

Titre: Méthodes de décomposition pour l'optimisation de la distribution
Title: des produits pétroliers.

Auteur: Abderrahman Bani
Author:

Date: 2024

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Bani, A. (2024). Méthodes de décomposition pour l'optimisation de la distribution
Citation: des produits pétroliers. [Thèse de doctorat, Polytechnique Montréal]. PolyPublie.
<https://publications.polymtl.ca/58322/>

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/58322/>
PolyPublie URL:

**Directeurs de
recherche:** Issmaïl El Hallaoui
Advisors:

Programme: Doctorat en mathématiques
Program:

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

**Méthodes de décomposition pour l'optimisation de la distribution des produits
pétroliers.**

ABDERRAHMAN BANI

Département de mathématiques et de génie industriel

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

Avril 2024

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

Cette thèse intitulée :

**Méthodes de décomposition pour l'optimisation de la distribution des produits
pétroliers.**

présentée par **Abderrahman BANI**

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

François SOUMIS, président

Issmaïl EL HALLAOUI, membre et directeur de recherche

Thibaut VIDAL, membre

Leandro C. COELHO, membre externe

DÉDICACE

Je dédie ce travail à ma mère . . .

REMERCIEMENTS

Au terme de ce parcours académique, il est temps de rendre hommage à ceux qui ont été des piliers essentiels tout au long de cette entreprise intellectuelle. Leur soutien indéfectible et leurs contributions ont largement favorisé la réalisation de cette thèse. Merci à tous ceux qui ont contribué de près ou de loin à la réalisation de cette thèse.

Je tiens tout d’abord à exprimer ma profonde gratitude envers mon directeur de recherche, Professeur Issmail El Hallaoui. Vos conseils éclairés, et votre volonté de partager votre expertise approfondie, ont joué un rôle déterminant dans le développement de mes compétences de chercheur. Je tiens également à vous remercier du fond du coeur pour votre disponibilité, vos anecdotes, votre patience, votre soutien financier et votre encouragement constant, même dans les moments les plus difficiles de cette thèse. Votre engagement envers l’excellence et le partage de votre passion pour la recherche qui a constitué une source d’inspiration pour moi.

Je remercie les membres du jury, les Professeurs François Soumis, Thibaut Vidal et Leandro C. Coelho, qui ont accepté d’évaluer et de critiquer ce travail. Je remercie aussi Polytechnique Montréal, *MITACS* et *Engage*, qui ont participé au financement de cette thèse.

Je tiens à exprimer ma reconnaissance à mes coauteurs, El Mehdi Er Raqabi, Professeur Ayoub Insa Corrêa et Professeur Adil Tahir, qui ont été des partenaires exceptionnels tout au long de cette aventure de recherche. Je remercie énormément El Mehdi qui m’a accompagné dans les quatre derniers mois, je lui suis profondément reconnaissant pour son énergie positive, sa disponibilité et ses conseils de rédaction. Un remerciement particulier au Professeur Ayoub Insa Corrêa qui a joué un rôle essentiel en m’introduisant le problème et les données qui ont été le déclencheur de cette thèse.

Mes remerciements vont également à l’ensemble de l’équipe du *GERAD*, dont les échanges fructueux et la collaboration ont enrichi mon expérience académique. Chacun de vous a apporté une contribution unique, et je suis reconnaissant d’avoir pu évoluer dans un environnement aussi stimulant. Je remercie particulièrement Safae Er-rbib et le Professeur Guy Desaulniers pour me donner l’opportunité de contribuer dans l’article intitulé : *Integrated and sequential solution methods for the cyclic bus driver rostering problem*.

Un grand merci à mes collègues à l’*IREQ* avec qui j’ai passé les quatres dernières années de cette thèse, spécialement mon chef Patrick Jeandroz pour m’avoir ouvert les portes en tant que stagiaire universitaire et pour m’offrir, après mon embauche en tant que chercheur en recherche opérationnelle, toutes les conditions favorables pour terminer cette aventure. Je

remercie également Mouad Morabit, Alexandre Blondin-Massé, Alarie Stéphane et Alexandre Besner pour leur encouragement tout au long de ce périple. Vos discussions, conseils et moments de détente ont été un équilibre bienvenu dans cette aventure parfois exigeante.

Je remercie le chercheur à l'*IREQ* Rachid Hassani, les Professeurs de *HEC Montréal* : Okan Arslan, Jean-François Cordeau et Yossiri Adulyasak, de m'avoir offert l'opportunité d'un stage *MITACS* de 11 mois sur le positionnement optimal des points de services d'Hydro-Québec. Une expérience très enrichissante qui m'a permis de profiter de leur expertise pour approfondir mes connaissances dans le *Network Redesign* et le développement de logiciel d'aide à la décision. Je tiens à exprimer ma sincère reconnaissance envers Rachid Hassani, mon encadrant de stage exceptionnel, sa disponibilité pour répondre à mes questions, discuter des défis et partager ses connaissances ont grandement enrichi mon expérience. Grâce à son mentorat, j'ai acquis des compétences techniques et émotionnelles précieuses qui resteront avec moi tout au long de ma carrière.

Je remercie l'équipe *B-CITI* qui m'ont accueillie pour un stage *Engage* de 12 mois sur la prédiction du taux d'occupation des places de stationnement, le fruit de cette collaboration est la publication du papier de conférence : *A deep learning approach to predict parking occupancy using cluster augmented learning method*. Je remercie Amine Amrouss pour sa guidance précieuse et son soutien continu tout au long de mon expérience au sein de l'équipe.

Je remercie mes anciens Professeurs Imad Hafidi et Abdelmoutalib Metrane qui m'ont ouvert les yeux sur le domaine de la recherche opérationnelle et l'algorithmique avancé. Un domaine que j'ai aimé et dans lequel je me suis lancé avec mon premier stage à Polytechnique Montréal avec Professeur Ilyas Himmich, que je remercie de m'avoir introduit le problème de plus court chemin avec contraintes de ressources et la programmation dynamique.

Enfin, ma famille mérite une reconnaissance spéciale pour son soutien inconditionnel. Votre amour, compréhension, patience et encouragement ont été les fondements sur lesquels reposent mes accomplissements académiques.

RÉSUMÉ

Le pétrole demeure une source d'énergie majeure dans la plupart des sociétés modernes, même si d'autres sources d'énergie alternatives (gaz, électricité et énergie solaire) s'avèrent de plus en plus utilisées. Les stations-service quant à elles sont des noeuds cruciaux dans la chaîne d'approvisionnement mondiale. Elles n'ont cessé d'évoluer depuis plus de quarante ans. En raison de cette croissance, le nombre et la taille des véhicules utilisant des produits pétroliers de stations-service ont augmenté de façon significative. Par conséquent, les entreprises du secteur pétrolier y ont investi massivement. Pour tirer parti de ces investissements, ces entreprises cherchent à optimiser leurs opérations. Celles-ci comprennent le réapprovisionnement, le stockage et le transport depuis les dépôts jusqu'aux points de vente.

Dans cette thèse, nous proposons des méthodes de décomposition exactes pour résoudre le problème d'optimisation du réapprovisionnement des stations-service (PRP : *Petrol Replenishment Problem*). Il fait partie de la famille des problèmes des tournées de véhicules compartimentés (MCVRP : *Multi-Compartment Vehicle Routing Problem*) bien connue pour se montrer *NP-difficile*. Dans la littérature, les variantes du PRP sont souvent résolues par des méthodes heuristiques, alors que très peu de travaux ont été consacrés à la création de méthodes exactes. Le but de cette thèse consiste à combler cette lacune en utilisant des méthodes exactes basées principalement sur la décomposition de Benders (BD : *Benders Decomposition*) et la décomposition de Dantzig-Wolfe (la génération de colonnes (CG : *Column Generation*) dans un encadrement de BP : *Branch-and-Price*).

Dans une première contribution à cette thèse, nous traitons une variante riche du problème de réapprovisionnement de stations-service avec multi-dépôts et multi-périodes (MDMPRP : *Multi-Depot Multi-Period Petrol Replenishment Problem*), ce qui implique la planification des tournées de plusieurs camions-citernes fortement compartimentés (jusqu'à 15 compartiments), afin de livrer des produits pétroliers à des stations-service pendant plusieurs périodes. L'objectif est de minimiser les coûts de transport tout en respectant les contraintes de capacité des camions-citernes et de temps de livraison.

Afin de résoudre ce problème, nous avons engagé une nouvelle méthode primale de branchement en profondeur (PDH : *Primal Diving Heuristic*) efficace pour les problèmes de partitionnement d'ensembles peu ou moyennement denses. Cette méthode permet de résoudre à chaque noeud de branchement un problème plus petit et donc plus facile. Nous avons également utilisé une approche BP exacte pour évaluer la qualité des solutions générées par PDH. Celle-ci nous a permis de trouver des solutions optimales ou presque optimales.

Quant à la génération de colonnes, nous avons résolu des sous-problèmes, chacun constituant un problème de plus court chemin élémentaire, avec contraintes de ressources (ESPPRC : *Elementary Shortest Path Problem with Resource Constraints*), où des centaines de millions d’extensions d’étiquettes sont effectuées ; chaque extension exige la résolution d’un programme en nombre entiers mixtes (MILP : *Mixed-Integer Linear Program*) qui s’assure de la faisabilité du chargement des commandes des stations-service dans le camion citerne à chaque extension (un compartiment doit être affecté au plus à une seule station-service) appelé problème de chargement des camions-citernes (TTLP : *Tank Truck Loading Problem*). Cela rend la résolution de cette variante impossible par les méthodes standards. Afin d’y remédier, nous proposons une technique de hachage réduisant drastiquement le nombre de MILP à résoudre, en utilisant l’intuition voulant que plusieurs *patterns* de TTLP puissent se répéter dans la génération de certaines étiquettes.

Enfin, nous avons réalisé une étude de cas pour démontrer l’implantabilité de l’approche que nous avons proposée. Cette étude se distingue de ce qui se fait dans le contexte nord américain, dans lequel une tournée contient une ou deux stations-service, car on livre relativement de petites quantités, ce qui augmente le nombre de stations-service par tournée (jusqu’à 7 par tournée), rendant ainsi la résolution difficile dans un réseau cyclique. Les résultats ont montré que notre méthode s’avère capable de résoudre efficacement des instances de taille réelle, dans un temps de calcul raisonnable. Les techniques de branchement primal, BP et de génération de tournées nous ont permis d’obtenir des résultats de haute qualité, tout en garantissant la faisabilité et l’efficacité de la méthode. L’article intitulé "*Solving a real-world multi-depot multi-period petrol replenishment problem with complex loading constraints*" qui décrit cette contribution, a été publié dans la revue European Journal of Operational Research, en 2023.

En guise de deuxième contribution, nous étendons les techniques développées dans le premier objectif afin de résoudre le MDMPRP avec une gestion d’inventaire appelé (MDMPRPIM : *Multi-Depot Multi-Period Petrol Replenishment Problem with Inventory Management*). Nous introduisons aussi la gestion du stock hebdomadaire au niveau des dépôts. Dans un premier temps, nous utilisons une formulation compacte (MILP) intégrant à la fois le TTLP et le problème des tournées de véhicules avec gestion de stock (IRP : *Inventory Routing Problem*). Compte tenu de sa complexité, qui provient essentiellement de l’écart d’intégralité élevé, cette formulation demeure très difficile à résoudre par des solveurs commerciaux, même pour de petites instances. Nous créons ainsi une approche de résolution exacte en deux phases combinant la décomposition de BD et de BP. Le problème maître de BD décide du niveau des stocks dans les dépôts et les stations-service, alors que les sous-problèmes de BD cherchent à résoudre le problème de réapprovisionnement en tenant compte du niveau des stocks. Avec cette hybridation, le l’écart d’intégralité n’est plus problématique. Dans la première phase,

nous résolvons des sous-problèmes de BD relaxés (sans contraintes d'intégralité) jusqu'à ce que les niveaux de stocks se stabilisent. Dans la seconde phase, nous résolvons à l'optimalité les sous-problèmes de BD à l'aide d'un algorithme de BP. Nous améliorons notre approche par diverses stratégies d'accélération, y compris les coupes initiales, le parallélisme et PDH. Des résultats étendus sur des instances réelles mettent en évidence la force de notre approche. Nous parvenons à des solutions quasi-optimales dans la plupart des cas et nous constatons que les stratégies d'accélération augmentent considérablement les performances de l'approche en deux phases. Nous générons plusieurs aperçus managériaux qui soulignent les avantages obtenus grâce à l'optimisation intégrée. Enfin, l'article intitulé *"Combining Benders Decomposition and Column Generation for the Petrol Station Replenishment Problem with Inventory Management"* traitant cette contribution, a été soumis à la revue *Management Science*, en 2024.

Dans une troisième contribution, nous capitalisons les connaissances acquises dans les deux premiers articles, afin de répondre à un cas d'étude réel chez une entreprise de distribution de produits pétroliers souhaitant optimiser son réseau de distribution. Nous avons montré qu'en utilisant adéquatement notre outil d'aide à la décision, nous pouvons évaluer la politique de tarification, le design du réseau, par exemple, en augmentant de 15% la capacité des réservoirs, que nous pouvons réduire le coût de 354 K\$ par trimestre et que cela est plus intéressant que d'aller plus loin que 25%, que nous pouvons aussi diminuer le nombre de camions-citernes à un ensemble minimal sans trop impacter le coût total. Les résultats numériques sur un réseau réel (jusqu'à quatre dépôts, cinq types de produits pétroliers, quatre groupes principaux de stations-service et une flotte de 44 camions-citernes hétérogènes et fortement compartimentée) prouvent l'efficacité et le potentiel élevé de l'approche proposée. Ils mettent également en évidence l'impact de l'optimisation en Afrique et de la pratique du réapprovisionnement des stations-service sur le continent africain. Enfin, l'article intitulé *"The Petrol Station Replenishment Problem : A Case Study from West Africa"* résumant cette contribution a été soumis, à la revue *African Transport Studies*, en 2024.

Dans cette thèse, nous avons montré, avec des résultats très prometteurs, que les méthodes de décomposition fondées sur des approches exactes constituent des solutions viables pour résoudre le problème MDMPRPIM, et que les techniques créées deviennent génériques et peuvent être étendues à d'autres problèmes de la famille du MCVRP.

ABSTRACT

Petroleum is still a major source of energy in most modern societies, even though other alternative sources of energy (gas, electricity, solar) are increasingly used. One of the key success factors in the petroleum industry is the efficiency of its complex distribution system. However, the management of its different activities is a constant and a real challenge. These activities include procurement, transportation of petrol products from depots to petrol stations, and inventory management at both (depots and petrol stations) points.

In this thesis, we propose exact decomposition methods for solving the petrol station replenishment problems (PRP), which are often solved in the literature with heuristics, while very few works have used exact methods. We fill this gap by introducing exact methods based mainly on Benders decomposition, and column generation embedded in the *Branch-and-Price* framework.

In the first contribution, we address a new variant of PRP, which is a rich real-world multi-depot multi-period (MDMPRP). We show that it is possible to solve this complex variant with an exact *Branch-and-Price* approach and some derived heuristics. On one side, this problem could be modeled as a set partitioning type problem with low to moderate density (the number of ones per column, i.e., clients to visit, is not large). Such problems have some nice polyhedral properties to consider for favoring integrality. On the other side, some complex handling rules apply due to the problem's context. A natural way is to address them in the column generation subproblem as an elementary shortest path problem with resource constraints, which constitutes the major bottleneck. To succeed in this challenge, we need to design some sophisticated techniques i) for branching to profit from the polyhedral properties and ii) for solving the column generation subproblem. Direct use of on-the-shelf algorithms does not work, unfortunately. Numerical results on a real network (four depots, five types of petrol products, four main groups of clients, heterogeneous fleet of highly compartmented tank trucks) prove the effectiveness and high potential of the proposed approach.

In the second contribution, we extend the techniques developed to solve another variant of MDMPRP with the integration of inventory management (MDMPRPIM). We investigate a petrol station replenishment problem (PRP) that integrates the tank truck loading problem (TTLP) and inventory routing problem (IRP). We introduce a compact mixed integer linear program (MILP) formulation for the problem. Given its complexity, the compact formulation cannot be solved by general-purpose solvers, even for small instances. We develop an exact two-phase solution approach that combines Benders decomposition and column gen-

eration. In the first phase, we solve relaxed (integrality) Benders subproblems until the inventory levels stabilize. In the second phase, we solve to optimality Benders decomposition subproblems using a *Branch-and-Price* framework. We enhance our approach with various acceleration strategies, including warm-start, parallelism, and primal diving heuristics. Extensive computational results using real-world instances from one of the major companies in West Africa highlight the strength of our approach. We reach near-optimal solutions in most of these instances and note that acceleration strategies significantly boost the performance of our two-phase method. We generate several managerial insights that highlight the benefits achieved through integrated optimization.

In the third contribution, we capitalize on our experience through a comprehensive overview of our case study at a fast-growing petroleum company in West Africa. This study explains how we used the work done in the first and second contributions to help this petroleum company optimize its distribution network using operational research tools. We approach this study in two steps. In the first step, we use the BP method to solve the weekly case where each customer can be visited more than once within a specified interval of days within the horizon (one week). In the second step, we use Benders decomposition to solve the case with a horizon of up to 12 weeks. The numerical results on a real network (up to four depots, five types of petrol products, four main customer groups, and a fleet of 44 heterogeneous and highly compartmented tank trucks) prove the efficiency and high potential of the proposed approach. They also highlight the impact of optimization on Africa and the practice of refueling service stations on the African continent.

This thesis has successfully showcased the viability of decomposition methods that rely on exact approaches in addressing the MDMPRPIM problem. The results obtained from our research have been highly promising. Furthermore, the techniques developed during this study are not only applicable to the MDMPRPIM problem but can also be extended to other problems within the multi-compartment vehicle routing problem family.

TABLE DES MATIÈRES

DÉDICACE	iii
REMERCIEMENTS	iv
RÉSUMÉ	vi
ABSTRACT	ix
TABLE DES MATIÈRES	xi
LISTE DES TABLEAUX	xv
LISTE DES FIGURES	xvi
LISTE DES SIGLES ET ABRÉVIATIONS	xvii
LISTE DES ANNEXES	xviii
CHAPITRE 1 INTRODUCTION	1
1.1 Contexte d'application et concepts de base	1
1.2 Éléments de la problématique	3
1.3 Objectifs de recherche	4
1.4 Plan de la thèse	4
CHAPITRE 2 REVUE DE LITTÉRATURE	6
2.1 Les variantes du PRP	6
2.1.1 Le PRP standard	6
2.1.2 Le PRP avec périodes multiples	7
2.1.3 Le PRP avec fenêtres de temps	7
2.2 Le problème du chargement optimal du camion-citerne	9
2.3 La gestion d'inventaire	9
2.4 Les méthodes heuristiques	10
2.5 Les méthodes exactes	11
2.5.1 <i>Branch-and-Price</i>	12
2.5.2 La décomposition de Benders	13

2.5.3	La combinaison de la Décomposition de Benders et la Génération de Colonne	13
2.6	Le positionnement	14
CHAPITRE 3 ORGANISATION DU TRAVAIL		15
CHAPITRE 4 ARTICLE 1: SOLVING A REAL-WORLD MULTI-DEPOT MULTI-PERIOD PETROL REPLENISHMENT PROBLEM WITH COMPLEX LOADING CONSTRAINTS		17
4.1	Introduction	17
4.2	Literature Review	19
4.3	Problem Statement	21
4.4	Mathematical Model	22
4.4.1	Sets and Elements, Parameters and Variables	22
4.4.2	Network Representation	24
4.4.3	MDMPPRP Compact Formulation	25
4.5	Solution Method	26
4.5.1	The Master Problem	27
4.5.2	The Column Generation Subproblem	28
4.5.3	Subproblem Labeling Algorithm	30
4.5.4	Integer Solutions by Primal Diving Heuristic.	34
4.5.5	Acceleration Strategies	37
4.6	Case Study	39
4.6.1	Context	39
4.6.2	Instances characteristics	40
4.6.3	TTLP Solving	43
4.6.4	Performance Comparison Between PDH and DH	46
4.6.5	PDH Acceleration Strategies	48
4.6.6	Exact Branch-and-Price	48
4.6.7	Functional Results	52
4.6.8	Comparison with PC Solutions	52
4.7	Conclusion	54
CHAPITRE 5 ARTICLE 2: COMBINING BENDERS DECOMPOSITION AND COLUMN GENERATION FOR THE PETROL STATION REPLENISHMENT PROBLEM WITH INVENTORY MANAGEMENT		56
5.1	Introduction	56

5.1.1	Our Contribution	57
5.2	Literature Review	59
5.2.1	The Truck Loading Problem	59
5.2.2	The Inventory Routing Problem	60
5.2.3	The Petrol Station Replenishment Problem	60
5.2.4	Combining Benders Decomposition and Column Generation	62
5.2.5	Paper Positioning	62
5.3	Problem Description	64
5.4	Mathematical Formulation	65
5.4.1	Network Representation	65
5.4.2	The MDMPPSRPIM Formulation	67
5.5	Solution Methodology	69
5.5.1	The Route Formulation	70
5.5.2	The Benders Reformulation	71
5.5.3	Column and Benders Cut Generation	74
5.5.4	Acceleration Strategies	76
5.6	Computational Experiments	78
5.6.1	Case Context: West Africa	78
5.6.2	Experimental Design	79
5.6.3	Computational Results	80
5.6.4	Comparison with Other Approaches	87
5.6.5	Managerial Insights	89
5.7	Conclusion	92
CHAPITRE 6 ARTICLE 3: THE PETROL STATION REPLENISHMENT PROBLEM : A CASE STUDY FROM WEST AFRICA		94
6.1	Introduction	94
6.1.1	Contributions	94
6.2	Literature Review	95
6.2.1	Tank Truck Loading Problem	95
6.2.2	Inventory Routing Problem	96
6.2.3	Petrol Station Replenishment Problem	97
6.2.4	State of Practice	98
6.3	Problem	100
6.4	Method	103
6.4.1	Framework	103

6.4.2	First Step	104
6.4.3	Second Step	104
6.5	Results	105
6.5.1	Experimental Design	105
6.5.2	Computational Results	106
6.5.3	Discussion	112
6.6	Conclusion	113
CHAPITRE 7 DISCUSSION GÉNÉRALE		114
CHAPITRE 8 CONCLUSION		116
8.1	Synthèse des travaux	116
8.2	Limitations de la solution proposée et Améliorations futures	117
RÉFÉRENCES		118
ANNEXES		126

LISTE DES TABLEAUX

Table 4.1	Main Characteristics of Past Studies vs This Paper.	19
Table 4.2	Main Business Rules.	22
Table 4.3	Overview of the Set of Instances.	41
Table 4.4	Tank Trucks Properties.	43
Table 4.5	Results of PDH: TTLP Statistics.	44
Table 4.6	Results of PDH and DH: Runtime and Solution Quality.	47
Table 4.7	Results of PDH: Performance Statistics.	49
Table 4.8	Results of PDH and DH with Acceleration Strategies.	50
Table 4.9	Results of Exact Method.	51
Table 4.10	Results of PDH, DH and Exact Method: Number of Routes and Vehicles.	53
Table 4.11	Comparison MS and PDHA Solutions.	54
Table 5.1	Past Close Studies	63
Table 5.2	General Rules	65
Table 5.3	MDMPPSRPIM Notation	66
Table 5.4	West Africa-Specific Rules	78
Table 5.5	Instances Properties	79
Table 5.6	Tank Trucks Properties	80
Table 5.7	Benders Decomposition Results	82
Table 5.8	Column Generation Results	83
Table 5.9	Benders Master Problem Acceleration Strategies Impact	85
Table 5.10	Benders Subproblem Acceleration Strategies Impact	86
Table 5.11	Comparison with Heuristics 1 and 2	88
Table 6.1	Instances Properties	105
Table 6.2	Tank Trucks Properties	106
Table 6.3	Direct Billing vs Standard Billing	107
Table 6.4	Increase Petrol Stations' Tanks for Frequently Visited Petrol Stations	109
Table 6.5	Impact of Reducing Fleet Size on Total Cost	112

LISTE DES FIGURES

Figure 4.1	Part of the MDMPPRP Network.	23
Figure 4.2	Distribution of Feasible Routes on the Instances.	42
Figure 4.3	Impact of TTLP-Hashing on Instance W3	46
Figure 5.1	Illustration of a Petrol Station Replenishment Process	57
Figure 5.2	Part of the MDMPPSRPIM Network	67
Figure 5.3	Flowchart of the Two-Phase Approach	75
Figure 5.4	Flowchart of the Colum Generation Procedure	76
Figure 5.5	UB and LB Improvement Along With the Number of Benders Iterations for Instance P4-C77-D4-S2	84
Figure 5.6	Number of Routes and Trucks Used per Week P12-C77-D4-S1 . . .	89
Figure 5.7	Average Free Space Percentage Within Trucks per Week for All 48 Instances	90
Figure 5.8	Number of Visits Required and Performed for Each Client-Product for Instance P12-C77-D4-S1	91
Figure 5.9	Inventory Levels (Variation) and Buying Pulses for Instance P12-C77- D4-S1	92
Figure 6.1	PC Petrol Stations (Blue) and Depots (Black)	101
Figure 6.2	Two-Step Approach	104
Figure 6.3	Constant Prices/Costs	108
Figure 6.4	Variable Prices/Costs	108
Figure 6.5	Total Cost when Changing Petrol Stations' Tank Capacities	110
Figure 6.6	Minimal Number of Tank Truck Configurations for the Optimal Cost	111

LISTE DES SIGLES ET ABRÉVIATIONS

BB	Branch-and-Bound
BD	Benders Decomposition
BP	Branch-and-Price
CG	Column Generation
CGMP	Column Generation Master Problem
CGSP	Column Generation Sub Problem
ESPPRC	Elementary Shortest Path Problem with Resource Constraints
IRP	Inventory Routing Problem
MCVRP	Multi-Compartment Vehicle Routing Problem
MDMPPRP	Multi-Depot Multi-Period Petrol Replenishment Problem
MDMPPRPIM	Multi-Depot Multi-Period Petrol Replenishment Problem with Inventory Management
MDPRPTW	Multi-Depot Petrol Replenishment Problem with Time Windows
MILP	Mixed-Integer Linear Program
MPPRP	Multi-Period Petrol Replenishment Problem
PRPTW	Petrol Replenishment Problem with Time Windows
PC	Petroleum Company
PDH	Primal Diving Heuristic
PPRP	Periodic Petrol Replenishment Problem
PRP	Petrol Replenishment Problem
RP	Routing Problem
TTLP	Tank Truck Loading Problem
VNS	Variable Neighborhood Search
VMI	Vendor-Managed Inventory
VRP	Vehicle Routing Problem
VRPTW	Vehicle Routing Problem with Time Windows

LISTE DES ANNEXES

Annexe A		126
----------	-------	-----

CHAPITRE 1 INTRODUCTION

Le pétrole demeure une source d'énergie majeure dans la plupart des sociétés modernes, même si d'autres sources d'énergie alternatives (gaz, électricité et énergie solaire) deviennent de plus en plus utilisées. Les stations-service quant à elles constituent des noeuds primordiaux dans la chaîne d'approvisionnement mondiale. Elles n'ont cessé d'évoluer depuis plus de quarante ans. En raison de cette croissance, le nombre et la taille des véhicules utilisant les produits pétroliers distribués aux stations-service ont augmenté de façon significative. Par conséquent, les entreprises du secteur pétrolier ont investi massivement. Pour tirer parti de ces investissements, ces entreprises cherchent à optimiser leurs opérations. Celles-ci comprennent l'approvisionnement, le stockage et le transport depuis les dépôts jusqu'aux points de vente. Ce chapitre permet d'introduire le sujet de cette thèse, de présenter les concepts de base nécessaires à sa compréhension, d'exposer les problèmes qui y sont associés et leur contexte d'application, de détailler les objectifs de recherche à atteindre, et enfin de présenter le plan de la thèse.

1.1 Contexte d'application et concepts de base

Sur une base hebdomadaire, une société pétrolière (PC : *Petroleum Company*) d'Afrique de l'Ouest doit ravitailler un réseau de stations-service à partir de plusieurs dépôts, en utilisant une flotte hétérogène de camions-citernes compartimentés. Ces camions-citernes sont possédés par des entreprises de transport privées qui font également affaire avec certains concurrents de la PC. Afin de faciliter la compréhension de ce document, nous utiliserons la terminologie suivante :

- *Multi-périodes* : Sur un horizon de plusieurs semaines, les jours de la semaine considérée sont du lundi au dimanche, au cours de laquelle la PC doit approvisionner un réseau de stations-service. Chaque période correspond à exactement une journée.
- *Multi-produits* : La PC se charge de distribuer des produits pétroliers de différents types qui peuvent être désignés chimiquement comme produits pétroliers marqués (*Diesel* et *Marine Zoom* et produits pétroliers non marqués (*Super* et *Gasoil*). En effet, les produits pétroliers sont marqués pour détecter les fraudes, car l'état les subventionne.
- *Stations-service* : Le réseau de stations-service est principalement constitué de : (i) stations-service régulières appartenant à la PC ; (ii) stations marines situées dans certains ports de pêche ; (iii) boulangeries. Cependant, de temps en temps, la PC sert des clients occasionnels (entreprises privées). Des frais supplémentaires d'escorte douanière

s'appliquent pour les livraisons de carburant aux stations marines. Une réduction de ces frais est appliquée lorsque deux visites consécutives dans des stations marines différentes sont effectuées.

- *Flotte de camions-citernes* : La distribution des produits pétroliers marqués et non marqués est effectuée par un ensemble de camions-citernes hétérogènes compartimentés appartenant à quatre entreprises de transport privées. Nous repérons deux types de camions-citernes selon leur capacité : (i) les petits camions-citernes ayant une capacité < 20000 litres ; (ii) les camions-citernes *jumbo* ayant une capacité ≥ 20000 litres. Certains camions-citernes étiquetés sont utilisés exclusivement pour transporter un type spécifique de produits pétroliers. Notons qu'un camion-citerne peut être à la fois *jumbo* et étiqueté ou aucun des deux. Les camions-citernes peuvent ne pas être tous disponibles en même temps.
- *Déchargement de produits pétroliers* : La livraison consiste à décharger complètement un ou plusieurs compartiments, en utilisant la gravité, puisque les camions-citernes ne sont pas équipés d'un débitmètre.
- *Contrat de transport* : Les contrats commerciaux entre la PC et chaque entreprise de transport sont distincts. Différents coûts sont appliqués en fonction des quatre statuts des camions-citernes : (i) consacrés aux produits pétroliers marqués ; (ii) peints aux couleurs de la PC ; (iii) acceptant d'être partiellement payés en produits pétroliers dans une station-service de la PC ; (ix) camions-citernes réguliers.
- *Dépôts* : Avant de visiter les stations-service de la PC, les camions-citernes doivent récupérer les produits pétroliers dans les dépôts. Nous supposons que le volume total des produits pétroliers disponibles dans les dépôts est suffisamment important pour satisfaire la demande. L'inventaire des produits pétroliers au dépôt se fait par semaine.
- *Produit-Client* : Un seul produit destiné à une station-service est désigné par *produit-client*. Une station-service peut avoir plusieurs types de produits pétroliers, tandis que les stations marines et les boulangeries ne peuvent avoir qu'un seul type de produit pétrolier. Le temps de chargement à chaque dépôt, le temps d'attente et le temps de déchargement dans chaque station-service sont considérés comme constants et ne dépendent pas des quantités à charger ou à décharger.
- *Sloshing* (Ballotement) : Certaines règles complexes de manutention doivent être appliquées afin d'assurer l'équilibre de tout camion-citerne chargé pendant sa tournée, une attention particulière étant portée au remplissage des produits pétroliers dans les compartiments. Une partie du produit pétrolier transportée lors des tournées des camions-citernes se transforme en gaz, en raison du frottement causé par le liquide sur les parois des compartiments, ce qui provoque ainsi une instabilité du camion-citerne.

