2.13}$$

$$T^{start}_{l't} = T^{start}_{lt}, \tag{6.2.14}$$

$$T^{arr}_{l't} = T^{time}_{l't} + t_{jn^{sk}_s}, \tag{6.2.15}$$

$$\mathcal{U}_{l't} = \mathcal{U}_{lt} \cup \{j\} \cup \{h \in N^C : T^{time}_{l't} + t_{jh} > \overline{w}_h \ \vee \ \max\{T^{pick}_{l't} + q^P_h, T^{max}_{l't} + q^D_h\} > Q_2\} \tag{6.2.16}$$

Règles de dominance Afin d'éviter l'énumération de tous les chemins possibles, nous appliquons la règle de dominance énoncée ci-dessous pour éliminer toute étiquette non-prometteuse.

Définition 6.1. *Une étiquette* $E_{lt} = (T^{rdc}_{lt}, T^{time}_{lt}, T^{pick}_{lt}, T^{start}_{lt}, T^{arr}_{lt}, U_{lt})$ *domine une étiquette* $E_{l't'} = (T^{rdc}_{l't'}, T^{time}_{l't'}, T^{pick}_{l't'}, T^{start}_{l't'}, T^{arr}_{l't'}, U_{l't'})$ *si les deux résident au même nœud et que les conditions suivantes sont satisfaites :*

$$T^{rdc}_{lt} \leq T^{rdc}_{l't'} + (Q^2 - T^{max}_{l't'}) \cdot \Delta_s(T^{start}_{lt}, T^{start}_{l't'}) + \phi(E_{lt}, E_{l't'}), \tag{6.2.17}$$

$$T^{time}_{lt} \leq T^{time}_{l't'}, \tag{6.2.18}$$

$$T^{pick}_{lt} \leq T^{pick}_{l't'}, \tag{6.2.19}$$

$$T^{max}_{lt} \leq T^{max}_{l't'} \tag{6.2.20}$$

$$\mathcal{U}_{lt} \subseteq \mathcal{U}_{l't'}. \tag{6.2.21}$$

où les termes Δ_s *et* ϕ *sont définis comme suit :*

$$\Delta_s(t, t') = \begin{cases} 0 & \textit{si } t \geq t' \\ \pi_{m^D(t)} - \pi_{m^D(t')} & \textit{autrement} \end{cases}$$

$$\phi(E_{lt}, E_{l't'}) = T^{pick}_{l't'} \cdot \delta_{m^P(T^{arr}_{l't'})} - T^{pick}_{lt} \cdot \delta_{m^P(T^{arr}_{lt})}.$$

Proposition 6.1. *Les règles de dominance* (6.2.17)-(6.2.21) *sont valides pour la variante du plus court chemin élémentaire avec contraintes de ressources considérée.*

Preuve 6.1. Dans ce qui suit, le symbole $\oplus$ est utilisé pour représenter la concaténation de deux chemins.

Soit ω une extension réalisable de l' and $\mathcal{C}_\omega$ le sous-ensemble de clients visités le long de cette dernière. Pour prouver la validité de cette proposition, il suffirait de prouver que i) toute extension réalisable ω de l' est également réalisable pour l et ii) le coût réduit du chemin $l \oplus \omega$ est inférieur ou égal à celui de $l' \oplus \omega$.

Les fonctions de prolongation (6.2.11)–(6.2.13) étant décroissantes, il est clair que le chemin $l \oplus \omega$ est réalisable par rapport aux contraintes sur les fenêtres de temps et de capacité. De plus, l'élémentarité de $l \oplus \omega$ ainsi que la condition (6.2.21) stipule que $\mathcal{C}_\omega \cap \mathcal{U}_{lt} = \emptyset$. Le chemin $l \oplus \omega$ est donc élémentaire.

Il ne reste plus qu'à montrer que $T^{rdc}_{l\oplus\omega,t} \leq T^{rdc}_{l'\oplus\omega,t'}$. En notant $c(\omega)$ le coût associé à ω, $\sigma(\omega) = \sum_{j\in\mathcal{C}_\omega} \sigma_j$ et $q^D(\omega) = \sum_{j\in\mathcal{C}_\omega} q^D_j$, les coûts réduits des chemins $l \oplus \omega$ et $l' \oplus \omega$ s'écrivent comme suit :

$$
\begin{aligned}
T^{rdc}_{l\oplus\omega,t} &= T^{rdc}_{lt} + c(\omega) - \sigma(\omega) - q^D(\omega) \cdot \pi_{m^D(T^{start}_{lt})} + T^{pick}_{lt} \cdot \delta_{m^P(T^{arr}_{l,t})} - T^{pick}_{l\oplus\omega,t} \cdot \delta_{m^P(T^{arr}_{l\oplus\omega,t})} \\
T^{rdc}_{l'\oplus\omega,t'} &= T^{rdc}_{l't'} + c(\omega) - \sigma(\omega) - q^D(\omega) \cdot \pi_{m^D(T^{start}_{l't'})} + T^{pick}_{l't'} \cdot \delta_{m^P(T^{arr}_{l',t'})} - T^{pick}_{l'\oplus\omega,t'} \cdot \delta_{m^P(T^{arr}_{l'\oplus\omega,t'})}
\end{aligned}
$$

Leur différence est égale à :

$$
\begin{aligned}
T^{rdc}_{l\oplus\omega,t} - T^{rdc}_{l'\oplus\omega,t'} = T^{rdc}_{lt} - T^{rdc}_{l't'} - q^D(\omega) \cdot (\pi_{m^D(T^{start}_{lt})} - \pi_{m^D(T^{start}_{l't'})}) \\
- \phi(E_{lt}, E_{l't'}) + \phi(E_{l\oplus\omega,t}, E_{l'\oplus\omega,t'})
\end{aligned} \tag{6.2.22}
$$

Étant donné que la fonction de prolongation (6.2.11) associée à la ressource de temps est croissante, on peut établir ce qui suit :

$$
\begin{aligned}
T^{time}_{lt} \leq T^{time}_{l't'} &\implies T^{time}_{l\oplus\omega,t} \leq T^{time}_{l'\oplus\omega,t'} \\
&\implies T^{arr}_{l\oplus\omega,t} \leq T^{arr}_{l'\oplus\omega,t'} \\
&\implies \delta_{m^P(T^{arr}_{l\oplus\omega},t)} \geq \delta_{m^P(T^{arr}_{l'\oplus\omega,t'})} \\
&\overset{\delta\leq 0}{\implies} T^{pick}_{l\oplus\omega,t} \cdot \delta_{m^P(T^{arr}_{l\oplus\omega,t})} \geq T^{pick}_{l'\oplus\omega,t'} \cdot \delta_{m^P(T^{arr}_{l'\oplus\omega,t'})} \\
&\implies \phi(E_{l\oplus\omega,t}, E_{l'\oplus\omega,t'}) \ = T^{pick}_{l'\oplus\omega,t'} \cdot \delta_{m^P(T^{arr}_{l'\oplus\omega,t'})} - T^{pick}_{l\oplus\omega,t} \cdot \delta_{m^P(T^{arr}_{l\oplus\omega,t})} \leq 0.
\end{aligned} \tag{6.2.23}
$$

Pour établir $T^{rdc}_{l\oplus\omega,t} \leq T^{rdc}_{l'\oplus\omega}$, on considère les deux cas suivants :

1. si $T_{lt}^{start} \geq T_{l't'}^{start}$, on a $\pi_{m^D(T_{lt}^{start})} \geq \pi_{m^D(T_{l't'}^{start})}$ et donc $\Delta_s(T_{l,t}^{start}, T_{l't'}^{start}) = 0$. Ainsi

$$T_{l\oplus\omega,t}^{rdc} - T_{l'\oplus\omega,t'}^{rdc} \overset{(6.2.23)}{\leq} T_{lt}^{rdc} - T_{l't'}^{rdc} - \phi(E_{lt}, E_{l't'}) \overset{(6.2.17)}{\leq} 0$$

2. si $T_{lt}^{start} < T_{l't'}^{start}$, on a $\pi_{m^D(T_{lt}^{start})} \leq \pi_{m^D(T_{l't'}^{start})}$. De plus, puisque $q^D(\omega) \leq (Q_2 - T_{l't'}^{max})$, on en déduit que :

$$\begin{aligned} T_{l\oplus\omega,t}^{rdc} - T_{l'\oplus\omega,t'}^{rdc} &\overset{(6.2.23)}{\leq} T_{lt}^{rdc} - T_{l't'}^{rdc} - (Q_2 - T_{l't'}^{max}) \cdot \Delta_s(T_{l,t}^{start}, T_{l't'}^{start}) - \phi(E_{lt}, E_{l't'}) \\ &\overset{(6.2.17)}{\leq} 0. \quad \square \end{aligned}$$

6.2.1.2 Stratégies d'accélération

Afin d'accélérer la génération de colonnes, nous utilisons la relaxation *ng-path* de Baldacci et al. (2011) ainsi que la relaxation décrementale de l'espace d'états (*decremental-state-space-relaxation*, DSSR, Boland et al. (2006); Righini and Salani (2008)). Nous expliquons brièvement ci-après ces deux stratégies d'accélération.

— La relaxation *ng-path* autorise la génération de routes contenant des cycles en définissant, pour chaque client $i \in N^C$, un voisinage $\mathcal{N}_i$. Ainsi, une route peut contenir un cycle $(i, i_1, \cdots, i_n, i)$ s'il existe un $k \in \{1, \ldots, n\}$ telque $i \notin \mathcal{N}_k$. En ajustant la taille des voisinages considérée, les cycles apparaissant dans les routes ne sont pas attractifs en raison des longs détours à effectuer et ont donc peu de chance de faire partie d'une solution optimale au PM. Lors de nos tests préliminaires, nous avons choisi de considérer des voisinages contenant les 15 clients les plus proches.

— La DSSR résout le ESPPRC en relâchant, dans un premier temps, les contraintes d'élémentarité sur les nœuds associés aux clients (ou aux tâches de partitionnements). Si aucun chemin à coût réduit négatif n'est trouvé, ou si au moins un chemin élémentaire à coût réduit négatif est identifié, la résolution s'arrête là. Sinon, les contraintes d'élémentarité sont imposées aux clients apparaissant plus d'une fois dans les chemins non élémentaires précédemment trouvés, avant de relancer l'algorithme d'étiquetage.

6.2.2 Stabilisation par inégalités duales

Afin d'accélérer la convergence de la génération de colonnes, nous utilisons des inégalités duales (Ben Amor et al., 2006; Gschwind and Irnich, 2016; Yarkony et al., 2020). Soit $\mu \in \mathbb{R}^n$ le vecteur des variables duales associées à un programme linéaire borné et réalisable et soit D^* l'ensemble de ses solutions duales-optimales. L'inégalité $\Lambda^T \mu \leq \lambda$, où $\Lambda \in \mathbb{R}^n$ et $\lambda \in \mathbb{R}$, est

une *inégalité duale-optimale* si elle est satisfaite par toutes les solutions de D^*. Un ensemble d'inégalités duales $A\mu \leq a$, avec $A \in \mathbb{R}^{m\times n}$ et $a \in \mathbb{R}^m$, sont des *inégalités duales-optimales profondes* (deep dual-optimal inequalities, DDOIs) si elles sont satisfaites par au moins une solution dans D^*.

Les inégalités duales considérées sont les *inégalités de transfert* (transfer inequalities, TIs), introduites pour le 2E-VRPTW par Mhamedi et al. (2022). Pour le 2E-VRPTW-SPD, ces inégalités sont les suivantes :

$$\pi_{m_1} \geq \pi_{m_2} \ , \ \forall (m_1, m_2) \in \mathcal{K}^D \tag{6.2.24}$$

$$\delta_{m_1} \geq \delta_{m_2} \ , \ \forall (m_1, m_2) \in \mathcal{K}^P \tag{6.2.25}$$

où les ensembles $\mathcal{K}^D$ et $\mathcal{K}^P$ sont définis comme suit :

$$\mathcal{K}^D = \{(m_1, m_2) \in \mathcal{M}^2 \ : \ S(m_1) \subsetneq S(m_2) \ \wedge \ t^s_{m_1} \geq t^s_{m_2} \ , \forall s \in S(m_1)\}$$
$$\mathcal{K}^P = \{(m_1, m_2) \in \mathcal{M}^2 \ : \ S(m_1) \subsetneq S(m_2) \ \wedge \ \bar{t}^s_{m_1} \leq \bar{t}^s_{m_2} \ , \forall s \in S(m_1)\}$$

L'ensemble des inégalités induites par les paires de routes dans $\mathcal{K}^D$ et $\mathcal{K}^P$ sont des DDOIs pour la relaxation linéaire de (6.1.1)-(6.1.7). Afin d'ajouter les TIs au dual de cette dernière, nous introduisons, pour chaque paire $(m_1, m_2) \in \mathcal{K}^h$, où $h \in \{D, P\}$, une variable non négative de coût nul et réécrivons les contraintes de synchronisation (6.1.3) et (6.1.4) comme suit :

$$\sum_{l\in\mathcal{L}} \sum_{o\in\mathcal{O}_l} Q^{D,l}_{mo} y_{lo} - \sum_{(m,m')\in\mathcal{K}^D} u^D_{mm'} + \sum_{(m',m)\in\mathcal{K}^D} u^D_{m'm} \leq Q^1 x^D_m, \qquad \forall m \in \mathcal{M}, \tag{6.2.26}$$

$$\sum_{l\in\mathcal{L}} \sum_{o\in\mathcal{O}_l} Q^{P,l}_{mo} y_{lo} - \sum_{(m,m')\in\mathcal{K}^P} u^P_{mm'} + \sum_{(m',m)\in\mathcal{K}^P} u^P_{m'm} \leq Q^1 x^P_m, \qquad \forall m \in \mathcal{M}. \tag{6.2.27}$$

Les variables ajoutées ci-dessus sont appelées *variables de transfert*. Dans la suite du chapitre, nous utiliserons les notations F et $\tilde{F}$ pour faire référence respectivement aux formulations données par (6.1.1)-(6.1.7) et (6.1.1)-(6.1.2),(6.1.5)-(6.1.7),(6.2.26)-(6.2.27).

6.2.2.1 Réalisabilité et processus de recouvrement

La formulation $\tilde{F}$ étant une relaxation de celle donnée par F, il est possible qu'une solution entière $(\tilde{x}, \tilde{y}, \tilde{u})$, où $\tilde{u} \neq \mathbf{0}$, ne soit pas réalisable pour F, c.-à-d. , elle ne peut être transformée en une solution réalisable pour F ayant le même coût. La figure 6.1 illustre cette situation pour une instance du 2E-VRPTW-SPD considérant 2 dépôts, 3 satellites, 20 clients, $Q^1 = 150$, $Q^2 = 40$, et pour chaque client $i \in N^C$, $q^D_i = 2q^P_i$. La solution associée $(\tilde{x}, \tilde{y}, \tilde{u})$ utilise les

routes de premier échelon m_2 et m_3 pour le transport des demandes de livraison ($\tilde{x}^D_{m_2} = \tilde{x}^D_{m_3} = 1$), m_4 pour le transport des demandes de collectes ($\tilde{x}^P_{m_4} = 1$) et huit routes de second échelon $l_i,\ i \in \{1, \ldots, 8\}$. De plus, deux variables de transfert associées aux paires (m_1, m_2) et (m_1, m_3) de $\mathcal{K}^D$ sont utilisées pour permettre de transférer implicitement des demandes de livraison de m_1 vers m_2 et m_3 ($\tilde{u}^D_{m_1m_2} \cdot \tilde{u}^D_{m_1m_3} > 0$ et $\tilde{x}^D_{m_1} = 0$). Les PCLs $\tilde{o}_{l_i}$ utilisées pour les routes l_i, $i \in \{1, \ldots, 8\}$, sont les suivantes :

$$
\begin{aligned}
\tilde{o}_{l_i} =& (m_1, m_4), & (Q^{D,l_i}_{m_1,\tilde{o}_{l_i}}, Q^{P,l_i}_{m_4,\tilde{o}_{l_i}}) &= (40, 20), & \forall i \in \{1, 2\},\\
\tilde{o}_{l_i} =& (m_2, m_4), & (Q^{D,l_i}_{m_2,\tilde{o}_{l_i}}, Q^{P,l_i}_{m_2,\tilde{o}_{l_i}}) &= (40, 20), & \forall i \in \{3, 4, 5\},\\
\tilde{o}_{l_i} =& (m_3, m_4), & (Q^{D,l_i}_{m_3,\tilde{o}_{l_i}}, Q^{P,l_i}_{m_3,\tilde{o}_{l_i}}) &= (30, 15), & \forall i \in \{6, 7, 8\}.
\end{aligned}
$$

Pour que la totalité des demandes de livraisons de l_1 et l_2 soit transférée de m_1 vers m_2 et m_3 et que $\tilde{x}^D_{m_1} = 0$, les variables de transfert utilisées associées aux paires (m_1, m_2) et (m_1, m_3) ne peuvent prendre que des valeurs $\tilde{u}^D_{m_1m_2} \in [20, 30]$ et $\tilde{u}^D_{m_1m_3} \in [50, 60]$, telles que $\tilde{u}^D_{m_1m_2} + \tilde{u}^D_{m_1m_3} \geq 80$. Cela n'est possible que si la demande totale de livraison de l_1 ou l_2 (i.e., 40 unités) est répartie entre m_2 et m_3. Ainsi, la solution illustrée n'est pas réalisable.

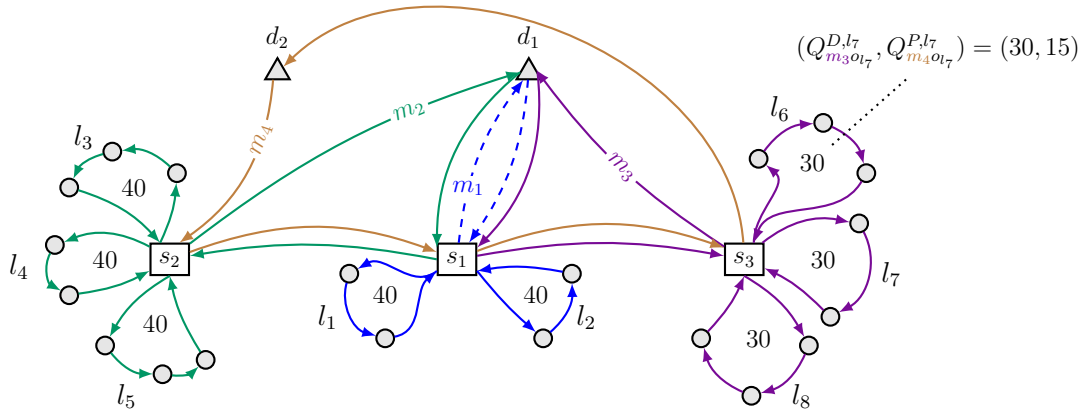

FIGURE 6.1 Exemple d'une solution entière non réalisable

Une solution entière $(\tilde{x}, \tilde{y}, \tilde{u})$, comprenant au moins une variable de transfert à valeur positive, peut possiblement être convertie en une solution réalisable de même coût pour F en utilisant le processus de recouvrement que nous décrivons ci-après. Soit $\tilde{\mathcal{M}}$ et $\tilde{\mathcal{L}}$ les sous-ensembles de routes de premier et second échelon respectivement utilisées dans la solution. Pour chaque route $l \in \tilde{\mathcal{L}}$, notons $\tilde{o}_l \in \mathcal{O}_l$ la PCL utilisée en conjonction avec cette dernière, soit $\tilde{y}_{l\tilde{o}_l} = 1$. Introduisons d'abord de la notation additionnelle.

Soit $h \in \{D, P\}$ un mode de maniement de fret associé à des transferts de demandes de livraisons (D) ou de collectes (P) au niveau des satellites. Pour un mode h donné, soit $\tilde{\mathcal{K}}^h = \{k \in \mathcal{K}^h : \tilde{u}_k^h > 0\}$ l'ensemble des paires dans $\mathcal{K}^h$ associées à des variables de transfert actives et $\tilde{\mathcal{M}}^h = \{m \in \tilde{\mathcal{M}} : \tilde{x}_m^h > 0\}$ l'ensemble des routes de premier échelon utilisées dans la solution pour le mode h. De plus, pour chaque mode h et route $l \in \tilde{\mathcal{L}}$, notons $\tilde{\mathcal{M}}_l^h$ l'ensemble des routes de premier échelon de $\tilde{\mathcal{M}}^h$ pouvant être associées à la route l. Cet ensemble est défini comme suit :

$$\tilde{\mathcal{M}}_l^h = \{m \in \tilde{\mathcal{M}}^h \mid m = m_{\tilde{o}_l}^h \vee \ m_{\tilde{o}_l}^h \overset{h}{\longmapsto} m\}.$$

où la notation $m \overset{h}{\longmapsto} m'$, pour $m, m' \in \mathcal{M}$, signifie qu'il existe au moins une variable active de transfert qui permettrait de transférer du fret de m vers m' lorsqu'en mode h (c.-à-d., $\exists j \geq 2$ et $m_1, \ldots, m_j \in \mathcal{M} \mid (m_i, m_{i+1}) \in \tilde{\mathcal{K}}^h \ \forall 1 \leq i \leq j-1 \wedge \ m_1 = m \ \wedge \ m_j = m'$).

Pour chaque mode $h \in \{D, P\}$, notons $G^h = (V^h, E^h)$ un graphe auxiliaire non orienté associé. Chaque route de premier échelon $m \in \tilde{\mathcal{M}}^h$ est associée à un sommet v_m dans V^h. L'ensemble des arêtes E^h contient une arête entre chaque paire de sommets v_m et $v_{m'}$ dans V^h tel que $(m, m') \in \tilde{\mathcal{K}}^h$ ou $(m', m) \in \tilde{\mathcal{K}}^h$.

Pour alléger le texte, on dira qu'étant donné un sous-ensemble $H \subseteq V^h$, une route de premier échelon $m \in \tilde{\mathcal{M}}^h$ appartient à H si et seulement si le sommet associé v_m est dans H. De manière similaire, une route de second échelon $l \in \tilde{\mathcal{L}}$ est dite appartenir à $H \subseteq V^h$ s'il existe une route de premier échelon $m \in \tilde{\mathcal{M}}_l^h$ appartenant à H (soit $v_m \in H$). Clairement, pour chaque route $l \in \tilde{\mathcal{L}}$, les routes de premier échelon dans $\tilde{\mathcal{M}}_l^h$ appartiennent toutes à la même composante connexe de G^h. Notons $\mathcal{C}(G^h)$ l'ensemble des composantes connexes de G^h.

Le problème de réalisabilité FP^h associé au mode de maniement de fret h utilise, pour chaque $l \in \tilde{\mathcal{L}}$ et $m \in \tilde{\mathcal{M}}_l^h$, une variable binaire λ_m^l qui prend une valeur de 1 lorsque la route de premier échelon m est affectée à la route de second échelon l lorsqu'en mode h, 0 autrement. De plus, FP^h fait appel à des variables non négatives s_l, $l \in \tilde{\mathcal{L}}$, qui prennent respectivement des valeurs non nulles lorsque la demande totale $Q_l^h = \sum_{i \in N^C} a_{il} q_i^h$ de l ne peut être affectée à aucune des routes de $\tilde{\mathcal{M}}_l^h$. Ainsi, FP^h est formulé comme suit :

$$\min \quad \sum_{H \in \mathcal{C}(G^h)} \sum_{l \in H} s_l \tag{6.2.28}$$

$$\text{s.t.} \quad \sum_{m \in \tilde{\mathcal{M}}_l^h} \lambda_m^l + s_l = 1, \qquad \forall l \in \tilde{\mathcal{L}} \tag{6.2.29}$$

$$\sum_{l \in \tilde{\mathcal{L}} : m \in \tilde{\mathcal{M}}_l^h} Q_l^h \lambda_m^l \leq Q_1, \qquad \forall m \in \tilde{\mathcal{M}}^h \tag{6.2.30}$$

$$\lambda_m^l \in \{0,1\}, \qquad \forall l \in \tilde{\mathcal{L}}, m \in \tilde{\mathcal{M}}_l^h \tag{6.2.31}$$

$$s_l \geq 0, \qquad \forall l \in \tilde{\mathcal{L}} \tag{6.2.32}$$

La fonction objectif (6.2.28) minimise le nombre d'affectations non réalisables de routes $l \in \tilde{\mathcal{L}}$ étant donné le mode de maniement de fret h considéré. Les contraintes (6.2.29) imposent que la route de second échelon l ne puisse être approvisionnée que par une seule route de l'ensemble $\tilde{\mathcal{M}}_l^h$. Lorsque cela est impossible, la variable d'écart s_l prendra la valeur 1. Les contraintes (6.2.30) assurent que les capacités associées aux routes de premier échelon utilisées en mode h ne soient pas dépassées. Ce problème peut être résolu par le biais d'un solveur commercial de programmation linéaire mixte en nombre entiers.

Dans le cas où FP^h admet une solution (λ^*, s^*) de coût nul, la solution $(\tilde{x}, \tilde{y}, \tilde{u})$ peut être convertie en une solution réalisable de même coût pour F, utilisant les mêmes routes pour les deux échelons et prenant en compte la solution obtenue pour FP^h afin de redéfinir les routes de premier échelon utilisées pour approvisionner celles de second échelon.

Dans le cas échéant, il faut d'abord identifier, pour chaque mode $h \in \{D, P\}$ associé à une solution (λ^*, s^*) de coût positif, les composantes connexes de G^h dites *non réalisables*, soit celles appartenant à $\{H \in \mathcal{C}(G^h) \mid \exists l \in H : s_l^* > 0\}$. Ensuite, pour chaque composante connexe non réalisable identifiée, les variables de transferts $u_{mm'}^h$, associées aux paires $(m, m') \in \{(m_1, m_2) \in \tilde{\mathcal{K}}^h \mid \exists l \in H : m_{\tilde{o}_l}^h \in \{m_1, m_2\}\}$, sont fixées à zéro avant de redémarrer la génération de colonnes.

6.2.3 Inégalités valides

Afin de renforcer les relaxations linéaires, nous considérons les quatre familles d'inégalités valides suivantes : inégalités de capacité arrondies (*rounded capacity inequalities*, RCIs), inégalités des deux-chemins (*two-path inequalities*, TPIs), inégalités des sous-ensembles de lignes (*subset-row inequalities*, SRIs) et les inégalités des satellites visités (*visited satellite inequalities*, VSIs). L'identification des inégalités violées à un nœud donné se fait dans l'ordre suivant : VSIs, RCIs, TPIs et SRIs. Bien que l'ajout d'inégalités valides puisse se faire en plusieurs rondes à un nœud de branchement, une seule famille d'inégalités est ajoutée par ronde. Afin d'alléger l'effort computationnel requis pour gérer les SRIs au niveau des sous-problèmes, la procédure de séparation pour les SRIs n'est invoquée que pour les nœuds de profondeur inférieure ou égale à 10 dans l'arbre de branchement et chaque ronde de séparation peut résulter en au plus 10 SRIs à ajouter.

Dans ce qui suit, nous présentons brièvement chaque famille d'inégalités et expliquons pour chacune la procédure de séparation utilisée et les modifications à apporter au besoin à l'al-

gorithme d'étiquetage. Pour tout sous-ensemble de clients $C \subseteq N^C$, soit $\delta(C)$ l'ensemble des arcs sortant de C, c.-à-d. , $\delta(C) = \{(i,j) \in A^2 : i \in C, j \notin C\}$. Aussi, pour chaque route de second échelon $l \in \mathcal{L}$ et arc $(i,j) \in A^2$, soit b^l_{ij} un paramètre binaire indiquant si la route l traverse ou non l'arc (i,j).

Inégalités des satellites visités. Les VSIs, introduites par Marques et al. (2020) pour le 2E-CVRP, assurent qu'à chaque fois qu'un client i est desservi à partir d'un satellite s, au moins une route de premier échelon visitant s est utilisée. Dans le contexte actuel, une paire de client i et satellite s donnée est associée aux VSIs suivantes :

$$\sum_{l \in \mathcal{L}_s} \sum_{o \in \mathcal{O}_l} a_{il} y_{lo} \leq \sum_{\{m \in \mathcal{M}_s \,:\, t^s_m + t_{si} \leq \overline{w}_i\}} x^D_m \tag{6.2.33}$$

$$\sum_{l \in \mathcal{L}_s} \sum_{o \in \mathcal{O}_l} a_{il} y_{lo} \leq \sum_{\{m \in \mathcal{M}_s \,:\, \bar{t}^s_m \geq t_{si} + \underline{w}_i\}} x^P_m \tag{6.2.34}$$

où $\mathcal{L}_s$ est le sous-ensemble de routes de second échelon partant du satellite s. Les VSIs sont séparées par énumération. Ces inégalités sont robustes, c'est-à-dire qu'elles n'entraînent aucun changement à la structure des sous-problèmes. Ainsi, la variable duale associée à une VSC associée à un client i et un satellite s est soustraite du coût réduit des arcs sortants de i dans V_s.