Ce phénomène est appelé *sloshing*. Il convient alors d'éviter autant que possible de laisser un espace vide dans un camion-citerne au début de sa tournée.

- *La facturation directe* : La facturation de chaque station-service approvisionnée s'appuie sur la distance se trouvant entre l'emplacement de la station-service et les dépôts tenant le produit pétrolier, même si le camion-citerne rend visite à d'autres stations-service sur la même route. La facturation d'une tournée à chaque station-service dépend également du coût au kilomètre par unité de volume défini pour chaque type de produit pétrolier et du volume du produit pétrolier transporté.

1.2 Éléments de la problématique

Le processus de réapprovisionnement d'une station-service peut se diviser en trois parties : le chargement, les tournées des véhicules et le déchargement. La route commence à partir d'un dépôt, dans lequel un camion-citerne compartimenté est chargé de produits pétroliers. Après le chargement, le camion-citerne visite certaines stations-service qui doivent être réapprovisionnées. Ces stations-service disposent généralement de réservoirs souterrains dans lesquels les produits pétroliers sont déchargés. Une fois la tournée terminée, le camion-citerne retourne au dépôt. Ce problème générique, qui est présenté dans de nombreuses variantes dans la littérature, est appelé : *problème de réapprovisionnement des stations-service* (PRP). Étant donné un ensemble de camions-citernes compartimentés, le PRP intègre deux principaux problèmes : le TTLP et le IRP.

Le TTLP vise à déterminer la quantité de chaque type de produit pétrolier pour chaque station-service, sans dépasser la capacité des compartiments et en satisfaisant simultanément la demande de chaque produit pétrolier à chaque station-service. Un camion-citerne peut transporter un ou plusieurs produits pétroliers, et chaque compartiment du camion-citerne peut être entièrement ou partiellement rempli que d'un seul produit pétrolier lorsque le camion-citerne quitte le dépôt. L'IRP cherche à optimiser à l'intérieur d'un horizon temporel donné le coût de stockage et de transport. La solution doit respecter la capacité et garantir que chaque station-service dispose, à chaque moment, d'une quantité suffisante pour satisfaire sa demande quotidienne.

Traditionnellement, le TTLP et l'IRP ont été abordés de manière séquentielle dans le PRP, le TTLP constituant une donnée d'entrée pour l'IRP. Cependant, cette approche peut conduire à des solutions sous-optimales et parfois non réalisables. En outre, si l'on considère la gestion des stocks dans les dépôts et dans les stations-service, le problème devient très difficile à être résolu à l'aide des méthodes exactes, c'est pourquoi des méthodes heuristiques sont souvent employés.

1.3 Objectifs de recherche

L'objectif ultime de cette thèse consiste à créer des méthodes de décomposition exactes, afin de résoudre une variante complexe et réaliste du PRP pour une grande flotte de camions-citernes très compartimentée (jusqu'à 15 compartiments). Notre étude comporte trois objectifs :

Comme premier objectif, nous résolvons le MDMPPRP hebdomadaire en utilisant la CG combinée à une nouvelle méthode primale de branchement en profondeur. Des stratégies d'accélération telles que des techniques de hachages, de parallélisme et de resserrement de bornes ont rendu possible la résolution efficace de ce problème, en un temps raisonnable et avec une garantie de la qualité de la solution obtenue. Nous résolvons d'une façon exacte des instances de taille moyenne et les plus grandes avec un petit écart d'optimalité.

Pour le deuxième objectif, nous proposons une approche progressive fondée sur la CG et la BD pour résoudre le MDMPPRPIM. À notre connaissance, il s'agit de la première tentative de résoudre le MDMPPRPIM par des méthodes exactes qui combinent la BD et la CG. La BD permet de découpler les décisions de gestion des stocks et de tournées des véhicules. Ensuite, nous pouvons résoudre les problèmes hebdomadaires en utilisant la technique créée dans la première étape. Des expériences approfondies démontrent l'efficacité de l'approche en deux étapes.

Comme troisième objectif, nous discutons de la mise en oeuvre de notre approche progressive dans le contexte africain afin de mener des expériences approfondies, mettant en lumière les points suivants : i) la comparaison entre la politique de facturation directe et la politique de facturation standard, ii) l'impact des fluctuations des prix du pétrole sur les décisions d'achat et de stockage les plus rentables pour la PC. iii) Il existe des capacités de réservoir optimales aux stations-service, et vi) parmi tous les types de camions-citernes disponibles, pour faciliter la gestion de la flotte des camions-citernes, on peut réduire le nombre de configurations à un ensemble minimal, sans trop impacter le coût total.

1.4 Plan de la thèse

Le plan de cette thèse se structure de la façon qui suit. La revue de la littérature est présentée dans le chapitre 2. Le chapitre 3 décrit brièvement l'organisation des principaux chapitres. Le chapitre 4 décrit le premier objectif qui s'intéresse à la résolution du MDMPPRP. Le chapitre 5 présente le deuxième objectif qui s'intéresse à la résolution du MDMPPRPIM. Le chapitre 6 est consacré au dernier objectif de cette thèse portant sur l'étude de cas. Le chapitre 7 entame une discussion globale synthétisant nos contributions. Enfin, les conclusions, les limitations

et les futures pistes de recherches clôturent cet ouvrage ; elles seront présentées au chapitre 8.

CHAPITRE 2 REVUE DE LITTÉRATURE

La revue de littérature porte sur le problème du PRP et ses variantes (voir Section 2.1) ainsi que sur les méthodes de résolution (voir Sections 2.4 et 2.5). Le positionnement de la thèse s'effectuera à la Section 2.6.

2.1 Les variantes du PRP

Au cours des trois dernières décennies, de nombreuses variantes du PRP ont été étudiées dans la littérature. Les premiers travaux ont été menés par Brown & Graves (1981), qui ont proposé un modèle MILP pour la répartition en temps réel de camions-citernes à partir de plusieurs dépôts. Pour obtenir une vue d'ensemble complète de l'histoire du PRP, nous invitons le lecteur à consulter les articles suivants : Cornillier et al. (2012); Benantar et al. (2016) et le mémoire de maîtrise Boers (2020).

Nous avons classé les variantes du PRP selon cinq variantes fondées sur trois caractéristiques, à savoir le nombre de périodes, le nombre de dépôts et la prise en compte ou non des fenêtres de temps. Les variantes sont les suivantes :

- Le PRP standard avec un dépôt, une seule période et sans fenêtres de temps (Brown & Graves, 1981);
- Le PRP avec multi-périodes, appelé MPPRP : *Multi-Period Petrol Replenishment Problem* (Triki, 2013; Archetti et al., 2015a));
- Le PRP avec fenêtres de temps, appelé PRPTW : *Petrol Replenishment Problem with Time Windows* (Cornillier et al., 2009);
- Le PRP avec multi-dépôts et fenêtres de temps, appelé MDPRTW : *Multi-Depot Petrol Replenishment Problem with Time Windows* (Cornillier et al., 2012);
- Le PRP avec multi-dépôts et multi-périodes, appelé MDMPRP (Carotenuto et al., 2018).

Il est à noter que Carotenuto et al. (2018) ont abordé le MDMPRP en considérant un seul produit pétrolier et des camions-citernes homogènes d'une capacité fixe.

2.1.1 Le PRP standard

Pour le PRP standard, Cornillier et al. (2008a) ont créé un algorithme exact. Celui-ci décompose ce problème en un problème de tournées des véhicules et un TTLP. Le nombre d'arrêts par tournée a été limité à deux, pour obtenir une solution.

Dans le modèle MILP de Popović et al. (2011), le PRP minimise à la fois les coûts de stockage et ceux du transport sur plusieurs périodes. Le nombre de stations-service qui peuvent être visitées dans une tournée est limité à trois, et une station-service ne peut être visitée que par un seul camion-citerne par période. En utilisant un taux de consommation fixé pour chaque réservoir, la demande est considérée comme déterministe.

Des méthodes heuristiques, telles que la recherche par voisinage variable (VNS : *Variable Neighborhood Search*), ont été proposées par Popović et al. (2012) pour résoudre le PRP avec des camions-citernes homogènes. De plus, Vidović et al. (2014) ont présenté deux modèles MILP pour minimiser les coûts de stockage et de transport en utilisant une flotte homogène de camions-citernes. Li et al. (2014) ont exposé une variante du PRP standard avec une flotte fixe de camions-citernes homogènes.

2.1.2 Le PRP avec périodes multiples

Une des caractéristiques clés des PRPs, est leur horizon multi-périodes, qui permet de résoudre ce problème pendant une durée de plusieurs jours plutôt qu'un seul. Cette approche présente l'avantage de prendre en compte la question du réapprovisionnement régulier des stations-service, ce qui conduit à des solutions plus efficaces. Une classification du MPPRP a été proposée par Cornillier et al. (2008b), et a ensuite été étendue par Triki (2013) pour inclure le problème de réapprovisionnement périodique des stations-service (PPRP : *Periodic Petrol Replenishment Problem*). Celui-ci vise à optimiser la livraison des produits pétroliers en déterminant le jour optimal pour visiter une station-service, tout en minimisant la distance totale de déplacement. Des formulations mathématiques ont été proposées pour le PPRP par Triki & Al-Hinai (2016), et ont été étendues par Al-Hinai & Triki (2020) pour comprendre le choix des fréquences de réapprovisionnement.

2.1.3 Le PRP avec fenêtres de temps

Plusieurs chercheurs se sont penchés sur le problème de la gestion des stations-service avec des fenêtres de temps, pendant lesquelles une station-service peut être réapprovisionnée. Ce problème peut être considéré comme un problème des tournées des véhicules avec fenêtre de temps (VRPTW : *Vehicle Routing Problem with Time Windows*). Dans leur étude, Benantar et al. (2016) ont examiné ce problème en prenant en compte des camions-citernes compartimentés pour une seule journée. Les quantités à livrer ont été déterminées en fonction de la demande prévue pour cette journée. Pour résoudre ce problème, les chercheurs ont utilisé un algorithme de recherche tabou qui définissait l'ordre de visite des stations-service comme voisinage.

Une variante du problème VRPTW consistant à permettre l'utilisation multiple des camions-citernes a été proposée par Macedo et al. (2011), ce qui signifie que chaque véhicule peut effectuer plusieurs tournées pendant une période de temps donnée. Dans une étude menée par Azi et al. (2010), cette problématique a été examinée en utilisant une flotte homogène de camions-citernes ayant une capacité fixe. Les chercheurs ont proposé un modèle mathématique s'appuyant sur un MILP et ont démontré comment résoudre le problème en utilisant un algorithme de BP. Cette approche permet de maximiser l'efficacité de la flotte de camions-citernes en optimisant leur utilisation, tout en respectant les contraintes de temps imposées par les stations-service.

Dans une étude de cas menée par Triki (2013), des cas réels ont été utilisés pour évaluer l'efficacité des méthodes de résolution proposées par les auteurs pour résoudre le PPRP. Les résultats ont démontré qu'en comparaison avec une planification effectuée par un opérateur humain, une économie de 17.7% a été réalisée en utilisant l'heuristique la plus efficace présentée par ces auteurs.

Dans une autre étude de cas menée aux États-Unis, Kazemi & Szmerekovsky (2015) ont utilisé une modélisation par MILP pour déterminer la conception optimale de la chaîne d'approvisionnement, en prenant en compte les méthodes de transport multimodal lors de la définition des emplacements des installations. Cette approche a permis de résoudre efficacement le problème de distribution d'une PC algérienne, comme l'ont démontré Benantar et al. (2016) dans leur étude. En utilisant un modèle pour le PRP avec des camions-citernes et des fenêtres de temps, les auteurs ont pu surpasser la solution existante de la PC, en réduisant à la fois le nombre de camions-citernes utilisés et la distance totale de déplacement. Cette étude de cas démontre ainsi l'efficacité de cette méthode dans l'optimisation des opérations logistiques pour les PCs américaines.

Dans une autre étude menée par Triki et al. (2016), le PPRP a été étudié pour une PC à Oman dans le but d'évaluer les performances d'un solveur utilisé pour trouver une solution exacte. Les résultats ont montré que le solveur était capable de trouver une solution pour un maximum de 22 stations-service. Cette modélisation sera ultérieurement utilisée pour préparer une offre pour des enchères d'approvisionnement en transport. Ces études de cas démontrent l'efficacité des méthodes de résolution proposées pour le PPRP et soulignent leur potentiel pour optimiser la planification et les opérations dans le domaine de la distribution des produits pétroliers.

2.2 Le problème du chargement optimal du camion-citerne

Dans le cadre du PRP, des camions-citernes sont utilisés pour distribuer les produits pétroliers. Un seul type de produit pétrolier peut être chargé dans le même compartiment, en raison de contraintes d'incompatibilité (Kaabi & Jabeur, 2015). Coelho & Laporte (2015) ont présenté différents modèles MILP pour ce problème, que les auteurs ont classé en livraisons *fractionnée* ou *non fractionnée*. La livraison fractionnée signifie que le contenu d'un compartiment peut être réparti entre plusieurs stations-service, tandis que la livraison non fractionnée veut dire que le contenu complet du compartiment doit être livré à une station-service. En outre, les chercheurs supposaient souvent que les camions-citernes n'étaient pas équipés de débitmètres, ce qui impliquait que le contenu complet d'un compartiment devait être vidé à une station-service (Macedo et al., 2011; Cornillier et al., 2008a,b; Benantar et al., 2016; Popović et al., 2012; Vidović et al., 2014). L'absence de débitmètre limite la flexibilité d'utilisation des camions-citernes, alors qu'il est aujourd'hui possible d'équiper les camions-citernes de débitmètres (Coelho & Laporte, 2015). Wang et al. (2020) ont introduit un TTLP différent pour le PRP avec des compartiments semblables. Ceux-ci sont équipés de débitmètres. Cette caractéristique permet d'affecter chaque compartiment à plusieurs commandes du même produit pétrolier. Cela facilite énormément le problème.

2.3 La gestion d'inventaire

Dans le domaine de la gestion de la chaîne d'approvisionnement, les coûts de transport et d'inventaire jouent un rôle majeur dans la rentabilité des PCs. Cependant, il est intéressant de noter que les approches traditionnelles de modélisation de la chaîne d'approvisionnement ont tendance à considérer ces deux aspects de manière indépendante. Pourtant, il existe une interdépendance évidente entre les décisions d'inventaire et des tournées des véhicules. En prenant en compte simultanément ces deux problèmes, il est possible d'obtenir de meilleures solutions, ce qui se traduit par une amélioration de la satisfaction des stations-service et une réduction des coûts de distribution, grâce à une utilisation plus efficace des camions-citernes disponibles.

Une étude menée par Vidović et al. (2014) met en évidence l'importance de cette approche intégrée. En utilisant un modèle MILP, les chercheurs ont pu résoudre simultanément les problèmes d'inventaire et ceux des tournées des véhicules pour une entreprise de logistique. Les résultats ont montré que cette approche permettait d'optimiser les décisions de stockage et de livraison, conduisant ainsi à une meilleure satisfaction des stations-service tout en réduisant les coûts de distribution. Cette étude de cas a donc démontré l'importance de

considérer de manière conjointe les décisions d’inventaire et de tournées de véhicules dans la gestion de la chaîne d’approvisionnement.

Les problèmes impliquant la prise de décisions concernant la gestion des stocks, le tournées des camions-citernes et l’ordonnancement peuvent être classés comme un IRP. Ces problèmes supposent l’application du concept d’inventaire géré par le fournisseur (VMI : *Vendor-Managed Inventory*), ce qui signifie que le fournisseur possède un accès en temps réel aux niveaux de l’inventaire et prend des décisions simultanées sur le moment et sur les quantités à livrer à chaque station-service, ainsi que sur les tournées à emprunter pour les servir. Cette stratégie présente des avantages tant pour le fournisseur que pour les vendeurs, car elle permet de minimiser les coûts globaux de distribution et de libérer les ressources des vendeurs qui n’ont plus désormais à se préoccuper de la gestion des stocks (Coelho et al., 2014).

Différentes politiques ont été étudiées dans la littérature pour déterminer les quantités à livrer aux stations-service. Parmi ces politiques, nous pouvons mentionner la politique *Maximum-Level* et la politique *Order-Up-to-Level*. Dans le cadre de la politique du *Maximum-Level*, la quantité commandée est flexible, mais elle respecte la capacité maximale de la station-service. Cette approche permet de maintenir un certain niveau de stock tout en évitant les surcharges. En revanche, la politique *Order-Up-to-Level* utilise une quantité de commandes fixes qui permet de maintenir le stock à un certain niveau prédéterminé.

Dans la littérature, la politique *Maximum-Level* s’avère la plus couramment utilisée pour le PRP. Des chercheurs tels que Cornillier et al. (2008a); Popović et al. (2011) ont appliqué cette politique en fixant des limites aux quantités considérées dans la livraison et en s’appuyant sur la capacité du réservoir de la station-service. D’autres chercheurs, comme Li et al. (2014), ont opté pour une politique *Order-Up-to-Level*. Cette approche permet de maximiser l’utilisation du réservoir, tout en évitant des coûts d’entreposage inutiles.

2.4 Les méthodes heuristiques

La prise en compte d’un plus grand nombre de caractéristiques réelles dans un modèle MILP pour le PRP entraîne pour conséquence de rendre le modèle encore plus complexe. Plusieurs études ont démontré cette difficulté, notamment celles menées par Li et al. (2014); Cornillier et al. (2008a); Vidović et al. (2014); Kaabi & Jabeur (2015); Al-Hinai & Triki (2020). Les matheuristiques se sont révélées efficaces pour résoudre des PRP complexes (Boers, 2020). Ces méthodes combinent des techniques de programmation mathématique et des méthodes heuristiques (Boschetti et al., 2009). Une autre méthode de résolution pour le PPRP a été proposée par Carotenuto et al. (2016), dans laquelle les auteurs ont utilisé un algorithme

génétique hybride pour résoudre le modèle de tournées des véhicules.

Une autre contribution importante dans le domaine de la gestion des stocks a été apportée par Cornillier et al. (2008a), qui a proposé une décomposition du problème en un TTLP et un problème RP : *Routing Problem*. L'objectif principal du TTLP consistait à maximiser la quantité totale de produits pétroliers livrés, tout en tenant compte des contraintes de capacité des camions-citernes et des réservoirs souterrains des stations-service. Une approche semblable a été présentée par Cornillier et al. (2008b) dans son heuristique étendue. En construisant le *planning* par jour et en décalant le jour de visite des stations-service, les auteurs ont réussi à améliorer considérablement les performances de leur heuristique. De plus, les deux approches heuristiques discutées dans ce paragraphe ont la particularité de limiter le nombre d'arrêts par tournée à un maximum de deux, ce qui simplifie grandement le problème.

Le modèle MILP inventé par Vidović et al. (2014) offre une solution innovante pour résoudre le PRP, multi-produits et multi-périodes avec gestion des stocks. Les auteurs ont adopté une approche de décomposition du problème en un problème d'inventaire et un problème des tournées des véhicules, permettant ainsi une résolution plus efficace. L'heuristique a permis de trouver une solution initiale, qui a ensuite été améliorée grâce à une procédure de recherche locale. Celle-ci repose sur la répartition du contenu du compartiment d'un camion-citerne entre les jours de la planification. Cette approche astucieuse permet d'optimiser les tournées et de maximiser l'efficacité de la livraison.

Il est intéressant de noter que cette même heuristique a été utilisée par Popović et al. (2012). Cette approche, combinée à un déplacement des compartiments, permet d'obtenir des résultats encore plus efficaces. Une simplification, fondée sur l'hypothèse que seuls les compartiments pleins peuvent être livrés aux stations-service, permet une résolution plus rapide et plus efficace, tout en garantissant des résultats satisfaisants.

2.5 Les méthodes exactes

Très peu d'articles ont tenté de fournir des modèles/méthodes de résolution exacts, même pour résoudre de petites instances de PRP. Benantar et al. (2016) ont résolu de petites instances (15 stations-service, deux types de produits pétroliers et deux compartiments) de MPPRP avec le solveur (ILOG, 2021). Cornillier et al. (2012) ont proposé un modèle exact de MDPRTW qui sélectionne, parmi un ensemble de trajets réalisables (un trajet est une tournée associée à un camion-citerne), le sous-ensemble qui satisfait la demande tout en maximisant le revenu journalier net. Comme le nombre de ces déplacements est souvent très important,

les auteurs ont proposé une approche heuristique alternative.

2.5.1 *Branch-and-Price*

La méthode de BP est une approche utilisée en optimisation, afin de résoudre des problèmes combinatoires complexes impliquant un grand nombre de variables et de contraintes. Elle résulte de la combinaison de deux méthodes distinctes, à savoir la méthode de BB : *Branch-and-Bound* (Morrison et al., 2016) et la méthode de CG (Desrochers et al., 1992; Desaulniers et al., 2006; Lübbecke & Desrosiers, 2005).

Dans la méthode de BB, le problème se subdivise en sous-problèmes plus petits grâce à un branchement sur une variable, et la solution de chaque sous-problème produit est bornée. Lorsque la borne d'un sous-problème s'avère moins favorable que la meilleure solution connue, cette branche peut être élaguée, et la recherche se poursuit dans d'autres branches. Diverses stratégies avancées d'exploration de l'arbre de branchement sont exposées dans la revue de Morrison et al. (2016).

Quant à la CG, elle représente une technique puissante et flexible pour résoudre des problèmes d'optimisation linéaire complexes. Cette méthode permet d'obtenir des solutions plus efficaces et plus précises, et même dans des cas difficiles, tout en réduisant le temps de calcul et les coûts associés à la résolution du problème. La CG s'applique dans le cadre de la programmation mathématique pour résoudre des problèmes d'optimisation linéaire complexes. Elle consiste à générer de nouvelles variables, appelées colonnes, au fur et à mesure que le programme linéaire est résolu, afin d'affiner la solution et d'améliorer l'efficacité du processus de résolution.

La méthode de CG s'avère particulièrement utile dans les situations dans lesquelles le nombre de variables se montre très important et où le problème ne peut être résolu de manière efficace par les techniques traditionnelles. Elle trouve fréquemment des applications dans les domaines du transport, de la production, de la planification, ainsi que dans les secteurs de la finance, de la logistique et de la distribution. En pratique, la CG implique la résolution itérative d'un sous-problème appelé problème maître (CGMP : *Column Generation Master Problem*). Ce problème maître utilise une formulation réduite du problème original, ne contenant qu'un sous-ensemble des variables possibles. Par la suite, un ou plusieurs sous-problèmes, appelé problème auxiliaire (CGSP : *Column Generation Sub Problem*), est résolu pour générer de nouvelles colonnes. Celes-ci sont ensuite ajoutées au problème maître, et ainsi le processus se répète jusqu'à ce que la solution optimale soit atteinte.

Cornillier et al. (2008a) ont décomposé le PRP standard en utilisant un schéma de CG *a priori* (autorisant seulement deux stations-service par tournée) en deux sous-problèmes : un

TTLP utilisé pour assigner des demandes aux compartiments en maximisant le volume des produits pétroliers livrés aux stations-service ; un RP utilisé pour sélectionner des tournées qui minimisent les coûts de transport globaux. Avella et al. (2004) ont proposé un algorithme de BP pour le PRP standard dans lequel, ils considèrent plusieurs stations-service par tournée. Les auteurs ont utilisé une approche heuristique afin de fournir un ensemble initial de colonnes pour démarrer l'algorithme de BP. Leur approche est testée sur des données réelles composées de 25 stations-service et de six camions-citernes de trois types différents, chacun possédant trois compartiments.

2.5.2 La décomposition de Benders

La méthode de BD, proposée par Benders (1962), est une technique algorithmique utilisée pour résoudre des problèmes d'optimisation combinatoire. Elle vise à décomposer un problème complexe en sous-problèmes plus simples, facilitant ainsi sa résolution. La méthode de BD a été appliquée pour résoudre des problèmes d'optimisation à grande échelle (Fischetti et al., 2017) avec des variables complexes, qui, lorsqu'elles sont temporairement fixées, produisent des sous-problèmes beaucoup plus faciles à résoudre. Une revue complète de cette méthode est disponible dans l'article de Rahmaniani et al. (2017).

Compte tenu de sa pertinence pratique, la BD a été appliquée dans de nombreux domaines, notamment pour la planification de la production (Adulyasak et al., 2015), celle de la maintenance (Canto, 2008), celle des soins de santé (Lin et al., 2016), la planification des compagnies aériennes (Zeighami & Soumis, 2019; Cordeau et al., 2001b), la gestion des ressources en eau (Cai et al., 2001), l'emplacement des hubs (Contreras et al., 2011), l'affectation des locomotives (Cordeau et al., 2001a), le voyageur de commerce (Laporte et al., 1994) et l'expansion de la capacité (Bloom, 1983). Pour le PRP, aucune étude pour le résoudre avec BD n'a été effectuée.

2.5.3 La combinaison de la Décomposition de Benders et la Génération de Colonnes

Dans les problèmes d'optimisation à grande échelle, la combinaison de la BD et de la CG apparaît comme une nouvelle approche exploitant les structures des problèmes primal et dual. Cordeau et al. (2001b) l'ont utilisé pour résoudre simultanément les problèmes des tournées des avions et d'affectation d'équipages. Le problème des tournées des avions constitue le problème maître de BD, alors que le problème d'affectation d'équipage devient le sous-problème de la BD. Les deux problèmes seront ensuite résolus à l'aide de la CG. Les auteurs ont introduit une approche en trois phases. Dans la première phase, les restrictions d'intégralité

sont relâchées. Dans la deuxième phase, elles sont ajoutées au problème maître de la BD. Et dans la troisième phase, les restrictions d'intégralité sont ajoutées au problème maître de la BD et au sous-problème. Zeighami & Soumis (2019) combinent la BD et la CG pour résoudre les problèmes de jumelage et d'affectation d'équipages dans la planification des compagnies aériennes. Le problème de jumelage d'équipage représente le problème maître de la BD, tandis que les sous-problèmes des (co)pilotes représentent les sous-problèmes de la BD, c'est-à-dire un problème pour les pilotes et un pour les copilotes. À notre connaissance, aucun article dans la littérature n'utilise cette combinaison pour résoudre le PRP. De plus, le caractère unique du MDMPPRPIM ne permet pas d'applications directes de cette combinaison.

2.6 Le positionnement

À la conclusion de cette revue, notre objectif était de préciser notre problématique et d'orienter nos contributions. Les points clés à retenir se présentent de la manière suivante :

- À notre connaissance, aucun article dans la littérature n'aborde de manière exacte le MDMPPRPIM et le MDMPPRP intégré (c'est-à-dire le TTLP et le RP en même temps) en utilisant un algorithme de BP avec plusieurs stations-service par tournée.
- Les méthodes heuristiques sont les techniques prédominantes, ce qui implique l'absence de garantie quant à la qualité des solutions.
- Les articles les plus proches des nôtres sont ceux de Cornillier et al. (2008b) et de Vidović et al. (2014). Cornillier et al. (2008b) abordent un problème semblable avec un entrepôt et deux stations-service par tournée en utilisant une heuristique multi-phases. Vidović et al. (2014) emploient deux heuristiques constructives pour résoudre le MDMPPRPIM pendant cinq jours, avec des camions homogènes à quatre compartiments. Ils ne tiennent pas compte de l'inventaire de l'entrepôt. Étant donné que nous découplons l'inventaire et l'acheminement, ces heuristiques, que nous améliorons, peuvent être utilisées comme coupes initiales dans notre approche exacte.

CHAPITRE 3 ORGANISATION DU TRAVAIL

D’après la revue de la littérature décrite dans le chapitre 2, nous pouvons déduire que le MDMPPRP et le MDMPPRPIM sont souvent résolus dans la littérature par des heuristiques, alors que très peu de travaux ont utilisé des méthodes exactes. Le but ultime de cette thèse vise le développement de méthodes exactes fondées principalement sur la combinaison de la BD et de la CG.

Dans le chapitre 4, nous examinons le MDMPPRP hebdomadaire. Nous supposons que les quantités de produits à livrer aux clients sont des multiples de 1000 L et les périodes de livraison sont connues avant la planification. Nous montrons qu’on peut résoudre cette variante complexe en utilisant la BP. D’un côté, ce problème peut être formulé comme un problème de partitionnement d’ensemble avec une densité de faible à modérée. Ces problèmes présentent des propriétés polyédrales intéressantes favorisant l’intégralité. D’un autre côté, des règles de manipulation complexes (comme un seul client-produit par compartiment) sont nécessaires, en raison du contexte spécifique du problème. Une approche naturelle consiste à les traiter dans le sous-problème de la CG, formulé comme le problème du plus court chemin élémentaire, avec des contraintes de ressources, constituant ainsi le principal goulot d’étranglement. Pour relever ce défi, nous avons développé des techniques sophistiquées, à savoir : i) des stratégies de branchement exploitant les propriétés polyédrales du problème ; ii) des méthodes d’accélération du sous-problème de CG.

Dans le chapitre 5, nous utilisons les techniques développées dans le chapitre 4 pour résoudre le MDMPPRPIM avec gestion hebdomadaire du stock dans les dépôts. Les quantités de produits à livrer aux clients et les périodes de livraisons deviennent des variables de décisions. Le modèle MILP que nous proposons pour ce problème, intègre le TTLP et IRP. Vu sa complexité, c’est impossible de le résoudre directement avec des solveurs commerciaux. Nous proposons une approche de résolution exacte en deux phases qui combine la BD et la BP. Nous résolvons des sous-problèmes de BD jusqu’à ce que les niveaux de stocks se stabilisent. Ensuite, nous résolvons à l’optimalité les sous-problèmes de BD avec BP. Nous développons un ensemble de stratégies d’accélération, incluant le *warm start*, le parallélisme et PDH. Des résultats numériques étendus utilisant des instances réelles mettent en évidence la force de notre approche. Nous obtenons des solutions optimales dans la plupart de ces cas et nous constatons que les stratégies d’accélération augmentent considérablement les performances de cette méthode en deux phases. Nous générons plusieurs aperçus managériaux qui soulignent les avantages obtenus grâce à l’optimisation intégrée.

Pour résoudre la problématique exposée dans cette thèse, nous avons utilisé des données réelles provenant d'une entreprise de distribution de produits pétroliers de l'Afrique de l'Ouest. Le troisième objectif de cette thèse, présentée dans le chapitre 6, consiste en une étude de cas réel combinant les méthodes exposées dans les chapitres 4 et 5, pour mettre en lumière les particularités du MDMPPRPIM dans le contexte africain. En utilisant notre outil d'aide à la décision développé dans les articles précédents, nous conduisons des expériences approfondies pour : i) effectuer une comparaison entre la politique de facturation directe et la politique de facturation standard ; ii) évaluer l'impact des scénarios de fluctuations des prix du pétrole sur les décisions d'achat et de stockage ; iii) montrer aussi qu'il existe des capacités de réservoirs optimales aux stations-service ; iv) faciliter la gestion de la flotte des camions-citernes en réduisant le nombre de configurations à un ensemble minimal sans trop impacter le coût total. Les résultats numériques obtenus sur un réseau réel (comprenant jusqu'à quatre dépôts, cinq types de produits pétroliers, quatre groupes principaux de stations-service, ainsi qu'une flotte de 44 camions-citernes hétérogènes et fortement compartimentés) confirment l'efficacité et le potentiel élevé de l'approche que nous proposons.