Inégalités de capacité arrondies. Soit $\mathcal{C} \subseteq N^C$ un sous-ensemble de clients donné et soit $\xi(\mathcal{C}) = \lceil \max\{\sum_{i \in \mathcal{C}} q^D_i, \sum_{i \in \mathcal{C}} q^P_i\}/Q^2 \rceil$ une borne inférieure sur le nombre de véhicules de second échelon requis pour desservir les clients appartenant à $\mathcal{C}$. La RCI pour $\mathcal{C}$ est :

$$\sum_{l \in \mathcal{L}} \sum_{o \in \mathcal{O}_l} \sum_{(i,j) \in \delta(\mathcal{C})} b^l_{ij} y_{lo} \geq \xi(\mathcal{C}). \tag{6.2.35}$$

L'identification de sous-ensembles de clients associés à des RCIs violées est effectuée en utilisant l'heuristique proposée par Lysgaard et al. (2004). Les RCIs sont également robustes. La variable duale associée à une RCI définie pour un sous-ensemble de clients $\mathcal{C}$ donné est soustraite du coût réduit des arcs appartenant à $\delta(\mathcal{C})$.

Inégalités des deux chemins. Introduites par Kohl et al. (1999) pour le VRPTW, ces inégalités ont été abondamment utilisées dans la littérature pour différentes variantes de VRPs. Soit $\mathcal{C} \subseteq N^C$ un sous-ensemble de clients qui ne peuvent être desservis par un seul véhicule de second échelon en raison des contraintes imposées par les fenêtres de temps. La

TPI associée à $\mathcal{C}$ s'exprime comme suit :

$$\sum_{l \in \mathcal{L}} \sum_{o \in \mathcal{O}_l} \sum_{(i,j) \in \delta(\mathcal{C})} b_{ij}^l y_{lo} \geq 2. \tag{6.2.36}$$

Nous utilisons l'heuristique proposée par Kohl et al. (1999) afin de générer des sous-ensembles de clients candidats. Les TPIs étant robustes, la variable duale associée à une TPI définie sur un sous-ensemble $\mathcal{C}$ donné est retranchée du coût réduit des arcs dans $\delta(\mathcal{C})$.

Inégalités des sous-ensembles de lignes. Introduites par Jepsen et al. (2008) pour le VRPTW, les SRIs sont valides pour le polytope d'empaquetage d'ensembles et correspondent à un sous-ensemble d'inégalités de Chvátal-Gomory de rang 1 basées sur un sous-ensemble de contraintes de partitionnement. Soit $\mathcal{C} \subseteq N^C$ un sous-ensemble de clients. Similairement à Jepsen et al. (2008), nous nous restreignons aux sous-ensembles de trois clients, c.-à-d. , $|\mathcal{C}| = 3$, car l'identification de tels sous-ensembles associés à des SRIs violées peut être faite par énumération. La SRI correspondante à $\mathcal{C}$ est définie de la manière suivante :

$$\sum_{l \in \mathcal{L}} \sum_{o \in \mathcal{O}_l} \left\lfloor \frac{\sum_{i \in \mathcal{C}} a_{il}}{2} \right\rfloor y_{lo} \leq 1. \tag{6.2.37}$$

Les SRIs sont non robustes. Chaque SRC associée à une valeur duale négative nécessite l'ajout d'une ressource binaire à la définition des étiquettes. De plus, la règle de dominance présentée en section 6.2.1.1 doit être modifiée afin de prendre en compte les ressources ajoutées. Pour plus de détails, voir Jepsen et al. (2008).

6.2.4 Branchement

Dans notre algorithme BPC, nous considérons une stratégie de branchement à six niveaux de priorité afin d'obtenir des solutions entières. Pour chaque règle de branchement considérée, une variable (entité) est choisie lorsqu'elle prend la *meilleure* valeur, soit la valeur fractionnaire la plus proche de 0.5. L'arbre de branchement est exploré par la stratégie meilleur d'abord. Soit $(\hat{x}, \hat{y}, \hat{u})$ une solution fractionnaire au PMR obtenue au nœud courant, et soit $h \in \{D, P\}$ un mode de maniement de fret associé aux demandes de livraisons (D) ou de collectes (P). Nous décrivons ci-dessous les règles de branchement considérées par ordre de priorité décroissant. Cet ordre de priorité a été déterminé lors de tests préliminaires.

Flot total sur les BFFERs. Dans un premier temps, nous branchons sur le flot total associé à une BFFER m utilisée en mode h lorsque celui-ci est fractionnaire. Pour imposer

des bornes supérieures et inférieures à ce flot, nous fixons certaines variables de premier échelon associées à des copies de m utilisées en mode h à zéro (copies précédant m) et à un (copies succédant à m) respectivement.

Nombre de véhicules. En deuxième niveau de priorité, nous branchons sur un ensemble d'entités calculant le nombre de véhicules utilisés lorsque ces dernières prennent des valeurs fractionnaires dans la solution. Les décisions sont imposées en ajoutant des contraintes au PM. De plus, les entités considérées sont réparties en deux sous-groupes de priorité. Dans un premier sous-groupe, nous considérons, en plus haute priorité, le nombre de véhicules de i) premier échelon ($\sum_{m\in\mathcal{M},h\in\{D,P\}} \hat{x}_m^h$), ii) premier échelon en mode h ($\sum_{m\in\mathcal{M}} \hat{x}_m^h$), iii) premier échelon visitant un satellite s ($\sum_{m\in\mathcal{M}_s,h\in\{D,P\}} \hat{x}_m^h$), iv) premier échelon visitant un satellite s en mode h ($\sum_{m\in\mathcal{M}_s} \hat{x}_m^h$), v) second échelon ($\sum_{l\in\mathcal{L},o\in\mathcal{O}_l} \hat{y}_{lo}$). Un deuxième sous-groupe considère le nombre de véhicule utilisés de i) second échelon débutant d'un satellite s ($\sum_{l\in\mathcal{L}_s,o\in\mathcal{O}_l} \hat{y}_{lo}$), ii) premier échelon débutant d'un dépôt d et visitant un satellite s ($\sum_{m\in\mathcal{M}_{ds},h\in\{D,P\}} \hat{x}_m^h$), iii) premier échelon en mode h débutant d'un dépôt d et visitant un satellite s ($\sum_{m\in\mathcal{M}_{ds}} \hat{x}_m^h$), où $\mathcal{M}_{ds} = \{m \in \mathcal{M}_s : d_m = d\}$.

Flot sur les routes de premier échelon. En troisième niveau de priorité, nous branchons sur le fait d'utiliser ou non une route de premier échelon m en mode h lorsque la variable associée prend une valeur fractionnaire dans la solution. Ces décisions sont imposées en fixant à zéro ou à un les bornes de la variable sélectionnée x_m^h.

Affectation client-satellite. Le quatrième niveau de priorité branche sur le fait de desservir ou non un client i à partir d'un satellite s. La paire client-satellite $(i, s) \in N^C \times N^S$ sélectionnée est associée à la meilleure valeur $\sum_{l\in\mathcal{L}_s,o\in\mathcal{O}_l} a_{il}\hat{y}_{lo}$. Imposer et interdire que le client i soit desservi à partir de s se fait en retirant le client i respectivement des ensembles de nœuds $N_{s'}$, où $s' \neq s$, et N_s. De plus, les variables de second échelon ne s'alignant pas avec les décisions de branchement imposées sont retirées du PM.

Flot sur les arcs. En cinquième niveau de priorité, nous branchons sur le flot des arcs de second échelon. L'arc (i, j) sélectionné est associé à la meilleure valeur $\sum_{l\in\mathcal{L},o\in\mathcal{O}_l} b_{ij}^l \hat{y}_{lo}$. Interdire l'utilisation de cet arc revient à retirer ce dernier de tous les ensembles A_s, $s \in N^S$. Pour imposer son utilisation, on retire de tous les ensembles A_s, $s \in N^S$, les arcs, autres que (i, j), entrant en et sortant de j et i, respectivement, soit les arcs (v, j), $v \neq i$, et (i, k), $k \neq j$. Dans les deux cas, nous retirons du PM les variables de second échelon qui ne s'alignent pas avec les décisions de branchement imposées.

Affectation client-satellite-route. Il est possible qu'une solution soit fractionnaire malgré que les flots sur les arcs de second échelon soient entiers et que les variables de premier échelon prennent des valeurs entières dans la solution. Ce cas de figure se produit lorsque la solution comprend des variables y à valeurs fractionnaires associées à une même route de second échelon et différentes PCLs. Le sixième niveau de priorité élimine une telle solution lorsqu'elle est prouvée ne pas être équivalente à une solution entière de même coût. À cette fin, des coupes combinatoires sont ajoutées au PM pour rejeter cette solution et empêcher sa régénération à des étapes ultérieures de la résolution. Reprenons ci-après les notations introduites en section 6.2.2.1 afin d'expliquer comment sont identifiés les cas nécessitant l'ajout de coupes combinatoires.

Pour chaque mode de maniement de fret $h \in \{D, P\}$ pour lequel il existe au moins une route $l \in \tilde{\mathcal{L}}$ telle que $|\tilde{\mathcal{M}}_l^h| > 1$, construisons, tel que mentionné en section 6.2.2.1, le graphe auxiliaire G^h et résolvons le problème de faisabilité FP^h associé. Soit (λ^*, s^*) la solution optimale de FP^h obtenue. Pour chaque composante $H \in \mathcal{C}(G^h)$ telle que $\sum_{l \in H} s_l^* > 0$, nous ajoutons au PM la contrainte suivante :

$$\sum_{m \in \tilde{\mathcal{M}}^h(H)} x_m^h + \sum_{l \in \tilde{\mathcal{L}}(H), o \in \mathcal{O}_l} \sum_{(i,j) \in A^2} b_{ij}^l y_{lo} \leq |\tilde{\mathcal{L}}(H)| + |\tilde{\mathcal{M}}^h(H)| - 1 \tag{6.2.38}$$

où $\tilde{\mathcal{L}}(H) = \{l \in \tilde{\mathcal{L}} : l \in H\}$ et $\tilde{\mathcal{M}}^h(H) = \{m \in \tilde{\mathcal{M}}^h : m \in H\}$. Les variables duales associées aux coupes combinatoires (6.2.38) sont soustraites des coûts réduits des arcs impliqués par ces dernières.

6.3 Expériences numériques

Dans cette section, nous présentons les résultats obtenus pour l'algorithme BPC décrit dans la section précédente. À la section 6.3.1, nous décrivons les instances du 2E-VRPTW-SPD utilisées pour nos tests. À la section 6.3.2, nous présentons un sommaire des résultats obtenus. Enfin, nous discutons à la section 6.3.3 l'impact de différentes composantes algorithmiques sur la performance computationnelle de l'algorithme développé.

Tous les tests numériques ont été effectués sur un ordinateur Linux équipé d'un processeur Intel® Core™ i7-8700 fonctionnant à 3,2 GHz et utilisant jusqu'à 66 GB de RAM. L'implémentation de notre algorithme est faite avec les langages de programmation `C/C++` et utilise la librairie GENCOL (version 4.5). De plus, la librairie CPLEX 22.1.0 est utilisée pour résoudre les PMRs et les problèmes de faisabilité. Enfin, un temps limite de calcul de trois heures est imposé par résolution.

6.3.1 Description des instances

Afin d'évaluer notre algorithme BPC, nous utilisons les instances introduites par Lehmann and Winkenbach (2024) pour une variante du 2E-VRPTW-SPD considérant des voyages multiples (*multi-trips*) au second échelon et de la consolidation de fret aux satellites. Ces instances sont dérivées de celles introduites par Dellaert et al. (2019). Lehmann and Winkenbach (2024) introduisent l'aspect des collectes et livraisons dans leurs instances en convertissant les clients dans les instances de Dellaert et al. (2019) en clients ayant soit une demande de livraison ou une demande de collecte, et en assurant une répartition égale entre ces deux groupes de clients. Les instances sont réparties en 12 groupes de 20 instances. Chaque groupe est caractérisé par un nombre de clients et une paire associée à un nombre de dépôts et de satellites. Le nombre de clients $|N^C|$ et les valeurs des paires $(|N^D|, |N^S|)$ considérées appartiennent respectivement à $\{15, 30, 50, 100\}$ et $\{(2, 3), (3, 5), (6, 4)\}$. De plus, chaque groupe d'instances comprend 4 sous-groupes de 5 instances, `Ca`, `Cb`, `Cc` et `Cd`, différant l'un de l'autre par la longueur des fenêtres de temps et les distributions des demandes considérées (voir Dellaert et al., 2019). Pour toutes les instances, les capacités des véhicules de premier et de second échelon sont de $Q^1 = 200$ et $Q^2 = 50$, alors que les coûts fixes d'utilisation pour les véhicules de premier et de second échelon sont de $f_1 = 50$ et $f_2 = 25$, respectivement. Finalement, les temps de service aux clients et satellites sont égaux à $\tau_i = 10$ pour tout $i \in N^C \cup N^S$.

6.3.2 Résultats numériques

Dans cette section, nous présentons les résultats obtenus par notre algorithme BPC. Nous commençons par un récapitulatif de ces résultats, puis exposons diverses statistiques relatives à l'exécution de l'algorithme conçu afin d'avoir une meilleure compréhension du processus de résolution. Enfin, nous discutons de certaines caractéristiques des solutions optimales obtenues.

6.3.2.1 Résultats récapitulatifs

Le tableau 6.1 présente les résultats récapitulatifs obtenus par notre algorithme BPC. Pour chaque nombre de clients, on considère trois groupes de 20 instances définis par le nombre de dépôts $|N^D|$ et de satellites $|N^S|$. Pour chaque groupe d'instances associé à un nombre de clients donné, les informations suivantes sont présentées : le temps de calcul moyen en secondes (T) calculé sur les instances résolues à l'optimalité et le nombre d'instances (# Opt) résolues à l'optimalité dans le temps limite de calcul de trois heures. Une entrée égale

à '-' placée à l'intersection d'une colonne et d'une ligne associées à des valeurs de $|N^C|$ et $(|N^D|, |N^S|)$ signifie qu'aucune des 20 instances définies par ces valeurs n'a été résolue par notre algorithme BPC dans le temps limite de calcul. Des résultats détaillés sont disponibles à l'Annexe C.1.

Les résultats du tableau 6.1 montrent que parmi les 180 instances testées, 125 sont résolues à l'optimalité par notre algorithme BPC. La complexité de résolution est positivement corrélée au nombre de clients considérés. En effet, pour une valeur donnée de $(|N^D|, |N^S|)$, les temps de calcul croissent rapidement avec l'augmentation de $|N^C|$. De plus, pour les instances considérant 100 clients, notre algorithme ne résout que 6 instances avec 2 dépôts et 3 satellites et aucune avec $(|N^D|, |N^S|) \in \{(3,5),(6,4)\}$. Pour les instances associées à $|N^C| \in \{30, 50\}$, le groupe d'instances le plus difficile à résoudre est celui avec 3 dépôts et 5 satellites. Cette difficulté, reflétée par des temps de calcul les plus élevés, provient principalement du grand nombre de routes de premier échelon à considérer pour $(|N^D|, |N^S|) = (3,5)$, ce qui implique une plus grande difficulté à résoudre les SPs et le PM.

Tableau 6.1 Récapitulatif des résultats numériques

		$\lvert N^C\rvert = 30$		$\lvert N^C\rvert = 50$		$\lvert N^C\rvert = 100$	
$\lvert N^D\rvert$	$\lvert N^S\rvert$	T (s)	# Opt	T (s)	# Opt	T (s)	# Opt
2	3	11,3	20	481,4	20	3362,9	6
3	5	340,3	20	3660,3	19	–	0
6	4	29,1	20	1988,5	20	–	0

Comme les instances ont des caractéristiques (lieux, coûts, capacités) semblables aux instances utilisées pour le 2E-VRPTW et le MC-2E-VRPTW et que les tests ont été réalisés sur les mêmes ordinateurs mais avec une version de CPLEX plus récente pour le 2E-VRPTW-SPD, nous pouvons comparer les résultats du tableau 6.1 avec ceux présentés dans les tableaux 4.1 et 5.2 pour le 2E-VRPTW et le MC-2E-VRPTW, respectivement. On peut ainsi remarquer que le 2E-VRPTW-SPD est plus difficile à résoudre que ces deux autres problèmes pour lesquels des totaux de 163 et 148 instances ont été résolues à l'optimalité dans le même temps limite. Ce n'est pas surprenant étant donné que, pour le 2E-VRPTW-SPD, il y a deux familles de routes de premier échelon à considérer, il y a deux groupes de contraintes de synchronisation dans le PM et l'algorithme d'étiquetage est plus complexe à cause de la gestion des collectes et livraisons simultanées. La résolution par notre algorithme BPC de 120 instances du 2E-VRPTW-SPD, notamment toutes les instances avec 30 et 50 clients sauf une, semble indiquer qu'il est efficace.

6.3.2.2 Statistiques sur l'exécution de l'algorithme

Nous présentons les statistiques relatives à l'exécution de notre algorithme BPC dans le tableau 6.2. Ces statistiques sont rapportées par groupes d'instances définis par les valeurs de $(|N^D|, |N^S|) \in \{(2,3),(3,5),(6,4)\}$ et du nombre de clients $|N^C| \in \{30, 50, 100\}$. Les statistiques moyennes sont obtenues à partir des instances résolues à l'optimalité. Pour chaque groupe d'instances, les statistiques présentées sont : le nombre moyen de nœuds de branchement (#Noeuds), le nombre moyen de routes de premier échelon énumérées ($|\mathcal{M}|$), le nombre moyen d'inégalités valides générées pour chaque famille d'inégalités considérée (#SRIs, #RCIs, #TPIs et #VSIs), et le nombre moyen de TIs ajoutées au PM au nœud racine (#TIs).

Les statistiques présentées s'alignent avec les remarques faites précédemment concernant l'impact du nombre de clients et de satellites sur la difficulté de résolution. Pour une valeur donnée de $(|N^D|, |N^S|)$, la résolution devient plus difficile avec l'augmentation du nombre de clients. Cette observation est corroborée par l'augmentation du nombre de nœuds de branchement requis et le nombre de SRIs, RCIs et TPIs générées. Les VSIs sont les plus générées dans l'ensemble et semblent être les plus efficaces pour les instances à 50 clients. Concernant les instances avec $|N^C| \in \{30, 50\}$, les plus difficiles à résoudre sont clairement celles avec 3 dépôts et 5 satellites, qui, comme mentionné précédemment, induisent un grand nombre de routes de premier échelon à considérer. Ceci complexifie la résolution i) des SPs dû à davantage d'étiquettes non dominées et ii) du PMR considérant un grand nombre de contraintes de synchronisation associées aux routes de collecte et de livraison. Enfin, les statistiques présentées mettent en évidence la corrélation positive existante entre le nombre de routes de premier échelon et le nombre de TIs : l'augmentation de $|\mathcal{M}|$ accroît considérablement le nombre de paires de routes de premier échelon satisfaisant les conditions d'appartenance à $\mathcal{K}^D$ et $\mathcal{K}^P$.

6.3.2.3 Statistiques sur les solutions optimales

Le tableau 6.3 présente un récapitulatif des statistiques sur les solutions optimales obtenues par notre algorithme. Les statistiques sont regroupées par groupes d'instances définis par les valeurs de $(|N^D|, |N^S|, |N^C|)$. Pour chaque groupe d'instances, on rapporte les nombres moyens de dépôts ($|N^D|^*$), de satellites ($|N^S|^*$), de routes de premier échelon ($|\mathcal{M}|^*$), et de routes de second échelon ($|\mathcal{L}|^*$) qui sont observés dans les solutions, de même que le pourcentage des BFFERs ($|\mathcal{M}^1|^*$) qui s'y retrouve.

Les statistiques présentées dans le tableau 6.3 indiquent que, plus le nombre de clients aug-

Tableau 6.2 Statistiques résumées sur l'exécution de l'algorithme

$\|N^D\|$	$\|N^S\|$	$\|N^C\|$	# Nœuds	$\|\mathcal{M}\|$	# SRIs	# RCIs	# TPIs	#VSIs	# TIs
2	3	30	21,5	19,3	43,1	9,2	2,6	50,9	49,4
3	5	30	32,3	363,0	51,6	7,1	5,0	125,1	2493,0
6	4	30	14,4	96,8	28,3	10,1	2,4	98,3	379,9
2	3	50	113,0	22,1	105,5	31,9	9,7	108,4	58,1
3	5	50	86,3	397,0	85,4	12,6	7,8	176,3	2800,9
6	4	50	210,4	99,9	98,2	20,9	14,7	168,7	422,1
2	3	100	535,0	27,0	153,3	49,8	19,2	80,5	97,2
3	5	100	–	–	–	–	–	–	–
6	4	100	–	–	–	–	–	–	–

mente, plus il y a de dépôts et de satellites utilisés. En effet, lorsque le nombre de clients passe de 30 à 50, le pourcentage de dépôts utilisés passe en moyenne de 75% à 100% (1,5 à 2,0) pour les instances avec $|N^D| = 2$, de 63% à 70% (1,9 à 2,1) pour les instances avec $|N^D| = 3$, et de 32% à 37% (1,9 à 2,2) pour les instances avec $|N^D| = 6$. De façon similaire, le pourcentage de satellites utilisés passe en moyenne de 50% à 73% (1,5 à 2,2) pour les instances avec $|N^S| = 3$, de 42% à 44% (2,1 à 2,2) pour les instances avec $|N^S| = 5$, et de 48% à 55% pour les instances avec $|N^S| = 4$. Ces augmentations proviennent du fait qu'un plus grand nombre de clients densifie la répartition des clients dans toutes les régions de la ville, justifiant l'utilisation de nouveaux satellites et de nouveaux dépôts. De plus, plus le nombre de clients augmente, plus le nombre de routes de premier et de deuxième échelon utilisées augmente. À titre d'exemple, pour les instances avec $(N^D, N^S) = (2, 3)$, lorsqu'on passe de 30 à 100 clients, le nombre de routes de premier échelon passe de 3,9 à 8,0, alors que le nombre de routes de deuxième échelon passe de 6,3 à 21,3. Ceci est directement corrélé à la capacité des véhicules et à la demande totale à servir qui dépend du nombre de clients. Par ailleurs, pour une valeur donnée de $|N^C|$, les nombres de véhicules de premier et de deuxième échelon ne sont pas grandement impactés par un changement des valeurs de $(|N^D|, |N^S|)$ étant donné que la demande totale varie peu.

6.3.3 Analyse de sensibilité

Dans cette section, nous avons mené une analyse de sensibilité sur quatre composantes algorithmiques importantes afin de déterminer l'impact de chacune sur la performance de l'algorithme BPC développé. Les composantes choisies sont les inégalités de transfert, les VSIs, les TPIs et le processus de recouvrement. Nous avons choisi d'écarter les autres compo-

Tableau 6.3 Statistiques moyennes sur les solutions optimales

$\lvert N^D \rvert$	$\lvert N^S \rvert$	$\lvert N^C \rvert$	$\lvert N^D \rvert^*$	$\lvert N^S \rvert^*$	$\lvert \mathcal{M} \rvert^*$	$\lvert \mathcal{M}^1 \rvert^*$ (%)	$\lvert \mathcal{L} \rvert^*$
2	3	30	1,5	1,5	3,9	98,3	6,3
3	5	30	1,9	2,1	3,8	90,8	6,2
6	4	30	1,9	1,9	3,8	98,3	6,6
2	3	50	2,0	2,2	4,2	94,5	9,9
3	5	50	2,1	2,2	4,3	95,8	9,9
6	4	50	2,2	2,2	4,2	93,0	10,2
2	3	100	1,8	2,5	8,0	100,0	21,3
3	5	100	–	–	–	–	–
6	4	100	–	–	–	–	–

santes parce que leurs impacts ont été bien établis dans nos travaux précédents (SRIs, RCIs et inégalités de bris de symétrie, voir les sections 4.4.3 et 5.4.3) ou que leurs impacts étaient mineurs par rapport aux composantes retenues (règles de branchement).

À cette fin, nous avons sélectionné, parmi les instances résolues à l'optimalité dans nos tests numériques, un sous-ensemble de 39 instances comprenant les plus difficiles à résoudre, soit les instances à trois dépôts, cinq satellites, où 20 et 19 instances avec 30 et 50 clients, respectivement, sont résolues à optimalité.

Pour évaluer et quantifier l'impact de chacune des trois premières composantes, nous avons réalisé des tests numériques sur ces 39 instances en omettant de l'algorithme BPC la composante dont l'impact est évalué. Concernant l'évaluation de l'impact de la procédure de recouvrement, les tests numériques ont été effectués en omettant la procédure de recouvrement présentée à la section 6.2.2.1 et en considérant une procédure de recouvrement simplifiée. Celle-ci consiste à fixer toutes les variables de transfert prenant des valeurs positives à zéro avant de redémarrer la génération de colonnes.

Le tableau 6.4 présente un récapitulatif des statistiques recueillies pour chaque composante omise. Pour les instances avec 30 et 50 clients, on présente le nombre d'instances résolues à optimalité (#Opt), le temps moyen de résolution en secondes (T) ainsi que la moyenne de la variation en pourcentage du temps de résolution (ΔT) calculé pour chaque instance par $(T' - T)/T$, où T' et T représentent le temps de résolution sans la composante omise et avec toutes les composantes (le BPC complet tel que proposé), respectivement. La dernière colonne indique le nombre total d'instances résolues à l'optimalité.

Parmi les quatre composantes testées, la composante ayant l'impact le plus important est

Tableau 6.4 Impact de l'omission de composantes algorithmiques

Composante omise	Instances à $\|N^C\| = 30$ # Opt	T (s)	ΔT (%)	Instances à $\|N^C\| = 50$ # Opt	T (s)	ΔT (%)
Inégalités de transfert	15	2525,1	3131,0	2	4405,8	1651,4
VSIs	20	1457,5	328,3	15	4138,7	13,0
TPIs	20	388,9	14,3	19	3583,0	-2,1
Processus de recouvrement	20	384,7	13,0	19	3450,7	-5,7

l'application des inégalités de transfert. En omettant cette composante, on passe de 20 à 15 instances résolues à optimalité avec 30 clients, et de 19 à 2 instances avec 50 clients. De plus, les temps de calcul sont en moyenne au moins 15 fois plus lents (augmentation moyenne de 3131,0% et 1651,4% avec 30 et 50 clients, respectivement).

Les VSIs impactent aussi fortement la performance algorithmique. Ces coupes sont efficaces en termes d'augmentation des bornes inférieures obtenues. En les omettant, le nombre d'instances résolues à l'optimalité reste égal à 20 pour les instances à 30 clients, et passe de 19 à 15 pour les instances à 50 clients. En outre, en omettant ces coupes, les temps de calcul croissent d'un facteur trois pour les instances à 30 clients et de 13% pour celles avec 50 clients.

L'impact des TPIs est mitigé dans la performance algorithmique. En les omettant, nous parvenons toujours à résoudre toutes les instances testées à l'optimalité. Les TPIs impactent positivement la performance algorithmique pour les instances avec 30 clients où leur omission est accompagnée d'une augmentation moyenne des temps de 14,3 %. Ceci dit, les temps de calcul se détériorent de 2,1% en moyenne pour les instances avec 50 clients lorsque les TPIs sont omises de l'algorithme BPC. Globalement, lorsque la totalité des instances est considérée, les temps de calcul augmentent en moyenne de 6,3% avec l'omission des TPIs.

Finalement, le processus de recouvrement a également un impact positif sur le processus de résolution pour les instances avec 30 clients. Bien que le nombre d'instances résolues à l'optimalité reste le même avec le processus de recouvrement simplifié, les temps de calcul augmentent de 13%. Pour les instances avec 50 clients, nous parvenons toujours à résoudre le même nombre d'instances à l'optimalité. Toutefois, les temps se détériorent en moyenne de 5,7% lorsque le processus de recouvrement est considéré. Sur l'ensemble des instances, une augmentation des temps de calcul de 3,9% reflète l'impact positif global du processus de recouvrement considéré dans l'algorithme BPC complet.

Ces analyses de sensibilité montrent que les deux premières composantes considérées contribuent de façon significative à l'efficacité de l'algorithme BPC développé. Concernant les composantes TPIs et processus de recouvrement, nous avons décidé de les inclure dans l'al-

gorithme BPC complet car leurs impacts demeurent légèrement positifs sur l'ensemble des instances considérées.

6.3.4 Conclusion

Dans ce chapitre, nous avons considéré une variante du 2E-VRPTW considérant des collectes et livraisons simultanées (2E-VRPTW-SPD). Dans ce contexte, une route de second échelon reçoit la totalité de la demande à livrer à partir d'une route de premier échelon et transfère la totalité de la demande collectée à une route de premier échelon. Nous avons développé un algorithme exact de type BPC pour ce problème, incluant un algorithme d'étiquetage spécifique permettant de prendre en compte l'aspect de synchronisation considéré. Nous utilisons des inégalités duales spécifiques aux phases de collectes et livraisons afin d'accélérer le processus de résolution. L'algorithme BPC développé inclut quatre familles d'inégalités valides ainsi qu'un processus de recouvrement assurant la faisabilité des solutions entières en présence des inégalités duales. Des instances incluant jusqu'à 100 clients avec deux dépôts et trois satellites sont résolues à l'optimalité en trois heures de temps de calcul. De plus, nous montrons à travers des analyses de sensibilité l'impact positif de certaines composantes algorithmiques sur la performance computationnelle globale, dont deux sont cruciaux.

CHAPITRE 7 DISCUSSION GÉNÉRALE

Dans cette thèse, nous nous sommes intéressés à trois variantes du problème de tournées de véhicules à deux échelons (le 2E-VRPTW, le MC-2E-VRPTW et le 2E-VRPTW-SPD). Pour ces trois variantes, nous avons proposé des algorithmes exacts basés sur la génération de colonnes avec plans coupants. Dans ce chapitre, nous mettons l'emphase sur trois contributions importantes de nos travaux. D'abord, nous décrivons comment chaque variante étudiée complexifie la résolution du sous-problème. Puis, nous expliquons comment le traitement de la synchronisation permet de développer des inégalités de transfert qui sont essentielles à la performance algorithmique pour l'ensemble des trois problèmes. Finalement, nous revenons sur les règles de branchement proposées, et plus précisément sur le branchement sur le flot total sur les BFFERs.

Premièrement, bien que pour les trois problèmes étudiés la demande d'un client doit provenir d'un seul véhicule de premier échelon, et d'un seul véhicule de deuxième échelon, cela n'est pas traité de la même façon dans la résolution des sous-problèmes. Pour le 2E-VRPTW, ceci implique également que chaque route de deuxième échelon soit réapprovisionnée par une et une seule route de premier échelon. Ce faisant, l'algorithme d'étiquetage développé pour résoudre le sous-problème permet de choisir la route de premier échelon qui réapprovisionne la route de deuxième échelon. Pour y arriver, l'algorithme d'étiquetage développé pour résoudre le sous-problème pour un satellite donné débute avec une étiquette pour chaque route de premier échelon qui visite le satellite. Lorsqu'on considère plusieurs commodités (dans le MC-2E-VRPTW), une route de deuxième échelon peut être réapprovisionnée par plusieurs routes de premier échelon. Par conséquent, l'algorithme d'étiquetage doit être modifié pour prendre cela en considération. Pour y arriver, on propose le concept de patrons de réapprovisionnement (*supply pattern*) qui permet d'identifier quelle combinaison de routes de premier échelon peut approvisionner la route de deuxième échelon. Puisque le nombre de patrons de réapprovisionnement pour chaque route de deuxième échelon peut être élevé, nous considérons alors l'ensemble T_s des temps d'arrivée au satellite s, et pour chaque valeur de T_s, nous pouvons identifier le meilleur patron de réapprovisionnement. L'algorithme d'étiquetage débute alors avec une étiquette pour chaque valeur de T_s. De plus, la règle de dominance est également modifiée afin de pouvoir comparer des étiquettes associées avec différentes heures de départ, pouvant potentiellement permettre de livrer des commodités associées à différents dépôts. Finalement, lorsqu'on considère des collectes et des livraisons simultanées, chaque route de deuxième échelon est réapprovisionnée par une et une seule route de premier échelon (livraisons), et réapprovisionne une et une seule route de premier échelon (collecte). Nous

introduisons la notion de PCLs associées aux routes de second échelon afin d'identifier, pour chaque route de second échelon, la paire de routes de premier échelon utilisées pour la livraison et la collecte de marchandises. De façon similaire au MC-2E-VRPTW, nous considérons pour les routes de livraison un ensemble T_s des temps d'arrivée au plus tôt au satellite s et, pour chaque valeur de T_s, nous pouvons identifier la meilleure route de premier échelon. Pour les routes de collecte, cela ne peut pas être fait a priori puisque le choix de la meilleure route de collecte pour chaque route de second échelon dépend du temps d'arrivée de celle-ci au satellite s. Ainsi, l'algorithme d'étiquetage débute alors avec une étiquette pour chaque valeur de T_s pour laquelle la meilleure route de livraison est identifiée. Cependant, le choix concernant la meilleure route de collecte doit être ajusté au fur et à mesure qu'on prolonge les étiquettes le long des arcs de G_s. En outre, la règle de dominance proposée est ajustée afin d'éliminer les étiquettes, associées à des combinaisons de routes partielles de second échelon et des PCLs associées, n'aboutissant pas à des chemins Pareto-optimaux.

Deuxièmement, les formulations considérées, impliquant des contraintes de capacité définies pour chaque route de premier échelon et des variables encapsulant l'information liée aux routes de second échelon et à la synchronisation exacte des opérations, permet l'exploitation des informations intrinsèques à chaque problème étudié et le développement des inégalités de transfert. Une inégalité de transfert est exprimée en fonction des variables duales associées aux contraintes de synchronisation des quantités de deux routes de premier échelon, m_1 et m_2, liées par des relations que nous définissons dépendamment du problème étudié. Ainsi, si une variable associée à une route de second échelon l et impliquant m_1 existe, il n'est pas nécessaire de générer la variable de second échelon associée à l et impliquant m_2 au lieu de m_1 puisque cette dernière est implicitement disponible pour le problème maître via des variables de transfert. Cette observation est en lien direct avec la réduction du nombre d'itérations de génération de colonnes. Ces inégalités ont été très efficaces, comme l'ont montré les analyses de sensibilité faites pour les trois problèmes considérés. Sans ces inégalités et sur les ensembles d'instances testés lors des analyses de sensibilité pour le 2E-VRPTW, le MC-2E-VRPTW et le 2E-VRPTW-SPD, les temps de calcul augmentent respectivement d'un facteur de 2, 10 et 30 tandis que le nombre d'instances résolues à l'optimalité décroît respectivement de 12%, 33% et 56%.

Troisièmement, dans la méthodologie développée pour les trois problèmes, les règles de branchement exploitent la structure des solutions. Plusieurs règles de branchement sont notamment reliées au nombre de véhicules de premier et de deuxième échelon. Notamment, une des règles de branchement novatrices concerne le flot sur les BFFERs. Dans la structure des solutions pour les trois problèmes, il est fréquent d'utiliser majoritairement des routes de type aller-retour au premier échelon. Pour les trois problèmes étudiés, cette règle de bran-

chement est essentielle à la réduction des temps de calcul et des effets de symétrie induites par l'utilisation de copies pour les routes aller-retour. Les analyses de sensibilité effectuées pour le 2E-VRPTW montrent qu'en omettant cette règle de branchement, les temps de calcul augmentent d'au moins 17%.

CHAPITRE 8 CONCLUSION

Dans cette thèse, nous avons étudié trois variantes du problème de tournées de véhicules à deux échelons. Pour chaque variante, nous avons développé et implanté un algorithme BPC pour résoudre le problème étudié à optimalité. Dans cette section, nous proposons d'abord une synthèse des travaux. Puis, nous présentons des limitations aux travaux effectués ainsi que des pistes de solutions afin de réduire ces limitations.

8.1 Synthèse des travaux

Dans le chapitre 4, nous avons proposé un nouvel algorithme BPC pour le 2E-VRPTW. Les contributions principales de cet article sont : (i) la synchronisation entre les routes de premier et de deuxième échelon qui est faite directement dans l'algorithme d'étiquetage utilisé pour résoudre les sous-problèmes en génération de colonnes, (ii) le développement d'inégalités duales (DDOIs) qui seront, par la suite, modifiées dans les chapitres suivants, et (iii) une nouvelle règle de branchement basée sur les routes aller-retour au premier échelon (BFFERs). Notre algorithme est au minimum trois fois plus rapide que le meilleur algorithme exact connu à ce jour Dellaert et al. (voir 2019) et permet ainsi de résoudre 41 nouvelles instances (parmi 180) à optimalité, laissant ainsi 17 instances non résolues à optimalité.

Dans le chapitre 5, nous avons repris plusieurs idées développés dans le chapitre 4 afin de les étendre à une variante avec commodités multiples (MC-2E-VRPTW). Afin de pouvoir traiter la composante de commodités multiples, nous avons dû modifier de façon importante l'algorithme d'étiquetage utilisé pour résoudre les sous-problèmes en génération de colonnes. Nous avons, entre autres, introduit le concept de patrons de réapprovisionnement (*supply pattern*) qui permet de traiter la synchronisation et de réduire le nombre d'étiquettes générées. Par ailleurs, nous avons également implanté une nouvelle règle de branchement (*depot-satellite supply*) qui permet d'imposer le réapprovisionnement entre un dépôt et un satellite. En comparaison avec les travaux de Dellaert et al. (2021), notre algorithme est plus rapide par un facteur entre 1.4 et 642.2, et nous résolvons 7 nouvelles instances (sur 40) à optimalité. Compte tenu du nombre limité d'instances, nous avons également proposé 140 nouvelles instances avec des caractéristiques différentes (notamment contenant davantage de dépôts et de satellites).

Dans le chapitre 6, nous introduisons une nouvelle variante du 2E-VRPTW où les clients peuvent avoir des demandes de collecte et de livraison, et où une route de deuxième échelon

peut faire à la fois des collectes et des livraisons (2E-VRPTW-SPD). Afin de générer les routes de deuxième échelon, nous devons ainsi adapter les algorithmes d'étiquetage afin de sélectionner une route de premier échelon pour la livraison et une autre route de premier échelon pour la collecte. Le choix de la route de premier échelon pour la livraison est faite de façon similaire aux travaux du chapitre 5, mais le choix de la route de premier échelon pour la collecte est plus compliquée et peut seulement se faire une fois la route de deuxième échelon générée. Cela complexifie ainsi l'algorithme BPC développé. De façon générale, notre algorithme semble assez compétitif pour les instances avec 30 et 50 clients (une instance non résolue à optimalité parmi 120), mais est moins performant pour les instances avec 100 clients (54 instances non résolues à optimalité parmi 60).

8.2 Limitations et améliorations futures

De façon générale, les algorithmes développés dans cette thèse sont efficaces et proposent des composantes novatrices. Néanmoins, ils comportent certaines limitations. Cette section répertorie quelques limitations de nos travaux et propose des pistes de solutions et d'améliorations futures.

Premièrement, dans les trois variantes étudiées, la synchronisation des opérations des véhicules aux satellites est exacte. Dans la littérature, plusieurs auteurs considèrent qu'il est possible de faire de la consolidation aux satellites (e.g., Marques et al., 2022). Sluijk et al. (2023) mentionnent ceci comme étant une limitation de l'article Mhamedi et al. (2022) présenté au chapitre 4. Bien que nous n'avons pas considéré ce type de synchronisation dans nos travaux, il serait possible de modifier nos algorithmes afin de considérer cette variante. Pour ce faire, il serait suffisant de remplacer les contraintes d'intégrité (4.2.8) sur les variables associées aux routes de second échelon (chacune combinée à une route de premier échelon l'approvisionnant) par des contraintes exigeant l'intégrité pour chaque route de second échelon mais seulement la non-négativité sur la route de premier échelon associée. Cela permettrait d'avoir des solutions fractionnaires supportées par des flots entiers sur les arcs de second échelon et des flots entiers sur les routes de premier échelon. De telles solutions sont réalisables pour les cas où la synchronisation des opérations est avec précédence, autorisant ainsi la consolidation de fret aux satellites. À noter que ces adaptations ne permettraient probablement pas d'avoir de meilleures performances que celles obtenues par l'algorithme BPC de Marques et al. (2022), qui reste à ce jour l'état de l'art pour la variante du 2E-VRPTW avec consolidation de fret aux satellites.

Deuxièmement, dans la troisième variante étudiée avec collecte et livraison simultanées, nous considérons que les véhicules de premier échelon peuvent soit faire des livraisons, soit faire

des collectes. En pratique, il est possible qu'un véhicule de premier échelon puisse faire dans une même route à la fois des livraisons et des collectes. Traiter cette particularité nécessiterait du travail supplémentaire. Pour y arriver, il faudrait d'abord bien définir le problème, c'est-à-dire qu'il faudrait déterminer si une route de premier échelon peut faire des livraisons et des collectes dans n'importe quel ordre, ou dans un ordre précis (par exemple, toutes les livraisons doivent être effectuées avant toutes les collectes), et il faudrait déterminer si une route de deuxième échelon peut être approvisionnée (pour la livraison) par une route de premier échelon différente de celle qu'elle réapprovisionne (pour la collecte). La formulation (6.1.1)-(6.1.7) devra inclure une contrainte de synchronisation pour chaque paire de route de premier échelon m et satellite s visité par m. Ces contraintes additionnelles, en plus des contraintes (6.1.3) et (6.1.4), permettront de gérer l'aspect de collectes et livraisons simultanées au niveau du premier échelon. Par la suite, il faudrait également modifier les algorithmes d'étiquetage développés pour générer des routes de deuxième échelon. Ceci impliquerait notamment une augmentation du nombre d'étiquettes, car il serait alors plus difficile de choisir la (les) route(s) de premier échelon qui approvisionnent et se font réapprovisionner par chaque route de deuxième échelon. Il serait ainsi intéressant d'essayer de voir s'il est possible d'étendre un concept similaire à celui du patron de réapprovisionnement développé au chapitre 5 afin de traiter cette variante.

Troisièmement, dans l'extension considérant multiples commodités, celles-ci sont considérées dans une configuration de un-à-un (*one-to-one setting*), c.-à-d. , une commodité fait référence à la demande d'un client et n'est disponible qu'à partir d'un dépôt spécifique. Une autre configuration pouvant être considérée pour les commodités est une de plusieurs-à-plusieurs (*many-to-many setting*, voir Gu et al., 2022). Dans un tel contexte, i) les commodités proviennent de différents dépôts, chacun étant associé à une quantité maximale pouvant être fournie pour chaque commodité, et ii) chaque client peut recevoir plusieurs commodités, chacune livrée lors d'une seule visite de véhicule de second échelon. Traiter une telle variante nécessiterait d'ajuster la notion de patron d'approvisionnement et d'adapter les algorithmes d'étiquetage afin d'être en mesure de prendre en compte simultanément les aspects de synchronisation et ceux liés à la livraison de commodités multiples. Cette variante pourrait être considérée comme une extension du MC-2E-DP considéré dans Petris et al. (2024), où les routes de premier échelon ne seraient pas limitées aux BFFERs et où la synchronisation aux satellites serait exacte.

Quatrièmement, nous pensons que plusieurs paramètres/hypothèses des problèmes pourraient être étudiés (e.g., temps de transfert aux satellites qui dépendent de la quantité transférée, voyages multiples pour les véhicules de premier échelon). Bien que ces variantes peuvent sembler simples à traiter, elles risquent d'apporter plusieurs défis algorithmiques. À titre

d'exemple, dans les trois variantes étudiées, les temps de transfert aux satellites sont considérés fixes et ne dépendent pas de la quantité de marchandise transférée. En pratique, plus la quantité transférée est élevée, plus cela prend du temps. Afin de pouvoir considérer cette variante, il faudrait être en mesure d'affecter des patrons de livraison aux routes de premier échelon spécifiant les quantités pouvant être transférées à chaque satellite. Cela impliquerait aussi qu'une route de premier échelon puisse être utilisée plusieurs fois avec différents patrons de livraison. Un autre exemple concerne le cas où les véhicules de premier échelon pourraient effectuer plusieurs voyages (*multi-trips*). En pratique, cela serait possible notamment lorsque les routes de premier échelon sont de simples aller-retour et que, compte tenu de la longueur d'un quart de travail, un conducteur pourrait effectuer plusieurs routes aller-retour au cours d'une même journée de travail. Ces deux exemples d'extension impliqueront clairement des modifications des formulations à considérer et des ajustements non-triviaux aux algorithmes d'étiquetage afin de générer des routes de premier et de second échelon respectant les aspects temporels additionnels imposés par ces nouvelles variantes.

Cinquièmement, nos résultats démontrent que nos algorithmes sont moins compétitifs lorsqu'il y a un nombre élevé de satellites (notamment dans les instances avec 3 dépôts et 5 satellites). En effet, plus le nombre de satellites est élevé, plus le nombre de routes de premier échelon est élevé. En pratique, il est fréquent que les routes de premier échelon soient des routes aller-retour. Si nous considérions une variante avec uniquement des routes aller-retour au premier échelon, cela permettrait de réduire la complexité algorithmique reliée au nombre élevé de satellites. De façon plus générale, en pratique, le nombre de satellites est souvent limité ce qui laisse présager que nos méthodes seraient tout de même performantes. De plus, considérer des routes aller-retour de premier échelon rendrait la conception d'algorithmes exacts pour des extensions potentielles (e.g., temps de transfert aux satellites et voyages multiples pour les véhicules de premier échelon) moins complexe. Une autre direction serait d'essayer de développer des méthodes de branchement ou des coupes spécifiques pouvant exploiter des propriétés intrinsèques des variantes étudiées pour éliminer davantage les symétries reliées à la considération de routes de premier échelon visitant plusieurs satellites. Pour la résolution d'instances avec beaucoup de satellites et routes de premier échelon, nos approches pourraient aussi potentiellement bénéficier de l'ajout dynamique de variables et de contraintes de synchronisation associées aux routes de premier échelon. Pour ce faire, l'approche algorithmique serait une de type génération de colonnes et contraintes simultanées (*simultaneous column-and-row generation*, voir Spliet, 2024, et les références y figurant). Cette approche de résolution inclurait des sous-problèmes additionnels pour la génération de variables de premier échelon, dont la résolution serait faite par inspection, ainsi que des sous-problèmes dédiés à la génération de contraintes de synchronisation.

Finalement, bien que nos algorithmes soient assez performants, nous pensons qu'inclure des techniques d'apprentissage machine pourraient permettre de les rendre encore plus efficaces. À titre d'exemple, nous pourrions développer des algorithmes d'apprentissage machine afin de sélectionner les satellites à desservir dans les routes de premier niveau ainsi que les arcs de premier échelon à conserver. Un autre exemple serait d'identifier quel(s) satellite(s) peut desservir chaque client ainsi que les arcs de second échelon à conserver. Dans nos instances, on observe qu'en cours de résolution, certains satellites ne sont jamais desservis par certains dépôts, et certains clients ne sont jamais desservis par certains satellites même si les routes pourraient être réalisables. Cela permettrait de réduire la taille des réseaux ainsi que le nombre de routes de premier et de deuxième échelon, ce qui devrait améliorer la performance algorithmique.

RÉFÉRENCES

Amarouche, Y., Guibadj, R. N., and Moukrim, A. (2018). A neighborhood search and set cover hybrid heuristic for the two-echelon vehicle routing problem. In Borndörfer, R. and Storandt, S., editors, *OpenAccess Series in Informatics*, pages 1–15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany.

Baldacci, R., Mingozzi, A., and Roberti, R. (2011). New route relaxation and pricing strategies for the vehicle routing problem. *Operations Research*, 59(5) :1269–1283.

Baldacci, R., Mingozzi, A., Roberti, R., and Calvo, R. W. (2013). An exact algorithm for the two-echelon capacitated vehicle routing problem. *Operations Research*, 61(2) :298–314.

Belenguer, J.-M., Benavent, E., Prins, C., Prodhon, C., and Calvo, R. W. (2011). A branch-and-cut method for the capacitated location-routing problem. *Computers & Operations Research*, 38(6) :931–941.

Ben Amor, H., Desrosiers, J., and Carvalho, J. (2006). Dual-optimal inequalities for stabilized column generation. *Operations Research*, 54 :454–463.

Boland, N., Dethridge, J., and Dumitrescu, I. (2006). Accelerated label setting algorithms for the elementary resource constrained shortest path problem. *Operations Research Letters*, 34(1) :58–68.

Breunig, U., Schmid, V., Hartl, R., and Vidal, T. (2016). A large neighbourhood based heuristic for two-echelon routing problems. *Computers & Operations Research*, 76 :208–225.

Contardo, C., Desaulniers, G., and Lessard, F. (2015). Reaching the elementary lower bound in the vehicle routing problem with time windows. *Networks*, 65(1) :88–99.

Costa, L., Contardo, C., and Desaulniers, G. (2019). Exact branch-price-and-cut algorithms for vehicle routing. *Transportation Science*, 53(4) :946–985.

Crainic, T. G., Mancini, S., Perboli, G., and Tadei, R. (2008). Clustering-based heuristics for the two-echelon vehicle routing problem. Technical Report CIRRELT-2008-46, CIRRELT, Canada.

Crainic, T. G., Mancini, S., Perboli, G., and Tadei, R. (2011). Multi-start heuristics for the two-echelon vehicle routing problem. In Merz, P. and Hao, J.-K., editors, *Evolutionary Computation in Combinatorial Optimization*, pages 179–190, Berlin, Heidelberg. Springer Berlin Heidelberg.

Crainic, T. G., Ricciardi, N., and Storchi, G. (2004). Advanced freight transportation systems for congested urban areas. *Transportation Research Part C : Emerging Technologies*, 12(2) :119–137.

Crainic, T. G., Ricciardi, N., and Storchi, G. (2009). Models for evaluating and planning city logistics systems. *Transportation science*, 43(4) :432–454.

Cuda, R., Guastaroba, G., and Speranza, M. (2015). A survey on two-echelon routing problems. *Computers & Operations Research*, 55 :185–199.

Dellaert, N. (2020). Private communication.

Dellaert, N., Dashty Saridarq, F., Van Woensel, T., and Crainic, T. G. (2019). Branch-and-price–based algorithms for the two-echelon vehicle routing problem with time windows. *Transportation Science*, 53(2) :463–479.

Dellaert, N., Van Woensel, T., Crainic, T. G., and Saridarq, F. D. (2021). A multi-commodity two-echelon capacitated vehicle routing problem with time windows : Model formulations and solution approach. *Computers & Operations Research*, 127 :105154.

Desaulniers, G., Desrosiers, J., and Spoorendonk, S. (2011). Cutting planes for branch-and-price algorithms. *Networks*, 58(4) :301–310.

Drexl, M. (2012). Synchronization in vehicle routing – A survey of VRPs with multiple synchronization constraints. *Transportation Science*, 46(3) :297–316.

Du Merle, O., Villeneuve, D., Desrosiers, J., and Hansen, P. (1999). Stabilized column generation. *Discrete Mathematics*, 194(1-3) :229–237.

Dumez, D., Tilk, C., Irnich, S., Lehuédé, F., Olkis, K., and Péton, O. (2023). A matheuristic for a 2-echelon vehicle routing problem with capacitated satellites and reverse flows. *European Journal of Operational Research*, 305(1) :64–84.

Gondzio, J., González-Brevis, P., and Munari, P. (2013). New developments in the primal-dual column generation technique. *European Journal of Operational Research*, 224(1) :41–51.

Gonzalez-Feliu, J. (2011). Two-echelon freight transport optimisation : unifying concepts via a systematic review. *Working Papers on Operations Management*, 2(1) :18–30.

Gonzalez-Feliu, J., Perboli, G., Tadei, R., and Vigo, D. (2008). The two-echelon capacitated vehicle routing problem.

Gschwind, T. and Irnich, S. (2016). Dual inequalities for stabilized column generation revisited. *INFORMS Journal on Computing*, 28(1) :175–194.

Gu, W., Archetti, C., Cattaruzza, D., Ogier, M., Semet, F., and Speranza, M. G. (2022). A sequential approach for a multi-commodity two-echelon distribution problem. *Computers & Industrial Engineering*, 163 :107793.

Gu, W., Archetti, C., Cattaruzza, D., Ogier, M., Semet, F., and Speranza, M. G. (2023). Vehicle routing problems with multiple commodities : A survey. *European Journal of Operational Research*.

Guastaroba, G., Speranza, M. G., and Vigo, D. (2016). Intermediate facilities in freight transportation planning : A survey. *Transportation Science*, 50(3) :763–789.

Hemmelmayr, V. C., Cordeau, J.-F., and Crainic, T. G. (2012). An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics. *Computers & Operations Research*, 39(12) :3215 – 3228.

Irnich, S. and Desaulniers, G. (2005). Shortest path problems with resource constraints. In Desaulniers, G., Desrosiers, J., and Solomon, M. M., editors, *Column Generation*, chapter 2, pages 33–65. Springer US, Boston, MA, 1st edition.

Jepsen, M., Petersen, B., Spoorendonk, S., and Pisinger, D. (2008). Subset-row inequalities applied to the vehicle-routing problem with time windows. *Operations Research*, 56(2) :497–511.

Jepsen, M., Spoorendonk, S., and Ropke, S. (2013). A branch-and-cut algorithm for the symmetric two-echelon capacitated vehicle routing problem. *Transportation Science*, 47(1) :23–37.

Jia, S., Deng, L., Zhao, Q., and Chen, Y. (2023). An adaptive large neighborhood search heuristic for multi-commodity two-echelon vehicle routing problem with satellite synchronization. *Journal of Industrial and Management Optimization*, 19(2) :1187–1210.

Kohl, N., Desrosiers, J., Madsen, O. B., Solomon, M. M., and Soumis, F. (1999). 2-path cuts for the vehicle routing problem with time windows. *Transportation Science*, 33(1) :101–116.

Laporte, G. and Nobert, Y. (1983). A branch and bound algorithm for the capacitated vehicle routing problem. *Operations-Research-Spektrum*, 5(2) :77–85.

Lehmann, J. and Winkenbach, M. (2024). A matheuristic for the two-echelon multi-trip vehicle routing problem with mixed pickup and delivery demand and time windows. *Transportation Research Part C : Emerging Technologies*, 160 :104522.

Li, J., Xu, M., and Sun, P. (2022). Two-echelon capacitated vehicle routing problem with grouping constraints and simultaneous pickup and delivery. *Transportation Research Part B : Methodological*, 162 :261–291.

Lübbecke, M. E. and Desrosiers, J. (2005). Selected topics in column generation. *Operations Research*, 53(6) :1007–1023.

Lysgaard, J., Letchford, A. N., and Eglese, R. W. (2004). A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Mathematical Programming*, 100(2) :423–445.

Marques, G., Sadykov, R., Deschamps, J.-C., and Dupas, R. (2020). An improved branch-cut-and-price algorithm for the two-echelon capacitated vehicle routing problem. *Computers & Operations Research*, 114 :104833.

Marques, G., Sadykov, R., Dupas, R., and Deschamps, J.-C. (2022). A branch-cut-and-price approach for the single-trip and multi-trip two-echelon vehicle routing problem with time windows. *Transportation Science*, 56(6) :1598–1617.

Mhamedi, T., Andersson, H., Cherkesly, M., and Desaulniers, G. (2022). A branch-price-and-cut algorithm for the two-echelon vehicle routing problem with time windows. *Transportation Science*, 56(1) :245–264.

Naddef, D. and Rinaldi, G. (2002). Branch-and-cut algorithms for the capacitated VRP. In Toth, P. and Vigo, D., editors, *The Vehicle Routing Problem. SIAM Monographs on Discrete Mathematics and Applications*, chapter 3, pages 53–84. Society for Industrial and Applied Mathematics, Philadelphia.

Oukil, A., Ben Amor, H., Desrosiers, J., and El Gueddari, H. (2007). Stabilized column generation for highly degenerate multiple-depot vehicle scheduling problems. *Computers & Operations Research*, 34(3) :817–834.

Perboli, G., Tadei, R., and Vigo, D. (2011). The two-echelon capacitated vehicle routing problem : models and math-based heuristics. *Transportation Science*, 45(3) :364–380.

Pessoa, A., Sadykov, R., Uchoa, E., and Vanderbeck, F. (2018). Automation and combination of linear-programming based stabilization techniques in column generation. *INFORMS Journal on Computing*, 30(2) :339–360.

Petris, M., Archetti, C., Cattaruzza, D., Ogier, M., and Semet, F. (2024). A branch-price-and-cut algorithm for the multi-commodity two-echelon distribution problem. *EURO Journal on Transportation and Logistics*, 13 :100139.

Poggi, M. and Uchoa, E. (2014). New exact algorithms for the capacitated vehicle routing problem. In *Vehicle Routing : Problems, Methods, and Applications, Second Edition*, pages 59–86. SIAM.

Righini, G. and Salani, M. (2008). New dynamic programming algorithms for the resource constrained elementary shortest path problem. *Networks*, 51(3) :155–170.

Rousseau, L.-M., Gendreau, M., and Feillet, D. (2007). Interior point stabilization for column generation. *Operations Research Letters*, 35(5) :660–668.

Santos, F. A., Mateus, G. R., and da Cunha, A. S. (2015). A branch-and-cut-and-price algorithm for the two-echelon capacitated vehicle routing problem. *Transportation Science*, 49(2) :355–368.

Savelsbergh, M. and Van Woensel, T. (2016). 50th anniversary invited article – city logistics : Challenges and opportunities. *Transportation Science*, 50(2) :579–590.

Semet, F., Toth, P., and Vigo, D. (2014). Classical exact algorithms for the capacitated vehicle routing problem. In *Vehicle Routing : Problems, Methods, and Applications, Second*

Edition, pages 37–57. SIAM.

Sluijk, N., Florio, A. M., Kinable, J., Dellaert, N., and Van Woensel, T. (2023). Two-echelon vehicle routing problems : A literature review. *European Journal of Operational Research*, 304(3) :865–886.

Soares, R., Marques, A., Amorim, P., and Parragh, S. N. (2024). Synchronisation in vehicle routing : classification schema, modelling framework and literature review. *European Journal of Operational Research*, 313(3) :817–840.

Spliet, R. (2024). A simple perspective on simultaneous column and row generation. In *Operations Research Forum*, volume 5, page 69. Springer.

Taniguchi, E. and Thompson, R. G. (2002). Modeling city logistics. *Transportation research record*, 1790(1) :45–51.

Yarkony, J., Adulyasak, Y., Desaulniers, G., and Singh, M. K. (2020). Data association via set packing for computer vision applications. *INFORMS Journal on Optimization*, 2(3) :167–191.