CHAPITRE 4 ARTICLE 1: SOLVING A REAL-WORLD MULTI-DEPOT MULTI-PERIOD PETROL REPLENISHMENT PROBLEM WITH COMPLEX LOADING CONSTRAINTS

A. Bani, I. El Hallaoui, A. I. Corr  a et A. Tahir ont   crit cet article, soumis le 4 novembre 2020 et publi   le 24 avril 2023 dans la revue *European Journal of Operational Research*.

4.1 Introduction

Petroleum is still a major source of energy in most modern societies even though other alternative sources of energy (gas, electricity, solar) are increasingly used. One of the key success factors in the petroleum industry is the efficiency of its complex distribution system. However, the sound management of its different activities is a constant and real challenge. These activities include procurement, transportation of petroleum products from depots to sales points and inventory management at both (depots and sales) points.

In this paper, we propose to solve a new complex variant of the petrol replenishment problem (PRP), integrating the tank truck loading problem (TTLP) and the routing problem (RP). The integrated problem consists in replenishing a network of distinct clients with different types of petroleum products from a set of depots by a heterogeneous fleet of multi-compartment tank trucks. Several handling rules apply due to collective agreements and the hazardous nature of the carried products. This problem is motivated by a real case faced by a West Africa based petroleum company. The main attributes of the problem are:

1. Rich vehicle routing problem (VRP) with multiple depots and multiple (subsidized and unsubsidized) products.
2. Limited number of capacitated vehicles consisting of a heterogeneous fleet of tank trucks with a number of variable size compartments (ranging from three to fifteen adding complexity to the problem).
3. Cyclic network on a weekly (multi-period) horizon and many clients per route. Each client can be visited more than once in a specified range of days within the horizon. From one planning period (horizon or instance) to another, the number and type of clients to be serviced may vary. Thus, we do not have a rolling horizon.
4. Distinct subfamilies of clients. The network comprises in our case: (i) petrol stations, (ii) marine stations, (iii) private bakeries, and (iv) occasional private companies.

These characteristics, often studied separately, make this variant complex and uncommon in

the literature. This variant is much harder than those addressed in the literature. In North America for instance, a tank truck serves generally only one or two clients per route, the number of compartments does not exceed three in general, and the number of product types is smaller too (e.g., no subsidized products). Most of the studies presented in the literature propose metaheuristic-based approaches or *a priori* column generation to solve the PRP, due to the fact that i) metaheuristics are easier to implement than exact methods, which is true in the case of the branch-and-price method for example; and ii) exact methods might not be appropriate to solve large scale problems (see, Laporte (1992)), which is not completely true. We show in this paper that it is possible, to some extent, to solve this complex problem with an exact branch-and-price approach and some derived heuristics. The PRP could be modeled as a set partitioning type problem with low to moderate density (the number of ones per column, i.e., clients to visit is not large). Such problems have some nice polyhedral properties to consider for favoring integrality. This challenge needs the design of some sophisticated techniques i) for branching to profit from these properties and ii) for solving the column generation subproblem. Direct use of on-the-shelf algorithms, like the one from the Boost library, does not work, unfortunately.

Our main contributions are the following:

- New complex variant of the petrol replenishment problem with multiple depots and multiple periods with heterogenous (compartmented) fleet and multiple products.
- Introduction of a new primal branch-and-price diving heuristic appropriate for low to moderately dense set partitioning type problems;
- Exact branch-and-price approach to assess the quality of the heuristic solutions;
- Efficient generation of routes (columns) with very complex loading constraints thanks to a hashing technique and other acceleration strategies;
- A case study showing the implementability of the proposed approach. Extensive numerical results show that our approach produces better quality solutions (having an average 0.31% of integrality gap in less than 3.5 minutes) than a diving branch-and-price heuristic usually used in practice and the manual solutions provided by the petroleum company (PC).

This paper is organized as follows: after the introduction (Section 4.1), we review the most significant contributions related to PRP in Section 4.2. In Section 4.3, we give a detailed description of the problem. A compact mathematical model is defined in Section 4.4, followed by a description of the proposed solution method (in Section 4.5). In Section 4.6, we present and discuss the computational results obtained on real-world instances provided by the PC. We then conclude and provide some future directions of research (in Section 4.7).

4.2 Literature Review

The PRP is among the most studied variants of the multi-compartment vehicle routing problem (MCVRP), see Derigs et al. (2011). The MCVRP is a generalization of the classical vehicle routing problem. It presents the following features: (i) it considers demand for multiple heterogeneous products instead of a single product; (ii) a vehicle has multiple compartments with different configurations and sizes instead of a single one; (iii) all products delivered on a route must be assigned to compartments of a vehicle; (iv) some products must not be loaded together in the same vehicle. A comprehensive overview of the history of the PRP can be found in the papers published by Cornillier et al. (2012) and Benantar et al. (2016). Relevant and recent literature is summarized in Table 4.1.

Table 4.1 Main Characteristics of Past Studies vs This Paper.

Paper	Limited Fleet	Hetero. Fleet	# Clients/route	Multi Depots	Time Windows	Multi-Periods	Multi Visits/day	Exact Method	Meta/Heuristic	Matheuristic
Brown & Graves (1981);	✓	✓	1	✓					✓	
Van der Bruggen et al. (1995)										
Brown et al. (1987)	✓	✓	≥ 1	✓					✓	
Ronen (1995)	✓	✓	1							
Malépart et al. (2003)	✓	✓	≥ 1						✓	
Abdelaziz et al. (2002)	✓	✓	≤ 2						✓	
Cornillier et al. (2008b)	✓	✓	≤ 2			✓			✓	
Cornillier et al. (2009)	✓	✓	≥ 1		✓					✓
Popović et al. (2011)	✓	✓	≤ 2			✓			✓	
Hanczar (2012)	✓	✓	≥ 1			✓			✓	
Cornillier et al. (2012)	✓	✓	≥ 1	✓	✓				✓	✓
Popović et al. (2012)		✓	≤ 3			✓			✓	
Vidović et al. (2014)		✓	≤ 4			✓				✓
Charusakwong & Lohatepanont (2016)	✓	✓	≤ 2		✓	✓	✓		✓	
Avella et al. (2004)	✓	✓	≥ 1					✓	✓	
Taqi'Allah et al. (2000)	✓	✓	$1/\geq 1$			✓		✓		
Ng et al. (2008)	✓	✓	≥ 1					✓		
Cornillier et al. (2008a)		✓	≤ 2					✓		
Benantar et al. (2016)	✓	✓	≥ 1		✓		✓	✓	✓	
Archetti et al. (2015b)	✓		≥ 1			✓		✓		
Triki (2013)	✓	✓	≥ 1			✓			✓	
Carotenuto et al. (2018)	✓		≥ 1	✓		✓			✓	
Our paper	✓	✓	≥ 1	✓		✓	✓	✓		✓

PRP variants. We can classify them into five variants according to three important features: multiple periods, multiple depots, time windows. The variants are i) standard PRP where we have one depot, a single-period and no time windows (Brown & Graves (1981)), ii) MPPRP, a multi-period PRP (Malépart et al. (2003), Triki (2013) and Archetti et al. (2015b)), iii) PRPTW, a PRP with time windows (Cornillier et al. (2009)), iv) MDPRTW, a multi-depot PRPTW (Cornillier et al. (2012)), v) MDMPPRP, a multi-depot MPPRP. Carotenuto et al. (2018) have tackled the MDMPPRP by considering a single product and a homogeneous tank trucks of given capacity. In the present paper, we introduce a more complex variant with multiple products and heterogeneous fleet of multi-compartmented tank trucks.

Heuristic/Matheuristic approaches. Most of the papers surveyed proposed heuristic or metaheuristic approaches to solve the PRP variants, due to the fact that the exact methods are more difficult to implement and also, they might not be appropriate to solve some large scale instances (see Laporte (1992)). The frequently used metaheuristics are: cluster first (based on the geographic limitation and expert considerations), route second; adapted VNS introduced by Hansen & Mladenović (2001); load first (assign orders to trucks), route second; tailored heuristics. In general, neither lower bounds or optimal solutions are known for the problems addressed and therefore, the quality of the obtained solutions cannot be assessed.

Exact models/approaches. Very few papers attempted exact models/solution methods even for solving small instances of PRP. Cornillier et al. (2008a) decomposed the standard PRP using *a priori* column generation scheme (allowing just two clients per route) into two subproblems: the tank truck loading problem (TTLP) and the routing problem (RP). The TTLP is used to assign orders to compartments maximizing the profit. RP is used to select routes that minimize overall transportation costs. Avella et al. (2004) proposed a branch-and-price algorithm for the standard PRP, where we allow multiple clients per route. Authors used a heuristic approach to provide an initial set of columns to start up the branch-and-price algorithm. Their approach is tested on real-world data consisting of 25 clients and 6 tank trucks of 3 different types with three compartments. Benantar et al. (2016) solved small instances (15 clients, 2 product types and 2 compartments) of MPPRP by CPLEX. Cornillier et al. (2012) proposed an exact model for MDPRTW that selects, among a set of feasible trips (a trip is a route associated with a truck), the subset that satisfies the demand while maximizing overall daily net revenue. Since the number of such trips is often very huge, the authors proposed an alternative heuristic approach. As we can see, none of these studies solved the complex integrated MDMPPRP (i.e., the TTLP and RP at once) using

a branch-and-price algorithm with several clients per route. This is our contribution in this paper.

4.3 Problem Statement

The MDMPPRP is defined over a one-week planning horizon (from Monday to Sunday), in which the PC must replenish a network of clients with different types of marked and unmarked petroleum products. In fact, some products are partially subsidized by the government and for this reason, they are chemically marked to detect fraud. Customs escort fees apply to petroleum products supplied to marine stations. A discount on these fees is granted when two marine stations are visited consecutively by the same tank truck during a trip. The idea is to encourage delivery under the supervision of the same Customs officer to reduce the number of supervision requests.

The distribution of the marked and unmarked products is carried out by a pool of heterogeneous fleet of tank trucks. We distinguish between two types of tank trucks (small and large), depending on their capacities. Some tank trucks are exclusively used to carry the marked products, while others can carry marked and unmarked products. The tank trucks may be not available all at the same time. Each tank truck is compartmented, and each compartment has constant capacity. For the remainder of this article, we will assume that the quantity available at the depots is always greater than the total demand.

An order placed by a client consists of a specific product denoted by *client-product*, its associated requested volume and a due date (the clients prefer actually be delivered during a certain range of days). A client may order one or several types of petroleum products. The orders are known prior to the week of delivery. The loading time at each depot, the waiting time and the unloading time at each client are considered constant and slightly overestimated to obtain solutions implementable in practice.

Some handling rules must be enforced (see Table 4.2). Furthermore, a certain percentage of the petroleum products carried during the trip transform into gas as a result of the friction caused by the liquid on the walls of the compartments, thus changing the shape and the volume of the compartments. This phenomenon is called *sloshing*. So, empty space in a tank truck should be avoided as much as possible.

The MDMPPRP consists in constructing a weekly transportation plan, specifying the types and volumes of the petroleum products, the tank trucks, the compartments used, and the delivery times. In addition to the pick-up periods at the depot, the exact volumes of petroleum products and the compartments filled in every tank truck must be provided. The objective function includes the following cost terms: (i) the sum of the fixed costs of using tank trucks; (ii) the sum of the Customs escort fees; (iii) the total distance traveled; (iv) a penalization

Table 4.2 Main Business Rules.

Rule	Description
$\mathcal{R}_1$	Marked and unmarked products cannot be loaded on the same tank truck (exclusion constraint).
$\mathcal{R}_2$	Some clients cannot accommodate large tank trucks due to accessibility constraints.
$\mathcal{R}_3$	Some tank trucks must only carry marked products.
$\mathcal{R}_4$	Demand satisfaction: each client-product must be served by a single tank truck.
$\mathcal{R}_5$	The quantities of products at the depots are limited.
$\mathcal{R}_6$	When a compartment of a tank truck is used, it must be completely filled.
$\mathcal{R}_7$	Only one client-product per compartment is permitted.
$\mathcal{R}_8$	We can only deliver what we collect from the depot.
$\mathcal{R}_9$	The daily total travel time must not exceed the total working time in a day.
$\mathcal{R}_{10}$	A truck cannot visit more than two marine stations in a row.
$\mathcal{R}_{11}$	The delivery should be done within $[a_i, b_i]$ specified by a client i (i.e., from day a_i to day b_i of the horizon) during which the client can be delivered at any time of the day.

of the empty space left in the tank trucks.

4.4 Mathematical Model

We first introduce some notations, followed by the network representation. Thereafter, we give a mixed-integer programming formulation along with an explanation of each constraint. We suppose without loss of generality that this problem is integer feasible and bounded, which is the case in practice.

4.4.1 Sets and Elements, Parameters and Variables

The sets and elements, parameters and decision variables of the proposed model are defined as follows:

Sets and elements

- $\mathbf{P}$: Set of all products;
- $\mathbf{N}$: Set of all client-product nodes: each client-product node corresponds to an order of one product by a client;
- $\mathbf{N}_p$: Subset of client-products ordering product $p \in \mathbf{P}$;
- $\mathbf{N}_f$: Subset of client-products related to marine stations;
- $\mathbf{E}$: Set of depots;
- $\mathbf{V}'$: Set of dummy nodes;
- $\mathbf{V} = \{\mathbf{N} \cup \mathbf{E} \cup \{\sigma, \delta\}\} \cup \mathbf{V}'$: Set of all nodes of the network including the source and destination nodes in addition to the dummy nodes;
- $\mathbf{S}$: Subset of vertices in the sub-tour;

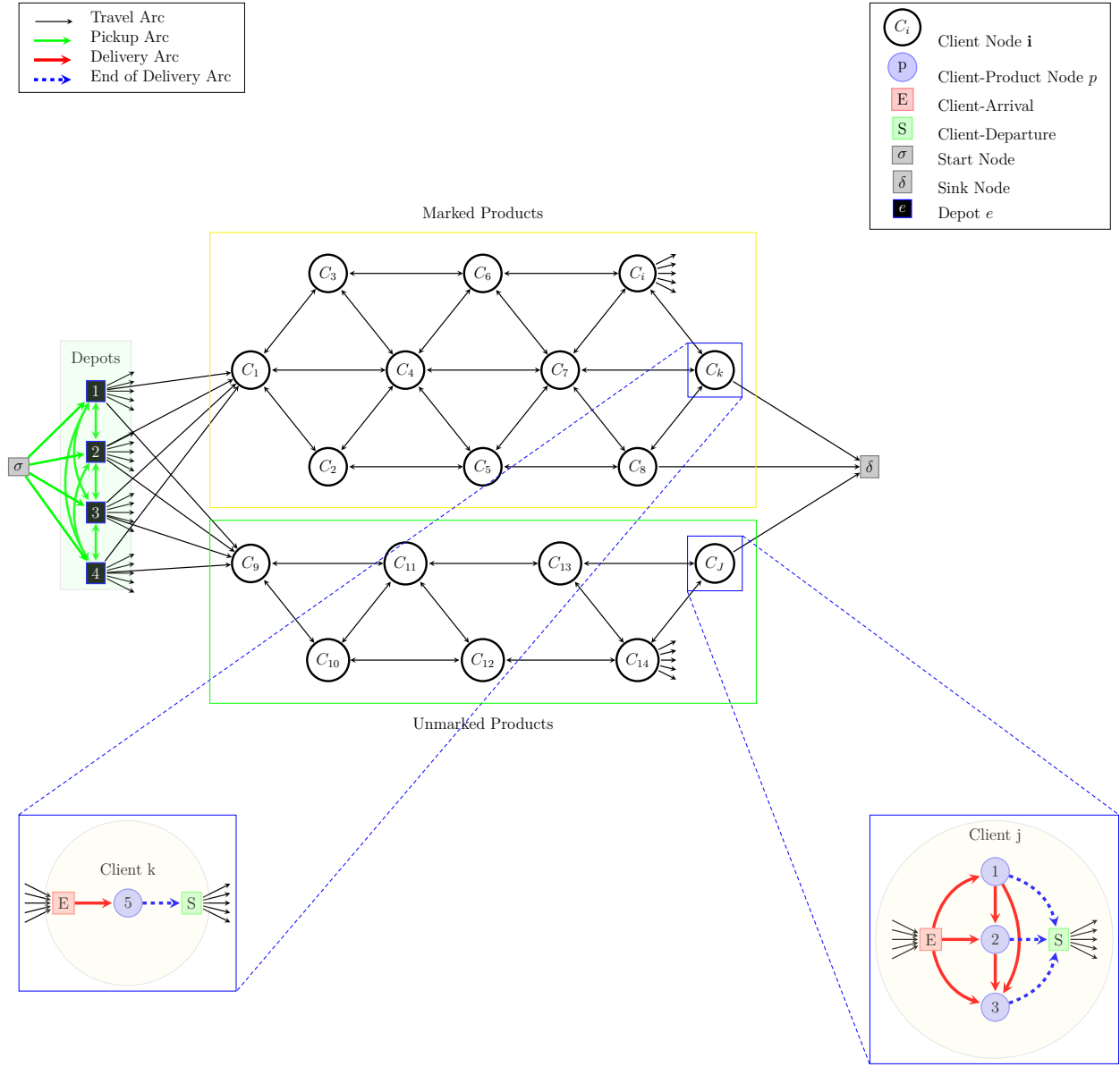


Figure 4.1 Part of the MDMPPRP Network.

- $\mathbf{A}$: Set of possible connections (arcs). Note that the Rule $\mathcal{R}_1$ is respected in the connection definition;
- $\mathbf{K}$: Set of tank trucks;
- $\mathbf{D}$: Set of days in a week (numbered from 0 to 6);
- $\mathbf{D}_k$: Set of availability days for each tank truck $k \in \mathbf{K}$;
- $\mathbf{L}_k$: Set of compartments of each tank truck $k \in \mathbf{K}$;
- $\mathbf{C}$: Set of compatible arc-truck-day combinations. A combination (i, j, k, d) is *com-*

patible if the tank truck k can visit both nodes i and j consecutively in day d without breaking the rules $\mathcal{R}_2$, $\mathcal{R}_3$ and $\mathcal{R}_{11}$.

Parameters

- Q_{pe} : Quantity of product p available at depot e ;
- C_{lk} : Capacity of compartment l in tank truck k ;
- c_{ij} : Cost of visiting node j after node i , including distance and Customs escort fees;
- τ_{ij} : Travel time from node i to node j , including the waiting time and the unloading time of the products;
- T_{max} : Length of a working day;
- o_n : Quantity of product in client-product $n \in \mathbf{N}$;
- Ψ_k : Penalty value of using tank truck k ;
- β : Penalty for leaving an empty space in the tank truck;
- Δ_k : Capacity of tank truck $k \in \mathbf{K}$, also equal to $\sum_{l \in \mathbf{L}_k} C_{lk}$.

Decision variables

- $u_{kd} = 1$ if a tank truck k is used on day d , 0 otherwise.
- $x_{ij}^{kd} = 1$ if $(i, j, k, d) \in \mathbf{C}$ and the tank truck k visits node j on day d after visiting the node i , 0 otherwise.
- $y_{ln}^{kd} = 1$ if the product in client-product n is loaded in compartment l of tank truck k on day d , 0 otherwise.
- $q_{pe}^{kd} \geq 0$ quantity of product p picked-up from depot e by tank truck k on day d .

4.4.2 Network Representation

Let $\mathbf{G} = (\mathbf{V}, \mathbf{A})$ be a directed multi-graph, where $\mathbf{V}$ is the node set and $\mathbf{A}$ the arc set. The network $\mathbf{G}$ is partially illustrated in Figure 4.1, where for clarity reasons, some arcs are truncated or omitted. In addition to one source node (σ) and one destination node (δ), this network contains five node types: *depot*, *client*, *client-departure*, *client-arrival* and *client-product*. Each client-product node $\mathbf{N}$ represents a product $p \in \mathbf{P}$ requested by a client. Each client has a pair of dummy nodes: the client-departure and client-arrival nodes. A client-arrival node represents the entry to the client site, and a client-departure node represents the exit from the client site. We denote by $\mathbf{V}'$ the subset of these dummy nodes.

The network involves four arc types connecting the different nodes mentioned above: *travel arc*, *pickup arc*, *delivery arc* and *end of delivery arc*. The *travel arcs* link the client-departure node of each client to the client-arrival node of other clients, also the depot nodes to client-

arrival nodes. The *pickup arcs* connect the source node with the depots and the depots between them. *Delivery arcs* link the client-arrival node with the client-product nodes of the same client. The arcs connecting the client-product nodes of a client with the client-departure node are the *end of delivery arcs*. It should be noted that the graph above contains two disjoint sub-graphs: the sub-graph of unmarked products and the sub-graph of marked products.

Our problem can be modeled as a large-scale mixed integer program using the following notation. Although this formulation is too large to be used in practice, it is helpful to understand the problem and the related decomposition method that follows.

4.4.3 MDMPPRP Compact Formulation

Given this notation, the problem, integrating TTLP and RP, can be modeled as follows:

$$\min_{x,y,u,q} \sum_{k \in \mathbf{K}} \sum_{d \in \mathbf{D}_k} (\Psi_k u_{kd}) + \left(\sum_{(i,j) \in \mathbf{A}} c_{ij} x_{ij}^{kd} \right) + \beta \left(\Delta_k u_{kd} - \sum_{l \in \mathbf{L}_k} \sum_{n \in \mathbf{N}} C_{lk} y_{ln}^{kd} \right) \quad (4.1)$$

$$\text{s.t.:} \quad \sum_{(i,n) \in \mathbf{A}} \sum_{k \in \mathbf{K}} \sum_{d \in \mathbf{D}_k} x_{in}^{kd} = 1 \quad \forall n \in \mathbf{N} \quad (4.2)$$

$$\sum_{k \in \mathbf{K}} \sum_{d \in \mathbf{D}_k} q_{pe}^{kd} \leq Q_{pe} \quad \forall p \in \mathbf{P}, \forall e \in \mathbf{E} \quad (4.3)$$

$$q_{pe}^{kd} \leq Q_{pe} \sum_{i \in \{\mathbf{E} \setminus \{e\}\} \cup \{\sigma\}} x_{ie}^{kd} \quad \forall p \in \mathbf{P}, \forall e \in \mathbf{E}, \forall k \in \mathbf{K}, \forall d \in \mathbf{D}_k \quad (4.4)$$

$$x_{ij}^{kd} \leq u_{kd} \quad \forall (i,j) \in \mathbf{A}, \forall k \in \mathbf{K}, \forall d \in \mathbf{D}_k \quad (4.5)$$

$$\sum_{l \in \mathbf{L}_k} C_{lk} y_{ln}^{kd} = o_n \sum_{(i,n) \in \mathbf{A}} x_{in}^{kd} \quad \forall n \in \mathbf{N}, \forall k \in \mathbf{K}, \forall d \in \mathbf{D}_k \quad (4.6)$$

$$\sum_{n \in \mathbf{N}} y_{ln}^{kd} \leq 1 \quad \forall k \in \mathbf{K}, \forall d \in \mathbf{D}_k, \forall l \in \mathbf{L}_k \quad (4.7)$$

$$\sum_{e \in \mathbf{E}} q_{pe}^{kd} = \sum_{n \in \mathbf{N}_p} \sum_{l \in \mathbf{L}_k} C_{lk} y_{ln}^{kd} \quad \forall p \in \mathbf{P}, \forall k \in \mathbf{K}, \forall d \in \mathbf{D}_k \quad (4.8)$$

$$\sum_{(i,j) \in \mathbf{A}} \tau_{ij} x_{ij}^{kd} \leq T_{max} \quad \forall k \in \mathbf{K}, \forall d \in \mathbf{D}_k \quad (4.9)$$

$$\sum_{(i,j) \in \{\mathbf{N} \times \mathbf{N}_f\}} x_{ij}^{kd} \leq 2 \quad \forall k \in \mathbf{K}, \forall d \in \mathbf{D}_k \quad (4.10)$$

$$\sum_{i \in \mathbf{V}} x_{ij}^{kd} - \sum_{i \in \mathbf{V}} x_{ji}^{kd} = \begin{cases} -1 & \text{if } j = \sigma \\ 0 & \text{if } j \in \mathbf{V} \setminus \{\sigma, \delta\} \\ 1 & \text{if } j = \delta \end{cases} \quad \forall k \in \mathbf{K}, \forall d \in \mathbf{D}_k \quad (4.11)$$

$$\sum_{i,j \in \mathbf{S}} x_{ij}^{kd} \leq |\mathbf{S}| - 1 \quad \forall k \in \mathbf{K}, \forall d \in \mathbf{D}_k \quad (\mathbf{S} \subset \mathbf{V}, 2 \leq |\mathbf{S}| \leq |\mathbf{V}| - 2) \quad (4.12)$$

$$x_{ij}^{kd} \in \{0, 1\} \quad \forall (i, j, k, d) \in \mathbf{C} \quad (4.13)$$

$$u_{kd} \in \{0, 1\} \quad \forall k \in \mathbf{K}, \forall d \in \mathbf{D}_k \quad (4.14)$$

$$y_{ln}^{kd} \in \{0, 1\} \quad \forall n \in \mathbf{N}, \forall k \in \mathbf{K}, \forall d \in \mathbf{D}_k, \forall l \in \mathbf{L}_k \quad (4.15)$$

$$q_{pe}^{kd} \geq 0 \quad \forall k \in \mathbf{K}, \forall d \in \mathbf{D}_k, \forall p \in \mathbf{P}, \forall e \in \mathbf{E} \quad (4.16)$$

The objective function (4.1) aims at minimizing the total cost, including the penalty cost of using tank trucks (first term), the total distance, unloading cost and Customs escort fees (second term). The total cost also includes the empty space in the tank trucks that we used (third term). Constraints (4.2) indicate that each client-product must be visited exactly once, as stipulated by Rule $\mathcal{R}_4$. The Rule $\mathcal{R}_5$ on quantities available at the depot is ensured by constraints (4.3). Constraints (4.4) ensure that a tank truck can only load products from the depot after visiting it. Constraints (4.5) ensure that a tank truck can be used only if available on day d . Constraints (4.6) indicate that the sum of the capacities of the compartments in which we want to put the order of client-product n must correspond exactly to the quantity demanded, as introduced in Rule $\mathcal{R}_6$. Constraints (4.7) impose that for each tank truck k , at most one client-product is allowed in the same compartment on a given day d ($\mathcal{R}_7$). These constraints with the constraints 4.6 define TTLP, the part of the problem concerning the loading of the tank trucks. Constraints (4.8) specify that the quantities of each product delivered to clients must be loaded from depots, as mentioned in Rule $\mathcal{R}_8$. Constraints (4.9) make sure that the daily total travel time must fall within the length of a working day for each tank truck driver, as requested by Rule $\mathcal{R}_9$. Constraints (4.10) state that a tank truck cannot visit more than two marine stations, as imposed by Rule $\mathcal{R}_{10}$. Constraints (4.11) and (4.12) are the flow conservation constraints and sub-tour elimination constraints, respectively. Finally, the domain of the decision variables are restricted by (4.13)–(4.16).

4.5 Solution Method

We propose a new branch-and-price heuristic to solve this hard problem, where each node in the branch-and-bound tree is evaluated using column generation (CG). CG is closely connected to Dantzig & Wolfe (1960) decomposition. It involves reformulating the problem as a restricted master problem (RMP) and one or more column generation subproblems (CGSPs). To perform CG, we reformulate the MDMPRP compact formulation described in Section 4.4.3 as a set partitioning master problem (MP) in Section 4.5.1 and define the CGSPs in Section 4.5.2. CG is an iterative approach where in every iteration, we consider only a subset of all possible columns (or routes) in the MP; that is why it is known as restricted MP. The RMP guides the CGSPs to generate new promising columns (routes) by providing them with a dual solution at each iteration. The CG procedure continues until no negative reduced cost columns (using this dual solution) are generated. To solve the CGSPs, we use a dynamic labeling algorithm given in Section 4.5.3. Finally, Section 4.5.4 outlines

how we obtain integer solutions.

4.5.1 The Master Problem

We reformulate the MDMPPRP as a set-partitioning model using Dantzig–Wolfe Decomposition Dantzig & Wolfe (1960). For this purpose, let $\Omega = \{1, \dots, |\Omega|\}$ be the set of feasible routes (columns). A binary variable ρ_ω is associated with each route $\omega \in \Omega$. ρ_ω takes the value 1 if route ω with cost c_ω is selected or 0 otherwise, where c_ω is the sum of Customs escort fees, penalties, and travel cost.

The coefficient $s_{n\omega}$ is set to 1 if the tank truck running the route $\omega \in \Omega$ serves the client-product $n \in N$, and 0 otherwise. The coefficient $u_{kd\omega}$ is equal to 1 if the route ω is served by tank truck k in day d , and 0 otherwise. The coefficient $q_{pe\omega}$ is the quantity of product p loaded from depot e to satisfy the order for product p on route ω . The mathematical formulation of our master problem is as follows:

$$\min_{\rho} \quad Z(\rho) = \sum_{\omega \in \Omega} c_\omega \rho_\omega \quad (4.17)$$

$$\text{s.t.:} \quad \sum_{\omega \in \Omega} s_{n\omega} \rho_\omega = 1 \quad \forall n \in N \quad (4.18)$$

$$\sum_{\omega \in \Omega} u_{kd\omega} \rho_\omega \leq 1 \quad \forall k \in K, \forall d \in D_k \quad (4.19)$$

$$\sum_{\omega \in \Omega} q_{pe\omega} \rho_\omega \leq Q_{pe} \quad \forall p \in P, \forall e \in E \quad (4.20)$$

$$\rho_\omega \in \{0, 1\} \quad \forall \omega \in \Omega \quad (4.21)$$

The objective (4.17) is to minimize the total cost Z . The set partitioning constraints (4.18) indicate that each client-product must be visited exactly once; note that we may perform multiple visits per day if we visit two client-products of the same client on the same day. The additional constraints (4.19) ensure that we can use tank truck k only once per period d . The additional constraints (4.20) ensure that the quantities loaded from each depot do not exceed what is available. Finally, integrality constraints (4.21) are imposed on the variables ρ_ω . These last constraints are relaxed when the CG is used to find the lower bound in a branch-and-bound node.

In summary, in the master problem, we manage two constraints of the original formulation: the demand satisfaction requirement (4.2) and the availability of products at the depots (4.3). The remaining constraints (4.4)–(4.16) are managed in the CGSPs.