ANNEXE A DETAILED RESULTS FOR THE 2E-VRPTW

A.1 Detailed computational results

In this section, we present detailed computational results that allow a comparison between our BPC algorithm and the algorithm of Dellaert et al. (2019). Tables A.1–A.3 report the results for the instances with $|N^C| = \{30, 50, 100\}$, respectively. In each table, the first column indicates the name of the instance as explained in Section 4.4.1. The second column reports the optimal solution value (z^*) when known. Note that for a few instances, we report optimal values that differs from those reported in Dellaert et al. (2019). In a private communication, Dellaert (2020) confirmed that our values are correct. The third column specifies the computational time (T) in seconds required by the algorithm of Dellaert et al. (2019) as reported in their paper. Then, for our BPC algorithm, we provide the computational time (T) in seconds, the integrality gap (*Gap*) computed as $(z^* - \underline{z})/\underline{z}$, where $\underline{z}$ is the lower bound achieved at the root node before adding cuts, the number of nodes explored in the search tree (# *Nodes*), the number of first-echelon routes considered ($|\mathcal{M}|$) and the numbers of DDOIs (4.3.13), RCIs (4.3.28), SRIs (4.3.30), and VSIs (4.3.31) generated. Finally, we indicate the number of depots ($|N^D|^*$), number of satellites ($|N^S|^*$), number of first-echelon routes ($|\mathcal{M}|^*$), percentage of first-echelon routes with only one satellite ($|\mathcal{M}^1|^*$), and number of second-echelon routes ($|\mathcal{L}|^*$) used in the optimal solution obtained by our BPC algorithm. A "–" indicates that no optimal solution was found within the 3-hour time limit.

Table A.1 – Detailed computational results with $|N^C| = 30$

Instance	z^*	Dellaert et al. † T(s)	This paper ‡ T(s)	Gap (%)	# Nodes	$\|\mathcal{M}\|$	# DDOIs	# RCIs	# SRIs	# VSIs	$\|N^D\|^*$	$\|N^S\|^*$	$\|\mathcal{M}\|^*$	$\|\mathcal{M}^1\|^*$ (%)	$\|\mathcal{L}\|^*$
Ca1-2,3,30	981.37	353.2	0.1	0.5	3	21	24	4	0	13	1	2	3	100.0	10
Ca2-2,3,30	968.32	2.6	0.1	0.4	2	25	24	27	0	11	2	3	3	100.0	10
Ca3-2,3,30	1,069.81	0.9	0.1	0.4	2	23	19	37	0	7	2	2	3	100.0	10
Ca4-2,3,30	948.05	4.2	0.6	2.1	15	25	24	35	17	15	2	3	3	66.67	9
Ca5-2,3,30	968.23	9.5	1.5	1.2	18	23	19	79	15	20	2	2	3	100.0	9
Cb1-2,3,30	985.76	5.7	0.4	1.0	7	21	24	29	9	20	1	2	3	100.0	10
Cb2-2,3,30	937.12	9.2	0.2	0.6	4	23	19	17	9	16	2	3	3	100.0	9
Cb3-2,3,30	1,076.25	1.1	0.1	0.1	2	20	15	14	9	17	2	2	3	100.0	10
Cb4-2,3,30	981.34	2.0	0.3	0.7	8	23	20	9	4	12	2	3	3	66.67	9
Cb5-2,3,30	834.85	7.7	0.3	0.9	3	22	18	8	19	25	2	2	2	100.0	8
Cc1-2,3,30	799.56	16.2	1.6	0.3	3	18	18	4	0	10	1	2	2	100.0	8
Cc2-2,3,30	937.71	10.8	3.0	0.1	11	24	20	12	0	31	2	3	3	100.0	10
Cc3-2,3,30	1,013.55	562.3	14.5	0.7	33	17	7	48	16	23	2	2	3	100.0	10
Cc4-2,3,30	943.28	130.9	3.7	1.1	8	23	23	44	12	14	2	3	3	66.67	10
Cc5-2,3,30	949.39	25.6	96.1	1.7	223	23	19	107	74	3	2	2	3	100.0	9
Cd1-2,3,30	807.25	3.1	0.2	0.0	3	18	18	0	0	13	1	2	2	100.0	8
Cd2-2,3,30	945.30	4.1	0.4	0.1	13	24	20	3	0	32	2	3	3	100.0	10
Cd3-2,3,30	1,030.61	9.6	0.3	0.7	7	17	7	10	3	16	2	2	3	100.0	10
Cd4-2,3,30	958.67	19.8	0.4	0.7	11	23	23	21	9	14	2	3	3	66.67	10
Cd5-2,3,30	965.89	55.5	7.3	1.8	107	23	19	71	32	14	2	2	3	100.0	9
Ca1-3,5,30	892.75	18.6	10.3	1.0	35	397	372	38	33	42	3	3	3	100.0	9
Ca2-3,5,30	960.53	24.6	4.3	0.4	9	392	367	28	10	34	3	4	3	66.67	10
Ca3-3,5,30	924.51	17.4	4.7	0.9	14	394	380	20	0	36	3	3	3	100.0	9
Ca4-3,5,30	909.87	18.3	4.0	0.4	12	364	348	23	0	42	3	4	3	66.67	9
Ca5-3,5,30	892.66	15.4	5.5	0.4	9	413	356	6	13	30	3	4	3	66.67	9
Cb1-3,5,30	928.05	48.1	2.8	1.6	5	374	366	5	21	35	3	4	3	66.67	10
Cb2-3,5,30	958.30	73.2	12.8	1.7	36	392	364	37	16	37	3	4	3	66.67	10
Cb3-3,5,30	900.90	8.4	2.3	0.4	6	347	327	17	0	33	3	3	3	100.0	9
Cb4-3,5,30	903.87	23.9	6.6	1.2	19	362	346	16	22	39	3	4	3	66.67	9
Cb5-3,5,30	977.91	45.5	25.6	1.2	55	361	279	50	20	47	2	3	3	100.0	11
Cc1-3,5,30	943.74	62.2	26.8	2.0	28	349	332	35	16	34	3	4	3	66.67	11
Cc2-3,5,30	897.77	65.2	17.9	0.4	11	363	341	49	3	46	3	5	3	33.33	9
Cc3-3,5,30	943.29	29.5	9.4	1.3	6	329	322	40	5	32	3	4	3	66.67	10

Continued on next page

Table A.1 – *Continued from previous page*

Instance	z^*	Dellaert et al. † T(s)	This paper ‡ T(s)	Gap (%)	# Nodes	$\lvert\mathcal{M}\rvert$	# DDOIs	# RCIs	# SRIs	# VSIs	$\lvert N^D\rvert^*$	$\lvert N^S\rvert^*$	$\lvert\mathcal{M}\rvert^*$	$\lvert\mathcal{M}^1\rvert^*$ (%)	$\lvert\mathcal{L}\rvert^*$
`Cc4-3,5,30`	890.65	325.5	37.8	1.3	14	354	336	32	33	39	2	4	3	66.67	9
`Cc5-3,5,30`	772.33	306.6	33.7	0.6	5	388	336	40	10	46	2	3	2	50.0	8
`Cd1-3,5,30`	955.08	36.8	5.3	1.6	12	349	332	32	6	30	3	4	3	66.67	11
`Cd2-3,5,30`	944.60	38.3	2.7	0.5	4	348	324	18	5	36	3	4	3	66.67	9
`Cd3-3,5,30`	907.53	29.2	3.5	0.8	10	344	333	6	8	30	3	4	3	66.67	10
`Cd4-3,5,30`	917.20	95.0	4.1	1.2	6	349	333	39	9	35	2	4	3	66.67	9
`Cd5-3,5,30`	917.01	83.7	12.1	1.5	18	390	347	63	9	46	2	3	3	100.0	9
`Ca1-6,4,30`	922.22	919.4	5.9	1.1	37	101	84	59	7	35	3	3	3	100.0	10
`Ca2-6,4,30`	956.35	2.4	3.3	2.2	35	108	69	63	15	30	3	4	3	66.67	10
`Ca3-6,4,30`	869.33	4.9	1.0	0.3	7	99	73	20	10	25	3	3	3	100.0	9
`Ca4-6,4,30`	867.42	11.4	2.3	1.1	17	106	94	36	12	32	3	4	3	66.67	9
`Ca5-6,4,30`	851.52	1.9	0.9	0.4	7	100	69	12	0	20	3	3	3	100.0	9
`Cb1-6,4,30`	968.05	4.6	5.0	1.0	47	102	88	26	8	46	3	3	3	100.0	10
`Cb2-6,4,30`	920.50	3.1	1.9	0.5	19	98	57	19	4	29	3	3	3	100.0	9
`Cb3-6,4,30`	929.48	3.6	1.2	0.6	12	104	75	18	5	28	3	3	3	100.0	10
`Cb4-6,4,30`	877.54	12.8	3.0	1.4	15	97	84	26	28	32	3	3	3	100.0	9
`Cb5-6,4,30`	895.74	9.3	2.0	1.8	9	104	73	20	10	29	3	3	3	100.0	10
`Cc1-6,4,30`	861.13	39.9	9.6	0.6	8	103	87	25	13	48	3	3	3	100.0	9
`Cc2-6,4,30`	912.38	10.4	6.2	0.0	9	108	77	0	0	27	3	4	3	66.67	10
`Cc3-6,4,30`	899.83	20.1	18.4	1.9	21	105	73	56	16	30	3	4	3	66.67	9
`Cc4-6,4,30`	833.64	190.0	9.7	0.7	10	99	85	41	8	30	3	3	3	100.0	9
`Cc5-6,4,30`	813.91	476.1	9.1	0.6	7	109	77	17	0	31	3	4	3	66.67	9
`Cd1-6,4,30`	918.39	21.6	5.7	0.8	33	97	76	41	20	42	3	3	3	100.0	10
`Cd2-6,4,30`	914.27	3.7	1.4	0.1	8	100	58	0	10	36	3	4	3	66.67	9
`Cd3-6,4,30`	880.59	8.4	1.5	0.9	7	95	60	11	8	30	3	3	3	100.0	9
`Cd4-6,4,30`	885.41	12.6	4.0	1.3	27	91	78	51	19	38	3	3	3	100.0	9
`Cd5-6,4,30`	943.68	16.2	8.0	1.1	67	90	59	72	12	40	3	3	3	100.0	11

† Tests reported with a computer equipped with an Intel Core i7-4770 processor.

‡ Tests reported with a computer equipped with an Intel Core i7-8700 processor.

Table A.2 – Detailed computational results with $|N^C| = 50$

Instance	z^*	Dellaert et al. † T(s)	This paper ‡ T(s)	Gap (%)	# Nodes	$\lvert\mathcal{M}\rvert$	# DDOIs	# RCIs	# SRIs	# VSIs	$\lvert N^D\rvert^*$	$\lvert N^S\rvert^*$	$\lvert\mathcal{M}\rvert^*$	$\lvert\mathcal{M}^1\rvert^*$ (%)	$\lvert\mathcal{L}\rvert^*$
Ca1-2,3,50	1,501.80	2.7	0.4	0.1	7	27	25	0	0	6	2	3	4	100.0	16
Ca2-2,3,50	1,586.21	–	15.8	0.5	112	28	30	68	13	21	2	3	4	100.0	16
Ca3-2,3,50	1,479.56	40.2	1.0	0.5	12	28	30	27	5	24	2	3	4	100.0	16
Ca4-2,3,50	1,485.70	3,427.1	2.5	0.7	25	28	30	34	15	18	2	3	4	100.0	16
Ca5-2,3,50	1,397.05	195.8	4.1	1.2	25	28	30	66	12	23	2	3	4	100.0	15
Cb1-2,3,50	1,455.11	27.4	3.1	0.8	29	28	30	27	12	13	2	3	4	100.0	15
Cb2-2,3,50	1,468.47	840.8	38.1	1.4	299	28	30	61	71	30	2	3	4	100.0	14
Cb3-2,3,50	1,492.25	3,365.1	47.9	1.4	345	28	30	90	32	28	2	3	4	100.0	16
Cb4-2,3,50	1,469.01	566.2	12.3	1.1	86	28	30	43	22	24	2	3	4	100.0	15
Cb5-2,3,50	1,415.23	1,147.9	22.0	1.1	123	28	30	77	44	28	2	3	4	100.0	15
Cc1-2,3,50	1,407.68	–	11.0	0.3	9	27	30	51	6	13	2	3	4	100.0	15
Cc2-2,3,50	1,468.88	–	13.3	0.4	14	27	30	46	9	18	2	3	4	100.0	15
Cc3-2,3,50	1,376.55	–	62.8	1.4	34	28	30	111	18	22	2	3	4	100.0	15
Cc4-2,3,50	1,425.01	–	3,868.3	1.9	3,595	27	30	81	41	33	2	3	4	100.0	15
Cc5-2,3,50	1,442.65	326.8	39.5	1.1	28	27	25	100	27	19	2	3	4	100.0	15
Cd1-2,3,50	1,458.04	30.7	1.4	0.3	12	27	29	44	6	14	2	3	4	100.0	15
Cd2-2,3,50	1,568.68	–	3.0	0.6	15	28	30	81	15	23	2	3	4	100.0	16
Cd3-2,3,50	1,449.70	133.4	1.9	0.9	19	27	30	49	4	20	2	3	4	100.0	16
Cd4-2,3,50	1,453.24	127.9	11.1	0.8	69	28	30	41	15	21	2	3	4	100.0	15
Cd5-2,3,50	1,462.30	31.0	4.7	1.3	21	28	30	75	20	19	2	3	4	100.0	16
Ca1-3,5,50	1,351.83	662.9	17.2	0.5	13	409	390	45	14	41	3	4	4	100.0	16
Ca2-3,5,50	1,354.12	172.1	19.0	1.1	27	475	429	65	11	44	3	5	4	75.0	16
Ca3-3,5,50	1,271.66	488.6	31.5	0.9	41	395	372	56	14	45	2	5	4	75.0	14
Ca4-3,5,50	1,299.86	371.1	207.5	1.2	181	382	323	118	37	54	3	4	4	100.0	15
Ca5-3,5,50	1,267.49	123.2	8.7	1.1	8	395	385	46	10	37	3	5	4	75.0	14
Cb1-3,5,50	1,355.50	631.5	461.5	1.9	441	413	386	112	183	50	2	4	4	100.0	15
Cb2-3,5,50	1,391.31	794.7	396.1	2.0	326	486	449	153	76	41	3	5	4	75.0	16
Cb3-3,5,50	1,368.97	847.4	104.6	1.5	85	397	372	122	51	53	2	4	4	100.0	16
Cb4-3,5,50	1,298.33	258.0	214.1	1.4	205	393	339	39	35	58	3	4	4	100.0	15
Cb5-3,5,50	1,270.39	462.3	128.5	1.2	185	382	370	34	38	57	3	5	4	75.0	14
Cc1-3,5,50	1,293.88	1,507.4	65.5	0.3	12	402	367	30	4	42	3	5	4	75.0	15
Cc2-3,5,50	1,317.82	150.0	83.3	0.7	14	456	390	64	0	28	3	5	4	75.0	15
Cc3-3,5,50	1,288.90	248.0	109.7	0.8	14	394	374	55	16	45	2	5	4	75.0	15

Continued on next page

Table A.2 – *Continued from previous page*

Instance	z^*	Dellaert et al. † T(s)	This paper ‡ T(s)	Gap (%)	# Nodes	$\lvert\mathcal{M}\rvert$	# DDOIs	# RCIs	# SRIs	# VSIs	$\lvert N^D\rvert^*$	$\lvert N^S\rvert^*$	$\lvert\mathcal{M}\rvert^*$	$\lvert\mathcal{M}^1\rvert^*$ (%)	$\lvert\mathcal{L}\rvert^*$
`Cc4-3,5,50`	1,348.83	–	402.1	2.2	45	360	308	137	36	59	3	4	4	100.0	17
`Cc5-3,5,50`	1,276.47	901.9	1,702.5	0.6	331	374	367	105	8	44	3	5	4	75.0	15
`Cd1-3,5,50`	1,314.59	4,997.0	88.4	1.3	81	407	367	108	34	62	2	4	4	100.0	15
`Cd2-3,5,50`	1,326.05	472.3	65.8	1.3	37	444	396	51	27	36	3	5	4	75.0	15
`Cd3-3,5,50`	1,351.50	8,526.9	96.3	1.4	57	383	365	99	44	57	2	5	4	75.0	16
`Cd4-3,5,50`	1,315.54	445.1	30.0	0.5	15	385	320	48	5	43	3	4	4	100.0	16
`Cd5-3,5,50`	1,289.71	2,068.8	96.3	1.1	117	388	381	68	26	52	3	5	4	75.0	15
`Ca1-6,4,50`	1,335.55	–	22.4	1.0	97	102	96	70	25	35	4	4	4	100.0	16
`Ca2-6,4,50`	1,376.13	5,749.5	9.2	2.1	18	102	88	102	39	22	3	3	4	100.0	15
`Ca3-6,4,50`	1,337.15	1,313.9	47.2	1.3	107	117	95	85	10	41	4	4	4	100.0	16
`Ca4-6,4,50`	1,242.56	29.6	5.0	0.4	26	112	84	50	2	41	4	4	4	100.0	15
`Ca5-6,4,50`	1,268.84	81.0	6.4	0.3	27	118	110	30	23	45	4	4	4	100.0	15
`Cb1-6,4,50`	1,355.92	63.8	7.3	1.0	49	101	96	33	19	36	4	4	4	100.0	16
`Cb2-6,4,50`	1,456.85	3,179.1	31.5	2.3	65	103	90	128	38	46	3	3	4	100.0	16
`Cb3-6,4,50`	1,345.02	75.0	43.5	1.0	175	122	103	46	17	49	4	4	4	100.0	15
`Cb4-6,4,50`	1,281.66	36.3	7.1	0.9	37	115	93	8	21	39	4	4	4	100.0	15
`Cb5-6,4,50`	1,341.25	244.4	89.1	1.1	247	114	96	91	49	53	4	4	4	100.0	15
`Cc1-6,4,50`	1,262.60	1,214.0	24.1	0.7	15	98	95	45	10	33	4	4	4	100.0	15
`Cc2-6,4,50`	1,304.68	7,221.0	382.0	1.5	89	100	80	124	70	41	4	4	4	100.0	15
`Cc3-6,4,50`	1,315.85	6,100.0	33.3	0.9	17	101	75	42	16	37	4	4	4	100.0	16
`Cc4-6,4,50`	1,206.64	–	561.2	0.5	289	110	84	43	8	44	4	4	4	100.0	15
`Cc5-6,4,50`	1,453.39	423.5	24.0	0.9	22	110	95	43	0	25	4	4	5	100.0	17
`Cd1-6,4,50`	1,300.53	59.1	11.5	0.4	33	102	96	71	27	41	4	4	4	100.0	15
`Cd2-6,4,50`	1,338.28	73.2	5.9	1.2	13	100	82	68	26	35	3	3	4	100.0	15
`Cd3-6,4,50`	1,375.06	50.1	20.0	1.1	63	113	92	34	17	42	4	4	4	100.0	16
`Cd4-6,4,50`	1,236.79	341.1	13.0	0.6	37	112	76	63	11	48	4	4	4	100.0	15
`Cd5-6,4,50`	1,332.86	1,229.7	87.1	1.2	251	105	83	64	31	46	4	4	4	100.0	16

† Tests reported with a computer equipped with an Intel Core i7-4770 processor.

‡ Tests reported with a computer equipped with an Intel Core i7-8700 processor.

Table A.3 – Detailed computational results with $|N^C| = 100$

Instance	z^*	Dellaert et al. † T(s)	This paper ‡ T(s)	Gap (%)	# Nodes	$\|\mathcal{M}\|$	# DDOIs	# RCIs	# SRIs	# VSIs	$\|N^D\|^*$	$\|N^S\|^*$	$\|\mathcal{M}\|^*$	$\|\mathcal{M}^1\|^*$ (%)	$\|\mathcal{L}\|^*$
Ca1-2,3,100	2,777.87	–	2,310.0	0.8	2,075	39	54	394	85	9	2	3	8	100.0	32
Ca2-2,3,100	2,863.08	–	494.0	0.9	407	40	54	321	76	0	2	3	8	100.0	32
Ca3-2,3,100	2,677.44	–	160.6	0.4	203	40	54	97	26	11	2	3	8	100.0	30
Ca4-2,3,100	3,037.91	8,647.7	21.4	0.8	15	36	54	218	29	0	1	3	8	100.0	31
Ca5-2,3,100	2,757.45	1,487.9	21.3	0.5	21	39	54	136	24	0	2	3	8	100.0	30
Cb1-2,3,100	2,490.59	–	4,051.9	1.1	3,913	36	48	98	130	13	2	3	7	100.0	27
Cb2-2,3,100	2,737.56	–	6,181.0	1.2	5,227	40	54	113	98	13	2	3	8	100.0	29
Cb3-2,3,100	2,716.63	–	4,848.5	0.8	6,371	40	54	124	76	13	2	3	8	100.0	30
Cb4-2,3,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cb5-2,3,100	2,798.85	–	7,382.9	1.0	9,393	39	54	138	133	11	2	3	8	100.0	29
Cc1-2,3,100	2,770.56	–	72.3	0.4	14	39	54	124	14	0	2	3	8	100.0	32
Cc2-2,3,100	2,661.94	–	184.2	0.8	18	39	45	229	36	0	2	3	8	100.0	30
Cc3-2,3,100	2,665.61	–	3,225.5	0.5	691	39	54	131	8	13	2	3	8	100.0	30
Cc4-2,3,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cc5-2,3,100	2,751.94	1,758.5	1,345.9	0.5	349	39	54	168	7	0	2	3	8	100.0	30
Cd1-2,3,100	2,666.54	497.0	41.0	0.3	41	37	45	69	24	9	2	3	8	100.0	30
Cd2-2,3,100	2,852.93	4,021.9	92.5	0.6	53	38	45	180	36	0	2	3	8	100.0	32
Cd3-2,3,100	2,633.61	–	7,644.1	0.9	10,503	40	54	348	106	16	2	3	8	100.0	29
Cd4-2,3,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cd5-2,3,100	2,794.66	1,269.4	310.3	0.5	227	39	54	232	30	0	2	3	8	100.0	30
Ca1-3,5,100	2,608.81	796.4	1,435.2	0.6	825	472	490	206	38	49	3	5	8	100.0	31
Ca2-3,5,100	2,483.29	–	2,938.1	1.5	976	413	443	541	169	42	3	5	8	100.0	30
Ca3-3,5,100	2,452.23	–	206.8	0.5	79	470	490	140	37	31	3	5	8	100.0	29
Ca4-3,5,100	2,648.21	–	2,029.4	0.7	633	496	489	182	33	32	3	5	8	100.0	32
Ca5-3,5,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cb1-3,5,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cb2-3,5,100	2,478.02	–	769.0	1.0	228	420	453	154	166	34	3	5	8	100.0	30
Cb3-3,5,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cb4-3,5,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cb5-3,5,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cc1-3,5,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cc2-3,5,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cc3-3,5,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

Continued on next page

Table A.3 – *Continued from previous page*

Instance	z^*	Dellaert et al. † T(s)	This paper ‡ T(s)	Gap (%)	# Nodes	$\|\mathcal{M}\|$	# DDOIs	# RCIs	# SRIs	# VSIs	$\|N^D\|^*$	$\|N^S\|^*$	$\|\mathcal{M}\|^*$	$\|\mathcal{M}^1\|^*$ (%)	$\|\mathcal{L}\|^*$
Cc4-3,5,100	2,571.43	–	3,015.6	0.6	89	491	487	192	36	24	3	5	8	100.0	31
Cc5-3,5,100	2,638.09	–	10,283.9	0.5	505	475	471	132	23	50	3	5	8	100.0	30
Cd1-3,5,100	2,530.20	4,985.9	2,461.7	0.8	1,165	472	490	197	59	46	3	5	8	100.0	30
Cd2-3,5,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cd3-3,5,100	2,523.66	2,352.4	40.5	0.4	16	471	490	69	24	32	3	5	8	100.0	30
Cd4-3,5,100	2,537.09	–	2,427.8	1.0	810	490	483	136	59	30	3	5	8	100.0	30
Cd5-3,5,100	2,666.30	–	3,341.7	0.7	611	479	486	212	76	22	3	5	8	100.0	30
Ca1-6,4,100	2,537.92	–	2,416.3	1.1	987	139	156	333	76	12	3	4	8	100.0	30
Ca2-6,4,100	2,380.65	429.1	65.5	0.5	27	129	121	149	25	12	4	4	7	100.0	28
Ca3-6,4,100	2,696.39	–	383.7	1.0	229	123	147	245	84	9	3	4	8	100.0	31
Ca4-6,4,100	2,375.64	–	549.9	0.5	388	121	114	174	39	23	3	4	7	100.0	28
Ca5-6,4,100	2,508.41	–	1,265.6	0.9	1,142	122	144	187	65	0	3	4	8	100.0	30
Cb1-6,4,100	2,375.52	–	6,805.1	1.0	2,658	135	144	171	124	26	3	4	7	100.0	28
Cb2-6,4,100	2,566.25	–	1,395.4	0.9	1,281	136	152	105	67	11	4	4	8	100.0	30
Cb3-6,4,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cb4-6,4,100	2,743.53	–	7,427.3	0.7	6,555	122	120	167	68	15	3	4	8	100.0	32
Cb5-6,4,100	2,569.62	–	1,369.0	1.0	1,000	123	144	121	81	0	3	4	8	100.0	31
Cc1-6,4,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cc2-6,4,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cc3-6,4,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cc4-6,4,100	2,559.58	–	625.7	0.7	45	120	122	204	32	10	3	4	8	100.0	30
Cc5-6,4,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cd1-6,4,100	2,591.01	–	9,296.9	0.9	7,934	137	136	228	82	9	3	4	8	100.0	31
Cd2-6,4,100	2,535.57	5,934.1	75.0	0.3	37	134	140	81	32	16	4	4	8	100.0	30
Cd3-6,4,100	2,409.16	–	145.5	0.4	81	120	135	93	28	10	3	4	7	100.0	28
Cd4-6,4,100	2,677.97	–	166.6	0.6	68	119	119	128	64	11	3	4	8	100.0	32
Cd5-6,4,100	2,575.22	–	842.6	0.9	417	118	143	185	63	0	3	4	8	100.0	31

† Tests reported with a computer equipped with an Intel Core i7-4770 processor.

‡ Tests reported with a computer equipped with an Intel Core i7-8700 processor.

A.2 Detailed computational results for the sensitivity analysis

In this section, we present the detailed computational results used to perform the sensitivity analysis described in Section 4.4.3. Tables A.4–A.7 report the results obtained when removing the DDOIs (4.3.13), the RCIs (4.3.28), the SRIs (4.3.30), and the VSIs (4.3.31), respectively. Then, Table A.8 give the results obtained when adding a second-copy variable for the first-echelon routes visiting more than one satellite.

For each table, the first column indicates the instance name. Then, we report the computational time (T) in seconds, the average computational time difference in percentage between the original BPC algorithm and the version of the algorithm tested (ΔT), where this difference is computed as $(T - T^*)/T^*$, with T^* being the total computational time required by the original algorithm. Next, we provide the number of nodes explored in the search tree (*# Nodes*), the number of first echelon routes enumerated ($|\mathcal{M}|$) and the numbers of DDOIs (4.3.13), RCIs (4.3.28), SRIs (4.3.30), and VSIs (4.3.31) generated whenever the family of inequalities is considered. A "–" indicates that no optimal solution was found within the 3-hour time limit.