4.5.2 The Column Generation Subproblem

To ease the understanding of this section, we add a superscript kd to a set $\mathbf{E}$ ($\mathbf{E}^{kd} \subseteq \mathbf{E}$) to designate the subset of its elements related to tank truck k and day $d \in \mathbf{D}_k$. Since constraints (4.4)–(4.16) are decomposed by tank truck k and day d , we modeled our pricing subproblems CGSPs as elementary shortest path problems with resource constraints (ESPPRC). It consists of a connected graph $\mathbf{G}^{kd}(\mathbf{V}^{kd}, \mathbf{A}^{kd})$; this subset includes nodes $\sigma, \delta, \mathbf{N}^{kd}, \mathbf{N}_p^{kd}, \mathbf{N}_f^{kd}$ and depot nodes. On the other hand, $\mathbf{A}^{kd} \subseteq \mathbf{A}$ is the subset of arcs induced by $\mathbf{V}^{kd}$. The ESPPRC finds a least negative reduced cost route among all the routes from σ to δ that satisfy the resource constraints imposed by the set of rules $(\mathcal{R}_5)$ – $(\mathcal{R}_{10})$.

Every resource-feasible route from σ to δ in the ESPPRC corresponds to a feasible route $\omega \in \mathbf{\Omega}$. The cost of this route is the sum of the costs of its arcs $\mathbf{A}^\omega$ and is equal to the corresponding route cost c_ω . However, in a CGSP, the arc costs need to be modified because the role of a CGSP is to find negative reduced cost routes, if at least one exists. Hence, the cost of a route should be changed to the reduced cost of the corresponding route variable. Let $\pi_n, \gamma_{dk}, \theta_{pe}$ be the dual variables associated with the master problem constraints (4.18), (4.19), (4.20), respectively. We must define the appropriate arc reduced cost $\bar{c}_{ij}$ for each arc $(i, j) \in \mathbf{A}$. To this end, we define four sets of arcs:

- $\mathbf{A}^\omega$: Set of arcs of route ω .
- $\mathbf{A}_n^u$: Set of unloading arcs that correspond to serving client-product n .
- $\mathbf{A}_{pe}^l$: Set of loading arcs that correspond to picking up a quantity q_{pe} of product p from depot e .
- $\mathbf{A}^o$: Set of other arcs.

The reduced cost $\bar{c}_\omega$ of a variable ρ_ω is then given by:

$$\bar{c}_\omega = c_\omega - \left(\sum_{n \in \mathbf{N}} s_{n\omega} \pi_n \right) - \left(\sum_{k \in \mathbf{K}} \sum_{d \in \mathbf{D}_k} u_{kd\omega} \gamma_{dk} \right) - \left(\sum_{p \in \mathbf{P}} \sum_{e \in \mathbf{E}} q_{pe\omega} \theta_{pe} \right) = \sum_{(i,j) \in \mathbf{A}^\omega} \bar{c}_{ij} \quad (4.22)$$

In this case, the (reduced) cost $\bar{c}_{ij}$ of arc $(i, j) \in \mathbf{A}^\omega$ is given by:

$$\bar{c}_{ij} = \begin{cases} c_{ij} - \pi_n & \text{if } (i, j) \in \mathbf{A}_n^u \\ c_{ij} - q_{pe} \theta_{pe} & \text{if } (i, j) \in \mathbf{A}_{pe}^l \\ c_{ij} & \text{if } (i, j) \in \mathbf{A}^o \end{cases} \quad (4.23)$$

The CGSP ^{kd} is formulated as follows:

$$\min_{x,y} \quad (\Psi_k - \gamma_{dk}) + \left(\sum_{(i,j) \in \mathbf{A}} \bar{c}_{ij} x_{ij}^{kd} \right) + \left(\Delta_k - \sum_{l \in \mathbf{L}_k} \sum_{n \in \mathbf{N}^{kd}} C_{lk} y_{ln}^{kd} \right) \quad (4.24)$$

$$\text{s.t.: } q_{pe}^{kd} \leq Q_{pe} \sum_{i \in \{E \setminus \{e\}\} \cup \sigma} x_{ie}^{kd} \quad \forall p \in \mathbf{P}, \forall e \in \mathbf{E} \quad (4.25)$$

$$\sum_{l \in \mathbf{L}_k} C_{lk} y_{ln}^{kd} = o_n \sum_{(i,n) \in \mathbf{A}^{kd}} x_{in}^{kd} \quad \forall n \in \mathbf{N}^{kd} \quad (4.26)$$

$$\sum_{n \in \mathbf{N}} y_{ln}^{kd} \leq 1 \quad \forall l \in \mathbf{L}_k \quad (4.27)$$

$$\sum_{e \in \mathbf{E}} q_{pe}^{kd} = \sum_{n \in \mathbf{N}_p^{kd}} \sum_{l \in \mathbf{L}_k} C_{lk} y_{ln}^{kd} \quad \forall p \in \mathbf{P} \quad (4.28)$$

$$\sum_{(i,j) \in \mathbf{A}^{kd}} \tau_{ij} x_{ij}^{kd} \leq T_{max} \quad (4.29)$$

$$\sum_{(i,j) \in \{\mathbf{N}^{kd} \times \mathbf{N}_f^{kd}\}} x_{ij}^{kd} \leq 2 \quad (4.30)$$

$$\sum_{i \in \mathbf{V}^{kd}} x_{ij}^{kd} - \sum_{i \in \mathbf{V}^{kd}} x_{ji}^{kd} = \begin{cases} -1 & \text{if } j = \sigma \\ 0 & \text{if } j \in \mathbf{V}^{kd} \setminus \{\sigma, \delta\} \\ 1 & \text{if } j = \delta \end{cases} \quad (4.31)$$

$$\sum_{i,j \in \mathbf{S}} x_{ij}^{kd} \leq |\mathbf{S}| - 1 \quad (\mathbf{S} \subset \mathbf{V}^{kd}, 2 \leq |\mathbf{S}| \leq |\mathbf{V}^{kd}| - 2) \quad (4.32)$$

$$y_{ln}^{kd} \in \{0, 1\} \quad \forall n \in \mathbf{N}^{kd}, \forall l \in \mathbf{L}_k \quad (4.33)$$

$$x_{ij}^{kd} \in \{0, 1\} \quad \forall (i, j) \in \mathbf{A}^{kd} \quad (4.34)$$

The objective function (4.24) minimizes the reduced cost. Constraints (4.25)–(4.34) are a restriction of constraints (4.4)–(4.16) to tank truck k and day d , respectively. They are equivalent.

Resource Constraint Reformulation.

We replace constraints (4.29)–(4.30) with a resource constraint reformulation suitable for the algorithm presented in Section 4.5.3. Let $\mathbf{R} = \{0, 1, 2, 3\}$ be the index set of resources, where resource 0 represents time, resource 1 represents the number of marine stations on the route, and resources 2 and 3 correspond to tank truck resources.

Time and marine stations resources: $[a^0, w^0] = [0, T_{max}]$ are the lower and upper bounds on the route length (time resource), while $[a^1, w^1] = [0, 2]$ are the lower and upper bounds on the number of marine stations to visit in a route (marine stations resource). The resource consumption for resource 0 is $r_{ij}^0 = \tau_{ij}, \forall (i, j) \in \mathbf{A}^{kd}$, while the resource consumption for resource 1 is $r_{ij}^1 = 1$ if node j is a marine station, and 0 otherwise.

Tank truck resources. We also use two additional resources to check the feasibility of a route, that are, capacity and number of full compartments, denoted by resources 2 and 3, respectively. For resource 2, $[a^2, w^2] = [0, \Delta_k]$ are the lower and upper bounds, and the

resource consumption is $r_{ij}^2 = o_j$ if node $j \in \mathbf{N}^{kd}$, and 0 otherwise. For resource 3, the lower and upper bounds are $[a^3, w^3] = [0, |\mathbf{L}_k|]$, and the resource consumption is given by r_{ij}^3 is equal to the minimum number of compartments required to load quantity o_j if node $j \in \mathbf{N}^{kd}$, and 0 otherwise.

We denote by R_i^m the cumulated consumption of resource $m \in \mathbf{R}$ over all the arcs composing a partial route Φ_i from σ to i . The new formulation is as follows:

$$(R_i^m + r_{ij}^m - R_j^m)x_{ij}^{kd} \leq 0 \quad \forall m \in \mathbf{R}, \forall (i, j) \in \mathbf{A}^{kd} \quad (4.35)$$

$$a_i^m \leq R_i^m \leq w_i^m \quad \forall m \in \mathbf{R}, \forall i \in \mathbf{V}^{kd} \quad (4.36)$$

In this formulation, constraints (4.35) model resource consumption along arc (i, j) whenever it is part of the route, while constraints (4.36) require resource consumption along the σ - i partial route to be within the corresponding resource interval. The following section describes the subproblem labeling algorithm to determine the possible optimal partial paths and eliminate unpromising labels using some dominance rules.

4.5.3 Subproblem Labeling Algorithm

Many articles covering the basics of solving ESPPRC by using dynamic programming based labeling algorithms already exist. In this section, we give the details of all modifications to classical ESPPRC to solve our subproblem. The ESPPRC is NP-hard in the strong sense, see Dror (1994). A relaxed version of the ESPPRC, the shortest path problem with resource constraints (SPPRC), is commonly used instead. The SPPRC can be solved with pseudo-polynomial algorithms, see Irnich & Desaulniers (2005). However, this relaxation provides a lower bound of poorer quality than those of the ESPPRC.

The basic dynamic programming algorithm was devised by Desrochers & Soumis (1988) as an extension of the well-known Ford-Bellman algorithm. The central part of the algorithm is the use of labels to store information about resource values for partial-routes rooted at source node σ . We use the *improved definition of label* introduced by Feillet et al. (2004). Hence, each label ζ has a set of attributes: (i) a unique id; (ii) the current vertex $\bar{v}(\zeta)$; (iii) the accumulated reduced cost $\bar{C}(\zeta)$; (iv) the cumulative travel cost C ; (v) the accumulated consumption of each resource given by $\bar{R}(\zeta) = (r^1, r^2, \dots, r^{|\mathbf{R}|})$; (vi) an ordered set of the last visited nodes $vis(\zeta) \subseteq \mathbf{V} \setminus \{\bar{v}\}$; (vii) a set of served client-products $N(\zeta)$; (viii) a set of nodes $U(\zeta)$ unreachable by the tank truck k on day d ; (ix) the number of unreachable nodes $s(\zeta)$; (x) the parent label id $id_p(\zeta)$; (xi) the associated tank truck $k(\zeta)$; and (xii) the corresponding TTLP, denoted by $TTLP(\zeta)$. To summarize, each label is represented by

$$\zeta = \{id, \bar{v}, \bar{C}, C, \bar{R}, vis, N, U, s, id_p, k, TTLP\}.$$

The concept of labeling algorithm, as presented in Algorithm 1, is to iteratively extend labels according to resource constraints in the following way, until there are no more labels left. When a label has been extended, it is considered "processed". Loading constraints are managed in the extension function (EF).

Algorithm 1: Labeling(G, σ, δ)

```

1  $\mathcal{U} \leftarrow \{\zeta_\sigma = \{0, \sigma, 0, 0, [0, 0, 0, 0], \emptyset, \emptyset, \emptyset, 0, k, \emptyset\}\}$  // Set of unprocessed labels
2  $Sol = \emptyset$  // List of Labels at destination.
3 while  $\mathcal{U} \neq \emptyset$  do
4   Remove-Dominated( $\mathcal{U}$ )
5    $\zeta \leftarrow$  pick a  $\zeta \in \mathcal{U}$  and  $\mathcal{U} \leftarrow \mathcal{U} \setminus \{\zeta\}$ 
6   foreach  $j \notin U(\zeta)$  // Nodes to which  $\zeta$  can be extended to.
7     do
8        $\zeta_j = \text{Extend-Label}(\zeta, j)$  // Defined in Algorithm 2
9       if  $\bar{v}(\zeta_j) = \delta$  then  $Sol \leftarrow Sol \cup \{\zeta_j\}$  else  $\mathcal{U} \leftarrow \mathcal{U} \cup \{\zeta_j\}$ 
10 return  $Sol$ 

```

We initialize the first label from the source node σ , which is then inserted into $\mathcal{U}$ in line 1. In lines 3–9, we loop as long as there are unprocessed labels left. In line 4, a dominance function Remove-Dominated is used to remove dominated labels as described in Section 10. It is clear that not removing dominated labels in line 4 results in a complete enumeration of all feasible paths. We pick out a label in line 5, which is then extended using the EF Extend-Label in line 8. In this function, client-products are assigned to the compartments according to the rules specified in Table 4.2. We use the TTLP-heuristic (defined in Section 19) as long as it finds a solution to the TTLP. If the TTLP-heuristic fails, the EF uses a MIP to solve the TTLP. The MIP formulation of the TTLP, denoted by TTLP-MIP, and the extension function are described in Section 10. Feasible routes that reach the destination node δ are saved in line 9; otherwise, the labels are marked as unprocessed.

Extension Function

Calling the function Extend-Label extends the existing labels at node $i \in V$ to its successor nodes $\{j \in V | (i, j) \in A\}$. This function first creates the new label ζ_j and updates the consumption of resources along the arc (i, j) , then checks its feasibility and discards it if infeasible. A label ζ_j is feasible if it corresponds to a partial route that respects the resource constraints and the corresponding $TTLP(\zeta_j)$ is feasible. If the resource constraints are satisfied, it solves $TTLP(\zeta_j)$ tentatively by a fast heuristic. If the heuristic fails, we solve it with a MIP solver.

Algorithm 2: Extend-Label(ζ, j)

Create a label ζ_j // Extend $\bar{C}, C, vis, N$ along the arc (i, j) and inherit U, s, k, id_p from ζ

Update the resources and check their feasibility:

$R_j^m \leftarrow R_i^m + r_{ij}^m, \forall m \in R$

if $\exists m : R_j^m \notin [a_j^m, w_j^m]$ **then**

$U(\zeta) \leftarrow U(\zeta) \cup \{j\}, s(\zeta) \leftarrow s(\zeta) + 1$
 return $\emptyset$

Check the feasibility of TTLP:

$TTLP(\zeta_j) \leftarrow$ Build TTLP from $(N(\zeta_j), k(\zeta_j))$

$h\text{-code} \leftarrow HF(TTLP(\zeta_j))$

if $h\text{-code}$ **is not in** HM **then**

if $TTLP\text{-Heuristic}(\zeta_j)$ **does not fail** **then**

$HM[h\text{-code}] \leftarrow$ result of $TTLP\text{-Heuristic}(\zeta_j)$

else

$HM[h\text{-code}] \leftarrow$ feasibility of $TTLP\text{-MIP}(\zeta_j)$

if $HM[h\text{-code}]$ **is infeasible** **then**

$U(\zeta) \leftarrow U(\zeta) \cup \{j\}, s(\zeta) \leftarrow s(\zeta) + 1$
 $\zeta_j = \emptyset$

return ζ_j

To solve the same TTLP only one time, we propose a method for storing the solved problem in a hash-map HM using a hash function (HF); the HF is used to determine if a TTLP problem is already solved or not. The HF assigns a unique key ($h\text{-code}$) to each TTLP solved. Concretely, we start by transforming the sorted list of $N(\zeta_j)$ into a string of characters, concatenate this string with the id of the vehicle k separated by a separator (";"), and then use this resulting string to compute the key. Note that we use $HM[h\text{-code}]$ as an indicator of the feasibility of $TTLP(\zeta_j)$.

TTLP-MIP

Tank trucks are not equipped with a debit-meter. In order to remove any ambiguity, we remind the following rules: (i) The delivery of client-product involves completely unloading one or more compartments by gravity; a compartment is either filled at 100% or empty. (ii) We can only assign one client-product per compartment. (iii) We can only ship a maximum of $|L_{k(\zeta_j)}|$ client-products. (iv) A client-product can however be assigned to multiple compartments.

The mathematical formulation of $TTLP\text{-MIP}(\zeta_j)$ takes the following form:

$$\min_y \sum_{n \in N(\zeta_j)} \sum_{l \in L_{k(\zeta_j)}} y_{ln}^{kd} \quad (4.37)$$

$$\text{s.t.:} \quad \sum_{n \in N(\zeta_j)} y_{ln}^{kd} \leq 1 \quad \forall l \in L_k(\zeta_j) \quad (4.38)$$

$$\sum_{l \in L_k(\zeta_j)} C_{lk} y_{ln}^{kd} = o_n \quad \forall n \in N(\zeta_j) \quad (4.39)$$

$$y_{ln}^{kd} \in \{0, 1\} \quad \forall l \in L_k(\zeta_j), \forall n \in N(\zeta_j) \quad (4.40)$$

The objective function (4.37) minimizes the number of filled compartments. Constraints (4.38) ensure at most one client-product per compartment, while constraints (4.39) match the compartments assigned to each client-product with the requested quantity.

TTLP-Heuristic

Here, we describe how to solve the $TTLP(\zeta_j)$ using the TTLP-Heuristic. The latter is an adaptation of the heuristic proposed by Cornillier et al. (2008a) to our context. The idea of this heuristic is based on the *Largest fit First* principle, where clients with the largest orders are assigned first to compartments. Generally, these orders require more than one compartment and are therefore more difficult to assign, which means that the infeasibility of TTLP in this case can be easily checked.

Algorithm 3: TTLP-Heuristic(ζ_j)

$\Gamma \leftarrow |N(\zeta_j)|$, $\Gamma^s \leftarrow |\{n \in N(\zeta_j) : o_n > \max_{l \in L_k(\zeta_j)} C_{lk}\}|$, $N \leftarrow N(\zeta_j)$, $L \leftarrow L_k(\zeta_j)$

Step 1:

if $\Gamma + \Gamma^s > |L_k(\zeta_j)|$ **or** $\sum_{n \in N(\zeta_j)} o_n > \Delta_{k(\zeta_j)}$ **then** no feasible solution exists: **return** *Infeasible*

Step 2:

Sort L in descending order of their capacities.

while $\Gamma > 0$ **do**

 Sort N in descending order.

$\hat{n} \leftarrow$ first element in N **and** $N \leftarrow N \setminus \{\hat{n}\}$

$\hat{l} \leftarrow$ first element in L **and** $L \leftarrow L \setminus \{\hat{l}\}$

if then

 Split the client-product $\hat{n}$: $o_n \leftarrow (o_{\hat{n}} - C_{\hat{l}k}(\zeta_j))$

 Insert the new client-product n of quantity o_n : $N \leftarrow N \cup \{n\}$.

$\Gamma \leftarrow \Gamma + 1$

else

if $o_{\hat{n}} < C_{\hat{l}k}(\zeta_j)$ **then**

return *Fail*

else

 Assign the client-product $\hat{n}$ to $\hat{l}$

$\Gamma \leftarrow \Gamma - 1$

return *Feasible*

Let Γ be the number of client-products in list $N(\zeta_j)$ and Γ^s be the number of client-products in list $N(\zeta_j)$ that are greater than the largest compartment. It is easy to see that $\Gamma + \Gamma^s$ is a lower bound on the needed compartments. In Step 1 of Algorithm 3, we start with a quick identification of simple cases of infeasibility using this lower bound. In Step 2, we start by sorting the $L_{k(\zeta_j)}$ in descending order; then, we loop as long as $\Gamma > 0$. We sort the list $N(\zeta_j)$ in descending order. We pick the client-product $\hat{n}$ with the highest order quantity from $N(\zeta_j)$. If $\hat{n}$ does not fit into the largest compartment $\hat{l}$ of list $L_{k(\zeta_j)}$, then we split $\hat{n}$ into two client-products and insert the new one in $N(\zeta_j)$. If a client-product cannot be assigned to the compartment $\hat{l}$ due to Rules 6 and 7, then TTLP-Heuristic fails.

Dominance Rules

To keep the number of labels as small as possible, it is decisive to perform a dominance step for eliminating unneeded labels.

A label ζ_1 dominates a label ζ_2 if they reside at the same vertex and if for every possible extension of ζ_2 , there is a feasible extension of ζ_1 better in terms of cost and resource consumption. Dominated labels need not be extended. A label that is not dominated by any other label is called undominated or Pareto-optimal. In this paper, we use two phases of dominance, each one has different rules to follow:

Phase 1 is a heuristic dominance formulation. ζ_1 dominates a label ζ_2 if and only if :

$$\overline{C}(\zeta_1) \leq \overline{C}(\zeta_2) \quad (4.41)$$

$$r^m(\zeta_1) \leq r^m(\zeta_2) \quad \forall m \in R \quad (4.42)$$

$$s(\zeta_1) \leq s(\zeta_2) \quad (4.43)$$

Phase 2 is an exact dominance formulation. ζ_1 dominates a label ζ_2 if and only if :

$$\overline{C}(\zeta_1) \leq \overline{C}(\zeta_2) \quad (4.44)$$

$$r^m(\zeta_1) \leq r^m(\zeta_2) \quad \forall m \in R \quad (4.45)$$

$$U(\zeta_1) \subseteq U(\zeta_2) \quad (4.46)$$

4.5.4 Integer Solutions by Primal Diving Heuristic.

As we mentioned earlier, to derive integer solutions from the RMP, we propose a method called primal diving heuristic (PDH) which improves on the diving heuristic (DH). DH is a branch-and-price heuristic that consists of deep branching only, which means there is no backtracking. DH starts the branch and bound process by solving the linear relaxation of the problem by column generation. If the obtained solution is an integer, we stop. Otherwise, we create only one child node where we round up (set to 1 in the case of binary variables for instance) the variable with the highest fractional value. The (residual) RMP and CGSPs

that remain after the rounding up operation are cleaned up before re-optimization by deleting all unnecessary rows, columns and arcs. The linear relaxation of the cleaned residual RMP is solved again by column generation and this process is repeated until finding an integer solution, hopefully. Such preprocessing is a key feature in diving heuristics. It helps to avoid the heuristic dead-ending with an infeasible solution and accelerates the solution process. DH is highly used in practice to solve large scale problems, namely in aircrew scheduling, because first, often we do not have enough time to backtrack and second, even though we could make some errors in the branching decisions, solving the linear relaxation by column generation at every branch and bound node generally repairs them and we end up with solutions acceptable in practice. More details could be found in Joncour et al. (2010a).

For low to moderately dense set partitioning type problems, solutions of the linear relaxation are not highly fractional and integer solutions could be easily obtained by solving the RMP by a MIP solver. These integer solutions are closer, in the sense of the hamming distance, to both the fractional and the optimal integer ones than in denser problems. In other words, many variables are common in their supports (i.e., the variables equal to 1). However, the DH method does not exploit this *primal* information available in such sparse problems, like the one we are dealing with in this paper. The idea of PDH is to reinforce DH by adding a new type of branching rules by considering the best integer solution available (incumbent). This branching rule belongs to the family of local search methods (see Joncour et al. (2010a)). We use the information provided by the incumbent integer solution to choose the variable on which to branch. This variable shows the minimum difference between its values in the current integer solution and the fractional solution. In other words, we prioritize branching on the variable that is the closest to the incumbent, which is, in its turn, expected to be close to an optimal integer solution. Here again, the closeness is in terms of the (hamming) distance between the vertices in the polyhedron and not in terms of the cost solely. Algorithm 5 describes the pseudo-code of the method.

We start by solving the root node to get fractional and integer solutions, denoted by ρ^{LP} and ρ^{IP} respectively. The solutions at the root and those of the children exist because i) the model is supposed to be feasible which is the case in practice and ii) the set Ω could be initialized in the worst case with artificial columns (one client-product per (fictive) tank-truck) of higher costs. If the integrality gap between ρ^{LP} and ρ^{IP} is smaller than a given value $Lgap$, we stop. Otherwise, we branch until we find an acceptable integer solution. A description of the generic algorithm of branch-and-price is presented by Barnhart et al. (1998). Algorithm 4 shows the solution of a branch and bound node with CG.

Algorithm 4: SolveNode(η, Ω)

```

//  $\eta$  is the node number
//  $\Omega$  is a set of columns
while true do
   $(\rho, \alpha) \leftarrow$  Solve RMP //  $\rho$  is the primal solution,  $\alpha$  is dual solution
  //  $\mathbf{K}$  is the set of tank trucks
  //  $\mathbf{D}$  is the set of periods
   $\bar{c} \leftarrow 0$ 
  foreach  $k \in \mathbf{K}$  and  $d \in \mathbf{D}$  do
    //  $\Omega_{kd}$ : is the set (maybe empty) of negative reduced cost columns generated by  $CGSP^{kd}$ 
    //  $\bar{c}_{kd}$ : is the minimum reduced cost of columns in  $\Omega_{kd}$ 
     $(\bar{c}_{kd}, \Omega_{kd}) \leftarrow$  Solve  $CGSP^{kd}(\alpha)$ 
     $\Omega \leftarrow \Omega \cup \Omega_{kd}$ 
     $\bar{c} \leftarrow \min\{\bar{c}, \bar{c}_{kd}\}$ 
  If  $(\bar{c} \geq 0)$  break
return  $(\rho)$ 

```

Algorithm 5: PDH

```

1 Initialize  $\Omega$  with an initial set of columns
2 Root solutions:
3  $\rho^{LP} \leftarrow$  SolveNode(0,  $\Omega$ ), // Defined in Algorithm 4.
4 if  $\rho^{LP}$  is Integer then
5    $\rho^{IP} \leftarrow \rho^{LP}$ , return  $(\rho^{IP})$ 
6  $Z^{LP} = Z(\rho^{LP})$ 
7  $\rho^{IP} \leftarrow$  SolveMIP( $\Omega$ )
8 Branching:
9  $\eta \leftarrow 1$ ,  $zIpNotI \leftarrow 0$ 
10 while  $\frac{Z(\rho^{IP}) - Z^{LP}}{Z^{LP}} \geq Lgap$  and  $zIpNotI \leq maxZIpNotI$  do
11    $nbVars \leftarrow 0$ 
12   while  $nbVars < nbMaxFVars$  do
13      $w' \leftarrow \arg \min_{w | \rho_w^{IP} = 1} (\rho_w^{IP} - \rho_w^{LP})$  and set  $\rho_{w'} = 1$ 
14     Cleanup and preprocessing to filter  $\Omega_\eta$ 
15      $\rho^{LP} \leftarrow$  SolveNode ( $\eta, \Omega_\eta$ )
16     if  $\rho^{LP}$  is Integer and  $Z(\rho^{LP}) < Z(\rho^{IP})$  then
17        $\rho^{IP} \leftarrow \rho^{LP}$ , return  $(\rho^{IP})$ 
18     if  $Z(\rho^{LP}) \geq Z(\rho^{IP})$  then
19       return  $(\rho^{IP})$ 
20    $\eta \leftarrow \eta + 1$ ,  $nbVars \leftarrow nbVars + 1$ ,  $\Omega \leftarrow \Omega \cup \Omega_\eta$ 
21    $\rho_M^{IP} \leftarrow$  SolveMIP( $\Omega$ )
22   if  $Z(\rho_M^{IP}) < Z(\rho^{IP})$  then  $\rho^{IP} \leftarrow \rho_M^{IP}$ ,  $zIpNotI \leftarrow 0$ 
23   else  $zIpNotI \leftarrow zIpNotI + 1$ 
24 return  $(\rho^{IP})$ 

```

Root solution. We solve the root node of the branch-and-price method to optimality to obtain generally a fractional solution $\rho^{LP} = \text{SolveNode}(0, \Omega)$. If by chance the obtained

solution is integer, we return this solution which is optimal and by doing that we end the solving process. If not, we solve the RMP as a MIP ($\text{SolveMIP}(\Omega)$), using only the columns generated during the solution of the root node to get an initial integer solution ρ^{IP} . This idea was suggested by Joncour et al. (2010b) to search for an integer solution by restricting the master problem to a subset of promising variables, hence solving a problem which is considerably of smaller size. Unlike Joncour et al. (2010b) who stops here, we use the obtained solutions to start up PDH that continues with branching.

Branching (Algorithm 5). In line 9, we initialize a node counter $\eta = 1$ and the number of times the current integer solution is not improved $zIpNotI = 0$, then we loop (line 10) as long as the integrality gap is beyond a predetermined threshold $Lgap$ and $zIpNotI \leq maxZIpNotI$ (where $maxZIpNotI$ is the maximum number of times that the objective function is not improved). In this loop, we initialize in line 11 the number of variables ($nbVars = 0$) fixed by branching. We loop in line 12 as long as the number of fixed variables does not exceed $nbMaxFVars$, which is the maximum number of variables to fix before solving the MIP. In other words, we evaluate an equivalent number of branching nodes to collect enough columns before calling again the MIP Solver. We choose the branching variable and fix its value to 1 (line 13) and create a new branch-and-bound node (η).

In line 14, we perform some preprocessing by removing i) conflictual columns with the branching decision, i.e., columns that overlap (cover same client-products) with that associated with the variable ρ'_w , ii) rows, nodes and arcs related to client-products covered by $\rho_{w'}$; then, we solve the child node in line 15. If the solution is integer and better (line 16), we update ρ^{IP} and return it because PDH cannot improve it (line 17). We also stop PDH and return the current ρ^{IP} when $Z(\rho^{LP}) \geq Z(\rho^{IP})$ because there is no hope to improve it too. Otherwise, we increase the node and $nbVars$ counters and ω in line 20 and loop again.

After fixing $nbMaxFVars$ variables, we solve the RMP as a MIP in line 21 by considering all variables generated and consequently, update the current solution in line 22 if the objective value is improved. Here, we would like to point out that we start up the MIP solver with the previous integer solution to accelerate the solving process. Otherwise, we increase $zIpNotI$ in line 23.

4.5.5 Acceleration Strategies

Below we present three acceleration strategies we developed to reduce the runtime of CGSPs. The first acceleration strategy is used to distribute the runtime among processors, which results in a reduction in the time needed to solve all CGSPs. The others are used to reduce

the runtime of TTLP-MIP. Instead of using Cplex each time, we use these strategies in the EF to decide whether TTLP-MIP is feasible or not (for up to five client-products).

i) Parallel computing. The CGSPs are independent; multi-threading is used to solve them in parallel.

ii) Fast order loading. Before describing the following acceleration strategies, we first define the set UC_k of the different compartment capacities in tank truck k , and let O be a set of orders and CC_k^o the set of possible compartment combinations associated with the order quantity $q_o, o \in O$ and the tank truck k . Moreover, two compartment combinations $cc \in CC_k^o$ and $cc' \in CC_k^o$ are said to be *compatible*, if for each $c \in UC_k$, the total number of compartments used by cc and cc' does not exceed the number of available compartments n_k^c .

To clarify these two notions (compartment combinations and compatibility), let us consider the following example. Let k be a tank truck having the following compartment configuration: 2-2-2-2-1-1-1-1-3-3 and suppose that the order quantity is 5. So, UC_k contains 3 elements $\{1, 2, 3\}$. We have $n_k^1 = 4, n_k^2 = 4$ and $n_k^3 = 2$ and $CC_k^5 = \{(3, 1, 0), (2, 0, 1), (1, 2, 0), (0, 1, 1)\}$. Basically, this set contains all integer solutions to the equation $x_1 + 2x_2 + 3x_3 = 5$ under constraints $x_1 \leq 4, x_2 \leq 4, x_3 \leq 2$. The size of this set is reasonable in practice (given the configurations in Table 4.4) and could be computed a priori before starting the column generation process. The combinations $(3, 1, 0)$ and $(2, 0, 1)$ are incompatible because $(3, 1, 0) + (2, 0, 1) = (5, 1, 1)$ exceeds the maximum $n_k^1 = 4$, while $(1, 2, 0)$ and $(0, 1, 1)$ are compatible because $(1, 2, 0) + (0, 1, 1) = (1, 3, 1)$ respects the number of available compartments ($n_k^1 = 4, n_k^2 = 4, n_k^3 = 2$).

The strategy simply tells EF to:

- Load order o into tank truck $k \in K$ if and only if $CC_k^o \neq \emptyset$. Otherwise, the loading is infeasible.
- Load two orders o_1 and o_2 into tank truck $k \in K$ if and only if there exist $cc \in CC_k^{o_1}$ and $cc' \in CC_k^{o_2}$ such that cc and cc' are compatible. Otherwise, the loading is infeasible.

iii) Resource bounding. For each tank truck k , the upper bounds on the number of orders (UB_k^O), on the capacity (UB_k^C), and on the number of compartments (UB_k^L) are computed a priori (before starting the column generation process) by maximizing $\sum_{o \in O} s_o$, $\sum_{o \in O} q_o s_o$, and $\sum_{l \in L_k} \sum_{o \in O} y_{lo}$ respectively over the set:

$$\sum_{o \in O} y_{lo} \leq 1 \quad \forall l \in L_k \quad (4.47)$$

$$\sum_{l \in L_k} C_{lk} y_{lo} = q_o s_o \quad \forall o \in O \quad (4.48)$$

$$y_{lo} \in \{0, 1\} \quad \forall l \in L_k, \forall o \in O \quad (4.49)$$

$$s_o \in \{0, 1\} \quad \forall o \in O \quad (4.50)$$

where y_{lo} are binary variables equal to 1 if compartment $l \in L_k$ is assigned to order $o \in O$, and 0 otherwise. s_o are binary variables equal to 1 if order $o \in O$ is completely satisfied by the tank truck $k \in K$, and 0 otherwise. Constraints (4.49) and (4.50) ensure the integrality of those variables. Constraints (4.47) ensure at most one order per compartment, while constraints (4.48) match the compartments assigned to each order with the requested quantity if s_o is equal to 1. These upper bounds are used by the EF to "kill" labels whenever we exceed them.

4.6 Case Study

In this section, we present a case study from Senegal, West Africa. At the beginning, we remind the context of application. Then, we go through the characteristics of the 29 instances featuring our test benchmark before reporting and discussing the computational results obtained from these experiments. Finally, we compare our results with those provided by the current solution approach used by the PC.

All our algorithms have been implemented using the C++ language. The optimization runs were performed on a machine equipped with an Intel (R) Core (TM) i7-6700 CPU at 3.40 GHz using a Linux operating system. All linear programs and mixed integer programs are solved by the IBM CPLEX commercial solver (version 12.9.0.0). We use the OsiSolverInterface (version 0.107.9) framework to communicate with CPLEX solver via the C++ language. All the CGSPs are ESPPRCs, solved by dynamic programming using the Boost library (version 1.57.0) Boost (2019). The computational times are in seconds. For the test settings, we use the following parameters: $Lgap = 0.25\%$, $nbMaxFVars = 3$ and $maxZipNotI = 3$.

4.6.1 Context

On a weekly basis, a petroleum company (PC) based in Dakar, Senegal has to replenish a network of 78 clients (petrol stations, marine stations, bakeries and some other occasional enterprises) with five types of marked and unmarked petroleum products from four depots using a heterogeneous fleet of compartmented tank trucks. The marked products are *diesel*, *marine zoom* and *kerosene*. The unmarked products are *super* and *gasoil*. A petrol station

may order several types of products, while marine stations and bakeries can order just one distinct marked petroleum product. Clients must place orders before Friday preceding the week of delivery. Every client specifies the latest day he/she should be delivered. This depends on an estimate of his/her consumption rate combined with the desired safety stock.

The PC rents on a weekly basis some tank trucks owned by four private transportation companies that are also doing business with some of the PC competitors. That is, a pool of petroleum companies is contracting with the same group of private transportation companies. The rental of a specific truck by a petroleum company from a transportation company is subject to its temporary availability since it might have already been rented by a competitor of the PC for the same targeted period. This implies the need for the PC to use a computer-based system for the planning, scheduling and optimization of its operations rather than relying on the empirical expertise of a planner/scheduler. Tanks trucks are classified into two categories with regards to capacity: (i) small tank trucks with capacity of less than 20000 liters, (ii) Jumbo tank trucks with capacity greater than or equal to 20000 liters. Some tank trucks carry only marked products are labeled 'Fishing' in this case study. Note that a tank truck can be both Jumbo and Fishing or none of them.

The mid-term goals of this research project are: (i) to provide near optimal routes and delivery schedules by considering the complex loading constraints, (ii) to rapidly re-optimize the solution in case of accidents, vehicle breakdowns or unexpected unavailability of drivers, and (iii) to pave way to the design of an effective decision support system (DSS). As we can see, we address the first point in this paper. As the resulting problem is uncommon in the literature, the PC cannot use existing DSSs; the design of a new efficient and customized solution approach is necessary to cope with the needs of a group of companies in the region following the same regulations.

4.6.2 Instances characteristics

The 29 real-world instances correspond to 29 planning weeks chosen from two years (2015-2016) of data shared gracefully by the PC, which covers a network of 78 distinct locations. The locations consist of 4 depots, 13 petrol stations, 11 marine stations, 7 enterprises, 42 bakeries. In Section 4.6.2, we first describe the planning weeks before presenting the characteristics of the tank trucks in Section 4.6.2.

Planning Weeks

Table 4.3 shows the instances and their characteristics, namely (from left to right), the instance type, the instance name (Inst.), the week from which it was generated, the number of client-products (#CP), the total number of stations (#Stat.), the number of petrol stations it contains (#Gas), the number of marine stations (#Mar), the number of bakeries (#Bak), and the number of enterprises (#Entr). We also give the volume of unmarked products (UnPr) and the volume of marked products (MaPr). Note that the volume of products is in cubic meters. The number of client-products is between 38 and 68, while the number of stations varies from 31 to 54.

Table 4.3 Overview of the Set of Instances.

Type	Inst.	Week	#CP	#Stat.	#Gas	#Mar	#Bak	#Entr	UnPr	MaPr
Small	W1	14-12-2015	41	33	16	2	12	3	18233	8029
	W2	04-01-2016	49	36	17	8	10	1	18250	8098
	W3	01-02-2016	46	33	16	6	8	3	18263	8100
	W4	22-06-2015	47	39	20	5	13	1	18285	8065
	W5	19-10-2015	48	31	18	2	10	1	18272	8040
	W6	18-01-2016	49	37	16	6	12	3	18293	8092
	W7	18-01-2015	50	38	16	6	13	3	18303	8092
	W8	02-11-2015	47	37	22	3	6	6	18310	8054
	W9	24-11-2015	51	34	21	2	9	2	18325	8034
	W10	16-11-2015	58	44	20	7	15	2	18299	8136
	W11	02-05-2016	56	44	20	4	17	3	18312	8075
	W12	14-03-2016	43	32	15	5	9	3	18246	8069
	W13	18-04-2016	68	54	18	7	24	5	18275	8124
	W14	05-04-2016	45	33	16	8	6	3	18247	8099
Medium	W15	16-05-2016	44	36	13	6	14	3	18250	8089
	W16	21-12-2015	51	39	18	6	15	0	18253	8083
	W17	09-11-2016	56	44	18	6	19	1	18287	8094
	W18	09-05-2016	53	42	18	8	14	2	18303	8117
	W19	29-03-2016	50	39	14	6	17	2	18228	8080
	W20	15-02-2016	46	35	14	7	12	2	18222	8090
	W21	12-10-2015	49	40	16	3	20	1	18222	8073
	W22	25-01-2016	52	39	16	5	18	0	18267	8083
	W23	08-02-2016	63	47	19	8	17	3	18306	8121
Large	W24	11-04-2016	67	49	23	7	17	2	18326	8100
	W25	11-01-2016	43	34	11	6	14	3	18149	8070
	W26	21-03-2016	64	50	18	9	23	0	18288	8143
	W27	11-06-2016	44	34	11	6	14	3	18150	8070
	W28	28-12-2015	52	39	17	3	17	2	18245	8062
	W29	30-11-2015	38	31	8	3	18	2	18113	8057

Week: the week from which the instance was generated; **#CP**: the number of client-products; **#Stat.**: the total number of stations; **#Gas**: the number of petrol stations; **#Mar**: the number of marine stations; **#Bak**: the number of bakeries; **#Entr**: the number of enterprises; **UnPr**: the volume of unmarked products; **MaPr**: the volume of marked products;

Based on density (Figure 4.2), the instances are divided into the following three groups:

small, medium and large size instances. Here, we mean by density the number of nonzeros per column, i.e., the number of client-products served per route. We believe that density is the inherent measure that influences the execution time both in subproblems (dominance, extension, ...) and master problem (matrix inversions, branching ...). In general VRP, we used to classify instances according to the number of nodes/arcs in the network because the density is highly correlated with the number of nodes/arcs. In this MDMPPRP, the context is quite different because of the loading constraints, the heterogeneity of available trucks (with different compartment configurations) and the clients themselves (ordering subsets of products) that we must serve that week. This is what impacts really the number of clients-products served per route (i.e., the density).

It is not surprising that the number of feasible routes is correlated with density as shown in Figure 4.2. What is surprising is when density increases a little bit, the number of feasible routes explodes. For instance, for the five largest instances (W24, W25, W26, W27, W29), we are not able to generate all feasible routes. The value 5×10^7 is given only to indicate the maximum number of feasible routes that we can generate using the available memory.

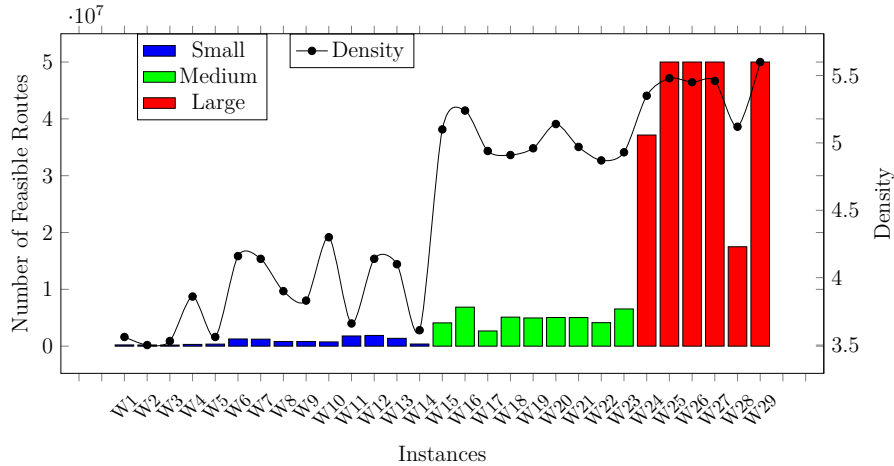


Figure 4.2 Distribution of Feasible Routes on the Instances.

Tank trucks description

Table 4.4 presents the heterogeneous tank trucks database and their characteristics, namely (from left to right), the id of tank truck (Id), the total capacity (Ca.), the ordered configuration of the compartments (Config.), where each digit represents the compartment capacity in cubic meters, and the number of compartments it contains (Co.). The last two columns indicate if the tank truck is labeled 'Fishing' or not (F.), and if it is labeled jumbo or not

Table 4.4 Tank Trucks Properties.

Id	Ca.	Config.	Co.	F.	J.	Id	Ca.	Config.	Co.	F.	J.
1	13	3-2-2-3-3	5	☐	☐	19	21	2-3-1-1-2-3-5-4	8	☐	☒
2	13	3-1-1-1-1-2-2-2	8	☐	☐	20	22	2-2-1-3-2-1-2-3-2-2-2	11	☐	☒
3	13	1-2-4-6	4	☒	☐	21	30	7-7-2-7-7	5	☐	☒
4	13	2-1-4-6	4	☒	☐	22	33	5-4-3-2-1-1-2-2-3-4-6	11	☐	☒
5	14	2-1-1-4-4-2	6	☐	☐	23	33	4-5-4-3-2-3-2-5-5	9	☐	☒
6	14	4-2-1-1-4-2	6	☐	☐	24	33	5-4-3-2-1-1-2-2-3-4-6	11	☐	☒
7	14	2-1-1-1-1-2-3-3	8	☒	☐	25	33	5-3-4-1.5-1.5-1-1-2-3-2-4-3-2	13	☒	☒
8	14	4-2-3-2-3	5	☒	☐	26	35	6-4-3-1-1-2-5-6-7	9	☒	☒
9	14	4-2-3-2-3	5	☐	☐	27	35	6-4-5-2-2-2-3-8-3	9	☒	☒
10	14	4-2-3-2-3	5	☐	☐	28	35	4-6-2-4-3-4-3-6-3	9	☐	☒
11	14	4-3-7	3	☐	☐	29	36	4-2-6-2-4-2-3-6-3	9	☐	☒
12	18	5-2-2-2-3-4	6	☐	☐	30	37	10-3-4-2-5-6-7	7	☐	☒
13	18	3-2-2-3-2-6	6	☐	☐	31	37	6-3-6-4-6-6-6	7	☒	☒
14	18	2-2-2-2-1-1-1-1-3-3	10	☐	☐	32	37	5-2-5-2-2-1-1-2-2-2-5-2-2-2	15	☒	☒
15	18	2-1.5-1-1-1.5-4-4-3	8	☐	☐	33	38	8-4-3-2-2-1-1-2-5-4-6	11	☒	☒
16	19	5-1-1-5-5-2	6	☐	☐	34	38	6-4-6-2-3-5-4-2-6	9	☐	☒
17	20	6-5-5-2-2	5	☐	☒	35	40	6-4-1-1-4-2-4-6-4-3-5	11	☒	☒
18	20	2-4-3-4-5-1-1	7	☐	☒	36	40	6-4-1-1-4-2-4-6-4-3-5	11	☐	☒

Id: the tank truck id; **Ca.**: the total capacity; **Config.**: the compartments configuration; **Co.**: the number of the tank truck compartments; **F.**: indicates if the tank truck is labeled 'Fishing' or not; **J.**: indicates if the tank truck is labeled jumbo or not;

(J.), respectively. To summarize, the database contains 36 tank trucks with capacities varying from 13000 to 40000 liters. Each tank truck contains from 3 to 15 compartments of capacities ranging from 1000 to 10000 liters. We have 11 tank trucks labeled 'Fishing' and 21 'jumbo'.

4.6.3 TTLP Solving

TTLP Results. To better understand the TTLP, we report in Table 4.5 the number of distinct TTLP-MIPs solved (#MIP) (this number includes those solved in the preprocessing phase, in which we solved all the combinations of 1 to 2 client-products for each tank truck in order to determine if we will add the corresponding arc to the graph or not), the number of calls to already solved TTLP-MIPs (#CAS), the maximum time that takes to solve a TTLP-MIP (MaxT(s)), the total time spent solving TTLP-MIPs (T(s)), and the percentage of the runtime of TTLP solving in total time (P(%)). From this table, we observe that the percentage of time required to solve the TTLP decreases when the size of the instances increases (32% on average for the small instances, 19% for the medium ones and 6% for the large ones). This is justified by the number of TTLP-MIPs solved in the first iteration. We

solve 99% of the TTLP-MIPs in the first iteration. In the following iterations, we just use the output of the already solved TTLP-MIPs in most cases. We also observe that #CAS increases with the size of the instances (3.8E+06, 1.0E+07, 7.3E+07, for the small, medium and large instances, respectively).

Table 4.5 Results of PDH: TTLP Statistics.

Type	Inst.	TTLP					TTLP-Heuristic	
		#MIP	#CAS	MaxT(s)	T(s)	P(%)	#Calls	Success(%)
Small	W1	2,904	1.1E+06	0.05	2	50	2391	36
	W2	3,508	8.3E+05	0.04	2	61	3174	42
	W3	4,265	8.6E+05	0.05	3	68	4211	39
	W4	2,967	2.1E+06	0.04	1	31	1711	47
	W5	5,828	1.5E+06	0.05	6	60	6313	41
	W6	4,241	4.3E+06	0.04	4	32	3419	38
	W7	4,101	4.0E+06	0.04	4	38	3062	42
	W8	6,847	4.5E+06	0.05	9	66	7223	29
	W9	7,989	4.1E+06	0.05	10	38	8732	33
	W10	7,517	5.2E+06	0.04	9	61	7831	38
	W11	8,598	4.3E+06	0.06	12	64	8366	25
	W12	8,407	8.4E+06	0.06	9	46	9326	34
	W13	6,513	9.4E+06	0.05	6	40	7001	40
	W14	5,858	2.3E+06	0.06	8	70	5110	38
Avg		5,682	3.8E+06	0.05	6	52	5562	37
Medium	W15	2,340	9.5E+06	0.03	3	30	1316	49
	W16	5,652	5.4E+06	0.04	4	56	5309	39
	W17	6,122	6.4E+06	0.08	9	42	6240	36
	W18	8,301	7.4E+06	0.05	7	58	8282	32
	W19	3,898	1.1E+07	0.04	6	37	3562	40
	W20	6,641	9.5E+06	0.05	12	32	7227	39
	W21	3,596	1.6E+07	0.63	6	41	2798	37
	W22	5,567	1.3E+07	0.04	9	30	4361	32
	W23	6,746	1.6E+07	0.05	11	37	7150	31
Avg		5,429	1.0E+07	0.11	7	40	5138	37
Large	W24	7,376	4.8E+07	0.07	18	34	8004	37
	W25	4,973	4.8E+07	0.04	14	29	6430	44
	W26	6,445	6.5E+07	0.06	48	24	5386	37
	W27	5,878	8.1E+07	0.05	25	25	7343	40
	W28	12,786	7.8E+07	0.05	32	35	15740	32
	W29	3,441	1.2E+08	0.05	29	25	3608	41
Avg		6,817	7.3E+07	0.05	28	29	7752	38

#MIP: the number of distinct TTLP-MIPs solved; #CAS: the number of calls to already solved TTLP-MIPs; MaxT(s): the maximum time that takes to solve a TTLP-MIP; T(s): the total time spent solving TTLP-MIPs; P(%): the percentage of the runtime of TTLP solving in the total time; #Calls: the number of calls to the TTLP-Heuristic; Success(%): the percentage of times that TTLP-Heuristic succeeded to solve the problem;

In the EF (see Section 10), in some cases, we solve a TTLP-MIP to check if a label is feasible or not, which takes a maximum of 0.11 seconds on average on the medium instances. The time needed to solve all the TTLP-MIPs ranges from 1 second (achieved by a small instance) to 48 seconds (observed in the large ones). The number of TTLP-MIPs solved is insignificant as compared to the value of $\#CAS$. It represents less than 0.50%. On average, we call one distinct TTLP-MIP 640 times for the small instances, and 2254 and 13308 times for the medium and large ones, respectively. In the following paragraph, we explain the impact of the hashing technique.

Impact of TTLP Hashing. To evaluate the impact of TTLP hashing, we ran the PDH method with and without TTLP hashing on all instances with a time limit of 1 hour. As mentioned in Section 10, without TTLP hashing, the same TTLP problem is solved many times. The results obtained are as follows: (i) PDH with TTLP hashing solves all instances in less than 8 minutes, (ii) PDH without TTLP hashing cannot solve any instance in less than one hour (some instances are not solved even after 6 hours). Figure 4.3 gives an overview of the progress of the two methods on instance **W3**. First, we report in Figure 4.3a the evolution of the number of TTLPs solved at each iteration without using TTLP hashing. Second, in Figure 4.3b we report the evolution of the number of TTLPs solved at each iteration using TTLP hashing.

We observe that the number of TTLPs solved in the first case is very big, jumping from 123418 in the first iteration to 919285 in the last iteration, with an augmentation of more than 100000 at each iteration. In the second case, the number of TTLPs solved is 75 times smaller and more stable (varying between 1636 and 1638). Note that we obtain the same solution in both cases when there is no time limit. In the first case, the time needed is more than one hour (3702 seconds), while it is 5 seconds in the second case. We observe that using TTLP hashing is 740 times faster than not using TTLP hashing. Therefore, we use TTLP hashing in the following sections.

Impact of TTLP-Heuristic. Table 4.5 reports, for each instance, the number of calls to the heuristic ($\#calls$) and the percentage of success ($Success(\%)$). Note that the number of TTLP-MIPs solved is not necessarily smaller than the value $\#Calls$ because we include the TTLP-MIPs that we solve in the preprocessing phase. We observe that TTLP-heuristic performs as well for the small instances as for the medium ones. It succeeds to solve the TTLP in 37% of the cases on average. It gives better performance for the large instances (+1% on average). The TTLP-Heuristic is a polynomial algorithm, while solving the TTLP-MIP is not polynomial. For this reason, this percentage of successes helps the EF to be faster than solving the TTLP-MIP all the time.

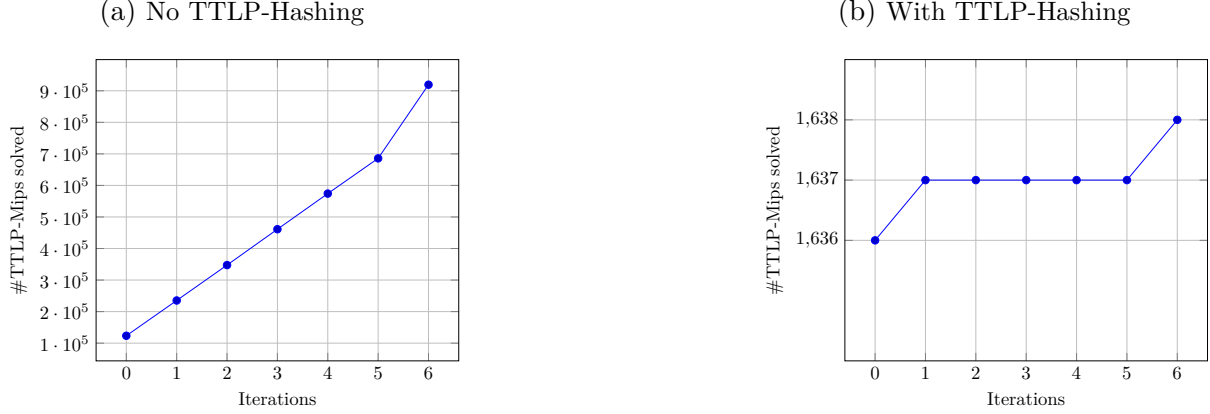


Figure 4.3 Impact of TTLP-Hashing on Instance **W3**.

4.6.4 Performance Comparison Between PDH and DH

The goal of this section is to demonstrate that PDH finds excellent solutions in a very short time. We provide a comparison between PDH and DH followed by performance analysis. In Table 4.6, we report the optimal value of the linear relaxation (LP) obtained at the root node, followed by the root node time (RT(s)). These values are the same for the two methods (PDH and DH). For each method, we report the integrality gap (GAP) in percentage between the cost of the best integer solution found and the linear relaxation optimal value (i.e., LB) and the total runtime (T(s)). In addition, we report: the number of generated columns (#Cols.), the number of iterations (It.) and the number of branch-and-bound nodes (No.). We can see that all the instances are solved by PDH in a smaller computing time. The average time for the large instances is 106 seconds and the largest one is 206 seconds. The DH fails to find a feasible solution for the four instances W6, W13, W14 and W22. We observe that PDH outperforms the DH in all instances.

When we exclude the time of the root node which is common for both methods, i.e., when comparing $T(s)$ minus $RT(s)$ for both, the PDH time is on average 14 times faster for the small instances, 30 times for the medium ones, and 106 times for the large ones. This is justified by the number of branching nodes required. On average, 5 times fewer nodes are needed for the small instances, and 6 times for the medium and large ones. Moreover, on average over all instances, the number of iterations is reduced by half. We stop after finding the initial integer solution in 65% of the instances. For the remaining 35%, we solve more nodes to find a solution better than the initial integer solution.

Since more branch-and-bound nodes are solved, the DH generates more columns than the PDH (from 79 to 180 on average for each type of instances). The solution quality, measured

Table 4.6 Results of PDH and DH: Runtime and Solution Quality.

Type	Inst.	Root Node		PDH					DH				
		LP	RT(s)	GAP(%)	T(s)	#Cols	It.	No.	GAP(%)	T(s)	#Cols	It.	No.
Small	W1	57033	4	0.37	5	4886	14	3	0.92	5	4882	11	2
	W2	168111	4	0.03	5	5701	9	1	0.22	6	5735	22	6
	W3	173060	4	0.06	5	6221	8	1	0.46	8	6287	29	9
	W4	142764	4	0.03	5	5228	16	1	0.06	6	5228	19	4
	W5	57934	10	0.03	10	7541	11	1	0.13	13	7549	18	5
	W6	173114	8	0.47	15	5995	29	10	-	-	-	-	-
	W7	174047	7	0.43	12	6067	21	7	0.86	21	6153	42	11
	W8	77341	14	0.12	14	7056	7	1	1.37	27	7140	24	8
	W9	57546	14	0.36	28	8323	31	13	0.8	33	8441	44	12
	W10	197748	15	0.02	16	9429	11	1	0.17	43	9515	44	11
	W11	82706	19	0.19	20	6595	15	1	1.22	37	6958	53	10
	W12	141953	19	0.03	19	6566	15	1	0.14	29	6590	23	4
	W13	148299	15	0.15	16	10525	16	1	-	-	-	-	-
	W14	153104	11	0.04	12	9174	9	1	-	-	-	-	-
		Avg	11	0.17	13	7093	15	3	0.58	21	6771	30	7
Medium	W15	122726	11	0.15	12	3947	15	1	0.57	22	3954	22	6
	W16	153645	8	0.16	8	6444	13	1	0.4	14	6499	30	7
	W17	138420	13	0.26	22	8944	24	9	0.65	24	8998	31	7
	W18	189546	12	0.06	13	7422	13	1	0.14	34	7876	50	9
	W19	163038	17	0.06	17	7371	17	1	0.49	37	7623	49	8
	W20	162835	18	0.32	40	6949	29	8	1.41	60	7046	46	12
	W21	92952	14	0.13	14	5749	21	1	0.13	15	5749	21	1
	W22	132679	19	0.29	31	7256	27	7	-	-	-	-	-
	W23	199063	30	0.13	31	7973	17	1	0.78	102	8182	60	13
		Avg	16	0.17	21	6895	20	3	0.57	39	6991	39	8
Large	W24	178067	54	0.11	56	12872	15	1	0.29	304	13048	55	11
	W25	157705	50	0.1	51	7609	22	1	0.84	366	7866	79	10
	W26	203309	86	0.27	206	9540	34	11	0.89	300	9548	33	7
	W27	157710	104	0.11	105	9295	27	1	0.89	286	9657	86	9
	W28	76924	71	0.73	96	10454	26	3	0.82	137	10657	36	4
	W29	71564	85	0.58	123	5917	35	4	1.73	158	5991	40	5
		Avg	75	0.32	106	9281	27	4	0.91	258	9461	55	8

LB: the optimal value of the linear relaxation (obtained at the root node); **RT(s)**: the root node time; **GAP(%)**: the integrality gap; **T(s)**: the total runtime; **#Cols**: the number of generated columns; **It.**: the number of iterations; **No.**: the number of branch-and-bound nodes.

by the integrality gap of the PDH, is 5 times higher than DH for small and large instances, and 4 times higher than DH for the medium ones. When we analyze the solutions of the two methods, we find that the DH solutions have more empty space on average (see Section 4.6.7).

PDH detailed performance statistics. We present detailed performance statistics of the PDH method. Table 4.7 shows from left to right: the instance type, the instance name (Inst.), the total number of labels (#Lbs), the total number of calls to the dominance function (#DC), the percentage of times that the dominance function returns true (DS(%)), the time

of the CGSPs (CGSPt(s)) and the time of the MP (Mpt(s)). From Table 7, we can observe is that the time of the MP solution is very small as compared to the time spent solving the CGSPs for all instances, it represents at most 2.40 seconds, while the CGSPs take 200 seconds for instance W28. This is due to the complexity of the CGSPs related to the complex loading constraints and the elementary routes. It is also due to the important number of managed labels (ranging on average from $2,1E+06$ for the small instances to $3,5E+06$ for the medium instances and $1,2E+07$ for the large instances). The reported value of DS(%) gives an idea about the complexity of the CGSPs and the search space. This value is a percentage of the calls to the dominance function reported in (#DC) that return true. The smaller this percentage, the more labels we keep. For example, this value starts at 0.88% on average for the small instances and decreases to 0.77% and 0.36% for the medium and the large ones, respectively.

4.6.5 PDH Acceleration Strategies

In this section, we analyze the impact of acceleration strategies on the PDH and DH methods, denoted by PDHA and DHA, respectively. Since the acceleration strategies do not affect solution quality, in Table 4.8, we only report the runtime (T(s)) and the runtime reduction in percentage (r.T(%)) for the PDHA and DHA. As can be seen from these results, runtime is reduced by more than 68% with the PDHA, which is an important amelioration of the solution time. For example, we need only 2.96 seconds on average to solve the small instances with PDHA, compared to 13 seconds without the acceleration strategies. A similar amelioration of solution time is also observed with the DHA. However, runtime reduction is slightly lower than with the PDHA (i.e., 1%, 2% and 5% less for the small, medium and large instances, respectively). The PDHA outperforms the DHA for all the instances, except for the smallest one (W1), since fewer nodes and a smaller number of iterations are needed (see Section 4.6.4).

For example, we only need, on average, 2.96 seconds to solve the small instances with PDHA rather than 4.85 seconds with DHA, see Table 4.8.

4.6.6 Exact Branch-and-Price

In this section, we present the solutions obtained by the exact branch-and-price method that serves to assess the effectiveness of the proposed heuristic approach. This method is implemented in C++ using the framework BCP (version 1.4) Coin-Or (2019) and using the common practice of branching only on inter-tasks (also called follow-ons), effectively forcing or forbidding two tasks to appear consecutively on the same route (see Anbil et al. (1998)). In Table 4.9, we report the same information for the 23 instances that we can optimally solve

Table 4.7 Results of PDH: Performance Statistics.

Type	Inst.	#Lbs	#DC	DS(%)	CGSPt(s)	Mpt(s)
Small	W1	1.3E+06	9.4E+07	1.06	4.30	0.09
	W2	9.3E+05	6.5E+07	1.08	3.90	0.23
	W3	7.8E+05	6.3E+07	0.94	4.30	0.40
	W4	1.8E+06	1.2E+08	1.16	4.30	0.12
	W5	1.9E+06	1.8E+08	0.84	9.40	0.15
	W6	4.5E+06	4.3E+08	0.85	13.00	1.00
	W7	3.4E+06	3.1E+08	0.92	11.00	0.55
	W8	1.9E+06	2.2E+08	0.74	13.00	0.40
	W9	6.1E+06	8.9E+08	0.57	27.00	0.90
	W10	2.8E+06	2.7E+08	0.85	15.00	0.33
	W11	2.9E+06	2.8E+08	0.87	19.00	0.33
	W12	4.0E+06	4.8E+08	0.72	18.00	0.13
	W13	4.4E+06	3.9E+08	0.92	15.00	0.50
	W14	1.7E+06	1.7E+08	0.77	11.00	0.56
Avg		2.8E+06	2.8E+08	0.88	12.01	0.41
Medium	W15	2.8E+06	3.8E+08	0.65	11.00	0.08
	W16	1.8E+06	1.2E+08	1.18	7.70	0.26
	W17	5.1E+06	6.0E+08	0.71	21.00	0.85
	W18	2.3E+06	1.8E+08	1.09	12.00	0.76
	W19	4.6E+06	4.7E+08	0.83	16.00	0.26
	W20	8.3E+06	1.7E+09	0.42	37.00	1.70
	W21	3.3E+06	3.2E+08	0.86	13.00	0.03
	W22	7.8E+06	1.1E+09	0.62	29.00	0.96
	W23	6.5E+06	9.8E+08	0.58	30.00	0.42
Avg		4.7E+06	6.5E+08	0.77	19.63	0.59
Large	W24	1.1E+07	1.8E+09	0.54	53.00	1.70
	W25	9.6E+06	1.9E+09	0.45	49.00	0.78
	W26	3.0E+07	1.3E+10	0.22	200.00	1.20
	W27	1.7E+07	4.7E+09	0.34	100.00	0.97
	W28	1.5E+07	3.9E+09	0.35	92.00	2.40
	W29	1.9E+07	6.8E+09	0.26	120.00	0.62
Avg		1.7E+07	5.3E+09	0.36	102.33	1.28

#Lbs: the total number of labels; #DC: the total number of calls to the dominance function; DS(%): the percentage of times that the dominance function returns true; CGSPt(s): the time of the CGSPs; Mpt(s): the time of the MP;

as in Table 4.6.