Table A.4 Detailed results when omitting DDOIs

Instance	T (s)	ΔT (%)	# Nodes	$\lvert\mathcal{M}\rvert$	# RCIs	# SRIs	# VSIs
Cc4-2,3,50	3,251.0	-16.0	2,871	27	92	57	25
Cc1-3,5,50	203.0	209.9	14	402	29	4	41
Cd2-3,5,50	264.5	302.0	53	444	56	29	36
Cc2-3,5,50	278.1	233.9	10	456	63	0	40
Cd1-3,5,50	487.6	451.6	75	407	103	31	67
Cd3-3,5,50	144.4	49.9	41	383	85	34	55
Cd5-3,5,50	365.3	279.3	119	388	75	25	46
Cb3-3,5,50	490.0	368.5	87	397	125	48	53
Cc3-3,5,50	209.1	90.6	16	394	44	17	45
Cb5-3,5,50	489.0	280.5	203	382	36	25	54
Ca4-3,5,50	642.9	209.8	309	382	80	27	59
Cb4-3,5,50	1,904.9	789.7	115	393	50	48	52
Cb2-3,5,50	1,980.4	400.0	413	486	150	70	41
Cc4-3,5,50	639.1	58.9	39	360	111	30	57
Cb1-3,5,50	1,532.1	232.0	391	413	116	168	44
Cc5-3,5,50	9,865.6	479.5	601	374	103	9	49
Cd5-6,4,50	166.4	91.0	245	105	61	31	49
Cb5-6,4,50	160.5	80.1	237	114	125	38	50
Cc2-6,4,50	473.2	23.9	103	100	121	68	40
Cc4-6,4,50	658.5	17.3	185	110	45	7	44
Ca2-2,3,100	1,160.5	134.9	893	40	301	50	10
Cc5-2,3,100	2,515.4	86.9	367	39	214	12	0
Ca1-2,3,100	2,794.2	21.0	2,196	39	380	87	16
Cc3-2,3,100	4,914.6	52.4	941	39	132	10	15
Cb2-3,5,100	3,978.5	417.4	285	420	157	159	37
Ca1-3,5,100	3,261.8	127.3	783	472	211	28	60
Ca4-3,5,100	–	–	–	–	–	–	–
Cd4-3,5,100	–	–	–	–	–	–	–
Cd1-3,5,100	8,027.7	226.1	907	472	251	73	36
Ca2-3,5,100	–	–	–	–	–	–	–
Cc4-3,5,100	–	–	–	–	–	–	–
Cd5-3,5,100	–	–	–	–	–	–	–
Ca3-6,4,100	255.0	-33.5	95	123	269	78	9
Ca4-6,4,100	988.0	79.7	428	121	195	45	22
Cc4-6,4,100	3,700.2	491.4	45	120	180	35	18
Cd5-6,4,100	1,555.0	84.5	531	118	223	47	11
Ca5-6,4,100	4,049.1	219.9	2,487	122	176	77	0
Cb5-6,4,100	2,268.0	65.7	1,061	123	121	81	0
Cb2-6,4,100	2,288.8	64.0	1,520	136	101	82	10
Ca1-6,4,100	3,447.1	42.7	919	139	328	91	13

Table A.5 Detailed results when omitting RCIs

Instance	T (s)	ΔT (%)	# Nodes	$\lvert\mathcal{M}\rvert$	# DDOIs	# SRIs	# VSIs
Cc4-2,3,50	–	–	–	–	–	–	–
Cc1-3,5,50	63.2	-3.5	14	402	367	19	48
Cd2-3,5,50	644.3	879.2	281	444	396	71	42
Cc2-3,5,50	85.2	2.3	17	456	390	23	38
Cd1-3,5,50	924.8	946.2	895	407	367	71	74
Cd3-3,5,50	1,194.5	1,140.4	1,013	383	365	107	62
Cd5-3,5,50	145.8	51.4	177	388	381	67	40
Cb3-3,5,50	117.5	12.3	97	397	372	79	51
Cc3-3,5,50	250.1	128.0	53	394	374	39	48
Cb5-3,5,50	124.2	-3.3	174	382	370	56	57
Ca4-3,5,50	311.0	49.9	247	382	323	56	55
Cb4-3,5,50	242.3	13.2	137	393	339	52	47
Cb2-3,5,50	283.6	-28.4	317	486	449	114	42
Cc4-3,5,50	740.5	84.2	77	360	308	57	46
Cb1-3,5,50	445.8	-3.4	419	413	386	155	43
Cc5-3,5,50	7,794.5	357.8	1,697	374	367	76	55
Cd5-6,4,50	116.5	33.8	269	105	83	53	51
Cb5-6,4,50	112.8	26.6	287	114	96	93	52
Cc2-6,4,50	595.9	56.0	167	100	80	124	38
Cc4-6,4,50	580.6	3.5	231	110	84	40	44
Ca2-2,3,100	1,322.4	167.7	2,555	40	54	153	10
Cc5-2,3,100	6,336.8	370.8	1,421	39	54	133	0
Ca1-2,3,100	5,655.7	144.8	9,941	39	54	157	11
Cc3-2,3,100	–	–	–	–	–	–	–
Cb2-3,5,100	484.3	-37.0	204	420	453	218	31
Ca1-3,5,100	5,058.6	252.5	3,201	472	490	87	56
Ca4-3,5,100	–	–	–	–	–	–	–
Cd4-3,5,100	4,864.6	100.4	1,679	490	483	109	35
Cd1-3,5,100	6,270.3	154.7	4,283	472	490	105	57
Ca2-3,5,100	5,489.1	86.8	2,185	413	443	277	41
Cc4-3,5,100	–	–	–	–	–	–	–
Cd5-3,5,100	–	–	–	–	–	–	–
Ca3-6,4,100	108.5	-71.7	107	123	147	175	10
Ca4-6,4,100	470.3	-14.5	346	121	114	116	21
Cc4-6,4,100	–	–	–	–	–	–	–
Cd5-6,4,100	5,827.8	591.6	6,496	118	143	154	20
Ca5-6,4,100	3,606.9	185.0	4,557	122	144	161	23
Cb5-6,4,100	2,117.9	54.7	1,783	123	144	112	0
Cb2-6,4,100	1,573.8	12.8	1,649	136	152	91	12
Ca1-6,4,100	2,141.1	-11.4	1,956	139	156	150	11

Table A.6 Detailed results when omitting SRIs

Instance	T (s)	ΔT (%)	# Nodes	$\lvert\mathcal{M}\rvert$	# DDOIs	# RCIs	# VSIs
Cc4-2,3,50	–	–	–	–	–	–	–
Cc1-3,5,50	65.7	0.3	21	402	367	36	43
Cd2-3,5,50	563.2	755.9	459	444	396	56	37
Cc2-3,5,50	82.0	-1.6	14	456	390	64	28
Cd1-3,5,50	618.0	599.1	489	407	367	150	58
Cd3-3,5,50	174.0	80.7	173	383	365	113	62
Cd5-3,5,50	167.9	74.4	257	388	381	76	52
Cb3-3,5,50	261.2	149.7	217	397	372	175	51
Cc3-3,5,50	115.5	5.3	20	394	374	54	45
Cb5-3,5,50	473.9	268.8	681	382	370	64	61
Ca4-3,5,50	2,660.0	1,181.9	3,101	382	323	64	67
Cb4-3,5,50	1,760.3	722.2	1,487	393	339	63	51
Cb2-3,5,50	795.5	100.8	819	486	449	208	39
Cc4-3,5,50	1,186.6	195.1	159	360	308	153	53
Cb1-3,5,50	861.1	86.6	907	413	386	167	52
Cc5-3,5,50	4,772.6	180.3	1,395	374	367	122	45
Cd5-6,4,50	292.8	236.2	763	105	83	96	48
Cb5-6,4,50	144.9	62.6	469	114	96	146	54
Cc2-6,4,50	1,606.4	320.5	597	100	80	176	47
Cc4-6,4,50	909.6	62.1	403	110	84	54	44
Ca2-2,3,100	–	–	–	–	–	–	–
Cc5-2,3,100	9,286.2	590.0	2,439	39	54	186	0
Ca1-2,3,100	–	–	–	–	–	–	–
Cc3-2,3,100	–	–	–	–	–	–	–
Cb2-3,5,100	2,097.9	172.8	1,128	420	453	137	35
Ca1-3,5,100	7,092.2	394.2	6,095	472	490	180	45
Ca4-3,5,100	–	–	–	–	–	–	–
Cd4-3,5,100	–	–	–	–	–	–	–
Cd1-3,5,100	9,978.0	305.3	3,871	472	490	325	45
Ca2-3,5,100	–	–	–	–	–	–	–
Cc4-3,5,100	–	–	–	–	–	–	–
Cd5-3,5,100	–	–	–	–	–	–	–
Ca3-6,4,100	4,066.3	959.8	3,389	123	147	407	11
Ca4-6,4,100	6,326.2	1,050.4	5,001	121	114	165	22
Cc4-6,4,100	–	–	–	–	–	–	–
Cd5-6,4,100	–	–	–	–	–	–	–
Ca5-6,4,100	–	–	–	–	–	–	–
Cb5-6,4,100	–	–	–	–	–	–	–
Cb2-6,4,100	–	–	–	–	–	–	–
Ca1-6,4,100	–	–	–	–	–	–	–

Table A.7 Detailed results when omitting VSIs

Instance	T (s)	ΔT (%)	# Nodes	$\|\mathcal{M}\|$	# DDOIs	# RCIs	# SRIs
Cc4-2,3,50	2,803.8	-27.5	2,991	27	30	118	55
Cc1-3,5,50	298.3	355.4	56	402	367	124	42
Cd2-3,5,50	112.9	71.6	93	444	396	140	45
Cc2-3,5,50	212.2	154.7	38	456	390	192	23
Cd1-3,5,50	137.1	55.1	71	407	367	191	95
Cd3-3,5,50	103.2	7.2	85	383	365	136	92
Cd5-3,5,50	158.4	64.5	197	388	381	198	66
Cb3-3,5,50	170.1	62.6	167	397	372	200	93
Cc3-3,5,50	273.5	149.3	50	394	374	126	36
Cb5-3,5,50	154.8	20.5	228	382	370	102	63
Ca4-3,5,50	212.0	2.2	269	382	323	113	50
Cb4-3,5,50	168.7	-21.2	129	393	339	82	53
Cb2-3,5,50	481.4	21.5	535	486	449	267	94
Cc4-3,5,50	378.2	-5.9	53	360	308	243	69
Cb1-3,5,50	825.9	79.0	525	413	386	377	261
Cc5-3,5,50	1,400.3	-17.8	301	374	367	309	30
Cd5-6,4,50	84.6	-2.9	189	105	83	107	39
Cb5-6,4,50	71.0	-20.3	181	114	96	133	52
Cc2-6,4,50	382.6	0.2	101	100	80	125	89
Cc4-6,4,50	196.3	-65.0	71	110	84	59	7
Ca2-2,3,100	410.2	-17.0	442	40	54	253	56
Cc5-2,3,100	552.5	-58.9	105	39	54	164	11
Ca1-2,3,100	3,777.4	63.5	3,748	39	54	306	83
Cc3-2,3,100	6,254.6	93.9	1,169	39	54	197	18
Cb2-3,5,100	1,138.1	48.0	338	420	453	403	200
Ca1-3,5,100	636.8	-55.6	325	472	490	245	34
Ca4-3,5,100	2,340.5	15.3	429	496	489	437	72
Cd4-3,5,100	1,994.6	-17.8	571	490	483	361	142
Cd1-3,5,100	2,242.2	-8.9	654	472	490	286	73
Ca2-3,5,100	2,948.2	0.3	1,155	413	443	662	236
Cc4-3,5,100	6,173.7	104.7	91	491	487	317	79
Cd5-3,5,100	2,676.6	-19.9	587	479	486	264	78
Ca3-6,4,100	272.4	-29.0	215	123	147	249	112
Ca4-6,4,100	486.7	-11.5	315	121	114	153	74
Cc4-6,4,100	532.2	-14.9	35	120	122	175	37
Cd5-6,4,100	551.2	-34.6	313	118	143	202	56
Ca5-6,4,100	1,031.1	-18.5	973	122	144	190	76
Cb5-6,4,100	1,463.9	6.9	1,173	123	144	110	82
Cb2-6,4,100	1,488.2	6.7	1,748	136	152	92	84
Ca1-6,4,100	1,688.1	-30.1	934	139	156	289	75

Table A.8 Detailed results when adding a second-copy variable for first-echelon routes visiting more than one satellite

Instance	T (s)	ΔT (%)	# Nodes	$\|\mathcal{M}\|$	# DDOIs	# RCIs	# SRIs	# VSIs
Cc4-2,3,50	6,254.6	61.7	5,355	42	72	83	43	33
Cc1-3,5,50	116.9	78.5	26	784	1,324	30	4	42
Cd2-3,5,50	206.5	213.8	51	868	1,440	58	30	36
Cc2-3,5,50	180.3	116.4	16	892	1,432	65	0	40
Cd1-3,5,50	153.5	73.6	63	794	1,324	72	37	59
Cd3-3,5,50	131.7	36.8	51	746	1,308	95	21	56
Cd5-3,5,50	161.9	68.1	127	756	1,364	76	37	48
Cb3-3,5,50	235.2	124.9	89	774	1,336	124	48	54
Cc3-3,5,50	163.9	49.4	18	768	1,336	73	11	51
Cb5-3,5,50	287.7	123.9	215	744	1,328	23	29	57
Ca4-3,5,50	575.9	177.5	266	744	1,156	76	24	54
Cb4-3,5,50	492.1	129.8	201	766	1,212	42	36	55
Cb2-3,5,50	997.3	151.8	517	952	1,636	95	61	41
Cc4-3,5,50	1,048.0	160.6	45	700	1,104	131	35	57
Cb1-3,5,50	751.9	62.9	417	806	1,392	109	166	44
Cc5-3,5,50	887.6	-47.9	121	728	1,308	151	8	49
Cd5-6,4,50	183.1	110.2	283	194	260	77	37	45
Cb5-6,4,50	150.6	69.0	269	212	304	104	38	49
Cc2-6,4,50	492.9	29.0	107	184	240	121	68	41
Cc4-6,4,50	314.3	-44.0	91	204	256	45	7	44
Ca2-2,3,100	1,136.6	130.1	907	56	120	266	63	0
Cc5-2,3,100	3,829.1	184.5	743	54	120	177	8	0
Ca1-2,3,100	6,029.5	161.0	4,125	54	120	384	78	18
Cc3-2,3,100	2,494.1	-22.7	395	54	120	153	14	11
Cb2-3,5,100	1,455.5	89.3	306	800	1,508	196	169	32
Ca1-3,5,100	2,454.1	71.0	931	904	1,640	189	28	55
Ca4-3,5,100	7,450.1	267.1	907	952	1,636	196	28	29
Cd4-3,5,100	8,498.9	250.1	1,323	940	1,612	136	64	40
Cd1-3,5,100	5,780.9	134.8	805	904	1,640	222	65	43
Ca2-3,5,100	6,154.4	109.5	1,193	786	1,468	521	166	42
Cc4-3,5,100	8,173.4	171.0	70	942	1,628	210	35	24
Cd5-3,5,100	7,197.3	115.4	901	918	1,624	189	45	20
Ca3-6,4,100	262.1	-31.7	111	214	396	303	72	9
Ca4-6,4,100	983.6	78.9	405	214	316	210	56	20
Cc4-6,4,100	1,432.8	129.0	63	208	328	200	36	18
Cd5-6,4,100	1,444.4	71.4	537	204	380	214	44	10
Ca5-6,4,100	3,519.1	178.1	2,447	212	384	227	67	0
Cb5-6,4,100	1,850.3	35.2	1,075	214	384	105	77	0
Cb2-6,4,100	1,776.3	27.3	1,310	240	416	103	68	12
Ca1-6,4,100	2,449.7	1.4	854	246	432	328	79	11

Table A.9 Detailed results when omitting branching level 1

Instance	T (s)	ΔT (%)	# Nodes	$\|\mathcal{M}\|$	# DDOIs	# RCIs	# SRIs	# VSIs
Cc4-2,3,50	4,151.4	7.3	4,035	27	30	95	38	31
Cc1-3,5,50	69.4	6.0	20	402	367	30	4	43
Cd2-3,5,50	76.2	15.8	55	444	396	74	26	36
Cc2-3,5,50	84.5	1.4	12	456	390	50	0	38
Cd1-3,5,50	83.0	-6.1	75	407	367	91	30	65
Cd3-3,5,50	169.2	75.7	181	383	365	74	12	60
Cd5-3,5,50	131.0	36.0	173	388	381	51	26	46
Cb3-3,5,50	83.3	-20.4	65	397	372	102	31	55
Cc3-3,5,50	145.1	32.3	51	394	374	54	5	55
Cb5-3,5,50	160.0	24.5	230	382	370	31	28	55
Ca4-3,5,50	338.0	62.9	351	382	323	96	28	59
Cb4-3,5,50	246.8	15.3	211	393	339	35	31	49
Cb2-3,5,50	390.5	-1.4	417	486	449	141	82	47
Cc4-3,5,50	384.5	-4.4	52	360	308	111	32	44
Cb1-3,5,50	449.4	-2.6	477	413	386	75	110	48
Cc5-3,5,50	1,940.4	14.0	437	374	367	113	8	44
Cd5-6,4,50	80.8	-7.2	243	105	83	59	33	47
Cb5-6,4,50	76.7	-13.9	239	114	96	80	33	47
Cc2-6,4,50	539.5	41.2	159	100	80	114	51	49
Cc4-6,4,50	951.0	69.5	407	110	84	42	6	44
Ca2-2,3,100	734.8	48.7	635	40	54	286	66	0
Cc5-2,3,100	1,095.9	-18.6	215	39	54	217	16	0
Ca1-2,3,100	5,049.1	118.6	5,315	39	54	229	52	11
Cc3-2,3,100	5,241.1	62.5	855	39	54	128	15	10
Cb2-3,5,100	970.5	26.2	406	420	453	152	126	34
Ca1-3,5,100	1,555.7	8.4	671	472	490	139	36	33
Ca4-3,5,100	4,692.1	131.2	697	496	489	176	26	25
Cd4-3,5,100	5,147.8	112.0	1,067	490	483	189	80	30
Cd1-3,5,100	3,641.3	47.9	795	472	490	196	54	27
Ca2-3,5,100	3,949.4	34.4	1,856	413	443	694	253	46
Cc4-3,5,100	8,523.3	182.6	190	491	487	202	35	22
Cd5-3,5,100	3,032.1	-9.3	559	479	486	176	37	22
Ca3-6,4,100	258.6	-32.6	147	123	147	292	80	12
Ca4-6,4,100	667.2	21.3	380	121	114	162	55	22
Cc4-6,4,100	341.0	-45.5	30	120	122	196	35	10
Cd5-6,4,100	1,024.5	21.6	629	118	143	199	39	11
Ca5-6,4,100	1,371.0	8.3	1,161	122	144	215	63	18
Cb5-6,4,100	1,921.9	40.4	1,296	123	144	83	69	0
Cb2-6,4,100	1,693.5	21.4	1,405	136	152	96	70	10
Ca1-6,4,100	2,970.0	22.9	1,399	139	156	282	74	15

Table A.10 Detailed results when omitting branching level 2

Instance	T (s)	ΔT (%)	# Nodes	$\|\mathcal{M}\|$	# DDOIs	# RCIs	# SRIs	# VSIs
Cc4-2,3,50	5,186.3	34.1	3,137	27	30	241	109	35
Cc1-3,5,50	2,580.2	3,839.2	216	402	367	361	174	68
Cd2-3,5,50	449.3	582.8	191	444	396	245	69	55
Cc2-3,5,50	269.7	223.8	34	456	390	226	7	42
Cd1-3,5,50	2,127.2	2,306.3	395	407	367	438	198	89
Cd3-3,5,50	640.0	564.6	179	383	365	185	97	74
Cd5-3,5,50	747.0	675.7	391	388	381	260	70	70
Cb3-3,5,50	543.4	419.5	273	397	372	158	32	67
Cc3-3,5,50	1,269.5	1,057.2	156	394	374	277	95	77
Cb5-3,5,50	2,322.3	1,707.2	1,948	382	370	275	243	99
Ca4-3,5,50	675.7	225.6	269	382	323	141	57	62
Cb4-3,5,50	870.5	306.6	239	393	339	89	49	67
Cb2-3,5,50	637.0	60.8	585	486	449	184	126	47
Cc4-3,5,50	713.7	77.5	65	360	308	106	34	59
Cb1-3,5,50	2,904.5	529.4	1,709	413	386	499	459	59
Cc5-3,5,50	4,212.7	147.4	425	374	367	286	17	59
Cd5-6,4,50	287.3	229.9	303	105	83	134	73	58
Cb5-6,4,50	371.8	317.3	505	114	96	496	159	53
Cc2-6,4,50	765.4	100.4	145	100	80	200	81	61
Cc4-6,4,50	1,006.9	79.4	189	110	84	72	19	43
Ca2-2,3,100	1,219.1	146.8	693	40	54	364	78	11
Cc5-2,3,100	8,112.4	502.7	689	39	54	1712	362	15
Ca1-2,3,100	4,026.8	74.3	1,685	39	54	519	122	13
Cc3-2,3,100	2,067.8	-35.9	193	39	54	315	79	11
Cb2-3,5,100	–	–	–	–	–	–	–	–
Ca1-3,5,100	9,836.4	585.4	1,250	472	490	1430	424	70
Ca4-3,5,100	5,215.6	157.0	601	496	489	214	29	27
Cd4-3,5,100	–	–	–	–	–	–	–	–
Cd1-3,5,100	–	–	–	–	–	–	–	–
Ca2-3,5,100	–	–	–	–	–	–	–	–
Cc4-3,5,100	–	–	–	–	–	–	–	–
Cd5-3,5,100	–	–	–	–	–	–	–	–
Ca3-6,4,100	1,008.9	162.9	249	123	147	595	194	16
Ca4-6,4,100	718.8	30.7	249	121	114	158	68	20
Cc4-6,4,100	–	–	–	–	–	–	–	–
Cd5-6,4,100	3,094.8	267.3	481	118	143	879	226	43
Ca5-6,4,100	–	–	–	–	–	–	–	–
Cb5-6,4,100	8,965.6	554.9	3,293	123	144	1039	428	46
Cb2-6,4,100	–	–	–	–	–	–	–	–
Ca1-6,4,100	–	–	–	–	–	–	–	–

Table A.11 Detailed results when omitting branching level 3

Instance	T (s)	ΔT (%)	# Nodes	$\|\mathcal{M}\|$	# DDOIs	# RCIs	# SRIs	# VSIs
Cc4-2,3,50	–	–	–	–	–	–	–	–
Cc1-3,5,50	51.1	-22.0	8	402	367	29	4	42
Cd2-3,5,50	125.2	90.3	81	444	396	49	28	36
Cc2-3,5,50	75.9	-8.9	11	456	390	63	0	40
Cd1-3,5,50	66.9	-24.3	57	407	367	87	25	61
Cd3-3,5,50	155.9	61.9	127	383	365	101	34	55
Cd5-3,5,50	308.1	219.9	447	388	381	155	44	44
Cb3-3,5,50	241.7	131.1	193	397	372	143	41	54
Cc3-3,5,50	131.5	19.9	17	394	374	57	16	45
Cb5-3,5,50	138.7	7.9	223	382	370	23	14	42
Ca4-3,5,50	689.0	232.0	937	382	323	60	24	46
Cb4-3,5,50	243.3	13.6	227	393	339	58	49	47
Cb2-3,5,50	1,674.1	322.6	2,229	486	449	94	64	40
Cc4-3,5,50	–	–	–	–	–	–	–	–
Cb1-3,5,50	683.2	48.0	825	413	386	35	63	45
Cc5-3,5,50	1,568.0	-7.9	645	374	367	97	10	41
Cd5-6,4,50	126.6	45.4	542	105	83	63	34	42
Cb5-6,4,50	352.9	296.1	1,611	114	96	57	15	49
Cc2-6,4,50	–	–	–	–	–	–	–	–
Cc4-6,4,50	2,587.5	361.1	1,485	110	84	42	6	47
Ca2-2,3,100	2,338.4	373.4	3,130	40	54	395	95	0
Cc5-2,3,100	–	–	–	–	–	–	–	–
Ca1-2,3,100	8,665.1	275.1	11,299	39	54	681	180	0
Cc3-2,3,100	2,541.4	-21.2	421	39	54	143	9	10
Cb2-3,5,100	4,564.2	493.5	2,524	420	453	95	93	31
Ca1-3,5,100	1,227.0	-14.5	739	472	490	204	29	53
Ca4-3,5,100	–	–	–	–	–	–	–	–
Cd4-3,5,100	–	–	–	–	–	–	–	–
Cd1-3,5,100	1,980.9	-19.5	836	472	490	200	66	47
Ca2-3,5,100	–	–	–	–	–	–	–	–
Cc4-3,5,100	–	–	–	–	–	–	–	–
Cd5-3,5,100	3,577.6	7.1	667	479	486	244	82	16
Ca3-6,4,100	285.7	-25.5	187	123	147	274	81	11
Ca4-6,4,100	1,254.7	128.2	1,233	121	114	162	35	42
Cc4-6,4,100	975.7	55.9	59	120	122	211	34	19
Cd5-6,4,100	3,488.5	314.0	2,599	118	143	197	38	6
Ca5-6,4,100	5,067.2	300.4	5,493	122	144	211	69	7
Cb5-6,4,100	1,557.7	13.8	1,815	123	144	102	70	3
Cb2-6,4,100	–	–	–	–	–	–	–	–
Ca1-6,4,100	–	–	–	–	–	–	–	–

Table A.12 Detailed results when omitting branching levels 2 and 3

Instance	T (s)	ΔT (%)	# Nodes	$\|\mathcal{M}\|$	# DDOIs	# RCIs	# SRIs	# VSIs
Cc4-2,3,50	–	–	–	–	–	–	–	–
Cc1-3,5,50	1,181.3	1,703.5	211	402	367	251	86	69
Cd2-3,5,50	369.7	461.9	197	444	396	191	66	43
Cc2-3,5,50	265.3	218.5	33	456	390	147	17	34
Cd1-3,5,50	675.6	664.3	249	407	367	222	126	72
Cd3-3,5,50	410.0	325.8	225	383	365	147	59	67
Cd5-3,5,50	469.3	387.3	421	388	381	159	58	73
Cb3-3,5,50	2,723.4	2,503.6	2,081	397	372	359	55	65
Cc3-3,5,50	855.4	679.8	153	394	374	142	83	70
Cb5-3,5,50	1,665.3	1,196.0	3,007	382	370	120	150	82
Ca4-3,5,50	675.1	225.3	541	382	323	116	48	49
Cb4-3,5,50	1,031.7	381.9	919	393	339	182	70	53
Cb2-3,5,50	4,011.5	912.7	5,047	486	449	138	79	47
Cc4-3,5,50	–	–	–	–	–	–	–	–
Cb1-3,5,50	2,841.8	515.8	3,969	413	386	101	145	46
Cc5-3,5,50	3,608.5	112.0	911	374	367	179	25	58
Cd5-6,4,50	203.5	133.6	505	105	83	84	58	50
Cb5-6,4,50	796.5	793.9	2,465	114	96	182	69	49
Cc2-6,4,50	–	–	–	–	–	–	–	–
Cc4-6,4,50	882.1	57.2	233	110	84	75	37	40
Ca2-2,3,100	2,041.6	313.3	1,535	40	54	458	91	11
Cc5-2,3,100	–	–	–	–	–	–	–	–
Ca1-2,3,100	–	–	–	–	–	–	–	–
Cc3-2,3,100	2,835.8	-12.1	353	39	54	339	69	11
Cb2-3,5,100	–	–	–	–	–	–	–	–
Ca1-3,5,100	6,386.9	345.0	1,319	472	490	1259	344	70
Ca4-3,5,100	–	–	–	–	–	–	–	–
Cd4-3,5,100	–	–	–	–	–	–	–	–
Cd1-3,5,100	–	–	–	–	–	–	–	–
Ca2-3,5,100	–	–	–	–	–	–	–	–
Cc4-3,5,100	–	–	–	–	–	–	–	–
Cd5-3,5,100	–	–	–	–	–	–	–	–
Ca3-6,4,100	–	–	–	–	–	–	–	–
Ca4-6,4,100	1,527.3	177.7	988	121	114	183	81	20
Cc4-6,4,100	–	–	–	–	–	–	–	–
Cd5-6,4,100	2,752.0	226.6	1,113	118	143	600	195	42
Ca5-6,4,100	7,502.1	492.8	4,285	122	144	1194	380	37
Cb5-6,4,100	–	–	–	–	–	–	–	–
Cb2-6,4,100	–	–	–	–	–	–	–	–
Ca1-6,4,100	–	–	–	–	–	–	–	–

Table A.13 Detailed results when omitting branching levels 1, 2 and 3

Instance	T (s)	ΔT (%)	# Nodes	$\|\mathcal{M}\|$	# DDOIs	# RCIs	# SRIs	# VSIs
Cc4-2,3,50	–	–	–	–	–	–	–	–
Cc1-3,5,50	1,048.0	1,500.0	445	402	367	140	79	98
Cd2-3,5,50	320.7	387.4	393	444	396	74	37	54
Cc2-3,5,50	525.1	530.4	283	456	390	120	13	40
Cd1-3,5,50	600.5	579.3	555	407	367	115	92	89
Cd3-3,5,50	670.7	596.5	767	383	365	108	34	86
Cd5-3,5,50	440.8	357.7	575	388	381	108	28	63
Cb3-3,5,50	2,579.3	2,365.9	2,967	397	372	95	8	65
Cc3-3,5,50	1,660.5	1,413.7	719	394	374	84	41	93
Cb5-3,5,50	1,915.4	1,390.6	3,327	382	370	66	73	86
Ca4-3,5,50	1,800.8	767.9	1,393	382	323	118	25	48
Cb4-3,5,50	3,248.2	1,417.1	3,081	393	339	37	18	51
Cb2-3,5,50	4,186.2	956.9	4,375	486	449	56	55	50
Cc4-3,5,50	–	–	–	–	–	–	–	–
Cb1-3,5,50	5,012.6	986.2	6,595	413	386	11	10	47
Cc5-3,5,50	2,360.7	38.7	597	374	367	166	18	71
Cd5-6,4,50	143.2	64.4	587	105	83	45	27	48
Cb5-6,4,50	1,014.3	1,038.4	3,747	114	96	38	11	47
Cc2-6,4,50	–	–	–	–	–	–	–	–
Cc4-6,4,50	1,073.0	91.2	493	110	84	58	14	45
Ca2-2,3,100	9,058.4	1,733.7	14,819	40	54	151	16	14
Cc5-2,3,100	–	–	–	–	–	–	–	–
Ca1-2,3,100	–	–	–	–	–	–	–	–
Cc3-2,3,100	6,146.3	90.6	1,149	39	54	121	13	11
Cb2-3,5,100	–	–	–	–	–	–	–	–
Ca1-3,5,100	6,937.8	383.4	5,892	472	490	50	13	62
Ca4-3,5,100	–	–	–	–	–	–	–	–
Cd4-3,5,100	–	–	–	–	–	–	–	–
Cd1-3,5,100	9,283.2	277.1	6,886	472	490	94	22	72
Ca2-3,5,100	–	–	–	–	–	–	–	–
Cc4-3,5,100	–	–	–	–	–	–	–	–
Cd5-3,5,100	–	–	–	–	–	–	–	–
Ca3-6,4,100	–	–	–	–	–	–	–	–
Ca4-6,4,100	4,752.7	764.3	4,115	121	114	82	26	36
Cc4-6,4,100	–	–	–	–	–	–	–	–
Cd5-6,4,100	–	–	–	–	–	–	–	–
Ca5-6,4,100	4,580.5	261.9	6,132	122	144	122	21	53
Cb5-6,4,100	–	–	–	–	–	–	–	–
Cb2-6,4,100	–	–	–	–	–	–	–	–
Ca1-6,4,100	–	–	–	–	–	–	–	–