The integrality gap of the exact method clearly shows that PDH finds solutions of high quality. Indeed, by comparing the GAP(%) reported in Tables 4.6 and 4.9, we find that for the majority of instances, PDH finds optimal solutions, or at worst, a solution with an integrality gap slightly higher than that shown by the exact method. From the 23 instances, PDHA succeeds to find an optimal solution for 19 of them, in a fraction of the time of

Table 4.8 Results of PDH and DH with Acceleration Strategies.

Type	#Inst	PDHA		DHA	
		T(s)	r.T(%)	T(s)	r.T(%)
Small	W1	1.27	-73	0.97	-79
	W2	1.10	-76	1.44	-76
	W3	1.21	-76	2.06	-74
	W4	1.39	-72	1.63	-71
	W5	1.79	-82	2.66	-80
	W6	4.33	-71	-	-
	W7	3.15	-73	5.23	-75
	W8	2.21	-85	5.94	-78
	W9	9.55	-66	8.59	-74
	W10	2.56	-84	9.53	-78
	W11	3.25	-84	8.72	-76
	W12	3.91	-80	6.63	-77
	W13	3.70	-77	-	-
	W14	1.97	-83	-	-
Avg		2.96	-77	4.85	-76
Medium	W15	3.66	-69	7.17	-68
	W16	1.82	-78	3.38	-77
	W17	4.61	-79	4.77	-80
	W18	3.36	-75	9.83	-71
	W19	3.44	-80	8.80	-76
	W20	11.21	-72	16.13	-73
	W21	3.51	-74	3.70	-76
	W22	8.10	-74	-	-
	W23	6.66	-78	26.84	-74
Avg		5.15	-76	10.08	-74
Large	W24	12.36	-78	86.97	-71
	W25	12.77	-75	119.06	-67
	W26	100.71	-51	150.12	-50
	W27	28.51	-73	101.64	-64
	W28	23.37	-76	37.38	-73
	W29	53.31	-57	72.52	-54
Avg		38.51	-68	94.62	-63

T(s): the total runtime; r.T(%): the runtime reduction in percentage.

the exact method. Too much time of the exact method has gone into closing the gap and certifying optimality without necessarily improving the PDH solutions.

Table 4.9 Results of Exact Method.

Type	#Inst	<i>Exact</i>				
		GAP(%)	T(s)	#Cols	It.	No.
Small	W1	0.37	21	6159	189	85
	W2	0.03	35	10978	239	105
	W3	0.06	261	15701	1475	957
	W4	0.01	4	6063	37	11
	W5	0.03	29	8604	158	79
	W6	-	-	-	-	-
	W7	-	-	-	-	-
	W8	0.12	1002	48843	3184	1621
	W9	-	-	-	-	-
	W10	0.02	181	26216	604	279
	W11	0.19	1188	55773	4437	2077
	W12	0.03	138	24720	587	177
	W13	0.14	26880	65992	35569	31243
	W14	0.04	132	19705	492	271
Avg		0.09	2715	26250	4270	3355
Medium	W15	0.15	115	4815	335	227
	W16	0.16	6447	94077	20251	13275
	W17	-	-	-	-	-
	W18	0.06	19373	102867	38165	29381
	W19	0.06	3233	45309	7360	5439
	W20	-	-	-	-	-
	W21	0.13	13	6588	94	29
	W22	-	-	-	-	-
	W23	0.13	33968	106562	50493	42836
Avg		0.12	10525	60036	19450	15198
Large	W24	0.11	28418	307768	41136	21537
	W25	0.1	11094	240422	30802	17925
	W26	0.27	29433	354224	47894	25062
	W27	0.11	8237	243893	23066	11055
	W28	0.66	38934	263212	55711	36054
	W29	0.57	11890	209383	39350	23797
Avg		0.30	21334	269817	39660	22572

GAP(%): the integrality gap; T(s): the total runtime; #Cols: the number of generated columns; It.: the number of iterations; No.: the number of branch-and-bound nodes.

4.6.7 Functional Results

In this section, we give an overview of the functional results of the PDHA, the DHA and the exact method. For each method, we report in Table 4.10 the number of routes ($\#Ro$), followed by the number of tank trucks used ($\#Ta$). In addition, we show the value of escort fees (Ef) and the percentage of escort fees saved ($Sef(\%)$) for each instance and each method. Another important characteristic of the solutions is the percentage of empty space ($Es(\%)$) in tank trucks.

For the PDHA, we save an average of 6% on escort fees for the small instances. For the medium and large ones, we save on average 9% and 10%, respectively. For instances where the value of escort fees saved is equal to zero, note that we cannot serve two marine stations consecutively by the same tank truck due to capacity/the compartments configuration or simply there is no economic interest (stations are too distant for instance). We also observe that the PDHA gives the best match between the number of vehicles, the number of routes used in the solution and the empty space for most instances. For the PDHA, we report a small difference in value (one unit), as compared to the exact method. It must be noted that for the medium instances, the average value reported for an exact solution is greater than for the PDHA, which is due to the number of instances solved.

The DHA shows the highest values of empty space, which explains why it finds a solution with a higher gap (reported in the Table 4.6) than the other methods. Empty space impacts directly the objective value of integer solutions (consequently, the value of the gap increases). This means that the solutions displayed by the DHA technique are of lower quality.

4.6.8 Comparison with PC Solutions

In this section, we compare the solutions of the PDHA to the manual solutions (MS) provided by the PC. These latter are obtained by an expert who spends two working days to find a solution for each week of scheduling. Note that for the MS, we do not have information about the order of visiting clients. Therefore, we only compare the functional aspect. For this reason, we report in Table 4.11 the same information ($\#Ro$, $\#Ta$, $Es(\%)$, $Sef(\%)$) for the MS as we did for PDHA in Table 4.10. In addition, for the PDHA, we report the savings obtained by PDHA vs MS. This is indicated in Table 4.11 by adding the prefix (r.).

From these results, we observe a significant reduction in escort fees, representing more than 200% on the small instances and 300% on the medium and large size instances. For the number of routes, we observe an average reduction of 15% routes for all types of instances. Looking at the numbers of vehicles used, one can notice that for the small instances, we have a small variation ($\pm 1\%$) in $\#Ta$. For the empty space, no significant reduction can be

Table 4.10 Results of PDH, DH and Exact Method: Number of Routes and Vehicles.

Type	Inst.	Ef	<i>PDHA</i>				<i>DHA</i>				<i>Exact</i>			
			#Ro	#Ta	Es(%)	Sef(%)	#Ro	#Ta	Es(%)	Sef(%)	#Ro	#Ta	Es(%)	Sef(%)
Small	W1	55000	13	10	3	0	14	12	4	0	13	10	3	0
	W2	185000	16	11	2	5	16	13	3	5	16	11	2	5
	W3	180000	17	11	1	6	19	11	4	6	17	10	1	6
	W4	140000	17	14	1	0	17	13	2	0	17	13	1	0
	W5	55000	14	12	2	0	15	12	2	0	14	11	2	0
	W6	195000	17	12	5	13	-	-	-	-	-	-	-	-
	W7	195000	18	12	8	13	19	14	10	13	-	-	-	-
	W8	90000	17	11	2	17	17	11	6	17	17	12	2	17
	W9	55000	16	13	2	0	16	12	2	0	-	-	-	-
	W10	195000	20	12	0	0	21	14	1	0	20	12	0	0
	W11	90000	16	13	3	0	17	14	5	0	16	13	3	0
	W12	140000	13	11	0	0	13	11	0	0	13	11	0	0
	W13	165000	19	13	3	6	-	-	-	-	19	13	3	6
	W14	185000	16	12	3	19	17	11	4	19	16	12	3	19
Avg		137500	16	12	3	6	17	12	4	5	16	12	2	5
Medium	W15	120000	17	13	1	0	17	12	4	0	17	12	1	0
	W16	175000	15	12	4	14	17	12	5	14	15	12	4	14
	W17	160000	17	12	4	9	20	14	6	16	-	-	-	-
	W18	220000	18	14	6	11	18	15	6	11	18	13	6	11
	W19	175000	15	10	2	9	15	10	5	9	15	10	2	9
	W20	170000	16	12	4	6	18	12	11	6	-	-	-	-
	W21	105000	14	11	5	14	14	11	5	14	14	10	5	14
	W22	145000	16	12	3	10	-	-	-	-	-	-	-	-
	W23	230000	20	15	3	7	21	15	7	11	20	14	3	7
Avg		166667	16	12	3	9	18	13	6	10	17	12	3	9
Large	W24	190000	20	13	0	8	19	12	1	8	20	13	0	8
	W25	180000	12	10	5	14	14	12	12	14	12	10	5	14
	W26	235000	20	14	3	11	20	16	7	11	20	14	3	11
	W27	180000	12	11	5	14	13	12	12	14	12	11	5	14
	W28	90000	15	10	4	17	14	11	4	17	15	10	4	17
	W29	70000	10	7	4	0	11	8	11	0	10	7	4	0
Avg		157500	15	11	4	10	15	12	8	10	15	11	4	10

Ef: the value of escort fees; **#Ro**: the number of routes; **#Ta**: the number of tank trucks used; **Es(%)**: the percentage of empty space; **Sef(%)**: the percentage of saved escort fees.

observed for the medium instances (1% on average). However, significant reductions of 20% and 28% on average are observed for the small and the large instances, respectively. These savings can be explained by the fact that PDHA finds optimal solutions almost all the time by choosing the tank trucks with the right capacities and compartment configurations that minimize overall cost, among many available ones. This is very hard to do manually due to the huge combinatorics. We can confirm this by the reduction of the time needed to find a (near-)optimal solution to the problem which is very significant: 99% (i.e., less than 100 seconds, as compared to the 14 hours of work needed by the PC expert).

Table 4.11 Comparison MS and PDHA Solutions.

Type	Inst.	<i>MS</i>				<i>PDHA vs MS</i>			
		#Ro	#Ta	Es(%)	Sef(%)	r.#Ro	r.#Ta	r.Es(%)	r.Sef(%)
Small	W1	17	8	11	0	-4	2	-8	0
	W2	21	10	1	0	-5	1	1	-5
	W3	21	13	2	0	-4	-2	0	-6
	W4	18	8	3	0	-1	6	-2	0
	W5	17	11	2	0	-3	1	0	0
	W6	20	14	4	8	-3	-2	1	-5
	W7	21	14	6	8	-3	-2	2	-5
	W8	19	12	2	0	-2	-1	1	-17
	W9	19	12	2	0	-3	1	-1	0
	W10	24	13	8	0	-4	-1	-8	0
	W11	20	13	3	0	-4	0	0	0
	W12	18	13	1	0	-5	-2	0	0
	W13	22	13	4	6	-3	0	-1	0
	W14	20	13	2	5	-4	-1	0	-14
Avg		20	12	4	2	-3	0	-1	-4
Medium	W15	18	10	2	0	-1	3	-1	0
	W16	20	10	1	0	-5	2	3	-14
	W17	19	12	5	9	-2	0	0	0
	W18	21	12	1	11	-3	2	6	0
	W19	17	11	0	0	-2	-1	2	-9
	W20	18	12	2	0	-2	0	2	-6
	W21	19	9	7	0	-5	2	-2	-14
	W22	21	10	7	0	-5	2	-5	-10
	W23	24	12	6	0	-4	3	-3	-7
Avg		20	11	3	2	-3	1	0	-7
Large	W24	24	12	0	0	-4	1	0	-8
	W25	15	9	6	6	-3	1	-1	-8
	W26	24	13	3	4	-4	1	0	-6
	W27	15	9	6	6	-3	2	-1	-8
	W28	18	10	4	0	-3	0	0	-17
	W29	12	7	8	0	-2	0	-3	0
Avg		18	10	4	3	-3	1	-1	-8

#Ro: the number of routes; **#Ta**: the number of used vehicles; **Es(%)**: the percentage of empty space; **Sef(%)**: the percentage of saved escort fees; **r.#Ro**: reduction (in %) of #Ro; **r.#Ta**: the reduction (in %) of #Ta; **r.Es(%)**: the reduction (in %) of Es(%); **r.Sef(%)**: the reduction (in %) of Sef(%).

4.7 Conclusion

In this paper, we present and discuss a real-world example of a multi-depot multi-period petrol replenishment problem of a petroleum company located in Senegal. This problem is

challenging because of: its specific complexity related to loading constraints (from 3 to 15 compartments); its multi-product, multi-depot and multi-trip aspects; some regional restrictions between some clients that could be served by the same tank truck; some exclusions between specific types of petroleum products that may be loaded in the same tank truck; the diversity of the clients' order patterns; clients that can order small quantities which increases the number of clients on a single route.

We propose a branch-and-price approach to find (near) optimal solutions to this problem. In this method, we solve the integrated problem of tank truck loading problem and the routing problem. We devised hashing technique and other techniques to accelerate the generation of potential routes, and, thus, we detect and fix a major bottleneck that tremendously harms the solution process. This paper is a good step towards a change of practice in this field since metaheuristics or a priori column generation are usually used to solve such problems. Indeed, extensive numerical results on 29 real-world instances show that our approach produces better quality solutions (with an average gap of 0.31%) than the manual solutions provided by the PC. These results show the effectiveness and high potential of the proposed approach.

Future developments could address a generalization of the hashing technique to general MCVRP. We believe that this method will be useful to accelerate the solution of CGSPs by a dynamic programming approach. We plan to test our approach on bigger real-world networks, with more complex constraints, such as integrating safer stowage, inventory management with deterministic and stochastic demands and the case where the total supply is less than the total demand (i.e., shortages caused by the refinery extended maintenance duration combined with some disruptions of the imports of gasoline products). Such improvements should lead to a method that may be applied by a broad range of distribution companies.

CHAPITRE 5 ARTICLE 2: COMBINING BENDERS DECOMPOSITION AND COLUMN GENERATION FOR THE PETROL STATION REPLENISHMENT PROBLEM WITH INVENTORY MANAGEMENT

A. Bani, E. M. Er Raqabi, I. El Hallaoui et A. I. Corr  a ont   crit cet article et soumis le 15 mars 2024    la revue Management Science.

5.1 Introduction

Despite the slow introduction of electric vehicles, petrol stations remain crucial nodes in the global supply chain and have been evolving continuously for more than 40 years. Due to this growth, the number and size of vehicles using petrol energy from petrol stations have increased significantly. Consequently, petroleum distribution companies made huge investments in the infrastructure. To capitalize on these investments, these companies rely heavily on business analytics. Nowadays, predictive and prescriptive data analytics are impossible to achieve without effective mathematical optimization tools. The emerging synergy between business analytics and operations research involves new challenges for the next generation of exact algorithms. Even with the remarkable development of general-purpose solvers over the past 10 years, finding optimal solutions for most of the significant combinatorial optimization problems is still out of reach, especially for mixed-integer linear programming (MILP) models with millions of variables.

This paper investigates a complex variant of one of the well-studied problems in the operations research literature: the petrol station replenishment problem (PSRP). Given a set of multi-compartment trucks, the PSRP integrates the truck loading problem (TLP) and inventory routing problem (IRP). The TLP aims to determine the quantity of each type of product to load into the truck’s compartments without exceeding the compartments’ capacities and simultaneously satisfying the demand for each product and each client. A vehicle may carry one or several products, and each truck compartment may be fully or partially filled with a single product when the truck leaves the depot. The IRP seeks to optimize, throughout a given time horizon and day by day, each client demand, by using optimal routes. The solution has to satisfy the route capacity and guarantee that each client has, in each period, a sufficient quantity for daily consumption. Figure 5.1 illustrates a petrol station replenishment process.

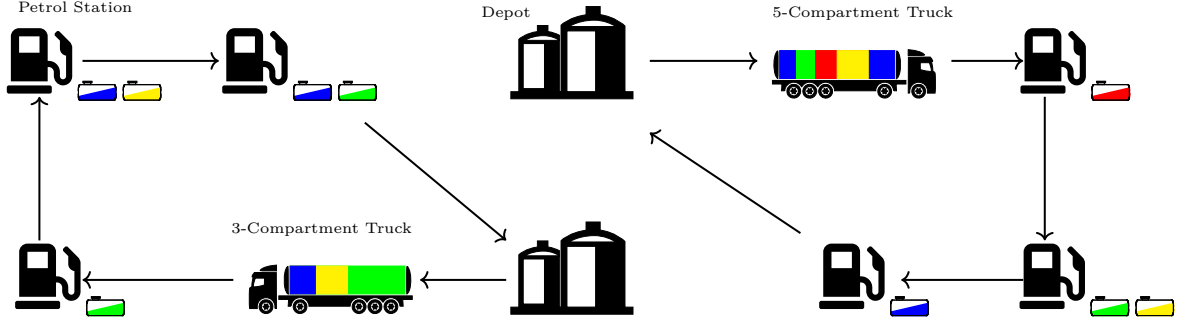


Figure 5.1 Illustration of a Petrol Station Replenishment Process

5.1.1 Our Contribution

Motivated by the recently emerging applications in inventory management (Bertsimas et al., 2016) and routing (Reed et al., 2022), our study addresses the methodological challenge of developing a highly scalable approach for solving large-scale PSRP optimization problems and other similar ones. While addressing a complex variant of the PSRP, this paper is a case study enhancing the development of exact MILP-based solutions for large-scale optimization problems involving inventory management and routing problems. Our approach is based on the intuition of *decoupling* inventory and routing decisions. This intuition comes from observing that when the inventory levels are fixed, the routing can be solved efficiently using existing operations research techniques. We believe our approach is a significant leap in the right direction for three reasons. First, the approach demonstrates the effectiveness of decoupling classical operations research models by using current technology. Indeed, the current technology provides a reason to criticize, review, and implement methods that decouple complex decisions since they are inherently more scalable. Second, several companies now keep a history of good solutions (obtained, for instance, using heuristics or machine learning). Using these solutions as an input to the mathematical model (e.g., warm-start, initial cuts, fixing some variables) will make it possible to reach optimal solution(s) quickly. Third, the insights presented in this paper are easily implementable, scalable, and generic enough to be applied to other problems that involve a synchronized inventory and the routing of multiple resources, and they can be tailored for such purposes.

The complex variant of the PSRP addressed in this paper is the multi-depot, multi-period PSRP with inventory management (MDMPPSRPIM). Thanks to our proposed approach, we can solve in under a minute models whose compact MILP formulations cannot be solved by a general-purpose solver, even for small instances. Our research has yielded five outcomes:

1. We combine Benders Decomposition (BD) and column generation (CG) in a two-phase (*Phase 1* and *Phase 2*) approach to solve exactly the MDMPPSRPIM, with inventory management at the depots and the clients.
2. We use BD to accomplish the *decoupling* intuition. We manage inventory in the Benders master problem and tackle the routing in the Benders subproblem(s). Since the inventory variables are continuous, the Benders master problem is solved quickly, without adding complexity to the problem. The Benders master problem can further be tightened with initial cuts obtained from existing heuristics in the literature or user-specific heuristics. The BD decomposition smartly reduces the complexity that is faced when considering the whole horizon, to a period-specific complexity, which is significantly better.
3. We use CG to solve the Benders subproblems and smartly tackle these subproblems' integrality using a two-phase approach. In *Phase 1*, we relax the integrality of the Benders subproblems and use an efficient label-correcting algorithm for generating columns. In *Phase 2*, we re-establish the integrality of the Benders subproblems and implement a tailored branching scheme to identify integer solutions and obtain integer Benders cuts. We leverage insights from Bani et al. (2023) to solve the Benders subproblems efficiently,
4. We propose several acceleration strategies that significantly improve the performance of the approach, including warm-start, parallelism, hashing techniques, and primal diving heuristics. In particular, warm-start makes it possible to decrease the execution time from hours to minutes for large instances.
5. As a proof of concept, we conduct extensive numerical experiments using real data instances from a geographical zone in West Africa. The results and managerial insights demonstrate the strength of our approach and show that it can obtain near-optimal solutions for real-world instances. We believe that our approach remains scalable for larger instances: decomposition by zone is always possible, the number of clients per route remains moderate due to real-life truck sizes, and the integrality of the Benders subproblems can still be well managed when increasing the size.

The remainder of this paper is organized as follows. We discuss the relevant literature in Section 5.2. We describe the problem and formulate it as a compact MILP model in Sections 5.3 and 5.4. Our solution approach is highlighted in 5.5, followed by computational experiments and their interpretation in Section 5.6. Section 5.7 presents the general conclusion with some directions for future research.

5.2 Literature Review

In the literature, among many problems related to petrol station replenishment operations, the ones significantly linked to the MDMPPSRPIM are the TLP, the IRP, and the PSRP. In the following, we review studies that focus on these problems. Then, we highlight the papers that combine BD and CG. Lastly, we position our paper in the literature and highlight the gaps filled.

5.2.1 The Truck Loading Problem

The TLP studied in this paper can be seen as a specific case of the container loading problem (Pisinger, 2002; Bortfeldt & Wäscher, 2013). The specificity is that each compartment can be allocated to, at most, a single product and a single client. The main reason for this is the prohibition against mixing multiple products, which is the case in several real contexts, including petrol distribution and olive oil collection (Androutsopoulos & Karouti, 2022).

In the petrol distribution context, Cornillier et al. (2008a) tackle the TLP as a subproblem of the PSRP, which was formulated as a set partitioning problem and solved using CG. Any route considered in the CG process involves a TLP that aims to assign the deliverable quantity of each product to the trucks' compartments. The objective of the TLP is to maximize the total petrol quantity delivered. A close TLP was tackled by the same group in Cornillier et al. (2008a) while solving a multi-period PSRP. To check the feasibility of any (heuristically) selected route, they solve the TLP introduced in Cornillier et al. (2008a). In another variant, the TLP arises as a subproblem for checking the feasibility of the constructed routes for the PSRP with time windows (Cornillier et al., 2009). The main difference is that this TLP objective maximizes the total revenue (instead of quantity) from the delivered petrol. Wang et al. (2020) introduce a different TLP for the PSRP with similar compartments. The latter are equipped with flow meters, which make it possible for compartments to be assigned to multiple orders of the same petrol product. Homogeneous compartments make the problem easier since the TLP becomes equivalent to assigning petrol quantities (to the compartments) instead of client-order quantities. Bani et al. (2023) design a hashing technique where, at each iteration, the solved TLPs are memorized, to be used in subsequent iterations. This hashing technique reduces the number of MILPs to be solved in a given iteration (tens of millions in one iteration of CG). Thus, the execution time of the CG subproblem labeling algorithm decreases significantly. Furthermore, after exploring the clients' total orders, they show that it is possible to tighten several resources (e.g., the tank truck capacity and the number of compartments required).

5.2.2 The Inventory Routing Problem

By the end of 2023, the IRP had been around for about 40 years. It can be seen as a combination of the vehicle routing problem (VRP) and the inventory management problem (IMP). The aim is to optimize the delivery of a set of products to a set of clients while respecting several constraints. The optimal solution provides integrated scheduling with the following insights: (1) the optimal inventory management policy, (2) the optimal routes for product delivery, and (3) the optimal delivery schedules. Several metaheuristics, matheuristics, and a few exact methods have been developed for the IRP. More details can be found in Coelho et al. (2014), Archetti & Ljubić (2022), and Cui et al. (2023).

The IRP is known to be a very complex problem, which made it highly popular within the operations research community. In terms of heuristic approaches, many recent contributions are based on metaheuristics or matheuristics methods (Archetti et al., 2012, 2017; Chitsaz et al., 2019; Coelho et al., 2012). As for exact approaches, the first approach for the IRP was developed by Archetti et al. (2007) for the single-vehicle case. More recently, exact approaches for the multi-vehicle case have emerged. Generally, these approaches are based on branch-and-cut frameworks (Adulyasak et al., 2014; Avella et al., 2018; Manousakis et al., 2021). Desaulniers et al. (2016) are the only ones who propose a branch-and-price algorithm with various valid inequalities. In the computational results, they compare their algorithm with the branch-and-cut algorithm proposed by Coelho & Laporte (2013). The results show that neither algorithm outperforms the other, with branch-and-price performing better for large instances with many vehicles and vice versa.

5.2.3 The Petrol Station Replenishment Problem

The PSRP, a known variant of the multi-compartment vehicle routing problem (MCVRP), has been studied intensively in the literature (Dantzig & Ramser, 1959; Ostermeier et al., 2021; Coelho & Laporte, 2015). Compared to the traditional vehicle routing problem with a single product and a single compartment per vehicle, the MCVRP is a generalization with the following characteristics: (1) a client demand might include multiple heterogeneous products; (2) each vehicle has multiple compartments with different configurations and sizes; (3) all products delivered on a route must be assigned a priori to a vehicle's compartments; (4) some products must not be mixed in one compartment. More details can be found in the surveys of Cornillier et al. (2012) and Wang et al. (2020).

The PSRP can be classified into five variants based on these three features: the number of periods, the number of depots, and the consideration of time windows or not. The variants are

(1) the basic PSRP with one depot, a single-period, and no time windows (Brown & Graves, 1981), (2) the multi-period PSRP, referred to as MPPSRP (Triki, 2013; Archetti et al., 2015b), (3) the PSRP with time windows, referred to as PSRPTW (Cornillier et al., 2009), (4) the multi-depot PSRP with time windows, referred to as MDPSRPTW (Cornillier et al., 2012), (5) the multi-depot, multi-period PSRP, referred to as MDMPPSRP (Carotenuto et al., 2018; Bani et al., 2023). Carotenuto et al. (2018) tackle the MDMPPSRP by considering a single product and homogeneous trucks of a fixed capacity. Bani et al. (2023) solve the MDMPPSRP with multiple products and a heterogeneous fleet of multi-compartmented tank trucks.

Most literature papers rely on heuristic approaches to solve the PSRP variants. The main metaheuristics used are (1) cluster first and route second, where the clustering is based on expert considerations and geographic limitation; (2) adapted variable neighborhood search (Hansen & Mladenović, 2001); (3) load first and route second; and (4) tailored heuristics. Vidović et al. (2014) decompose the problem into an inventory and a routing problem. The initial solution found by the heuristic is improved using a local search procedure based on shifting the content of the compartment of a vehicle between days in the planning period. Generally, neither lower bounds nor optimal solutions are confirmed for the variants addressed.

Few papers have attempted to tackle the PSRP exactly, even for small instances. Cornillier et al. (2008a) decompose the basic PSRP using an a priori CG scheme into two subproblems: the TLP and the routing problem (RP). They consider just two clients per route. The TLP is used to assign orders to compartments to maximize the profit. The RP is used to select routes that minimize total transportation costs. Avella et al. (2004) propose a branch-and-price algorithm for the basic PSRP, where they consider multiple clients per route. The authors use a heuristic approach to provide an initial set of columns to initialize the branch-and-price algorithm. Their approach is tested on realistic instances, which consist of 25 clients and 6 trucks of 3 different types with 3 compartments each. Benantar et al. (2016) solve small MPPSRP instances with 15 clients, 2 product types, and 2 compartments using CPLEX. Cornillier et al. (2012) propose an exact model for MDPSRPTW that selects, among a set of feasible routes, the subset that satisfies the demand while maximizing the total daily net revenue. Since the number of possible routes is often exponential, the authors propose an alternative heuristic approach. Bani et al. (2023) address the MDMPPSRP and show that this complex variant can be solved with an exact branch-and-price approach and some derived heuristics.

5.2.4 Combining Benders Decomposition and Column Generation

BD has been applied to tackle large-scale optimization problems (Fischetti et al., 2017) with complicating variables, which, when temporarily fixed, yield problems that are significantly easier to solve. A comprehensive review of the method is available in Rahmaniani et al. (2017). CG is an efficient algorithm for solving large linear programs (LPs) with an exponential number of variables (columns). A detailed review is available in Desaulniers et al. (2006) and Lübbecke & Desrosiers (2005).

In large-scale contexts, combining BD and CG has emerged as a new approach that exploits the primal and the dual problem structures. Cordeau et al. (2001b) employ it for simultaneous aircraft routing and crew scheduling problems. The aircraft routing is the Benders master problem, while the crew scheduling problem is the Benders subproblem. Then, both are solved using CG. The authors introduce a three-phase approach. In the first phase, the integrality restrictions are relaxed. In the second phase, the integrality restrictions are added to the Benders master problem. In the third phase, the integrality restrictions are added to the Benders master problem and subproblem. Zeighami & Soumis (2019) combine BD and CG to tackle crew pairing and crew assignment problems in airline planning. The crew pairing problem represents the Benders master problem, while the (co)pilots subproblems represent the Benders subproblems, i.e., one for the pilots and one for the copilots. Given the exponential number of variables in the subproblems, they use CG to solve them. Wu et al. (2022) investigate a vessel service planning problem (VSPP) in seaports that integrates berth allocation and pilotage planning. The berth allocation problem is the Benders master problem, while the Benders subproblem is the pilotage scheduling problem. Both problems are solved using CG. Unlike Cordeau et al. (2001b), where the last phase is solved just once (which does not guarantee optimality), Zeighami & Soumis (2019) solve the last phase by generating integer Benders cuts until the optimality condition is reached, and Wu et al. (2022) propose a branching scheme that enables their approach to determine an optimal solution to the VSPP.

5.2.5 Paper Positioning

To the best of our knowledge, no paper in the literature tackles the MDMPPSRPIM exactly (See Table 5.1). Heuristics are the dominating techniques, which implies no guarantee of the solutions' quality. The closest papers to ours are Cornillier et al. (2008b) and Vidović et al. (2014). Cornillier et al. (2008b) tackle a similar problem, with one depot and two clients per route, using a multi-phase heuristic. Vidović et al. (2014) use two constructive heuristics to solve the MDMPPSRPIM over five days using homogeneous four-compartment

trucks. They do not consider the depot inventory. Since we decouple inventory and routing, these heuristics, which we improve, can be used as initial cuts or upper bounds in our exact approach.

Table 5.1 Past Close Studies

Paper	Heterogenous Fleet	Multi-Depot	Depot Inventory	Station Inventory	Multi-Period	Multi Visits/Day	#Clients/Route	Exact Method	Heuristic Method
Cornillier et al. (2008b)	✓			✓	✓		≤ 2		✓
Cornillier et al. (2012)	✓	✓					≥ 1		✓
Vidović et al. (2014)	✓			✓	✓		≤ 4		✓
Carotenuto et al. (2018)	✓	✓		✓	✓		≥ 1		✓
Boers et al. (2020)	✓			✓	✓	✓	≥ 1		✓
Bani et al. (2023)	✓	✓			✓	✓	≥ 1	✓	
This paper	✓	✓	✓	✓	✓	✓	≥ 1	✓	

In this paper, we leverage the insights in Bani et al. (2023) and tackle the MDMPPSRPIM exactly by combining BD and CG. Our paper extends the works highlighted above in four directions. First, we consider more realistic cases with inventory management taking place both at the depots and the clients, a heterogeneous fleet of highly compartmentalized trucks, up to four months in the planning horizon, up to four depots, up to four products, up to 65 clients, and no limit on the number of clients per route. The trucks have up to 15 compartments each. Second, we design a tailored two-phase approach that makes it possible to find an optimal solution to the MDMPPSRPIM. While several papers combine BD and CG in the literature (Cordeau et al., 2001b; Zeighami & Soumis, 2019; Wu et al., 2022), the uniqueness of the MDMPPSRPIM does not allow for direct applications of the existing solution approaches based on this combination. Third, we propose several acceleration strategies that have proven to be very effective in improving the performance of the approach. Fourth, the proposed method is generic for a large family of problems with inventory variables and large integrality gaps. It tackles the Benders subproblems integrality well and requires few adjustments that need specific complex development (e.g., the user rules can be easily implemented).

5.3 Problem Description

The MDMPPSRPIM seeks to find an optimal petrol replenishment schedule over a planning horizon (e.g., week, month, quarter) from a set of depots to a set of clients, using a heterogeneous fleet of multi-compartment trucks. An optimal schedule provides the loading of these products into the compartments of the trucks, the timing and quantities of products to be delivered to each client, the delivery routes for the trucks, the inventory levels at the depots and clients at each period, and the assignment of routes to available tank trucks. The optimal petrol replenishment schedule minimizes the total costs involved, including the total distance cost, unloading cost, penalty cost for using tank trucks, penalty for free space in tank trucks, and inventory costs.

The first step is to load different product types onto trucks from the depots. We assume the depots have storage tanks and can order petrol products to fulfill the total demand. A truck may carry more than one product, and each truck compartment may be fully or partially filled with a single product when leaving the depot. The product distribution is carried out by a heterogeneous fleet of tank trucks. We distinguish between several types of tank trucks depending on their capacities and number of compartments. The tank trucks may not all be available at the same time. Each tank truck has several compartments, and each compartment has a fixed capacity.

A client may have one or several types of petrol products. For each Client, the average daily consumptions are known and constant. The inventory of each petrol station is managed by the petroleum distribution company. The auditors of the petroleum company collaborate daily with the managers of the petrol stations to have up-to-date information (e.g., levels of the tanks, sales, and consumption rates). The final decision to replenish a petrol station to a certain level depends on the volumes of petrol available at the petroleum distribution company depots. In other words, the stations do not specify the visit periods or the quantities of each product type to be delivered. The loading time at each depot, the waiting time, and the unloading time at each client are assumed constant.

Table 5.2 General Rules

Rule	Definition
$\mathcal{R}_1$	Some clients cannot accommodate large tank trucks due to accessibility constraints.
$\mathcal{R}_2$	Demand satisfaction: each product ordered by a client must be served by a single tank truck.
$\mathcal{R}_3$	The quantities of products at the depots are limited.
$\mathcal{R}_4$	Only one product ordered by a client is permitted per compartment.
$\mathcal{R}_5$	The daily total travel time must not exceed the total working time in a day.

Some general rules must be enforced (see Table 5.2). Furthermore, a certain percentage of the petroleum products carried transforms into gas en route, due to friction between the liquid and the compartment walls, thereby changing the shape and volume of the compartments. This phenomenon is called sloshing. So, avoiding free space in a tank truck is highly recommended. Thus, we penalize free space in the trucks, as mentioned when discussing the various costs to minimize.

5.4 Mathematical Formulation

In this section, we first describe the MDMPPSRPIM and then highlight the mathematical formulation. We first introduce the MDMPPSRPIM network representation before presenting the MDMPPSRPIM's MILP formulation. For the notation, summarized in Table 5.3, sets are in calligraphic style, parameters are in bold, and decision variables are in italics.

5.4.1 Network Representation

Let $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ be a directed multi-graph, where $\mathcal{V}$ and $\mathcal{A}$ are the node and arc sets, respectively. We illustrate $\mathcal{G}$ in Figure 5.2. In addition to the source (σ) and destination (δ) nodes, $\mathcal{G}$ contains five node types: *depot*, *client*, *client-departure*, *client-arrival* and *client-product*. From a modeling perspective, the last node type (client-product) allows for split delivery if it is less costly. It also fits with the practice where a client's orders (products) are treated separately: each product is stored in a separate tank at the client's site, and the consumption rate differs between products. It follows that each client-product node $n \in \mathcal{N}$ represents a product $p \in \mathcal{P}$ requested by a client. Each client has a pair of dummy nodes: the client-departure and client-arrival nodes. A client-arrival node represents the entry to the client location, and a client-departure node represents the exit from the client location. We denote by $\mathcal{V}'$ the subset of these dummy nodes.

Table 5.3 MDMPPSRPIM Notation

Notation	Definition
Sets	
$\mathcal{P}$	Set of all products $p \in \mathcal{P}$
$\mathcal{N}$	Set of all client-product nodes $n \in \mathcal{N}$: each client-product node corresponds to an order of one product by a client
$\mathcal{N}_p$	Subset of client-products ordering product $p \in \mathcal{P}$
$\mathcal{E}$	Set of depots $e \in \mathcal{E}$
$\mathcal{V}'$	Set of dummy nodes
$\mathcal{V}$	Set of all nodes of the network including the source and destination nodes in addition to the dummy nodes ($\mathcal{V} = \{\mathcal{N} \cup \mathcal{E} \cup \{\sigma, \delta\}\} \cup \mathcal{V}'$)
$\mathcal{S}$	Subset of vertices in the sub-tour
$\mathcal{A}$	Set of possible connections (arcs) that satisfies the Rule $\mathcal{R}_6$
$\mathcal{K}$	Set of tank trucks $k \in \mathcal{K}$
$\mathcal{W}$	Set of planning weeks $w \in \mathcal{W}$
$\mathcal{D}^w$	Set of days in a week $w \in \mathcal{W}$ (numbered from 0 to 6)
$\mathcal{D}_k^w$	Set of availability days $d \in \mathcal{D}_k^w$ for each tank truck $k \in \mathcal{K}$ at week $w \in \mathcal{W}$
$\mathcal{D}$	Set of all planning weekdays $d \in \mathcal{D}$
$\mathcal{L}_k$	Set of compartments $l \in \mathcal{L}_k$ of each tank truck $k \in \mathcal{K}$
$\mathcal{C}$	Set of compatible arc-truck-day combinations. A combination (i, j, k, d) is <i>compatible</i> if the tank truck $k \in \mathcal{K}$ can visit both nodes i and j consecutively in day $d \in \mathcal{D}$ without breaking the rules $\mathcal{R}_1$
Parameters	
$\hat{Q}_{pe}$	Maximum quantity of product $p \in \mathcal{P}$ allowed at depot $e \in \mathcal{E}$
U_{pe}^w	Unit price for a liter of product $p \in \mathcal{P}$ at depot $e \in \mathcal{E}$ at week $w \in \mathcal{W}$
H_{pe}^w	Storage unit cost for a liter of product $p \in \mathcal{P}$ at depot $e \in \mathcal{E}$ for a week $w \in \mathcal{W}$
C_{lk}	Capacity of compartment $l \in \mathcal{L}_k$ in tank truck $k \in \mathcal{K}$
c_{ij}	Cost of visiting node j after node i , including distance
τ_{ij}	Travel time from node i to node j , including the waiting time and the unloading time of the products
T_{max}	Length of a workday
$\hat{Q}_n$	Tank capacity of client-product $n \in \mathcal{N}$
$\hat{O}_n^d$	Daily consumption of client-product $n \in \mathcal{N}$
Ψ_k	Penalty value of using tank truck $k \in \mathcal{K}$
β	Penalty for leaving free space in the tank truck
Δ_k	Capacity of tank truck $k \in \mathcal{K}$, also equal to $\sum_{l \in \mathcal{L}_k} C_{lk}$
Decision Variables	
u_{kd}	= 1 if a tank truck $k \in \mathcal{K}$ is used on day $d \in \mathcal{D}$, 0 otherwise
x_{ij}^{kd}	= 1 if $(i, j, k, d) \in \mathcal{C}$ and the tank truck $k \in \mathcal{K}$ visits node j on day $d \in \mathcal{D}$ after visiting the node i , 0 otherwise
y_{ln}^{kd}	= 1 if the product in client-product $n \in \mathcal{N}$ is loaded in compartment $l \in \mathcal{L}_k$ of tank truck $k \in \mathcal{K}$ on day $d \in \mathcal{D}$, 0 otherwise
q_{pe}^{kd}	≥ 0 quantity of product $p \in \mathcal{P}$ picked up from depot $e \in \mathcal{E}$ by tank truck $k \in \mathcal{K}$ on day $d \in \mathcal{D}$
I_{pe}^w	≥ 0 inventory level of product $p \in \mathcal{P}$ at depot $e \in \mathcal{E}$ at the end of week $w \in \mathcal{W}$
z_{pe}^d	≥ 0 quantity of product $p \in \mathcal{P}$ to be bought at depot $e \in \mathcal{E}$ at day $d \in \mathcal{D}$
o_n^{kd}	≥ 0 quantity of product to be delivered to client-product $n \in \mathcal{N}$ at day $d \in \mathcal{D}$ using tank truck $k \in \mathcal{K}$
I_n^d	inventory level of client-product $n \in \mathcal{N}$ at the end of day $d \in \mathcal{D}$ $[0, \hat{Q}_n]$

The network involves four arc types connecting the different nodes mentioned above: *travel arc*, *pickup arc*, *delivery arc*, and *end of delivery arc*. The travel arcs link the client-departure node of each client to the client-arrival node of the other clients. They also link the depot nodes to client-arrival nodes. The pickup arcs connect the source node with the depots and the depots between them. The delivery arcs link the client-arrival node with the client-product nodes of the same client. The arcs connecting the client-product nodes of a client with the client-departure node are the end of delivery arcs.

We also manage inventory at the depots and the clients' locations. Each lower part of the

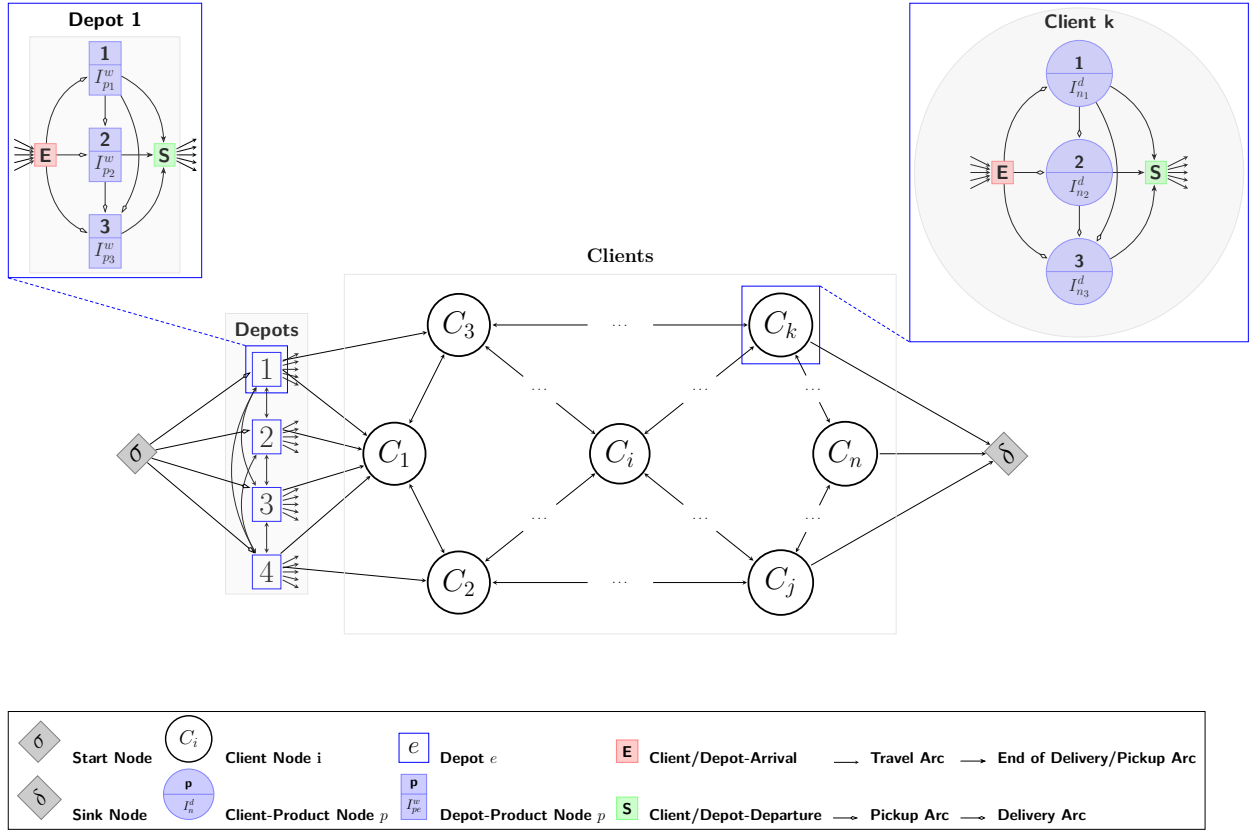


Figure 5.2 Part of the MDMPPSRPIM Network

client-product ($n \in \mathcal{N}$) node represents the inventory level for the period ($d \in \mathcal{D}$). If the quantity $I_n^d \leq \hat{\mathbf{O}}_n^d$, then a delivery of quantity o_n^{kd} must be made for the client-product $n \in \mathcal{N}$ for the period $d \in \mathcal{D}$ using truck $k \in \mathcal{K}$ and we update the inventory level as follows: $I_n^{d+1} = I_n^d - \hat{\mathbf{O}}_n^d + o_n^{kd}$. Otherwise, we update the client-product inventory for the next period as follows: $I_n^{d+1} = I_n^d - \hat{\mathbf{O}}_n^d$. For the depot ($e \in \mathcal{E}$) and a given product $p \in \mathcal{P}$, the update is done weekly ($w \in \mathcal{W}$) as follows: $I_{pe}^{w+1} = I_{pe}^w - q_{pe}^w + z_{pe}^w$ if we buy quantity z_{pe}^w . Otherwise, we update as follows: $I_{pe}^{w+1} = I_{pe}^w - q_{pe}^w$, where q_{pe}^w is the quantity of product $p \in \mathcal{P}$ delivered from the depot $e \in \mathcal{E}$ during $w \in \mathcal{W}$.

5.4.2 The MDMPPSRPIM Formulation

We present a compact MILP formulation for the MDMPPSRPIM, which recalls an extension of the MDMPPRP formulation studied by Bani et al. (2023). Using the notation in Table 5.3, the compact formulation integrates the TLP and the IRP as follows:

$$\min_{\substack{x,y,u,q \\ I,o,z}} \sum_{k \in \mathcal{K}} \sum_{d \in \mathcal{D}_k} \left(\sum_{(i,j) \in \mathcal{A}} c_{ij} x_{ij}^{kd} + \Psi_k u_{kd} + \beta \left(\Delta_k u_{kd} - \sum_{l \in \mathcal{L}_k} \sum_{n \in \mathcal{N}} C_{lk} y_{ln}^{kd} \right) \right) + \sum_{w \in \mathcal{W}} \sum_{p \in \mathcal{P}} \sum_{e \in \mathcal{E}} (U_{pe}^w z_{pe}^w + H_{pe}^w I_{pe}^w) \quad (\text{MDMPPSRPIM})$$

$$\text{s.t.: } I_{pe}^w - \sum_{k \in \mathcal{K}} \sum_{d \in \mathcal{D}_k} q_{pe}^{kd} + z_{pe}^w = I_{pe}^{w+1} \quad \forall p \in \mathcal{P}, \forall e \in \mathcal{E}, \forall w \in \mathcal{W} \quad (5.1)$$

$$I_n^d - \hat{O}_n^d + \sum_{k \in \mathcal{K}} o_n^{kd} = I_n^{d+1} \quad \forall n \in \mathcal{N}, \forall d \in \mathcal{D} \quad (5.2)$$

$$\sum_{e \in \mathcal{E}} (I_{pe}^w - I_{pe}^{w+1} + z_{pe}^w) = \sum_{n \in \mathcal{N}_p} \sum_{d \in \mathcal{D}^w} (\hat{O}_n^d + I_n^{d+1} - I_n^d) \quad \forall p \in \mathcal{P}, \forall w \in \mathcal{W} \quad (5.3)$$

$$o_n^{kd} \leq \hat{Q}_n^d \sum_{(i,n) \in \mathcal{A}^{kd}} x_{in}^{kd} \quad \forall n \in \mathcal{N}, \forall k \in \mathcal{K}, \forall d \in \mathcal{D}_k \quad (5.4)$$

$$\sum_{k \in \mathcal{K}} C_{lk} y_{ln}^{kd} = o_n^{kd} \quad \forall n \in \mathcal{N}, \forall k \in \mathcal{K}, \forall d \in \mathcal{D}_k \quad (5.5)$$

$$\sum_{(i,n) \in \mathcal{A}} \sum_{k \in \mathcal{K}} \sum_{d \in \mathcal{D}_k} x_{in}^{kd} \leq 1 \quad \forall n \in \mathcal{N}, \forall w \in \mathcal{W} \quad (5.6)$$

$$q_{pe}^{kd} \leq \hat{Q}_{pe} \sum_{i \in \{\mathcal{E} \setminus \{e\}\} \cup \{\sigma\}} x_{ie}^{kd} \quad \forall p \in \mathcal{P}, \forall e \in \mathcal{E}, \forall k \in \mathcal{K}, \forall d \in \mathcal{D}_k \quad (5.7)$$

$$x_{ij}^{kd} \leq u_{kd} \quad \forall (i,j) \in \mathcal{A}, \forall k \in \mathcal{K}, \forall d \in \mathcal{D}_k \quad (5.8)$$

$$\sum_{n \in \mathcal{N}} y_{ln}^{kd} \leq 1 \quad \forall k \in \mathcal{K}, \forall l \in \mathcal{L}_k, \forall d \in \mathcal{D}_k \quad (5.9)$$

$$\sum_{e \in \mathcal{E}} q_{pe}^{kd} = \sum_{n \in \mathcal{N}_p} \sum_{l \in \mathcal{L}_k} C_{lk} y_{ln}^{kd} \quad \forall p \in \mathcal{P}, \forall k \in \mathcal{K}, \forall d \in \mathcal{D}_k \quad (5.10)$$

$$\sum_{(i,j) \in \mathcal{A}} \tau_{ij} x_{ij}^{kd} \leq T_{max} \quad \forall k \in \mathcal{K}, \forall d \in \mathcal{D}_k \quad (5.11)$$

$$\sum_{i \in \mathcal{V}} x_{ij}^{kd} - \sum_{i \in \mathcal{V}} x_{ji}^{kd} = \begin{cases} -1 & \text{if } j = \sigma \\ 0 & \text{if } j \in \mathcal{V} \setminus \{\sigma, \delta\} \\ 1 & \text{if } j = \delta \end{cases} \quad \forall k \in \mathcal{K}, \forall d \in \mathcal{D}_k \quad (5.12)$$

$$\sum_{i,j \in \mathcal{S}} x_{ij}^{kd} \leq |\mathcal{S}| - 1 \quad \forall k \in \mathcal{K}, \forall d \in \mathcal{D}_k \quad (\mathcal{S} \subset \mathcal{V}, 2 \leq |\mathcal{S}| \leq |\mathcal{V}| - 2) \quad (5.13)$$

$$x_{ij}^{kd} \in \{0, 1\} \quad \forall (i, j, k, d) \in \mathcal{C} \quad (5.14)$$

$$u_{kd} \in \{0, 1\} \quad \forall k \in \mathcal{K}, \forall d \in \mathcal{D}_k \quad (5.15)$$

$$y_{ln}^{kd} \in \{0, 1\} \quad \forall n \in \mathcal{N}, \forall k \in \mathcal{K}, \forall l \in \mathcal{L}_k, \forall d \in \mathcal{D}_k \quad (5.16)$$

$$q_{pe}^{kd} \geq 0 \quad \forall k \in \mathcal{K}, \forall p \in \mathcal{P}, \forall e \in \mathcal{E}, \forall d \in \mathcal{D}_k \quad (5.17)$$

$$0 \leq z_{pe}^w \leq \hat{Q}_{pe} \quad \forall p \in \mathcal{P}, \forall e \in \mathcal{E}, \forall w \in \mathcal{W} \quad (5.18)$$

$$0 \leq I_{pe}^w \leq \hat{Q}_{pe} \quad \forall p \in \mathcal{P}, \forall e \in \mathcal{E}, \forall w \in \mathcal{W} \quad (5.19)$$

$$0 \leq I_n^d \leq \hat{Q}_n \quad \forall n \in \mathcal{N}, \forall d \in \mathcal{D} \quad (5.20)$$

$$o_n^{kd} \geq 0 \quad \forall n \in \mathcal{N}, \forall k \in \mathcal{K}, \forall d \in \mathcal{D}_k \quad (5.21)$$

The objective function minimizes the total cost. It includes the total distance cost, unloading cost (first term), the cost of using tank trucks (second term), the penalty for free space in

tank trucks (third term), and the inventory costs (last term). Constraints (5.1) manage the inventory at the depot while satisfying Rule $\mathcal{R}_3$ on available quantities. Constraints (5.2) manage the inventory at the clients' sites. Constraints (5.3) guarantee that, for a given $p \in \mathcal{P}$ and $w \in \mathcal{W}$, the total quantity that goes out of the depots must be equal to the total quantity delivered to clients. Constraints (5.4) ensure that the delivery can take place only when the client $n \in \mathcal{N}$ is visited. Constraints (5.5) indicate that the sum of the compartment capacities fulfilling the order of client-product $n \in \mathcal{N}_p$ must correspond exactly to the quantity requested by the client. Constraints (5.6) specify that each client-product $n \in \mathcal{N}_p$ can be visited at most once a week, as indicated by Rule $\mathcal{R}_2$. Constraints (5.7) ensure that a tank truck can load products from the depot $e \in \mathcal{E}$ only after visiting it. Constraints (5.8) ensure that a tank truck can only be used if it is available on day $d \in \mathcal{D}$. Constraints (5.9) impose that for each tank truck $k \in \mathcal{K}$, at most one client-product $n \in \mathcal{N}_p$ is allowed in the same compartment on a given day $d \in \mathcal{D}$ (Rule $\mathcal{R}_4$). Constraints (5.10) specify that the quantities of each product $p \in \mathcal{P}$ delivered to clients must be loaded from depots. Constraints (5.11) make sure that the daily total travel time falls within the length of a workday $d \in \mathcal{D}$ for each tank truck driver, as requested by Rule $\mathcal{R}_5$. Constraints (5.12) and (5.13) are the flow conservation constraints and sub-tour elimination constraints, respectively. Finally, the decision variable conditions are given by (5.14)–(5.21).

We observe that the second and third terms of the objective function with constraints (5.5)–(5.10), (5.15), and (5.16) define the TLP. The remaining terms in the objective and constraints model the IRP. Without loss of generality, we assume that the above problem is integer feasible and bounded, which is the case in real life. The compact MILP formulation is too large and complex to be used in practice. Still, it helps to understand the problem and motivates the decomposition method in Section 5.5.

5.5 Solution Methodology

In the previous section, we formulated the MDMPPSRPIM using a compact MILP formulation. For this formulation, the complexity comes essentially from the very large integrality gap, which is due to the mix of continuous and binary variables in the same constraints and the big-M constraints (e.g., Constraints 5.4, 5.5, and 5.7). Thus, it is practically impossible to solve even relatively small instances using a general-purpose optimization solver. In this section, we propose an exact approach combining BD and CG to solve the MDMPPSRPIM. We first present a route formulation for MDMPPSRPIM. Then, we motivate the BD choice. Following that, we describe the exact approach combining BD and CG. Lastly, we highlight some acceleration strategies.

5.5.1 The Route Formulation

We reformulate the MDMPPSRPIM using routes. For this purpose, let $\Omega = \{1, \dots, |\Omega|\}$ be the set of feasible routes (columns). We define a binary variable ρ_ϱ for each route $\varrho \in \Omega$. It takes the value 1 if route ϱ with cost c_ϱ is selected, and 0 otherwise. The cost $\mathbf{c}_\varrho$ corresponds to the sum of all the costs associated with route $\varrho \in \Omega$.

We introduce the following additional notation. The coefficient $\tilde{\mathbf{s}}_\varrho^{nw}$ is set to 1 if the tank truck used for route $\varrho \in \Omega$ serves the client-product $n \in \mathcal{N}$ during the week $w \in \mathcal{W}$, and 0 otherwise. The coefficient $\tilde{\mathbf{u}}_\varrho^{kwd}$ is equal to 1 if the route ϱ is served by tank truck $k \in \mathcal{K}$ in day $d \in \mathcal{D}$ of week $w \in \mathcal{W}$, and 0 otherwise. The coefficient $\tilde{\mathbf{q}}_\varrho^{pe}$ is the quantity of product $p \in \mathcal{P}$ loaded from depot $e \in \mathcal{E}$ to satisfy the order for the same product on route $\varrho \in \Omega$. The coefficient $\tilde{\mathbf{o}}_\varrho^{nd}$ is the quantity of product p delivered to client-product n in day d using route $\varrho \in \Omega$. The resulting route reformulation of MDMPPSRPIM is as follows:

$$\min_{\rho, z, I} \quad \sum_{\varrho \in \Omega} \mathbf{c}_\varrho \rho_\varrho + \sum_{w \in \mathcal{W}} \sum_{p \in \mathcal{P}} \sum_{e \in \mathcal{E}} (\mathbf{U}_{pe}^w z_{pe}^w + \mathbf{H}_{pe}^w I_{pe}^w) \quad (\text{MDMPPSRPIM}_\rho)$$

$$\text{s.t.:} \quad I_{pe}^w - \sum_{\varrho \in \Omega} \tilde{\mathbf{q}}_\varrho^{pe} \rho_\varrho + z_{pe}^w = I_{pe}^{w+1} \quad \forall p \in \mathcal{P}, \forall e \in \mathcal{E}, \forall w \in \mathcal{W} \quad (5.22)$$

$$I_n^d - \hat{\mathbf{O}}_n^d + \sum_{\varrho \in \Omega} \tilde{\mathbf{o}}_\varrho^{nd} \rho_\varrho = I_n^{d+1} \quad \forall n \in \mathcal{N}, \forall d \in \mathcal{D} \quad (5.23)$$

$$\sum_{e \in \mathcal{E}} (I_{pe}^w - I_{pe}^{w+1} + z_{pe}^w) = \sum_{n \in \mathcal{N}_p} \sum_{d \in \mathcal{D}^w} (\hat{\mathbf{O}}_n^d + I_n^{d+1} - I_n^d) \quad \forall p \in \mathcal{P}, \forall w \in \mathcal{W} \quad (5.24)$$

$$\sum_{\varrho \in \Omega} \tilde{\mathbf{s}}_\varrho^{nw} \rho_\varrho \leq 1 \quad \forall n \in \mathcal{N}, \forall w \in \mathcal{W} \quad (5.25)$$

$$\sum_{\varrho \in \Omega} \tilde{\mathbf{u}}_\varrho^{kwd} \rho_\varrho \leq 1 \quad \forall k \in \mathcal{K}, \forall w \in \mathcal{W}, \forall d \in \mathcal{D}_k^w \quad (5.26)$$

$$0 \leq z_{pe}^w \leq \hat{\mathbf{Q}}_{pe} \quad \forall p \in \mathcal{P}, \forall e \in \mathcal{E}, \forall w \in \mathcal{W} \quad (5.27)$$

$$0 \leq I_{pe}^w \leq \hat{\mathbf{Q}}_{pe} \quad \forall p \in \mathcal{P}, \forall e \in \mathcal{E}, \forall w \in \mathcal{W} \quad (5.28)$$

$$0 \leq I_n^d \leq \hat{\mathbf{Q}}_n \quad \forall n \in \mathcal{N}, \forall d \in \mathcal{D} \quad (5.29)$$

$$\rho_\varrho \in \{0, 1\} \quad \forall \varrho \in \Omega \quad (5.30)$$

The objective function minimizes the total cost. Constraints (5.22) manage the inventory at the depot for each product. Constraints (5.23) manage the inventory at the client. Constraints (5.24) equals the quantities that go out of the depots with the quantities received by the clients. Constraints (5.25) indicate that each client-product must be visited at most once a week. We note that we may perform multiple visits per day if we visit two client-products of the same client on the same day. Constraints (5.26) ensure that we can use tank truck $k \in \mathcal{K}$ at most once per day $d \in \mathcal{D}$. Constraints (5.28) ensure that the quantities loaded from each depot do not exceed what is available. Constraints (5.29) ensure that the daily

quantities delivered to each client-product n respect their maximum capacity limit. Finally, integrality constraints (5.30) are imposed on the route variables ρ_ρ for each route $\rho \in \Omega$.

In terms of the integrality gap, the route formulation is better than the compact one at the expense of a combinatorial number of routes. In practice, generating a subset of routes using CG would be enough. However, tackling model MDMPPSRPIM $_\rho$ using the Dantzig-Wolfe decomposition (Vanderbeck, 2000) remains complex because of the following:

- If we keep the constraints (5.22–5.23) in the Dantzig-Wolfe master problem, although the subproblems can be easily broken down by day and by truck and can be solved in parallel, these subproblems are more complex because we do not know a priori the clients who will be delivered to on a given day. In addition, the dynamic programming algorithm (used to solve the CG subproblem, which is an elementary shortest path problem with resource constraints) is very long because we generate a large number of labels since all the clients are present in the cyclic network. Furthermore, this decomposition also suffers from a major loss of information on the inventory at the clients' sites and at the depots, which makes the generated routes not compatible with the master problem despite their negative reduced cost: the routes obtained from the subproblem consider only a single period without access to the actual client inventories, and the delivered quantities by routes do not respect inventory balance constraints. For this reason, the master problem does not find a combination of routes that satisfies all the constraints. Thus, the decomposition will require a lot of iterations and will lead to a huge number of routes in the master problem.
- If we keep the constraints (5.22–5.23) in the Dantzig-Wolfe subproblem(s), we ensure that all the generated routes are usable in the master problem without any loss of information. However, the subproblem(s) become even more complex for the dynamic programming algorithm because, in addition to the fact that it contains all the clients (the network is larger, with more nodes, more arcs, and more labels), we cannot break it down by periods because of the presence of constraints that link days d and $d + 1$.

The analysis highlighted above, aligned with the decoupling intuition, leads us to observe that when we fix the inventory variables (z, I) , the problem becomes relatively easier. One of the ways to decouple inventory and routing decisions is BD, which is presented next.

5.5.2 The Benders Reformulation

In this section, we present the Benders reformulation by highlighting the Benders subproblem and the Benders master problem.