Table A.14 Detailed results when prioritizing branching over levels 2 and 3 over branching level 1

Instance	T (s)	ΔT (%)	# Nodes	$\|\mathcal{M}\|$	# DDOIs	# RCIs	# SRIs	# VSIs
Cc4-2,3,50	3,010.7	-22.2	3,473	27	30	93	48	30
Cc1-3,5,50	60.0	-8.4	15	402	367	31	4	48
Cd2-3,5,50	76.3	16.0	55	444	396	52	26	36
Cc2-3,5,50	80.7	-3.1	11	456	390	51	0	33
Cd1-3,5,50	98.5	11.4	63	407	367	97	37	60
Cd3-3,5,50	85.4	-11.3	55	383	365	78	38	49
Cd5-3,5,50	128.1	33.0	159	388	381	42	24	41
Cb3-3,5,50	85.6	-18.2	67	397	372	87	30	54
Cc3-3,5,50	106.4	-3.0	17	394	374	53	16	45
Cb5-3,5,50	154.9	20.5	241	382	370	32	29	55
Ca4-3,5,50	227.3	9.5	243	382	323	98	20	59
Cb4-3,5,50	220.1	2.8	183	393	339	40	28	50
Cb2-3,5,50	280.9	-29.1	283	486	449	143	91	46
Cc4-3,5,50	278.6	-30.7	53	360	308	107	32	45
Cb1-3,5,50	384.2	-16.7	363	413	386	59	123	47
Cc5-3,5,50	1,776.8	4.4	377	374	367	105	8	45
Cd5-6,4,50	85.9	-1.4	273	105	83	59	33	50
Cb5-6,4,50	64.3	-27.8	221	114	96	70	34	50
Cc2-6,4,50	594.1	55.5	181	100	80	114	51	47
Cc4-6,4,50	305.6	-45.5	151	110	84	46	10	44
Ca2-2,3,100	890.6	80.3	839	40	54	277	66	0
Cc5-2,3,100	1,869.4	38.9	395	39	54	196	13	0
Ca1-2,3,100	2,973.3	28.7	2,915	39	54	233	61	8
Cc3-2,3,100	2,857.2	-11.4	565	39	54	125	10	12
Cb2-3,5,100	1,009.3	31.2	361	420	453	135	136	32
Ca1-3,5,100	2,621.6	82.7	955	472	490	223	44	29
Ca4-3,5,100	3,851.3	89.8	909	496	489	153	29	24
Cd4-3,5,100	4,563.8	88.0	983	490	483	169	82	24
Cd1-3,5,100	2,175.9	-11.6	631	472	490	206	50	28
Ca2-3,5,100	4,097.8	39.5	1,897	413	443	657	220	40
Cc4-3,5,100	3,614.2	19.9	113	491	487	190	35	25
Cd5-3,5,100	3,061.4	-8.4	505	479	486	192	40	22
Ca3-6,4,100	253.7	-33.9	147	123	147	319	74	16
Ca4-6,4,100	904.9	64.6	457	121	114	149	50	25
Cc4-6,4,100	411.4	-34.2	33	120	122	188	31	21
Cd5-6,4,100	1,131.3	34.3	683	118	143	190	50	11
Ca5-6,4,100	2,100.2	65.9	2,225	122	144	193	59	6
Cb5-6,4,100	1,610.0	17.6	1,359	123	144	69	69	16
Cb2-6,4,100	1,835.6	31.5	1,392	136	152	110	67	10
Ca1-6,4,100	2,506.6	3.7	1,175	139	156	376	82	16

A.3 Detailed results for managerial insights

In this section, we present the detailed computational results obtained for establishing managerial insights. Tables A.15–A.17 report the results gathered when considering only back-and-forth first-echelon routes for the instances with $|N^C| = 30$, 50 and 100, respectively. For each table, the first column indicates the instance name. The second and third columns report, respectively, the optimal solution value (z^*) when known and the relative difference between this value and the optimal value (z^*_{all}) obtained when considering all first-echelon routes ($\Delta z^* = (z^* - z^*_{all})/z^*_{all}$). Then, we report the computational time (T) in seconds, the relative difference between this time and the time (T_{all}) required when considering all first-echelon routes ($\Delta T = (T - T_{all})/T_{all}$), the integrality gap ($Gap$) computed as $(z^* - \underline{z})/\underline{z}$, where $\underline{z}$ is the lower bound achieved at the root node before adding cuts, the number of nodes explored in the search tree (# *Nodes*), and the numbers of DDOIs (4.3.13), RCIs (4.3.28), SRIs (4.3.30), and VSIs (4.3.31) generated. For these tables, we do not report the number of DDOIs, as there are none when using only first-echelon routes visiting exactly one satellite. A "–" indicates that no optimal solution was found within the 3-hour time limit.

Table A.15 – Detailed results when considering only back-and-forth first-echelon routes for $|N^C| = 30$

Instance	z^*	Δz^* (%)	T (s)	Δ T (%)	Gap (%)	# Nodes	$\|\mathcal{M}\|$	# RCIs	# SRIs	# VSIs
Ca1-2,3,30	981.37	0.0	0.10	0.0	0.5	3	9	4	0	9
Ca2-2,3,30	968.32	0.0	0.20	100.0	0.4	9	9	27	0	29
Ca3-2,3,30	1,069.81	0.0	0.10	0.0	0.4	1	9	31	0	7
Ca4-2,3,30	950.39	0.2	0.20	-66.7	0.9	3	9	40	14	21
Ca5-2,3,30	968.23	0.0	1.10	-26.7	1.2	17	9	87	19	18
Cb1-2,3,30	985.76	0.0	0.30	-25.0	1.0	6	9	28	9	10
Cb2-2,3,30	937.12	0.0	0.10	-50.0	0.6	3	9	15	9	16
Cb3-2,3,30	1,076.25	0.0	0.10	0.0	0.1	2	9	10	9	13
Cb4-2,3,30	991.04	1.0	0.50	66.7	0.9	19	9	31	4	19
Cb5-2,3,30	834.85	0.0	0.20	-33.3	0.9	4	6	8	19	29
Cc1-2,3,30	799.56	0.0	1.50	-6.3	0.3	2	6	4	0	10
Cc2-2,3,30	937.71	0.0	2.00	-33.3	0.1	8	9	11	0	22
Cc3-2,3,30	1,013.55	0.0	12.40	-14.5	0.7	41	9	56	13	19
Cc4-2,3,30	957.29	1.5	1.60	-56.8	0.3	5	9	22	3	23
Cc5-2,3,30	949.39	0.0	91.40	-4.9	1.7	247	9	96	79	3
Cd1-2,3,30	807.25	0.0	0.20	0.0	0.0	5	6	0	0	24
Cd2-2,3,30	945.30	0.0	0.30	-25.0	0.1	10	9	3	0	31
Cd3-2,3,30	1,030.61	0.0	0.20	-33.3	0.7	9	9	11	3	16
Cd4-2,3,30	972.56	1.4	0.30	-25.0	0.7	9	9	27	6	23
Cd5-2,3,30	965.89	0.0	3.40	-53.4	1.8	81	9	61	34	17
Ca1-3,5,30	892.75	0.0	0.60	-94.2	0.6	12	15	24	17	38
Ca2-3,5,30	962.18	0.2	0.50	-88.4	0.3	9	15	19	5	41
Ca3-3,5,30	924.51	0.0	0.30	-93.6	0.3	6	15	3	0	42

Continued on next page

Table A.15 – *Continued from previous page*

Instance	z^*	Δz^* (%)	T (s)	Δ T (%)	Gap (%)	# Nodes	$\lvert\mathcal{M}\rvert$	# RCIs	# SRIs	# VSIs
Ca4-3,5,30	912.07	0.2	0.60	-85.0	0.1	10	15	21	0	49
Ca5-3,5,30	894.65	0.2	0.30	-94.5	0.1	11	15	5	0	35
Cb1-3,5,30	944.83	1.8	1.50	-46.4	1.6	35	15	13	28	53
Cb2-3,5,30	961.66	0.4	1.30	-89.8	1.6	25	15	44	12	53
Cb3-3,5,30	900.90	0.0	0.30	-87.0	0.4	8	15	17	0	29
Cb4-3,5,30	908.12	0.5	0.90	-86.4	1.2	12	15	9	12	46
Cb5-3,5,30	977.91	0.0	3.40	-86.7	1.2	61	15	53	20	51
Cc1-3,5,30	947.60	0.4	4.50	-83.2	1.2	15	15	18	6	38
Cc2-3,5,30	907.61	1.1	21.30	19.0	0.9	35	15	100	12	54
Cc3-3,5,30	943.50	0.0	3.20	-66.0	1.3	7	15	41	8	36
Cc4-3,5,30	895.49	0.5	14.40	-61.9	1.4	14	15	48	21	48
Cc5-3,5,30	778.16	0.8	7.10	-78.9	0.5	5	10	62	4	59
Cd1-3,5,30	967.91	1.3	0.60	-88.7	1.2	10	15	51	5	36
Cd2-3,5,30	945.64	0.1	0.30	-88.9	0.4	6	15	27	3	47
Cd3-3,5,30	913.35	0.6	0.50	-85.7	1.0	11	15	20	0	46
Cd4-3,5,30	922.59	0.6	0.60	-85.4	1.2	10	15	46	6	45
Cd5-3,5,30	917.01	0.0	0.60	-95.0	1.1	8	15	58	1	31
Ca1-6,4,30	922.22	0.0	1.60	-72.9	1.1	33	12	55	7	36
Ca2-6,4,30	966.86	1.1	0.10	-97.0	0.1	3	12	12	8	25
Ca3-6,4,30	869.33	0.0	0.20	-80.0	0.3	4	12	22	10	27
Ca4-6,4,30	869.07	0.2	0.40	-82.6	0.9	9	12	25	4	31
Ca5-6,4,30	851.52	0.0	0.20	-77.8	0.2	5	12	6	0	32
Cb1-6,4,30	968.05	0.0	1.50	-70.0	1.0	47	12	25	7	44
Cb2-6,4,30	920.50	0.0	0.60	-68.4	0.5	19	12	20	4	41

Continued on next page

Table A.15 – *Continued from previous page*

Instance	z^*	Δz^* (%)	T (s)	Δ T (%)	Gap (%)	# Nodes	$\lvert\mathcal{M}\rvert$	# RCIs	# SRIs	# VSIs
`Cb3-6,4,30`	929.48	0.0	0.30	-75.0	0.5	7	12	17	0	26
`Cb4-6,4,30`	877.54	0.0	0.90	-70.0	1.1	17	12	20	15	34
`Cb5-6,4,30`	895.74	0.0	0.50	-75.0	1.5	7	12	29	8	38
`Cc1-6,4,30`	861.13	0.0	4.40	-54.2	0.6	9	12	26	12	44
`Cc2-6,4,30`	920.91	0.9	0.90	-85.5	0.0	5	12	0	0	27
`Cc3-6,4,30`	900.76	0.1	5.90	-67.9	0.9	10	12	41	16	31
`Cc4-6,4,30`	833.64	0.0	6.00	-38.1	0.7	14	12	40	3	33
`Cc5-6,4,30`	818.31	0.5	4.20	-53.8	0.8	16	12	20	6	35
`Cd1-6,4,30`	918.39	0.0	1.20	-78.9	0.8	35	12	44	16	42
`Cd2-6,4,30`	915.31	0.1	0.30	-78.6	0.1	9	12	4	9	21
`Cd3-6,4,30`	880.59	0.0	0.20	-86.7	0.1	6	12	5	0	34
`Cd4-6,4,30`	885.41	0.0	1.10	-72.5	1.3	21	12	35	16	38
`Cd5-6,4,30`	943.68	0.0	3.10	-61.3	1.1	79	12	51	14	39

Table A.16 – Detailed results when considering only back-and-forth first-echelon routes for $|N^C| = 50$

Instance	z^*	Δz^* (%)	T (s)	Δ T (%)	Gap (%)	# Nodes	$\lvert\mathcal{M}\rvert$	# RCIs	# SRIs	# VSIs
Ca1-2,3,50	1,501.80	0.0	0.50	25.0	0.0	9	12	0	0	12
Ca2-2,3,50	1,586.21	0.0	9.30	-41.1	0.5	99	12	72	8	22
Ca3-2,3,50	1,479.56	0.0	1.10	10.0	0.5	18	12	25	5	25
Ca4-2,3,50	1,485.70	0.0	1.60	-36.0	0.7	19	12	38	14	16
Ca5-2,3,50	1,397.05	0.0	2.80	-31.7	1.2	29	12	48	12	23
Cb1-2,3,50	1,455.11	0.0	2.90	-6.5	0.8	29	12	26	12	20
Cb2-2,3,50	1,468.47	0.0	33.20	-12.9	1.3	291	12	60	43	30
Cb3-2,3,50	1,492.25	0.0	39.60	-17.3	1.4	321	12	80	28	27
Cb4-2,3,50	1,469.01	0.0	7.40	-39.8	1.1	63	12	34	27	17
Cb5-2,3,50	1,415.23	0.0	16.30	-25.9	1.1	125	12	61	31	28
Cc1-2,3,50	1,407.68	0.0	8.50	-22.7	0.3	8	12	51	6	6
Cc2-2,3,50	1,468.88	0.0	11.50	-13.5	0.4	12	12	65	10	18
Cc3-2,3,50	1,376.55	0.0	38.00	-39.5	1.4	26	12	113	17	22
Cc4-2,3,50	1,425.01	0.0	5,167.20	33.6	1.9	5,115	12	80	38	32
Cc5-2,3,50	1,442.65	0.0	35.90	-9.1	1.1	25	12	89	24	21
Cd1-2,3,50	1,458.04	0.0	1.30	-7.1	0.3	8	12	49	8	23
Cd2-2,3,50	1,568.68	0.0	4.20	40.0	0.6	23	12	86	14	23
Cd3-2,3,50	1,449.70	0.0	1.60	-15.8	0.9	11	12	53	6	20
Cd4-2,3,50	1,453.24	0.0	7.30	-34.2	0.8	45	12	39	19	28
Cd5-2,3,50	1,462.30	0.0	6.70	42.6	1.3	45	12	64	13	22
Ca1-3,5,50	1,351.83	0.0	2.90	-83.1	0.3	15	20	23	7	62
Ca2-3,5,50	1,378.60	1.8	2.90	-84.7	0.8	14	20	84	5	55
Ca3-3,5,50	1,277.35	0.4	6.70	-78.7	0.6	17	20	96	39	54

Continued on next page

Table A.16 – *Continued from previous page*

Instance	z^*	Δz^* (%)	T (s)	Δ T (%)	Gap (%)	# Nodes	$\lvert\mathcal{M}\rvert$	# RCIs	# SRIs	# VSIs
Ca4-3,5,50	1,299.86	0.0	75.00	-63.9	1.1	275	20	98	37	59
Ca5-3,5,50	1,274.81	0.6	3.90	-55.2	1.1	13	20	36	16	39
Cb1-3,5,50	1,355.50	0.0	38.70	-91.6	1.2	171	20	20	16	39
Cb2-3,5,50	1,414.08	1.6	22.60	-94.3	1.8	143	20	59	33	48
Cb3-3,5,50	1,368.97	0.0	28.10	-73.1	1.0	101	20	63	5	51
Cb4-3,5,50	1,298.33	0.0	58.20	-72.8	1.2	237	20	27	23	62
Cb5-3,5,50	1,273.00	0.2	36.90	-71.3	1.0	129	20	41	25	70
Cc1-3,5,50	1,296.36	0.2	32.00	-51.1	0.3	16	20	27	14	67
Cc2-3,5,50	1,336.05	1.4	32.20	-61.3	1.0	13	20	49	10	40
Cc3-3,5,50	1,291.82	0.2	38.20	-65.2	0.3	11	20	62	19	55
Cc4-3,5,50	1,348.83	0.0	73.10	-81.8	2.2	27	20	113	25	55
Cc5-3,5,50	1,281.55	0.4	537.20	-68.4	0.6	229	20	120	10	65
Cd1-3,5,50	1,314.59	0.0	28.60	-67.6	1.3	73	20	101	34	56
Cd2-3,5,50	1,347.04	1.6	34.60	-47.4	1.6	149	20	73	47	62
Cd3-3,5,50	1,357.85	0.5	31.60	-67.2	1.0	85	20	85	32	70
Cd4-3,5,50	1,315.54	0.0	3.50	-88.3	0.4	22	20	25	0	54
Cd5-3,5,50	1,290.50	0.1	12.90	-86.6	0.8	61	20	43	9	59
Ca1-6,4,50	1,335.55	0.0	11.10	-50.4	1.0	65	16	103	21	39
Ca2-6,4,50	1,376.13	0.0	5.40	-41.3	2.1	17	16	99	35	30
Ca3-6,4,50	1,337.15	0.0	13.70	-71.0	1.3	65	16	91	6	43
Ca4-6,4,50	1,242.56	0.0	2.00	-60.0	0.4	24	16	45	2	41
Ca5-6,4,50	1,268.84	0.0	3.30	-48.4	0.3	28	16	14	20	45
Cb1-6,4,50	1,355.92	0.0	4.80	-34.2	1.0	45	16	34	19	36
Cb2-6,4,50	1,456.85	0.0	20.30	-35.6	2.3	60	16	116	44	52

Continued on next page

Table A.16 – *Continued from previous page*

Instance	z^*	Δz^* (%)	T (s)	Δ T (%)	Gap (%)	# Nodes	$\lvert\mathcal{M}\rvert$	# RCIs	# SRIs	# VSIs
`Cb3-6,4,50`	1,345.02	0.0	32.90	-24.4	1.0	191	16	46	17	50
`Cb4-6,4,50`	1,281.66	0.0	3.80	-46.5	0.9	35	16	8	21	40
`Cb5-6,4,50`	1,341.25	0.0	50.70	-43.1	1.1	221	16	92	34	56
`Cc1-6,4,50`	1,262.60	0.0	16.70	-30.7	0.7	12	16	45	10	36
`Cc2-6,4,50`	1,304.68	0.0	249.50	-34.7	1.5	131	16	123	59	38
`Cc3-6,4,50`	1,315.85	0.0	24.80	-25.5	0.8	16	16	51	10	41
`Cc4-6,4,50`	1,206.64	0.0	299.10	-46.7	0.5	215	16	57	6	53
`Cc5-6,4,50`	1,453.39	0.0	17.00	-29.2	0.9	13	20	37	0	30
`Cd1-6,4,50`	1,300.53	0.0	5.70	-50.4	0.4	29	16	65	26	56
`Cd2-6,4,50`	1,338.28	0.0	2.60	-55.9	1.2	12	16	68	26	35
`Cd3-6,4,50`	1,375.06	0.0	15.60	-22.0	1.1	87	16	35	13	46
`Cd4-6,4,50`	1,236.79	0.0	6.80	-47.7	0.6	43	16	39	10	51
`Cd5-6,4,50`	1,332.86	0.0	45.90	-47.3	1.2	235	16	59	31	49

Table A.17 – Detailed results when considering only back-and-forth first-echelon routes for $|N^C| = 100$

Instance	z^*	Δz^* (%)	T (s)	Δ T (%)	Gap (%)	# Nodes	$\|\mathcal{M}\|$	# RCIs	# SRIs	# VSIs
Ca1-2,3,100	2,777.87	0.0	2,313.30	0.1	0.8	2,342	24	252	67	16
Ca2-2,3,100	2,863.08	0.0	425.90	-13.8	0.9	496	24	294	60	0
Ca3-2,3,100	2,677.44	0.0	83.30	-48.1	0.4	101	24	103	22	13
Ca4-2,3,100	3,037.91	0.0	11.10	-48.1	0.8	10	24	126	21	0
Ca5-2,3,100	2,757.45	0.0	20.60	-3.3	0.5	23	24	122	18	0
Cb1-2,3,100	2,490.59	0.0	2,804.50	-30.8	1.1	2,801	21	103	120	12
Cb2-2,3,100	2,737.56	0.0	2,699.20	-56.3	1.2	3,897	24	96	113	12
Cb3-2,3,100	2,716.63	0.0	1,650.20	-66.0	0.8	2,698	24	95	59	12
Cb4-2,3,100	–	–	–	–	–	–	–	–	–	–
Cb5-2,3,100	2,798.85	0.0	4,930.10	-33.2	1.0	6,849	24	98	134	10
Cc1-2,3,100	2,770.56	0.0	77.60	7.3	0.4	13	24	126	11	0
Cc2-2,3,100	2,661.94	0.0	140.80	-23.6	0.8	14	24	209	25	0
Cc3-2,3,100	2,665.61	0.0	2,782.10	-13.7	0.5	533	24	140	8	11
Cc4-2,3,100	–	–	–	–	–	–	–	–	–	–
Cc5-2,3,100	2,751.94	0.0	1,698.50	26.2	0.5	503	24	157	12	0
Cd1-2,3,100	2,666.54	0.0	40.80	-0.5	0.3	37	24	92	22	6
Cd2-2,3,100	2,852.93	0.0	49.20	-46.8	0.6	43	24	134	24	0
Cd3-2,3,100	2,633.61	0.0	3,443.00	-55.0	0.9	6,181	24	171	55	17
Cd4-2,3,100	2,904.74	–	6,665.20	–	1.1	10,224	24	293	61	8
Cd5-2,3,100	2,794.66	0.0	214.90	-30.7	0.5	205	24	174	16	0
Ca1-3,5,100	2,608.81	0.0	943.00	-34.3	0.6	1,131	40	191	29	55
Ca2-3,5,100	2,483.29	0.0	603.40	-79.5	1.5	637	40	263	119	41
Ca3-3,5,100	2,452.23	0.0	63.50	-69.3	0.5	87	40	125	26	32

Continued on next page

Table A.17 – *Continued from previous page*

Instance	z^*	Δz^* (%)	T (s)	Δ T (%)	Gap (%)	# Nodes	$\lvert\mathcal{M}\rvert$	# RCIs	# SRIs	# VSIs
`Ca4-3,5,100`	2,648.21	0.0	848.10	-58.2	0.7	853	40	147	28	31
`Ca5-3,5,100`	2,621.71	–	3,738.80	–	0.9	2,307	40	185	64	34
`Cb1-3,5,100`	2,646.86	–	9,915.60	–	1.2	12,342	40	167	110	44
`Cb2-3,5,100`	2,478.02	0.0	161.60	-79.0	1.0	140	40	76	76	32
`Cb3-3,5,100`	2,547.65	–	8,424.90	–	1.1	9,335	40	156	95	38
`Cb4-3,5,100`	2,723.97	–	7,006.70	–	1.2	6,328	40	255	209	56
`Cb5-3,5,100`	2,760.24	–	4,002.80	–	1.1	3,551	40	160	70	29
`Cc1-3,5,100`	2,563.82	–	6,371.70	–	0.9	769	40	376	31	39
`Cc2-3,5,100`	2,435.40	–	1,478.70	–	1.3	145	40	312	59	34
`Cc3-3,5,100`	–	–	–	–	–	–	–	–	–	–
`Cc4-3,5,100`	2,571.43	0.0	604.20	-80.0	0.6	70	40	175	43	24
`Cc5-3,5,100`	2,638.09	0.0	1,673.80	-83.7	0.5	179	40	143	27	60
`Cd1-3,5,100`	2,530.20	0.0	1,049.00	-57.4	0.8	614	40	247	63	38
`Cd2-3,5,100`	2,440.95	–	1,133.40	–	1.1	1,445	40	113	39	21
`Cd3-3,5,100`	2,523.66	0.0	23.30	-42.5	0.4	21	40	98	26	35
`Cd4-3,5,100`	2,537.09	0.0	1,233.10	-49.2	1.0	959	40	138	81	40
`Cd5-3,5,100`	2,666.30	0.0	1,017.40	-69.6	0.7	491	40	174	32	14
`Ca1-6,4,100`	2,537.92	0.0	1,156.70	-52.1	1.1	859	32	281	65	13
`Ca2-6,4,100`	2,380.65	0.0	28.80	-56.0	0.5	14	28	152	33	12
`Ca3-6,4,100`	2,696.39	0.0	192.60	-49.8	1.0	183	32	257	77	9
`Ca4-6,4,100`	2,375.64	0.0	374.40	-31.9	0.5	344	28	160	47	23
`Ca5-6,4,100`	2,508.41	0.0	1,022.70	-19.2	0.9	1,301	32	224	67	0
`Cb1-6,4,100`	2,375.52	0.0	3,313.40	-51.3	1.0	2,816	28	124	99	13
`Cb2-6,4,100`	2,566.25	0.0	1,100.40	-21.1	0.9	1,547	32	108	72	11

Continued on next page

Table A.17 – *Continued from previous page*

Instance	z^*	Δz^* (%)	T (s)	Δ T (%)	Gap (%)	# Nodes	$\lvert\mathcal{M}\rvert$	# RCIs	# SRIs	# VSIs
`Cb3-6,4,100`	–	–	–	–	–	–	–	–	–	–
`Cb4-6,4,100`	2,743.53	0.0	3,747.90	-49.5	0.7	5,707	32	125	66	15
`Cb5-6,4,100`	2,569.62	0.0	941.50	-31.2	1.0	1,181	32	95	65	0
`Cc1-6,4,100`	–	–	–	–	–	–	–	–	–	–
`Cc2-6,4,100`	–	–	–	–	–	–	–	–	–	–
`Cc3-6,4,100`	–	–	–	–	–	–	–	–	–	–
`Cc4-6,4,100`	2,559.58	0.0	449.40	-28.2	0.7	41	32	205	40	19
`Cc5-6,4,100`	–	–	–	–	–	–	–	–	–	–
`Cd1-6,4,100`	2,591.01	0.0	3,474.90	-62.6	0.9	5,795	32	227	86	11
`Cd2-6,4,100`	2,535.57	0.0	57.30	-23.6	0.3	53	32	92	23	10
`Cd3-6,4,100`	2,409.16	0.0	76.30	-47.6	0.4	61	28	90	30	14
`Cd4-6,4,100`	2,677.97	0.0	134.10	-19.5	0.6	164	32	122	62	11
`Cd5-6,4,100`	2,575.22	0.0	390.60	-53.6	0.9	371	32	178	44	11

ANNEXE B SUPPLEMENT FOR THE MC-2E-VRPTW

B.1 Proof of Proposition 5.1

In this proof, we use the symbol $\oplus$ to denote a concatenation of two paths. More precisely, let l be a path and ω a possible path extension of l. Then, $l \oplus \omega$ represents the path resulting from the concatenation of l and ω.

Proof B.1. Let ω be a feasible extension of l' and $\mathcal{C}_\omega$ the subset of customers visited along this extension. To prove the proposition's statement, it suffices to show that i) any feasible extension ω of path l' such that $d_i \in D_s^{T_{lt}^{start}}$ for all $i \in \mathcal{C}_\omega$ (otherwise, condition (5.3.11) is violated and label E_{lt} does not dominate label $E_{l't'}$) is also feasible for path l and ii) the reduced cost of path $l \oplus \omega$ is less than or equal to that of path $l' \oplus \omega$. Given that time and load REFs (5.3.4)–(5.3.5) are non-decreasing, establishing the feasibility of path $l \oplus \omega$ with respect to the time window and capacity constraints is straightforward. Moreover, elementarity of path $l' \oplus \omega$ implies that no customers in $\mathcal{C}_\omega$ is in U_{lt} and thus $l \oplus \omega$ is elementary as well.

Let us now show that $T^{rdc}_{l\oplus\omega,t} \leq T^{rdc}_{l'\oplus\omega,t'}$. For convenience, let $c(\omega)$ denote the total routing cost of ω and $\sigma(\omega) = \sum_{j\in\mathcal{C}_\omega} \sigma_j$. With this notation, we get:

$$T^{rdc}_{l\oplus\omega,t} = T^{rdc}_{lt} + c(\omega) - \sigma(\omega) - \sum_{j\in\mathcal{C}_\omega} q_j \cdot \pi_{m(d_j,T_{lt}^{start})}$$
$$T^{rdc}_{l'\oplus\omega,t'} = T^{rdc}_{l't'} + c(\omega) - \sigma(\omega) - \sum_{j\in\mathcal{C}_\omega} q_j \cdot \pi_{m(d_j,T_{l't'}^{start})}$$

and the difference between these two reduced costs is:

$$T^{rdc}_{l\oplus\omega,t} - T^{rdc}_{l'\oplus\omega,t'} = T^{rdc}_{lt} - T^{rdc}_{l't'} - \sum_{j\in\mathcal{C}_\omega} q_j \cdot (\pi_{m(d_j,T_{lt}^{start})} - \pi_{m(d_j,T_{l't'}^{start})}). \tag{B.1.1}$$

Let us consider two cases. First, if $T_{lt}^{start} \geq T_{l't'}^{start}$, then $\mathcal{M}_s^{T_{lt}^{start}} \supseteq \mathcal{M}_s^{T_{l't'}^{start}}$ and

$$\pi_{m(d_j,T_{lt}^{start})} = \max_{m\in\mathcal{M}_{d_j}\cap\mathcal{M}_s^{T_{lt}^{start}}}\{\pi_m\} \geq \max_{m\in\mathcal{M}_{d_j}\cap\mathcal{M}_s^{T_{l't'}^{start}}}\{\pi_m\} = \pi_{m(d_j,T_{l't'}^{start})} \quad \forall j \in \mathcal{C}_\omega.$$

Establishing that $T^{rdc}_{l\oplus\omega,t} \leq T^{rdc}_{l'\oplus\omega,t'}$ in this case is deduced from (B.1.1) and the fact that condition (5.3.8) implies $T^{rdc}_{lt} \leq T^{rdc}_{l't'}$.

Second, if $T_{lt}^{start} < T_{l't'}^{start}$, the following ensues:

$$D_s^{T_{lt}^{start}} \subseteq D_s^{T_{l't'}^{start}} \implies \Delta(T_{lt}^{start}, T_{l't'}^{start}) = \min_{d \in D_s^{T_{lt}^{start}}} \pi_{m(d,T_{lt}^{start})} - \pi_{m(d,T_{l't'}^{start})} \leq 0.$$

and we can deduce that

$$\begin{aligned}\sum_{j \in \mathcal{C}_\omega} q_j \cdot (\pi_{m(d_j,T_{lt}^{start})} - \pi_{m(d_j,T_{l't'}^{start})}) &\geq \sum_{j \in \mathcal{C}_\omega} q_j \cdot \min_{d \in D_s^{T_{lt}^{start}}} (\pi_{m(d,T_{lt}^{start})} - \pi_{m(d,T_{l't'}^{start})}) \\ &\geq (Q_2 - T_{l't'}^{load}) \cdot \Delta(T_{lt}^{start}, T_{l't'}^{start})\end{aligned}$$

because $\sum_{j \in \mathcal{C}_\omega} q_j \leq Q_2 - T_{l't'}^{load}$ for a feasible extension ω. Consequently, relation (B.1.1) and condition (5.3.8) yield $T_{l \oplus \omega,t}^{rdc} - T_{l' \oplus \omega,t'}^{rdc} \leq T_{lt}^{rdc} - T_{l't'}^{rdc} - (Q_2 - T_{l't'}^{load}) \cdot \Delta(T_{lt}^{start}, T_{l't'}^{start}) \leq 0$, i.e., $T_{l \oplus \omega,t}^{rdc} \leq T_{l' \oplus \omega,t'}^{rdc}$. $\square$

B.2 Detailed example of solution feasibility with transfer inequalities

Figure B.1 illustrates an MC-2E-VRPTW example with $|N^D| = 2$, $|N^S| = 3$, $|N^C| = 20$, $Q_1 = 100$, and $Q_2 = 40$, where it is not possible to convert a solution into a same-cost feasible one by transferring demands between first-echelon routes. The green and grey customers are assigned to depots d_1 and d_2, respectively, i.e., $N_{d_2}^C = \{2, 4, 6, 7, 9, 13, 19\}$ and $N_{d_1}^C = N^C \setminus N_{d_2}^C$. The solution uses three first-echelon routes m_2, m_3 and m_4 ($(\tilde{x}_2, \tilde{x}_3, \tilde{x}_4) = (1, 1, 1)$) and eight second-echelon routes l_i, $i \in \{1, \ldots, 8\}$. Furthermore, active transfer variables associated with route pairs (m_1, m_2) and (m_1, m_3) allow for an implicit use of first-echelon route m_1, i.e., $\tilde{u}_{m_1 m_2} = \tilde{u}_{m_1 m_3} = 20$ and $\tilde{x}_1 = 0$. The supply patterns $\tilde{p}_{l_i} \in \mathcal{P}_{l_i}$ retained for the second-echelon routes l_i, $i \in \{1, \cdots, 8\}$, along with the total loads associated with their supplying first-echelon routes are:

$$\begin{aligned}
\tilde{p}_{l_1} &= \{(m_1, 1), (m_4, 2)\}, & (Q_{l_1 \tilde{p}_{l_1}}^{m_1}, Q_{l_1 \tilde{p}_{l_1}}^{m_4}) &= (30, 10) \\
\tilde{p}_{l_2} &= \{(m_1, 3), (m_4, 4)\} & (Q_{l_2 \tilde{p}_{l_2}}^{m_1}, Q_{l_2 \tilde{p}_{l_2}}^{m_4}) &= (10, 20) \\
\tilde{p}_{l_3} &= \{(m_2, 5), (m_4, 6), (m_4, 7)\} & (Q_{l_3 \tilde{p}_{l_3}}^{m_2}, Q_{l_3 \tilde{p}_{l_3}}^{m_4}) &= (20, 20) \\
\tilde{p}_{l_4} &= \{(m_2, 8), (m_4, 9)\} & (Q_{l_4 \tilde{p}_{l_4}}^{m_2}, Q_{l_4 \tilde{p}_{l_4}}^{m_4}) &= (20, 20) \\
\tilde{p}_{l_5} &= \{(m_2, 10), (m_2, 11), (m_2, 12)\} & Q_{l_5 \tilde{p}_{l_5}}^{m_2} &= 40 \\
\tilde{p}_{l_6} &= \{(m_4, 13), (m_3, 14), (m_3, 15)\} & (Q_{l_6 \tilde{p}_{l_6}}^{m_3}, Q_{l_6 \tilde{p}_{l_6}}^{m_4}) &= (20, 10) \\
\tilde{p}_{l_7} &= \{(m_3, 16), (m_3, 17)\} & Q_{l_7 \tilde{p}_{l_7}}^{m_3} &= 30 \\
\tilde{p}_{l_8} &= \{(m_3, 18), (m_4, 19), (m_3, 20)\} & (Q_{l_8 \tilde{p}_{l_8}}^{m_3}, Q_{l_8 \tilde{p}_{l_8}}^{m_4}) &= (30, 10).
\end{aligned}$$

According to these supply patterns, first-echelon routes m_1, m_2, m_3, and m_4 have total loads of 40, 80, 80, and 90, respectively. The transfer variable values $\tilde{u}_{m_1 m_2} = \tilde{u}_{m_1 m_3}$ indicate that 20 units of load are transferred from m_1 to m_2 and from m_1 to m_3, reducing the load of m_1 to 0 and increasing the loads of m_2 and m_3 to $Q_1 = 100$. Given that the demand of customer 1, initially assigned to route m_1, is $q_1 = 30$, it cannot be transferred in its entirety to either route m_2 or m_3. Consequently, solution $(\tilde{x}, \tilde{y}, \tilde{u})$ cannot be transformed into a same-cost solution that is feasible for formulation F.

B.3 Adjustments to the labeling algorithm

To impose the decisions related to the *customer-satellite-route assignment* branching rule, the labeling algorithm described in Section 5.3.1.1 is modified as follows:

a. To impose that customer i is supplied from satellite s by first-echelon route m, customer i is removed from vertex sets $N_{s'}$ such that $s' \neq s$:

   i. In the initial label E_{0t}, customer i is included in sets U_{0t} associated with time $t \in T_s$ such that $t < t_s^m$. Note that the unreachability status of i remains unchanged when considering ng-paths.

   ii. When extending a label E_{lt}, associated with a partial path l ending at vertex $h \in N_s \setminus \{n_s^{sk}\}$ and a time $t \in T_s$, along an arc $(h, j) \in A_s$ such that $T_{lt}^{start} \geq t_s^m$, the REF (5.3.3), computing this reduced cost of label $E_{l't}$ resulting from the extension, is adjusted as:

$$T_{l't}^{rdc} = T_{lt}^{rdc} + \bar{c}_{hj} - \begin{cases} q_j \cdot \pi_{m(d_j, T_{lt}^{start})}, & \text{if } j \notin \{n_s^{sk}, i\}, \\ q_j \pi_m, & \text{if } j = i, \\ 0 & \text{otherwise.} \end{cases} \tag{B.3.1}$$

b. To impose that customer i is supplied by a first-echelon route $m' \neq m$ whenever i is serviced through satellite s:

   i. No changes need to be made when extending a label E_{lt}, associated with a partial path l ending at vertex $h \in N_s \setminus \{n_s^{sk}\}$ and a time $t \in T_s$ such that $T_{lt}^{start} < t_s^m$.

   ii. When extending a label E_{lt}, associated with a partial path l ending at vertex $h \in N_s \setminus \{n_s^{sk}\}$ and a time $t \in T_s$, along an arc $(h, j) \in A_s$ such that $T_{lt}^{start} = t \geq t_s^m$,

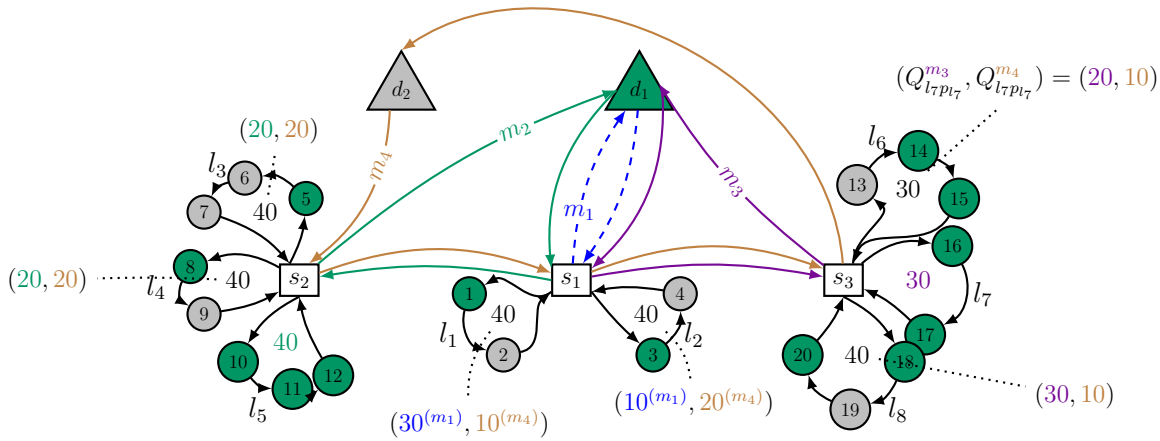

Figure B.1 Example of an integer infeasible solution

the REF (5.3.3) for the reduced cost is adjusted as:

$$T^{rdc}_{l't} = T^{rdc}_{lt} + \bar{c}_{hj} - \begin{cases} q_j \cdot \pi_{m(d_j, T^{start}_{lt})}, & \text{if } j \notin \{n^{sk}_s, i\} \text{ or } (j = i \text{ and } m(d_j, T^{start}_{lt}) \neq m), \\ q_j \cdot \pi_{m_i}, & \text{if } j = i \text{ and } m(d_j, T^{start}_{lt}) = m, \\ 0 & \text{otherwise} \end{cases}, \tag{B.3.2}$$

where $m_i \in \arg\max \left\{\pi_m : m \in \hat{\mathcal{M}}^{T^{start}_{lt}}_s \cap (\mathcal{M}_{d_i} \setminus \{m\})\right\}$, and $\hat{\mathcal{M}}^t_s$ refers to routes in $\mathcal{M}^t_s$ that are not fixed to zero in the current branching node.

B.4 Detailed computational results

This section presents the detailed computational results obtained by our BPC algorithm. Results for instances with $|N^C| = 30, 50$ and 100 are reported in Tables B.1–B.3, respectively. For each row in each table, the first column indicates the instance name in the format `MC-Cxs-`$|N^D|, |N^S|, |N^C|$, where `x` $\in \{a, b, c, d\}$ is the instance category and `s` $\in \{1, 2, 3, 4, 5\}$ the instance number in this category. The next two columns specify the optimal value z^* found by our algorithm, and the computational time (T) in seconds as reported by Dellaert et al. (2021), respectively. Then for our BPC algorithm, we report the computational time (T) in seconds, the integrality gap (Gap) in percentage computed as $100(z^* - \underline{z})/\underline{z}$ where $\underline{z}$ is the root node lower bound before adding cuts, the number of branch-and-bound nodes explored (# Nodes), and the number of enumerated first-echelon routes ($|\mathcal{M}|$). Columns # TIs, # RCIs, # SRIs and # VSIs give, respectively, the number of TIs (5.3.14), RCIs (5.3.21),

SRIs (5.3.23) and lifted VSIs (5.3.20) considered during the solution process. The last four columns describes some characteristics of the computed optimal solution, namely, the number of satellites used ($|N^S|^*$), the number of first-echelon routes used ($|\mathcal{M}|^*$), the percentage of these routes that are back-and-forth ones ($|\mathcal{M}^1|^*$), and the number of second-echelon routes used ($|\mathcal{L}|^*$).

Table B.1 – Detailed computational results for instances with $|N^C| = 30$

Instance	z^*	Dellaert et al. (2021) † T(s)	Our BPC algorithm ‡ T (s)	Gap (%)	# Nodes	$\|\mathcal{M}\|$	# TIs	# RCIs	# SRIs	# VSIs	$\|N^S\|^*$	$\|\mathcal{M}\|^*$	$\|\mathcal{M}^1\|^*$ (%)	$\|\mathcal{L}\|^*$
MC-Ca1-2,3,30	1257.82	76.5	0.6	0.77	16	36	36	4	60	80	2	4	100.0	10
MC-Ca2-2,3,30	1189.67	4.6	0.7	1.21	15	29	26	10	92	77	3	4	100.0	10
MC-Ca3-2,3,30	1207.82	0.8	0.2	0.44	1	29	26	0	21	77	1	4	50.0	10
MC-Ca4-2,3,30	1101.32	2.3	0.3	0.65	3	32	29	4	43	78	1	3	66.7	9
MC-Ca5-2,3,30	1210.97	2.0	0.4	1.01	6	31	29	8	31	82	2	4	75.0	9
MC-Cb1-2,3,30	1259.51	59.9	4.1	3.79	123	27	22	31	90	79	2	3	100.0	10
MC-Cb2-2,3,30	1078.84	1.2	0.5	1.36	8	33	30	19	7	79	3	3	100.0	9
MC-Cb3-2,3,30	1136.04	2.4	2.2	0.82	43	26	22	6	46	77	1	3	66.7	10
MC-Cb4-2,3,30	1172.14	1.8	0.3	1.18	10	36	36	8	8	81	2	4	75.0	10
MC-Cb5-2,3,30	1030.38	2.9	0.7	1.94	13	30	26	23	24	80	2	3	66.7	9
MC-Cc1-2,3,30	989.81	15.9	1.9	1.39	5	32	29	0	22	90	2	3	66.7	8
MC-Cc2-2,3,30	1168.48	1108.1	2.4	1.24	24	36	36	0	23	90	3	4	75.0	10
MC-Cc3-2,3,30	1038.19	211.7	2.8	0.38	7	24	17	6	92	90	2	3	100.0	10
MC-Cc4-2,3,30	1103.19	68.3	1.5	0.48	5	29	26	0	28	90	2	4	75.0	10
MC-Cc5-2,3,30	1052.6	1.7	1.1	0.22	2	28	25	0	12	90	2	3	100.0	9
MC-Cd1-2,3,30	1036.89	1.0	0.3	0.04	3	32	29	10	0	90	2	3	100.0	8
MC-Cd2-2,3,30	1090.12	11.3	0.1	0.03	2	30	27	0	13	87	2	3	66.7	10
MC-Cd3-2,3,30	1094.12	2.1	0.4	0.98	6	22	16	5	10	88	2	3	100.0	10
MC-Cd4-2,3,30	1076.57	15.1	0.3	1.47	3	28	23	2	11	90	2	3	33.3	10
MC-Cd5-2,3,30	1093.93	2.0	0.4	1.06	4	27	23	1	20	88	2	3	66.7	9
MC-Ca1-3,5,30	1247.26	-	29.3	4.26	83	644	625	42	36	137	3	3	33.3	9
MC-Ca2-3,5,30	1220.89	-	27.9	5.72	101	470	449	100	87	125	4	3	66.7	10
MC-Ca3-3,5,30	1231.52	-	12.8	1.66	17	629	599	14	95	139	3	3	0.0	9
MC-Ca4-3,5,30	1151.18	-	5.7	4.77	17	490	466	13	14	134	3	3	100.0	9
MC-Ca5-3,5,30	1146.37	-	5.7	0.73	6	444	426	10	9	130	3	4	100.0	9
MC-Cb1-3,5,30	1212.19	-	24.1	4.73	69	465	430	36	35	141	3	3	66.7	10
MC-Cb2-3,5,30	1216.03	-	16.2	3.03	33	598	572	19	37	135	4	3	33.3	10
MC-Cb3-3,5,30	1264.63	-	8.2	1.02	39	495	469	5	17	129	3	4	50.0	10
MC-Cb4-3,5,30	1173.25	-	5.6	2.30	19	500	464	9	22	120	4	3	66.7	10
MC-Cb5-3,5,30	1166.95	-	19.6	4.07	49	480	447	37	43	143	4	3	33.3	11
MC-Cc1-3,5,30	1304.17	-	73.2	6.07	112	395	382	59	198	150	4	4	50.0	11
MC-Cc2-3,5,30	1130	-	175.9	6.75	138	497	465	102	140	150	3	3	33.3	9
MC-Cc3-3,5,30	1198.81	-	240.3	6.13	258	393	366	39	76	150	4	3	0.0	10

Continued on next page

Table B.1 – *Continued from previous page*

Instance	z^*	Dellaert et al. (2021) †	Our BPC algorithm ‡											
		T (s)	T (s)	Gap (%)	# Nodes	$\lvert\mathcal{M}\rvert$	# TIs	# RCIs	# SRIs	# VSIs	$\lvert N^S\rvert^*$	$\lvert\mathcal{M}\rvert^*$	$\lvert\mathcal{M}^1\rvert^*$ (%)	$\lvert\mathcal{L}\rvert^*$
MC-Cc4-3,5,30	1080.66	-	65.4	3.35	31	617	592	35	23	150	3	3	66.7	9
MC-Cc5-3,5,30	1076.84	-	32.6	1.43	15	423	412	10	36	150	3	4	75.0	8
MC-Cd1-3,5,30	1250.54	-	25.4	6.52	75	421	389	39	102	148	4	3	0.0	11
MC-Cd2-3,5,30	1275.63	-	27.9	5.00	88	400	380	44	72	148	3	4	50.0	9
MC-Cd3-3,5,30	1151.79	-	26.4	4.75	63	436	416	32	76	148	2	3	33.3	10
MC-Cd4-3,5,30	1188.72	-	5.4	1.45	9	414	394	9	30	148	3	4	75.0	9
MC-Cd5-3,5,30	1115.27	-	93.4	6.02	86	522	495	83	154	148	3	3	66.7	9
MC-Ca1-6,4,30	1518.43	-	10.8	3.67	69	195	160	45	94	108	4	6	83.3	11
MC-Ca2-6,4,30	1521.19	-	1.1	1.36	7	193	162	10	11	100	4	6	66.7	10
MC-Ca3-6,4,30	1462.26	-	4.3	1.30	33	173	138	17	4	106	3	6	83.3	10
MC-Ca4-6,4,30	1477.12	-	8.8	2.03	55	184	157	26	36	111	2	6	100.0	10
MC-Ca5-6,4,30	1526.62	-	1.7	1.37	11	175	139	10	4	107	4	6	66.7	10
MC-Cb1-6,4,30	1601.61	-	12.2	1.96	81	239	213	20	71	112	2	6	83.3	10
MC-Cb2-6,4,30	1469.43	-	3.8	2.19	28	189	162	21	29	103	4	6	66.7	9
MC-Cb3-6,4,30	1604.21	-	9.7	3.77	97	201	170	27	72	110	3	6	83.3	10
MC-Cb4-6,4,30	1440.76	-	8.8	2.54	75	151	113	20	21	109	2	6	100.0	10
MC-Cb5-6,4,30	1533.93	-	2.1	2.50	21	178	142	6	11	103	3	6	83.3	11
MC-Cc1-6,4,30	1405.72	-	45.1	4.24	50	196	162	63	140	120	2	6	83.3	9
MC-Cc2-6,4,30	1471.62	-	35.4	4.61	58	188	154	48	123	120	4	6	66.7	10
MC-Cc3-6,4,30	1435.69	-	18.2	3.44	28	182	148	24	15	120	4	6	83.3	9
MC-Cc4-6,4,30	1416.02	-	56.2	4.10	87	182	145	34	92	120	3	6	50.0	9
MC-Cc5-6,4,30	1430.97	-	102.0	3.09	128	181	152	101	130	120	3	6	83.3	9
MC-Cd1-6,4,30	1425.04	-	9.7	2.74	45	167	134	31	32	120	2	6	100.0	9
MC-Cd2-6,4,30	1438.07	-	7.8	2.24	29	141	108	10	76	117	2	6	100.0	9
MC-Cd3-6,4,30	1475.18	-	8.0	1.82	33	166	133	19	13	118	3	6	100.0	9
MC-Cd4-6,4,30	1472.13	-	9.1	1.99	35	158	123	25	80	116	3	6	83.3	9
MC-Cd5-6,4,30	1507.26	-	6.5	1.79	63	161	132	16	87	117	3	6	83.3	11

† Tests ran on a computer equipped with an Intel Core i7-4770 pocessor.

‡ Tests ran on a computer equipped with an Intel Core i7-8700 processor.

Table B.2 – Detailed computational results for instances with $|N^C| = 50$

Instance	z^*	Dellaert et al. (2021) †	Our BPC algorithm ‡											
		T (s)	T (s)	Gap (%)	# Nodes	$\lvert\mathcal{M}\rvert$	# TIs	# RCIs	# SRIs	# VSIs	$\lvert N^S\rvert^*$	$\lvert\mathcal{M}\rvert^*$	$\lvert\mathcal{M}^1\rvert^*$ (%)	$\lvert\mathcal{L}\rvert^*$
MC-Ca1-2,3,50	1808.2	43.0	1.0	1.16	7	39	42	8	46	132	3	5	60.0	16
MC-Ca2-2,3,50	1743.18	-	5.7	1.95	51	34	33	23	216	133	2	4	75.0	16
MC-Ca3-2,3,50	1766.5	186.0	0.9	0.33	11	37	38	2	27	136	3	5	100.0	16
MC-Ca4-2,3,50	1607.81	30.0	1.1	0.58	8	34	33	7	56	138	3	4	100.0	16
MC-Ca5-2,3,50	1617.26	116.0	8.7	0.63	85	36	36	24	64	132	3	4	75.0	15
MC-Cb1-2,3,50	1722.39	-	128.1	1.61	567	39	42	75	132	143	3	5	80.0	15
MC-Cb2-2,3,50	1638.38	-	9.9	1.46	51	36	36	32	22	134	3	4	75.0	14
MC-Cb3-2,3,50	1694.12	-	208.0	2.68	1165	34	33	210	311	142	3	4	100.0	16
MC-Cb4-2,3,50	1648.86	-	24.1	1.45	143	36	36	29	44	129	3	4	100.0	15
MC-Cb5-2,3,50	1694.42	-	30.4	1.15	171	39	42	77	52	143	3	5	40.0	15
MC-Cc1-2,3,50	1547.9	-	12.7	0.67	11	34	33	37	34	150	3	4	50.0	15
MC-Cc2-2,3,50	1573.41	-	5.6	0.24	4	32	30	0	29	150	3	4	75.0	15
MC-Cc3-2,3,50	1529.43	-	21.5	0.81	27	36	36	21	81	150	3	4	75.0	15
MC-Cc4-2,3,50	1566.67	-	245.1	1.47	241	29	26	58	102	150	3	4	75.0	15
MC-Cc5-2,3,50	1579.08	-	19.8	0.99	19	32	30	21	52	150	3	4	75.0	15
MC-Cd1-2,3,50	1602.73	-	7.7	0.91	47	36	36	32	98	146	3	4	75.0	15
MC-Cd2-2,3,50	1787.01	-	10.6	0.92	65	37	40	34	109	148	3	5	60.0	16
MC-Cd3-2,3,50	1724.4	-	4.3	0.72	17	36	37	17	74	147	3	5	40.0	16
MC-Cd4-2,3,50	1708.93	-	15.6	1.52	73	35	37	37	107	150	3	5	80.0	15
MC-Cd5-2,3,50	1780.23	-	3.1	1.31	25	33	33	21	62	145	3	5	100.0	16
MC-Ca1-3,5,50	1814.7	4386.0	773.8	2.32	1479	689	706	90	314	229	4	6	66.7	16
MC-Ca2-3,5,50	1828.37	4758.0	80.4	3.08	163	622	627	74	209	210	5	6	50.0	16
MC-Ca3-3,5,50	1664.81	2158.0	213.4	3.28	190	834	841	106	276	231	4	5	40.0	14
MC-Ca4-3,5,50	1711.82	200.0	143.4	2.74	178	572	570	160	225	218	4	4	25.0	15
MC-Ca5-3,5,50	1685.09	2469.0	911.6	6.63	1334	624	613	194	506	229	3	4	75.0	14
MC-Cb1-3,5,50	1822.96	-	77.6	2.30	195	760	761	23	71	223	5	6	83.3	16
MC-Cb2-3,5,50	1849.23	-	655.0	3.79	1451	710	710	472	505	223	4	5	60.0	16
MC-Cb3-3,5,50	1839.83	-	121.6	2.38	149	798	821	23	110	225	4	6	66.7	17
MC-Cb4-3,5,50	1726.29	-	201.7	2.47	339	727	737	113	202	220	4	5	40.0	16
MC-Cb5-3,5,50	1683.58	-	408.2	3.58	733	740	731	102	92	224	4	4	25.0	14
MC-Cc1-3,5,50	1549.84	-	899.5	3.50	163	584	570	126	269	250	4	4	25.0	15
MC-Cc2-3,5,50	1683.01	-	551.3	3.16	138	760	767	112	302	250	4	5	40.0	15
MC-Cc3-3,5,50	1613.64	-	193.7	2.56	29	762	769	29	74	250	5	5	40.0	15

Continued on next page

Table B.2 – *Continued from previous page*

Instance	z^*	Dellaert et al. (2021) †	Our BPC algorithm ‡											
		T (s)	T (s)	Gap (%)	# Nodes	$\lvert\mathcal{M}\rvert$	# TIs	# RCIs	# SRIs	# VSIs	$\lvert N^S\rvert^*$	$\lvert\mathcal{M}\rvert^*$	$\lvert\mathcal{M}^1\rvert^*$ (%)	$\lvert\mathcal{L}\rvert^*$
MC-Cc4-3,5,50	1827.61	-	528.7	3.62	177	657	664	118	334	250	5	6	66.7	17
MC-Cc5-3,5,50	1730.19	-	238.1	3.89	102	477	463	40	146	250	4	6	50.0	15
MC-Cd1-3,5,50	1738.81	-	124.3	2.92	73	757	769	87	160	248	4	6	83.3	15
MC-Cd2-3,5,50	1764.77	-	124.4	4.07	108	666	682	122	182	250	5	6	50.0	15
MC-Cd3-3,5,50	1652.72	-	85.0	2.76	62	660	655	50	142	250	4	5	80.0	16
MC-Cd4-3,5,50	1717.1	-	82.5	3.84	75	694	691	63	121	244	5	5	40.0	16
MC-Cd5-3,5,50	1663.41	-	207.3	3.37	188	620	624	124	324	243	5	5	40.0	15
MC-Ca1-6,4,50	2110.31	-	72.6	2.71	203	243	216	96	268	172	4	6	83.3	16
MC-Ca2-6,4,50	2017.13	-	26.4	1.32	47	245	216	14	82	180	3	6	83.3	16
MC-Ca3-6,4,50	2214.22	-	20.2	1.63	48	238	220	37	72	176	4	8	87.5	16
MC-Ca4-6,4,50	1973.93	-	7.8	2.70	21	228	199	21	43	166	4	6	66.7	15
MC-Ca5-6,4,50	2075.04	-	42.4	2.56	96	240	217	94	73	176	3	7	85.7	15
MC-Cb1-6,4,50	2173.65	-	45.1	2.74	165	263	234	72	168	167	4	6	83.3	17
MC-Cb2-6,4,50	2177.47	-	6.5	1.86	15	216	187	16	23	176	3	7	100.0	17
MC-Cb3-6,4,50	2167.62	-	23.2	2.19	71	256	233	49	28	177	4	7	100.0	16
MC-Cb4-6,4,50	1985.13	-	54.8	4.01	137	279	254	102	113	175	3	6	83.3	16
MC-Cb5-6,4,50	2075.33	-	128.9	3.80	357	188	157	107	331	176	4	6	50.0	16
MC-Cc1-6,4,50	1960.1	-	913.9	5.05	245	225	195	232	284	200	4	6	50.0	15
MC-Cc2-6,4,50	1904.53	-	87.6	2.18	39	224	197	32	28	200	3	6	83.3	15
MC-Cc3-6,4,50	1948.02	-	76.0	1.90	27	180	139	32	44	200	4	6	50.0	16
MC-Cc4-6,4,50	1870.07	-	199.6	3.73	81	215	184	49	130	200	4	6	50.0	15
MC-Cc5-6,4,50	2166.48	-	197.8	4.22	116	217	190	60	198	200	4	7	85.7	17
MC-Cd1-6,4,50	1980.33	-	63.8	3.26	81	196	165	81	207	197	2	6	66.7	15
MC-Cd2-6,4,50	1960.23	-	319.3	3.23	459	252	223	137	536	197	3	6	50.0	15
MC-Cd3-6,4,50	2038.08	-	144.9	3.00	168	220	192	77	411	195	3	6	66.7	16
MC-Cd4-6,4,50	1867.35	-	31.3	2.85	46	225	192	43	47	197	4	6	66.7	15
MC-Cd5-6,4,50	1980.33	-	63.2	3.26	81	196	165	81	207	197	2	6	66.7	15

† Tests ran on a computer equipped with an Intel Core i7-4770 pocessor.

‡ Tests ran on a computer equipped with an Intel Core i7-8700 processor.

Table B.3 – Detailed computational results for instances with $|N^C| = 100$

Instance	z^*	Dellaert et al. (2021) † T (s)	Our BPC algorithm ‡ T (s)	Gap (%)	# Nodes	$\|\mathcal{M}\|$	# TIs	# RCIs	# SRIs	# VSIs	$\|N^S\|^*$	$\|\mathcal{M}\|^*$	$\|\mathcal{M}^1\|^*$ (%)	$\|\mathcal{L}\|^*$
MC-Ca1-2,3,100	3126.86	-	1172.8	0.39	1357	51	66	48	71	268	3	9	66.7	32
MC-Ca2-2,3,100	3270.5	-	1244.9	0.72	845	51	66	26	134	265	3	9	66.7	32
MC-Ca3-2,3,100	3004.22	-	69.3	1.00	57	46	55	58	217	249	3	8	37.5	30
MC-Ca4-2,3,100	3305.93	-	4827.3	0.88	3688	47	59	55	98	280	3	8	75.0	31
MC-Ca5-2,3,100	3095.43	-	365.6	0.86	182	48	60	35	166	277	3	8	62.5	30
MC-Cb1-2,3,100	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MC-Cb2-2,3,100	3017.68	-	10624.2	0.95	8406	46	55	147	88	268	3	8	75.0	30
MC-Cb3-2,3,100	3000.1	-	9720.1	1.31	5805	48	60	137	182	270	3	8	75.0	30
MC-Cb4-2,3,100	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MC-Cb5-2,3,100	3035.56	-	5796.1	0.78	4993	48	60	78	56	267	3	8	75.0	29
MC-Cc1-2,3,100	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MC-Cc2-2,3,100	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MC-Cc3-2,3,100	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MC-Cc4-2,3,100	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MC-Cc5-2,3,100	2944.71	-	1391.5	0.67	173	48	60	15	165	300	3	8	75.0	30
MC-Cd1-2,3,100	2932.14	-	1508.5	0.72	335	49	61	50	218	298	3	9	66.7	30
MC-Cd2-2,3,100	3157.21	-	1617.5	1.83	484	49	61	71	665	293	3	9	66.7	32
MC-Cd3-2,3,100	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MC-Cd4-2,3,100	3111.62	-	3607.8	0.75	932	48	60	51	182	296	3	8	75.0	30
MC-Cd5-2,3,100	2976.11	-	265.8	1.08	53	41	45	66	438	295	3	8	50.0	30
MC-Ca1-3,5,100	3206.67	2919.0	9686.1	1.48	5453	966	1051	207	235	424	5	10	60.0	31
MC-Ca2-3,5,100	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MC-Ca3-3,5,100	3029.88	-	10049.4	1.22	3946	978	1051	212	329	428	5	9	66.7	29
MC-Ca4-3,5,100	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MC-Ca5-3,5,100	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MC-Cb1-3,5,100	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MC-Cb2-3,5,100	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MC-Cb3-3,5,100	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MC-Cb4-3,5,100	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MC-Cb5-3,5,100	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MC-Cc1-3,5,100	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MC-Cc2-3,5,100	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MC-Cc3-3,5,100	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Continued on next page

Table B.3 – *Continued from previous page*

Instance	z^*	Dellaert et al. (2021) † T (s)	Our BPC algorithm ‡ T (s)	Gap (%)	# Nodes	$\lvert\mathcal{M}\rvert$	# TIs	# RCIs	# SRIs	# VSIs	$\lvert N^S\rvert^*$	$\lvert\mathcal{M}\rvert^*$	$\lvert\mathcal{M}^1\rvert^*$ (%)	$\lvert\mathcal{L}\rvert^*$
MC-Cc4-3,5,100	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MC-Cc5-3,5,100	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MC-Cd1-3,5,100	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MC-Cd2-3,5,100	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MC-Cd3-3,5,100	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MC-Cd4-3,5,100	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MC-Cd5-3,5,100	3048.75	-	4008.8	1.15	363	953	1026	249	864	490	3	9	100.0	30
MC-Ca1-6,4,100	3364.65	-	3810.1	1.99	1657	350	352	362	827	358	4	10	40.0	30
MC-Ca2-6,4,100	3465.14	-	4823.9	2.27	3348	324	325	1485	1527	352	4	10	50.0	28
MC-Ca3-6,4,100	3960.03	-	110.8	1.49	83	310	312	82	168	341	4	12	83.3	31
MC-Ca4-6,4,100	3374.9	-	7514.2	2.18	2999	333	330	698	1086	355	4	9	66.7	29
MC-Ca5-6,4,100	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MC-Cb1-6,4,100	3402.62	-	5883.6	2.01	2887	341	343	724	210	362	4	10	60.0	28
MC-Cb2-6,4,100	3678.83	-	762.1	1.67	768	362	383	231	302	352	4	12	75.0	31
MC-Cb3-6,4,100	3725.96	-	4876.3	1.50	3615	311	320	196	151	358	4	11	81.8	32
MC-Cb4-6,4,100	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MC-Cb5-6,4,100	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MC-Cc1-6,4,100	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MC-Cc2-6,4,100	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MC-Cc3-6,4,100	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MC-Cc4-6,4,100	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MC-Cc5-6,4,100	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MC-Cd1-6,4,100	3474.89	-	3036.7	1.74	429	329	333	256	574	393	4	11	36.4	31
MC-Cd2-6,4,100	3374.88	-	999.5	1.79	541	337	338	359	269	390	4	10	80.0	30
MC-Cd3-6,4,100	3280.52	-	7594.8	2.78	1997	302	300	411	913	396	4	10	60.0	28
MC-Cd4-6,4,100	3545.37	-	6757.5	2.48	2359	263	252	316	979	389	4	11	81.8	32
MC-Cd5-6,4,100	3469.71	-	10275.3	1.46	1215	265	267	253	723	393	4	11	54.5	31

† Tests ran on a computer equipped with an Intel Core i7-4770 pocessor.

‡ Tests ran on a computer equipped with an Intel Core i7-8700 processor.

ANNEXE C RÉSULTATS DÉTAILLÉS POUR LE 2E-VRPTW-SPD

C.1 Résultats numériques détaillés

Dans cette section, nous présentons les résultats numériques détaillés obtenus par notre algorithme BPC. Les résultats pour les instances avec 30, 50 et 100 clients sont présentés dans les tableaux C.1, C.2 et C.3, respectivement. Pour chaque ligne de ces tableaux, la première colonne indique le nom de l'instance. La deuxième colonne indique la valeur de la solution optimale (z^*) lorsque celle-ci est connue. Nous rapportons ensuite le temps de calcul (T) en secondes, le gap d'intégrité (Gap) en pourcentage, les nombres de nœuds de branchement (#Nœuds), de routes de premier échelon énumérées ($|\mathcal{M}|$), de TIs ajoutées au PM au nœud racine (# TIs), et d'inégalités valides générées pour chaque famille d'inégalités considérée (# RCIs, # SRIs, # VSIs et # TPIs). Les colonnes restantes donnent des statistiques sur la solution optimale obtenue, à savoir les nombres de dépôts ($|N^D|^*$), de satellites ($|N^S|^*$), de routes de premier échelon ($|\mathcal{M}|^*$), de routes de premier échelon pour la livraison ($|\mathcal{M}|^*_D$) et la collecte ($|\mathcal{M}|^*_P$) de marchandises, et de routes de second échelon ($|\mathcal{L}|^*$) qui se trouvent dans la solution. Nous rapportons aussi le pourcentage des BFFERs ($|\mathcal{M}^1|^*$) parmi les routes de premier échelon utilisées.

Tableau C.1 – Résultats numériques détaillés pour les instances avec $|N^C| = 30$

Instance	z^*	T (s)	Gap (%)	# Nœuds	$\|\mathcal{M}\|$	# TIs	# RCIs	# SRIs	# VSIs	# TPIs	$\|N^D\|^*$	$\|N^S\|^*$	$\|\mathcal{M}\|^*$	$\|\mathcal{M}\|^*_D$	$\|\mathcal{M}\|^*_P$	$\|\mathcal{M}^1\|^*$ (%)	$\|\mathcal{L}\|^*$
Ca1-2,3,30	894.14	1.1	0.77	9	18	60	13	27	20	3	1	1	4	2	2	100.0	6
Ca2-2,3,30	953.44	0.7	0.49	8	22	57	2	27	56	1	2	2	4	2	2	100.0	7
Ca3-2,3,30	1044.91	1.9	0.43	21	20	47	10	39	26	6	1	1	4	2	2	100.0	7
Ca4-2,3,30	895.06	0.6	1.12	3	22	55	2	22	40	3	1	1	4	2	2	100.0	7
Ca5-2,3,30	946.68	4.1	1.45	26	20	47	2	63	65	3	2	2	4	2	2	100.0	7
Cb1-2,3,30	971.01	3.5	1.71	33	18	60	7	53	47	4	1	1	4	2	2	100.0	7
Cb2-2,3,30	897.56	6.3	2.05	23	20	49	20	51	85	4	2	2	4	2	2	100.0	6
Cb3-2,3,30	1016.38	0.8	0.91	5	17	38	13	23	44	2	1	1	4	2	2	100.0	6
Cb4-2,3,30	934.84	0.2	0.00	1	20	48	0	0	30	0	1	1	4	2	2	100.0	7
Cb5-2,3,30	918.01	0.6	0.69	6	22	56	0	29	42	2	2	2	3	2	1	66.7	6
Cc1-2,3,30	738.51	7.8	0.92	5	18	60	8	18	54	3	1	1	3	2	1	100.0	5
Cc2-2,3,30	851.73	67.5	1.87	37	21	53	9	94	88	0	2	2	4	2	2	100.0	6
Cc3-2,3,30	928.53	41.1	2.15	25	14	27	26	97	43	0	1	1	4	2	2	100.0	6
Cc4-2,3,30	887.4	65.0	2.60	147	20	48	42	127	52	3	1	1	4	2	2	100.0	6
Cc5-2,3,30	862.24	10.5	1.82	15	20	47	12	37	42	2	2	2	4	2	2	100.0	5
Cd1-2,3,30	793.05	6.8	1.26	23	18	60	12	56	56	6	1	1	3	2	1	100.0	5
Cd2-2,3,30	925.85	0.2	0.02	3	21	53	0	10	47	0	2	2	4	2	2	100.0	7
Cd3-2,3,30	1011.53	1.0	0.51	5	14	27	3	24	52	2	1	1	4	2	2	100.0	6
Cd4-2,3,30	944.69	3.5	1.75	20	20	48	3	36	43	5	2	2	4	2	2	100.0	7
Cd5-2,3,30	931.62	1.9	0.89	15	20	47	0	29	86	3	2	2	4	2	2	100.0	6
Ca1-3,5,30	973.22	4.2	0.16	6	392	2782	4	0	121	0	2	3	4	2	2	50.0	6
Ca2-3,5,30	949.22	23.2	1.49	13	387	2674	1	49	115	4	2	2	4	2	2	100.0	6
Ca3-3,5,30	992.52	59.2	1.39	21	389	2949	0	36	124	4	2	2	4	2	2	100.0	6
Ca4-3,5,30	892.2	2230.1	2.45	147	359	2712	14	206	141	19	2	2	4	2	2	100.0	6
Ca5-3,5,30	919.49	9.9	1.03	7	408	2735	0	25	55	6	2	2	4	2	2	100.0	6
Cb1-3,5,30	946.66	39.0	3.33	41	369	2507	3	45	131	5	2	2	4	2	2	100.0	6
Cb2-3,5,30	962.36	24.7	1.16	13	387	2567	10	36	129	2	2	2	4	2	2	100.0	6
Cb3-3,5,30	977.83	64.6	3.90	32	342	2500	3	53	114	7	2	2	4	2	2	100.0	6
Cb4-3,5,30	884.63	162.2	0.76	41	357	2702	1	29	88	14	2	2	4	2	2	100.0	6
Cb5-3,5,30	994.27	426.4	1.27	77	356	2080	2	51	136	5	2	3	4	2	2	50.0	8
Cc1-3,5,30	924.8	36.5	1.21	12	344	2324	10	36	116	0	2	2	4	2	2	100.0	7
Cc2-3,5,30	842.33	597.3	3.97	71	358	2486	17	113	184	3	1	1	3	2	1	100.0	6
Cc3-3,5,30	857.75	43.4	0.84	12	324	2243	37	10	124	0	2	2	4	2	2	100.0	6
Cc4-3,5,30	756.84	218.5	2.45	27	349	2570	8	23	154	0	1	2	3	1	2	66.7	7

Suite à la page suivante

Tableau C.1 – *Suite de la page précédente*

Instance	z^*	T (s)	Gap (%)	# Nœuds	$\|\mathcal{M}\|$	# TIs	# RCIs	# SRIs	# VSIs	# TPIs	$\|N^D\|^*$	$\|N^S\|^*$	$\|\mathcal{M}\|^*$	$\|\mathcal{M}\|^*_D$	$\|\mathcal{M}\|^*_P$	$\|\mathcal{M}^1\|^*$ (%)	$\|\mathcal{L}\|^*$
`Cc5-3,5,30`	663.24	2341.6	1.98	23	383	2017	2	113	183	0	1	1	2	1	1	100.0	4
`Cd1-3,5,30`	1010.74	15.0	1.34	12	344	2324	1	23	103	6	2	3	4	2	2	50.0	8
`Cd2-3,5,30`	965.74	37.2	1.30	24	343	2230	7	48	120	7	2	2	4	2	2	100.0	7
`Cd3-3,5,30`	893.46	8.3	0.59	7	339	2468	4	26	104	4	2	2	4	2	2	100.0	7
`Cd4-3,5,30`	889.37	404.7	1.07	45	344	2485	9	60	132	5	2	2	4	2	2	100.0	6
`Cd5-3,5,30`	900.39	60.5	1.34	14	385	2504	8	50	127	8	2	2	4	2	2	100.0	6
`Ca1-6,4,30`	901.88	2.1	0.16	5	97	405	0	0	92	4	2	2	4	2	2	100.0	7
`Ca2-6,4,30`	952.77	12.0	1.11	19	104	374	5	54	56	3	2	2	4	2	2	100.0	7
`Ca3-6,4,30`	914.61	3.0	0.33	6	95	358	0	10	83	3	2	2	4	2	2	100.0	7
`Ca4-6,4,30`	857.63	43.3	1.59	31	102	461	0	50	111	3	2	2	4	2	2	100.0	6
`Ca5-6,4,30`	872.75	1.7	0.96	5	96	353	5	0	83	3	2	2	4	2	2	100.0	7
`Cb1-6,4,30`	897.78	2.6	0.80	6	98	402	2	8	79	2	2	2	4	2	2	100.0	7
`Cb2-6,4,30`	978.32	13.9	2.03	25	94	318	4	16	112	2	2	2	4	2	2	100.0	7
`Cb3-6,4,30`	942.9	4.1	1.15	10	100	396	5	23	94	4	2	2	4	2	2	100.0	7
`Cb4-6,4,30`	838.55	31.2	1.60	33	93	394	7	29	110	2	1	1	3	2	1	100.0	6
`Cb5-6,4,30`	941.03	1.8	1.38	7	100	379	0	0	74	1	2	2	4	2	2	100.0	8
`Cc1-6,4,30`	792.57	208.2	4.52	34	99	420	59	99	155	3	2	2	4	2	2	100.0	6
`Cc2-6,4,30`	908.25	17.5	0.62	8	104	384	1	34	99	1	2	2	4	2	2	100.0	6
`Cc3-6,4,30`	857.45	34.3	1.05	15	101	404	0	44	120	0	2	2	4	2	2	100.0	5
`Cc4-6,4,30`	736.2	49.6	2.44	15	95	428	47	27	121	1	1	1	3	1	2	100.0	6
`Cc5-6,4,30`	746.07	125.9	2.60	17	105	416	22	88	96	1	1	1	3	2	1	100.0	5
`Cd1-6,4,30`	892.74	13.2	1.21	17	93	371	1	29	108	3	2	2	4	2	2	100.0	7
`Cd2-6,4,30`	924.6	8.3	3.48	14	96	326	9	19	114	4	3	2	3	2	1	66.7	7
`Cd3-6,4,30`	900.51	1.9	0.41	3	91	337	33	6	80	3	2	2	4	2	2	100.0	6
`Cd4-6,4,30`	896.79	3.1	1.48	11	87	371	0	19	100	2	2	2	4	2	2	100.0	7
`Cd5-6,4,30`	894.46	3.4	0.67	6	86	300	1	10	78	3	2	2	4	2	2	100.0	7

Tableau C.2 – Résultats numériques détaillés pour les instances avec $|N^C| = 50$

Instance	z^*	T (s)	Gap (%)	# Nœuds	$\|\mathcal{M}\|$	# TIs	# RCIs	# SRIs	# VSIs	# TPIs	$\|N^D\|^*$	$\|N^S\|^*$	$\|\mathcal{M}\|^*$	$\|\mathcal{M}\|^*_D$	$\|\mathcal{M}\|^*_P$	$\|\mathcal{M}^1\|^*$ (%)	$\|\mathcal{L}\|^*$
Ca1-2,3,50	1434.5	33.8	1.47	61	24	70	8	79	90	21	2	2	5	3	2	100.0	11
Ca2-2,3,50	1349.9	242.8	1.69	245	22	55	6	146	104	9	2	2	4	2	2	100.0	10
Ca3-2,3,50	1296.71	31.6	0.59	39	22	56	4	48	98	4	2	2	4	2	2	100.0	10
Ca4-2,3,50	1321.59	72.1	1.44	79	22	57	5	123	104	15	2	2	4	2	2	100.0	11
Ca5-2,3,50	1330.13	15.9	0.49	40	22	60	1	44	89	3	2	3	5	3	2	80.0	11
Cb1-2,3,50	1335	461.4	1.71	305	22	56	17	152	107	23	2	2	4	2	2	100.0	10
Cb2-2,3,50	1296.46	72.8	1.56	69	22	55	1	75	97	12	2	2	4	2	2	100.0	10
Cb3-2,3,50	1401.86	171.5	1.06	225	22	56	12	81	108	18	2	2	4	2	2	100.0	12
Cb4-2,3,50	1314.45	212.6	1.85	159	22	57	19	157	108	7	2	2	4	2	2	100.0	11
Cb5-2,3,50	1266.3	196.3	1.38	231	22	60	17	77	110	28	2	3	4	2	2	50.0	10
Cc1-2,3,50	1136.4	1787.4	2.04	101	21	52	61	171	136	4	2	2	4	2	2	100.0	9
Cc2-2,3,50	1186.51	2154.0	1.46	111	21	52	100	180	104	0	2	2	4	2	2	100.0	9
Cc3-2,3,50	1131.44	69.5	0.62	8	22	56	15	41	119	1	2	2	4	2	2	100.0	8
Cc4-2,3,50	1163.55	2800.2	2.28	239	21	53	98	193	145	1	2	2	4	2	2	100.0	8
Cc5-2,3,50	1164.09	992.9	1.27	113	21	52	242	200	119	2	2	2	4	2	2	100.0	8
Cd1-2,3,50	1270.95	15.8	1.59	19	21	52	2	36	109	10	2	2	4	2	2	100.0	9
Cd2-2,3,50	1306.92	53.2	2.09	29	22	55	7	102	111	8	2	2	4	2	2	100.0	10
Cd3-2,3,50	1380.01	26.6	0.48	39	24	71	1	50	92	5	2	3	5	2	3	80.0	11
Cd4-2,3,50	1276.36	212.7	1.37	135	22	57	6	133	116	18	2	2	4	2	2	100.0	10
Cd5-2,3,50	1267.79	4.7	0.35	12	25	80	16	21	101	4	2	3	5	2	3	80.0	9
Ca1-3,5,50	1292.71	2555.3	1.08	75	404	3321	0	84	154	10	2	3	5	3	2	80.0	11
Ca2-3,5,50	1281.74	2692.6	2.27	111	465	3101	3	29	138	16	2	2	4	2	2	100.0	11
Ca3-3,5,50	1144.02	4854.1	1.33	155	385	2727	13	135	219	3	2	2	4	2	2	100.0	10
Ca4-3,5,50	1216.86	7197.3	1.54	315	372	2401	16	136	200	10	2	2	4	2	2	100.0	10
Ca5-3,5,50	1208.71	222.3	3.29	23	385	2758	3	79	162	8	2	2	4	2	2	100.0	10
Cb1-3,5,50	1286.62	5040.0	1.18	147	403	2865	15	120	149	5	2	3	5	3	2	80.0	10
Cb2-3,5,50	1272.97	3311.4	2.21	106	476	3230	42	59	208	11	3	3	5	3	2	80.0	10
Cb3-3,5,50	1250.04	253.7	0.54	21	392	3182	0	28	124	6	2	2	5	2	3	100.0	10
Cb4-3,5,50	1169.34	3665.5	1.05	69	383	2511	11	76	125	11	2	2	4	2	2	100.0	10
Cb5-3,5,50	1194.12	1443.4	2.40	61	372	2665	8	39	213	8	2	2	4	2	2	100.0	10
Cc1-3,5,50	1054.26	7107.6	0.80	35	392	2679	6	94	188	0	2	2	4	2	2	100.0	8
Cc2-3,5,50	1165.4	3043.6	2.47	35	446	2846	9	81	168	4	2	2	4	2	2	100.0	10
Cc3-3,5,50	1106.14	8003.1	1.17	87	384	2727	9	126	201	5	2	2	4	2	2	100.0	9
Cc4-3,5,50	1219.53	4610.0	0.99	43	355	2664	22	74	167	1	2	2	5	2	3	100.0	10

Suite à la page suivante

Tableau C.2 – *Suite de la page précédente*

Instance	z^*	T (s)	Gap (%)	# Nœuds	$\|\mathcal{M}\|$	# TIs	# RCIs	# SRIs	# VSIs	# TPIs	$\|N^D\|^*$	$\|N^S\|^*$	$\|\mathcal{M}\|^*$	$\|\mathcal{M}\|^*_D$	$\|\mathcal{M}\|^*_P$	$\|\mathcal{M}^1\|^*$ (%)	$\|\mathcal{L}\|^*$
Cc5-3,5,50	1157.2	6014.7	3.29	119	364	2589	15	141	206	8	2	2	4	2	2	100.0	10
Cd1-3,5,50	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cd2-3,5,50	1218.41	2785.4	2.05	75	434	2823	44	52	173	12	2	2	4	2	2	100.0	9
Cd3-3,5,50	1280.33	249.4	1.08	18	378	3025	3	56	178	7	2	3	5	2	3	80.0	11
Cd4-3,5,50	1167.97	5904.5	1.42	107	375	2412	13	143	172	12	2	2	4	2	2	100.0	10
Cd5-3,5,50	1162.32	591.6	3.27	37	378	2691	7	71	205	12	2	2	4	2	2	100.0	9
Ca1-6,4,50	1300.03	341.9	2.65	81	94	389	3	98	117	7	3	3	5	3	2	80.0	11
Ca2-6,4,50	1192.13	117.2	0.57	31	94	384	0	31	120	0	2	2	4	2	2	100.0	9
Ca3-6,4,50	1250.26	324.1	1.20	41	109	478	7	39	166	8	2	2	4	2	2	100.0	11
Ca4-6,4,50	1202.48	1550.1	1.10	157	104	402	18	92	139	9	2	2	4	2	2	100.0	10
Ca5-6,4,50	1196.84	444.7	1.68	125	110	546	3	179	182	8	2	3	4	2	2	50.0	11
Cb1-6,4,50	1406.51	41.8	1.78	37	93	384	3	49	120	3	3	3	5	3	2	80.0	13
Cb2-6,4,50	1328.37	8935.4	2.84	1829	95	389	48	204	234	89	2	2	4	2	2	100.0	11
Cb3-6,4,50	1253.77	2881.0	1.74	479	114	510	8	142	175	18	2	2	4	2	2	100.0	10
Cb4-6,4,50	1191.49	638.5	1.57	65	107	433	7	79	138	4	2	2	4	2	2	100.0	10
Cb5-6,4,50	1229.59	2165.8	1.76	349	106	494	9	127	166	47	3	3	4	2	2	50.0	11
Cc1-6,4,50	1137.64	528.5	1.13	21	90	366	13	68	158	0	2	2	4	2	2	100.0	10
Cc2-6,4,50	1054.32	393.6	0.97	17	92	365	19	67	183	1	2	2	4	2	2	100.0	8
Cc3-6,4,50	1129.94	6815.0	1.67	152	93	336	60	150	218	3	2	2	4	2	2	100.0	10
Cc4-6,4,50	1051.19	2412.5	1.32	47	102	389	40	160	180	5	2	2	4	2	2	100.0	8
Cc5-6,4,50	1169.91	4204.0	1.78	210	103	545	93	106	225	0	2	2	5	2	3	100.0	10
Cd1-6,4,50	1142.92	164.3	1.41	31	94	389	8	82	185	4	2	2	4	2	2	100.0	9
Cd2-6,4,50	1226.06	241.5	1.46	29	92	373	1	65	142	7	2	2	4	2	2	100.0	11
Cd3-6,4,50	1232.89	6992.4	2.27	435	105	433	59	74	201	68	2	2	4	2	2	100.0	10
Cd4-6,4,50	1156.26	548.6	1.38	57	104	391	11	121	172	11	2	2	4	2	2	100.0	10
Cd5-6,4,50	1213.47	29.4	0.72	15	97	445	8	30	152	2	2	2	4	2	2	100.0	10

Tableau C.3 – Résultats numériques détaillés pour les instances avec $|N^C| = 100$

Instance	z^*	T (s)	Gap (%)	# Nœuds	$\|\mathcal{M}\|$	# TIs	# RCIs	# SRIs	# VSIs	# TPIs	$\|N^D\|^*$	$\|N^S\|^*$	$\|\mathcal{M}\|^*$	$\|\mathcal{M}\|^*_D$	$\|\mathcal{M}\|^*_P$	$\|\mathcal{M}^1\|^*$ (%)	$\|\mathcal{L}\|^*$
Ca1-2,3,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Ca2-2,3,100	2445.79	3532.8	2.05	445	28	95	33	261	95	40	2	2	8	4	4	100.0	25
Ca3-2,3,100	2116.7	4440.4	1.85	543	28	97	117	145	71	20	2	2	8	4	4	100.0	20
Ca4-2,3,100	2525.36	1064.6	0.94	313	24	108	7	145	95	22	1	3	8	4	4	100.0	22
Ca5-2,3,100	2260.97	6256.8	0.86	1361	27	93	52	119	95	9	2	3	8	4	4	100.0	19
Cb1-2,3,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cb2-2,3,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cb3-2,3,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cb4-2,3,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cb5-2,3,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cc1-2,3,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cc2-2,3,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cc3-2,3,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cc4-2,3,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cc5-2,3,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cd1-2,3,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cd2-2,3,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cd3-2,3,100	2156.47	3720.4	1.90	305	28	97	65	148	50	10	2	2	8	4	4	100.0	22
Cd4-2,3,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cd5-2,3,100	2256.8	1162.4	0.86	243	27	93	25	102	77	14	2	3	8	4	4	100.0	20
Ca1-3,5,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Ca2-3,5,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Ca3-3,5,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Ca4-3,5,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Ca5-3,5,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cb1-3,5,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cb2-3,5,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cb3-3,5,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cb4-3,5,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cb5-3,5,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cc1-3,5,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cc2-3,5,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cc3-3,5,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cc4-3,5,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

Suite à la page suivante

Tableau C.3 – *Suite de la page précédente*

Instance	z^*	T (s)	Gap (%)	# Nœuds	$\|\mathcal{M}\|$	# TIs	# RCIs	# SRIs	# VSIs	# TPIs	$\|N^D\|^*$	$\|N^S\|^*$	$\|\mathcal{M}\|^*$	$\|\mathcal{M}\|^*_D$	$\|\mathcal{M}\|^*_P$	$\|\mathcal{M}^1\|^*$ (%)	$\|\mathcal{L}\|^*$
Cc5-3,5,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cd1-3,5,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cd2-3,5,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cd3-3,5,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cd4-3,5,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cd5-3,5,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Ca1-6,4,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Ca2-6,4,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Ca3-6,4,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Ca4-6,4,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Ca5-6,4,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cb1-6,4,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cb2-6,4,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cb3-6,4,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cb4-6,4,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cb5-6,4,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cc1-6,4,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cc2-6,4,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cc3-6,4,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cc4-6,4,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cc5-6,4,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cd1-6,4,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cd2-6,4,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cd3-6,4,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cd4-6,4,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cd5-6,4,100	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–