The Benders Subproblem

After fixing the complicating variables $\bar{I}_{pe}^w$, $\bar{I}_n^d$, and $\bar{z}_{pe}^w$ (a fixed variable has a bar on it), the Benders primal subproblem is the following:

$$\min_{\rho} \sum_{\varrho \in \Omega} c_{\varrho} \rho_{\varrho} \quad (\text{BPSP})$$

$$\text{s.t.:} \quad \sum_{\varrho \in \Omega} \tilde{q}_{\varrho}^{pe} \rho_{\varrho} = \bar{I}_{pe}^w - \bar{I}_{pe}^{w+1} + \bar{z}_{pe}^w \quad \forall p \in \mathcal{P}, \forall e \in \mathcal{E}, \forall w \in \mathcal{W} \quad (5.31)$$

$$\sum_{\varrho \in \Omega} \tilde{o}_{\varrho}^{nd} \rho_{\varrho} = \hat{O}_n^d + \bar{I}_n^{d+1} - \bar{I}_n^d \quad \forall n \in \mathcal{N}, \forall d \in \mathcal{D} \quad (5.32)$$

$$\sum_{\varrho \in \Omega} \tilde{s}_{\varrho}^{nw} \rho_{\varrho} \leq 1 \quad \forall n \in \mathcal{N}, \forall w \in \mathcal{W} \quad (5.33)$$

$$\sum_{\varrho \in \Omega} \tilde{u}_{\varrho}^{kwd} \rho_{\varrho} \leq 1 \quad \forall k \in \mathcal{K}, \forall w \in \mathcal{W}, \forall d \in \mathcal{D}_k^w \quad (5.34)$$

$$\rho_{\varrho} \in \{0, 1\} \quad \forall \varrho \in \Omega \quad (5.35)$$

At each iteration, we are supposed to solve the dual of BPSP to obtain either an optimality or feasibility cut. Let $\boldsymbol{\pi}^{(5.31)} = (\pi_{pew}^{(5.31)} \mid p \in \mathcal{P}, e \in \mathcal{E}, w \in \mathcal{W})$, $\boldsymbol{\pi}^{(5.32)} = (\pi_{nd}^{(5.32)} \mid n \in \mathcal{N}, d \in \mathcal{D})$, $\boldsymbol{\pi}^{(5.33)} = (\pi_{nw}^{(5.33)} \leq 0 \mid n \in \mathcal{N}, w \in \mathcal{W})$ and $\boldsymbol{\pi}^{(5.34)} = (\pi_{kwd}^{(5.34)} \leq 0 \mid k \in \mathcal{K}, w \in \mathcal{W}, d \in \mathcal{D}_k^w)$ be the dual variables associated with constraints (5.31)–(5.34), respectively. The dual of the relaxed BPSP is then

$$\min_{\boldsymbol{\pi}} \quad \sum_{p \in \mathcal{P}} \sum_{e \in \mathcal{E}} \sum_{w \in \mathcal{W}} (\bar{I}_{pe}^w - \bar{I}_{pe}^{w+1} + \bar{z}_{pe}^w) \pi_{pew}^{(5.31)} + \sum_{n \in \mathcal{N}} \sum_{d \in \mathcal{D}} (\hat{O}_n^d + \bar{I}_n^{d+1} - \bar{I}_n^d) \pi_{nd}^{(5.32)} \quad (\text{BDSP})$$

$$\begin{aligned} & + \sum_{n \in \mathcal{N}} \sum_{w \in \mathcal{W}} \pi_{nw}^{(5.33)} + \sum_{k \in \mathcal{K}} \sum_{w \in \mathcal{W}} \sum_{d \in \mathcal{D}_k^w} \pi_{kwd}^{(5.34)} \\ \text{s.t.:} \quad & \sum_{p \in \mathcal{P}} \sum_{e \in \mathcal{E}} \sum_{w \in \mathcal{W}} \tilde{q}_{\varrho}^{pe} \pi_{pew}^{(5.31)} + \sum_{n \in \mathcal{N}} \sum_{d \in \mathcal{D}} \tilde{o}_{\varrho}^{nd} \pi_{nd}^{(5.32)} \\ & + \sum_{n \in \mathcal{N}} \sum_{w \in \mathcal{W}} \tilde{s}_{\varrho}^{nw} \pi_{nw}^{(5.33)} + \sum_{k \in \mathcal{K}} \sum_{w \in \mathcal{W}} \sum_{d \in \mathcal{D}_k^w} \tilde{u}_{\varrho}^{kwd} \pi_{kwd}^{(5.34)} \leq c_{\varrho} \quad \forall \varrho \in \Omega \end{aligned} \quad (5.36)$$

$$\pi_{nw}^{(5.33)} \leq 0 \quad \forall n \in \mathcal{N}, \forall w \in \mathcal{W} \quad (5.37)$$

$$\pi_{kwd}^{(5.34)} \leq 0 \quad \forall k \in \mathcal{K}, \forall w \in \mathcal{W}, \forall d \in \mathcal{D}_k \quad (5.38)$$

Instead of solving the BDSP, we solve an equivalent problem in the literature (Bani et al., 2023). The next theorem highlights this equivalence.

Theorem 1. *Solving the BDSP is equivalent to solving $|\mathcal{W}|$ dual SPPs.*

Proof. Since we fixed inventory levels, we observe that the BDSP can be decomposed into $|\mathcal{W}|$ problems. Furthermore, we know the total quantity delivered to each client-product for each day d . We note this quantity $\bar{o}_n^d = \hat{O}_n^d + \bar{I}_n^{d+1} - \bar{I}_n^d$. We also know the quantities

of each product p that must be taken from the depot e each week w , which we denote $\bar{q}_{pe}^w = \bar{I}_{pe}^w - \bar{I}_{pe}^{w+1} + \bar{z}_{pe}^w$. For a given week $w \in \mathcal{W}$, the weekly problem is this:

$$\max_{\pi, w} \sum_{p \in \mathcal{P}} \sum_{e \in \mathcal{E}} \bar{q}_{pe}^w \pi_{pew}^{(5.31)} + \sum_{n \in \mathcal{N}} \sum_{d \in \mathcal{D}^w} \bar{o}_n^d \pi_{nd}^{(5.32)} + \sum_{n \in \mathcal{N}} \pi_{nw}^{(5.33)} + \sum_{k \in \mathcal{K}} \sum_{d \in \mathcal{D}_k^w} \pi_{kwd}^{(5.34)} \quad (\text{WBDSPP})$$

$$\begin{aligned} \text{s.t.:} \quad & \sum_{p \in \mathcal{P}} \sum_{e \in \mathcal{E}} \bar{q}_{pe}^w \pi_{pew}^{(5.31)} + \sum_{n \in \mathcal{N}} \sum_{d \in \mathcal{D}^w} \bar{o}_n^d \pi_{nd}^{(5.32)} \\ & + \sum_{n \in \mathcal{N}} \bar{s}_n^w \pi_{nw}^{(5.33)} + \sum_{k \in \mathcal{K}} \sum_{d \in \mathcal{D}_k^w} \bar{u}_k^{kd} \pi_{kwd}^{(5.34)} \leq c_\rho \quad \forall \rho \in \Omega \end{aligned} \quad (5.39)$$

$$\pi_{nw}^{(5.33)} \leq 0 \quad \forall n \in \mathcal{N} \quad (5.40)$$

$$\pi_{kwd}^{(5.34)} \leq 0 \quad \forall k \in \mathcal{K}, \forall d \in \mathcal{D}_k^w \quad (5.41)$$

The dual of SPP (Appendix A) of Bani et al. (2023) is

$$\max_{\pi} \sum_{p \in \mathcal{P}} \sum_{e \in \mathcal{E}} \bar{q}_{pe} \alpha_{pe}^{(1)} + \sum_{n \in \mathcal{N}} \alpha_n^{(2)} + \sum_{k \in \mathcal{K}} \sum_{d \in \mathcal{D}_k} \alpha_{kd}^{(3)} \quad (\text{DSPP})$$

$$\text{s.t.:} \quad \sum_{p \in \mathcal{P}} \sum_{e \in \mathcal{E}} \bar{q}_{pe} \alpha_{pe}^{(1)} + \sum_{n \in \mathcal{N}} \bar{s}_n \alpha_n^{(2)} + \sum_{k \in \mathcal{K}} \sum_{d \in \mathcal{D}_k} \bar{u}_k^{kd} \alpha_{kd}^{(3)} \leq c_\rho \quad \forall \rho \in \Omega \quad (5.42)$$

$$\alpha_{pe}^{(1)} \leq 0 \quad \forall n \in \mathcal{N} \quad (5.43)$$

$$\alpha_{kd}^{(3)} \leq 0 \quad \forall k \in \mathcal{K}, \forall d \in \mathcal{D}_k \quad (5.44)$$

The SPP is always feasible and bounded (Bani et al., 2023). Thus, the DSPP is always feasible and bounded. For a given $w \in \mathcal{W}$, let

$$\begin{aligned} \alpha_{pe}^{(1)} &= \pi_{pew}^{(5.31)} & \forall p \in \mathcal{P}, \forall e \in \mathcal{E} & \quad (\text{Dual Equalities}) \\ \alpha_n^{(2)} &= \sum_{d \in \mathcal{D}^w} \bar{o}_n^d \pi_{nd}^{(5.32)} + \pi_{nw}^{(5.33)} & \forall n \in \mathcal{N} \\ \bar{s}_n \alpha_n^{(2)} &= \sum_{d \in \mathcal{D}^w} \bar{o}_n^d \pi_{nd}^{(5.32)} + \bar{s}_n^w \pi_{nw}^{(5.33)} & \forall n \in \mathcal{N} \\ \alpha_{kd}^{(3)} &= \pi_{kwd}^{(5.34)} & \forall k \in \mathcal{K}, \forall d \in \mathcal{D}_k^w \end{aligned}$$

Using the Dual Equalities above, we deduce that the WBDSPP is always feasible and bounded. Furthermore, the DSPP and the WBDSPP have the same optimal value, which implies that solving the DSPP is equivalent to solving the WBDSPP. Thus, solving $|\mathcal{W}|$ DSPPs is equivalent to solving the BDSP. $\square$

Theorem 1 is interesting because, instead of solving the BDSP, it involves solving $|\mathcal{W}|$ DSPP problems (one problem per week), and then using the Dual Equalities equations to find the corresponding dual solutions for the BDSP. These dual solutions will be used to construct the Benders optimality cuts for the Benders master problem. We recall that the insight is

that BD makes it possible to decouple the inventory management decisions from the routing and loading problems, and thus each of the Benders subproblems obtained is similar to the weekly problem tackled by Bani et al. (2023).

The Benders Master Problem

Let $\mathcal{F}$ be the feasible region of the BDSP and let $\Upsilon_{\mathcal{F}}$ be the set of extreme points of $\mathcal{F}$. Introducing the additional free variable μ , the Benders master problem can be formulated as follows:

$$\min_{z, I} \quad \sum_{w \in \mathcal{W}} \sum_{p \in \mathcal{P}} \sum_{e \in \mathcal{E}} (U_{pe}^w z_{pe}^w + H_{pe}^w I_{pe}^w) + \mu \quad (\text{BMP})$$

$$\begin{aligned} \text{s.t.:} \quad \mu &\geq \sum_{p \in \mathcal{P}} \sum_{e \in \mathcal{E}} \sum_{w \in \mathcal{W}} (I_{pe}^w - I_{pe}^{w+1} + z_{pe}^w) \pi_{pew}^{(5.31)} \\ &\quad + \sum_{n \in \mathcal{N}} \sum_{d \in \mathcal{D}} (\hat{O}_n^d + I_n^{d+1} - I_n^d) \pi_{nd}^{(5.32)} \\ &\quad + \sum_n \sum_w \pi_{nw}^{(5.33)} + \sum_{k \in \mathcal{K}} \sum_{w \in \mathcal{W}} \sum_{d \in \mathcal{D}_k^w} \pi_{kwd}^{(5.34)} \quad (\pi^{(5.31)}, \pi^{(5.32)}, \pi^{(5.33)}, \pi^{(5.34)}) \in \Upsilon_{\mathcal{F}} \end{aligned} \quad (5.45)$$

$$\sum_{e \in \mathcal{E}} (I_{pe}^w - I_{pe}^{w+1} + z_{pe}^w) = \sum_{n \in \mathcal{N}_p} \sum_{d \in \mathcal{D}^w} (\hat{O}_n^d + I_n^{d+1} - I_n^d) \quad \forall p \in \mathcal{P}, \forall w \in \mathcal{W} \quad (5.46)$$

$$0 \leq z_{pe}^w \leq \hat{Q}_{pe} \quad \forall p \in \mathcal{P}, \forall e \in \mathcal{E}, \forall w \in \mathcal{W} \quad (5.47)$$

$$0 \leq I_{pe}^w \leq \hat{Q}_{pe} \quad \forall p \in \mathcal{P}, \forall e \in \mathcal{E}, \forall w \in \mathcal{W} \quad (5.48)$$

$$0 \leq I_n^d \leq \hat{Q}_n \quad \forall n \in \mathcal{N}, \forall d \in \mathcal{D} \quad (5.49)$$

We recall that since each DSPP is always feasible and bounded (see Theorem 1), the feasible region $\mathcal{F}$ does not contain extreme rays, and consequently we only have extreme points, i.e., Benders optimality cuts. Model BMP contains more constraints than the LP relaxation of MDMPPSRPIM $_{\rho}$, but most optimality cuts are inactive in the optimal solution. Therefore, these Benders cuts do not need to be generated exhaustively. Instead, an iterative approach is used to generate a subset of cuts, which is sufficient to identify an optimal solution. We highlight the two-phase approach next.

5.5.3 Column and Benders Cut Generation

We are now ready to describe the overall algorithm, referred to as the two-phase Benders decomposition, first introduced by Cordeau et al. (2001b). We adapt it to our context where we solve $|\mathcal{W}|$ DSPPs using CG. We highlight the algorithm in Figure 5.3.

The algorithm consists of two phases: *Phase 1* and *Phase 2*. In Phase 1, the integrality conditions are relaxed, and at each iteration of the BD, the LP relaxation of the BMP and

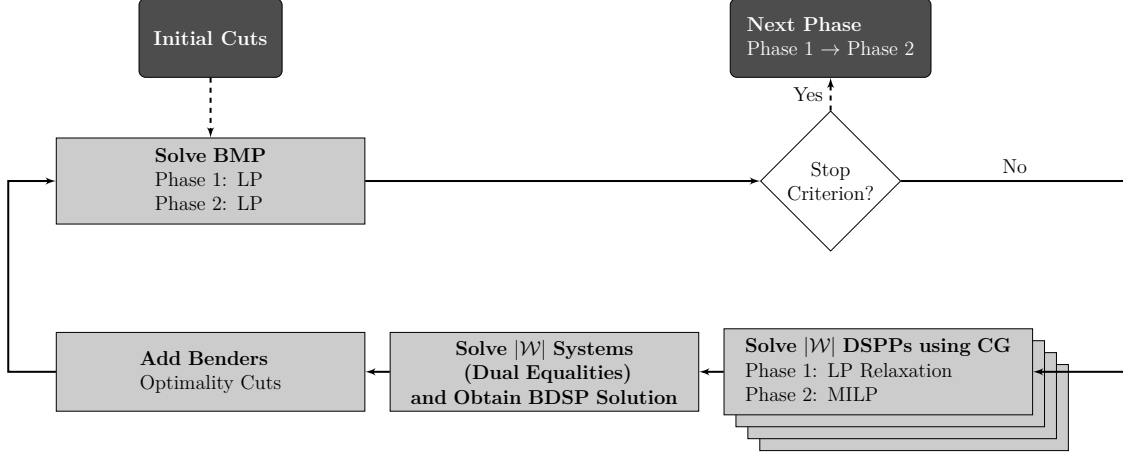


Figure 5.3 Flowchart of the Two-Phase Approach

the $|\mathcal{W}|$ SPPs are solved. Each SPP is solved using CG. All the Benders optimality cuts generated in the first phase are kept in the BMP. In Phase 2, the integrality constraints are introduced for the SPP variables. The resulting MILP is solved using CG embedded in a branch-and-bound framework. We recall that we solve the Dual Equalities system (see Proof of Theorem1) to obtain the BDSP solution, which is used to obtain the Benders optimality cuts.

An important aspect to highlight is the following. In Phase 2, we face two cases. The first case is when the relaxed solution of the SPP is an integer. This case mainly happens when the problem columns are very sparse. In such a case, we collect the dual solution of the SPP and obtain the corresponding Benders cut. The second case is when the relaxed solution obtained from the CG is fractional. In such a case, we embed CG in a branch-and-bound framework until reaching a near-optimal feasible node. Then, we fix the routes from the standard branching constraints and keep others free. Such a procedure maintains the problem structure (in the sense of the number of constraints). We then collect the dual solution corresponding to that node and obtain the Benders cut to be added into BMP, as in Er Raqabi et al. (2023b).

We also note that instead of solving Phase 2 once, as in Cordeau et al. (2001b), we solve it until the optimality criterion is reached, as in Zeighami & Soumis (2019) (see Section 5.2). In other words, our approach is theoretically exact. However, in practice, we often stop each phase before the optimality conditions are met. As we approach optimality, new cuts have little or no effect on the optimal BD solution. To avoid the well-known tailing-off effect (Rahmaniani et al., 2017), we generate new cuts until the relative difference between the lower and upper bounds is less than or equal to 0.1% for Phase 1 and 0.05% for Phase 2.

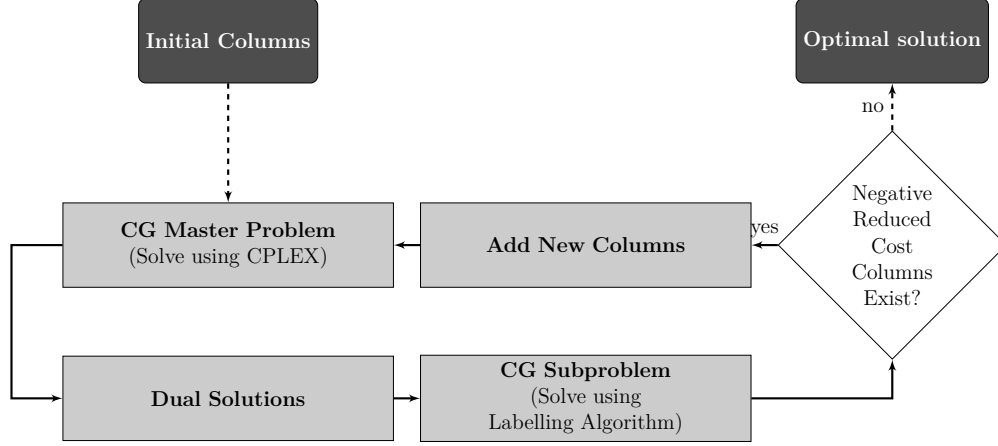


Figure 5.4 Flowchart of the Column Generation Procedure

The number of feasible routes $\varrho \in \Omega$ is exponential. To avoid a large model, we generate these routes iteratively using a CG method (embedded in a branch-and-price scheme for Phase 2). The CG decomposes the main problem into a restricted master problem (RMP) and several pricing problems. At each iteration of the CG procedure, we solve the RMP using an LP solver over a subset of the routes (columns). Based on the dual solution, we solve the pricing subproblems to find negative reduced-cost route(s). The pricing subproblem is detailed in Appendix A. We then add these routes to the RMP. We iterate until no negative reduced-cost route(s) are identified, as highlighted in Figure 5.4. Similarly to BD, in practice, as we approach optimality, new routes with negative reduced costs have little or no effect on the optimal value of the RMP. Therefore, we stop the CG when the optimal value of the RMP improves by less than 0.1% over five iterations. To make the algorithm work, we enhance it with several acceleration strategies. The latter are presented next.

5.5.4 Acceleration Strategies

We now describe the acceleration strategies that improve the efficiency of the combined BD and CG approach.

Warm-start. To tighten the Benders MP, we add a set of initial Benders cuts corresponding to initial solutions. These solutions are based on the *maximum level policy* heuristic, which consists of two stages: In the first stage, we start by finding the depletion day $d \in \mathcal{D}$ for each client-product $n \in \mathcal{N}$. Then, we find an available truck ($k \in \mathcal{K}$) that best fits the free space in the client-product tank. In other words, we fill the client-product tank to its maximum capacity. After that, we update the inventory values and set the next depletion day for the same client-product. We iterate over the client-products until all daily consumption of

client-products is covered for the whole horizon. In the second stage, we assign each quantity ordered of each client-product to the closest depot. To guarantee the solution’s feasibility, we set the inventory levels at the depots for each product $p \in \mathcal{P}$ and week $w \in \mathcal{W}$ to zero. In the final stage, we solve the routing problem in the Benders subproblem to find combinations of the client-products that can be put together in the same truck $k \in \mathcal{K}$, if any. After solving the BDSP, we use the solution to obtain an initial Benders cut, which is added to the BMP. We recall that any heuristic (in the literature or user-specific) that finds feasible solutions can be used to define initial Benders cuts to our problem.

Parallelism. An interesting aspect of decomposition methods is that we can solve their subproblems in parallel. In our case, as per Theorem 1, and since we solve the Benders subproblems using CG, we can profit from two levels of parallelism. The first level consists of solving the independent $|\mathcal{W}|$ SPPs (the Benders subproblems) in parallel. The second level consists of solving in parallel the independent CG subproblems in each Benders subproblem, where each CG subproblem corresponds to one truck $k \in \mathcal{K}$ and one day $d \in \mathcal{D}_k^w$, as in Bani et al. (2023). For both levels, we use multi-threading to solve the subproblems in parallel. It is worth mentioning that we use a subnetwork for each subproblem (both the Benders subproblems and the CG subproblems) to avoid any access conflict between threads. Each subnetwork has an independent set of arcs, nodes, and a data labeling structure. We limit CPLEX to a single thread to solve the resulting TLP problem inside the extension function.

Hashing Technique. Solving a set of TLPs inside the labeling algorithm can be very time-consuming. Thus, at each iteration, we use a hashing technique to memorize the solved TLPs, to use them in the subsequent iterations. This technique helps reduce the execution time of the CG subproblem labeling algorithm. Furthermore, after exploring the total orders, it is possible to tighten the resources, which are the tank truck capacity and the number of compartments required. The hashing technique is useful when solving similar MILPs several times and when several routes’ components are regenerated at each iteration, i.e., we can reuse the information and solutions history.

Primal Diving Heuristic. To accelerate finding the CG integer solution in Phase 2, we use the primal diving heuristic introduced in Bani et al. (2023), which is a mix of local branching (Fischetti & Lodi, 2003) around primal feasible integer solution(s) and a diving heuristic (Joncour et al., 2010b). The primal diving heuristic consists of fixing fractional variables close to the integer primal solution. This technique, proven to find the closest integer solution to the optimal one for problems that have some nice polyhedral properties that favor integrality (e.g., set partitioning– type problems with low to moderate density), helps reduce the number of branching nodes and iterations. It limits the number of iterations

without solution improvement.

5.6 Computational Experiments

In this section, we run computational experiments on real instances. We first describe the case context. Then, we present the experimental design. After that, we highlight the computational results, the comparison with two heuristics, and the managerial insights.

5.6.1 Case Context: West Africa

We consider a petroleum distribution company from West Africa facing the MDMPPSRPIM in a geographical zone. The specificity is that the petrol products are broken down into the categories of marked and unmarked products. Marked products are partially subsidized by the government. For this reason, they are chemically marked to detect fraud. Furthermore, the company has a limited fleet of heterogeneous trucks, with many variably sized compartments. It also serves distinct subfamilies of clients: (i) petrol stations, (ii) marine stations, (iii) private bakeries, and (iv) occasional private companies. We assume a fixed consumption rate for each client-product, which makes their demand deterministic. In real life, we cannot know the stations' fuel consumption beforehand. The company uses safety stock levels or emergency deliveries to correct the resulting uncertainty, as described in the use case of Popović et al. (2011). In addition to the problem description in Section 5.3, some additional specific business rules are summarized in Table 5.4 below.

Table 5.4 West Africa-Specific Rules

Rule	Description
$\mathcal{R}_6$	Marked and unmarked products cannot be loaded onto the same tank truck (exclusion constraint).
$\mathcal{R}_7$	Some tank trucks must only carry marked products.
$\mathcal{R}_8$	A truck cannot visit more than two marine stations in a row.

There are also these additional constraints:

$$\sum_{(i,j) \in \mathcal{N} \times \mathcal{N}_f} x_{ij}^{kd} \leq 2 \quad \forall k \in \mathcal{K}, \forall d \in \mathcal{D}_k \quad (5.50)$$

where $\mathcal{N}_f$ is the subset of client-products related to marine stations. Constraints (5.50) state that a tank truck cannot visit more than two marine stations, as imposed by Rule $\mathcal{R}_8$.

Given that the company considers two types of products (marked and unmarked), we highlight that the graph in Figure 5.2 above contains two disjoint subgraphs: the subgraph of unmarked products and the subgraph of marked products. Since the graph becomes bipartite, we tackle each class separately to ensure that Rules $\mathcal{R}_6$ and $\mathcal{R}_7$ are met. By doing so, we highlight the approach’s flexibility in incorporating user-specific and context-specific decision rules.

5.6.2 Experimental Design

For our tests, we consider 48 realistic instances. These instances belong to two seasons (2016 and 2017). The instances, presented in Table 5.5, have the following features: the number of client-products ($|\mathcal{N}|$), the number of clients ($\#C$), the number of products ($\mathcal{P}$), the number of weeks/days ($|\mathcal{W}|/|\mathcal{D}|$), the number of depots ($|\mathcal{E}|$), and the instance name. In total, we have four types of petrol products. The instance name includes the number of weeks, the number of client-products, the number of depots, and the season. For instance, Instance **P4-C24-D1-S1** has 4 weeks, 24 client-products, 1 depot, and belongs to season 1 (2016). We classify instances into four classes based on $|\mathcal{N}|$.

Table 5.5 Instances Properties

$ \mathcal{N} $	$\#C$	$\mathcal{P}$	$ \mathcal{W} / \mathcal{D} $	$ \mathcal{E} $	Instance	$ \mathcal{N} $	$\#C$	$\mathcal{P}$	$ \mathcal{W} / \mathcal{D} $	$ \mathcal{E} $	Instance
24	12	[0, 1]	4/28	1	P4-C24-D1-S1	47	47	[2, 3]	4/28	1	P4-C47-D1-S1
					P4-C24-D1-S2						P4-C47-D1-S2
				4	P4-C24-D4-S1					4	P4-C47-D4-S1
					P4-C24-D4-S2						P4-C47-D4-S2
			8/56	1	P8-C24-D1-S1				8/56	1	P8-C47-D1-S1
					P8-C24-D1-S2						P8-C47-D1-S2
				4	P8-C24-D4-S1					4	P8-C47-D4-S1
					P8-C24-D4-S2						P8-C47-D4-S2
			12/84	1	P12-C24-D1-S1				12/84	1	P12-C47-D1-S1
					P12-C24-D1-S2						P12-C47-D1-S2
				4	P12-C24-D4-S1					4	P12-C47-D4-S1
					P12-C24-D4-S2						P12-C47-D4-S2
34	22	[0, 1, 2]	4/28	1	P4-C34-D1-S1	77	65	[0, 1, 2, 3]	4/28	1	P4-C77-D1-S1
					P4-C34-D1-S2						P4-C77-D1-S2
				4	P4-C34-D4-S1					4	P4-C77-D4-S1
					P4-C34-D4-S2						P4-C77-D4-S2
			8/56	1	P8-C34-D1-S1				8/56	1	P8-C77-D1-S1
					P8-C34-D1-S2						P8-C77-D1-S2
				4	P8-C34-D4-S1					4	P8-C77-D4-S1
					P8-C34-D4-S2						P8-C77-D4-S2
			12/84	1	P12-C34-D1-S1				12/84	1	P12-C77-D1-S1
					P12-C34-D1-S2						P12-C77-D1-S2
				4	P12-C34-D4-S1					4	P12-C77-D4-S1
					P12-C34-D4-S2						P12-C77-D4-S2

The vehicles’ features, which include the capacity (Ca.) in m^3 , the compartment configuration (Config.), the number of compartments (Co.), if the truck is suitable for marked

and unmarked products (F.), and whether or not the truck is jumbo (J.), are highlighted in Table 5.6.

Table 5.6 Tank Trucks Properties

Id	Ca.	Config.	Co.	F.	J.	Id	Ca.	Config.	Co.	F.	J.
k_1	13	[2, 1, 4, 6]	4	✓	□	k_{23}	21	[2, 5, 5, 5, 4]	5	✓	✓
k_2	13	[1, 2, 4, 6]	4	✓	□	k_{24}	21	[2, 3, 1, 1, 2, 3, 5, 4]	8	□	✓
k_3	13	[3, 1, 1, 1, 1, 2, 2, 2]	8	✓	□	k_{25}	21	[6, 4, 6, 2, 3, 5, 4, 2, 6]	9	□	✓
k_4	13	[2, 1, 1, 1, 1, 5, 2]	7	✓	□	k_{26}	22	[2, 2, 1, 3, 2, 1, 2, 3, 2, 2, 2]	11	✓	✓
k_5	13	[3, 2, 2, 3, 5, 3]	5	✓	□	k_{27}	30	[7, 7, 2, 7, 7]	5	□	✓
k_6	14	[2, 1, 1, 4, 4, 2]	6	□	□	k_{28}	33	[5, 3, 4, 1.5, 1.5, 1, 1, 2, 3, 2, 4, 3, 2]	13	✓	✓
k_7	14	[2, 2, 3, 1, 2, 2, 2]	7	✓	□	k_{29}	33	[5, 4, 3, 2, 1, 1, 2, 2, 3, 4, 6]	11	□	✓
k_8	14	[4, 2, 3, 2, 3]	5	✓	□	k_{30}	33	[5, 4, 3, 2, 1, 1, 2, 2, 3, 4, 6]	11	□	✓
k_9	14	[2, 1, 1, 1, 1, 1, 1, 2, 1, 2]	11	□	□	k_{31}	33	[4, 5, 4, 3, 2, 3, 2, 5, 5]	9	□	✓
k_{10}	14	[2, 1, 1, 1, 1, 2, 3, 3]	8	✓	□	k_{32}	35	[6, 4, 5, 2, 2, 2, 3, 8, 3]	9	✓	✓
k_{11}	14	[4, 2, 1, 1, 4, 2]	6	✓	□	k_{33}	35	[4, 6, 2, 4, 3, 4, 3, 6, 3]	9	□	✓
k_{12}	18	[3, 2, 2, 3, 2, 6]	6	□	□	k_{34}	35	[6, 4, 3, 1, 1, 2, 5, 6, 7]	9	□	✓
k_{13}	18	[5, 2, 2, 2, 3, 4]	6	✓	□	k_{35}	36	[4, 2, 6, 2, 4, 2, 4, 3, 6, 3]	10	□	✓
k_{14}	18	[4, 2, 5, 4, 3]	5	□	□	k_{36}	37	[5, 2, 5, 2, 2, 1, 1, 2, 2, 2, 2, 5, 2, 2, 2]	15	✓	✓
k_{15}	18	[2, 2, 2, 2, 1, 1, 1, 1, 3, 3]	10	□	□	k_{37}	37	[6, 3, 6, 4, 6, 6, 6]	7	✓	✓
k_{16}	18	[2, 1.5, 1, 1, 1.5, 4, 4, 3]	8	□	□	k_{38}	37	[10, 3, 4, 2, 5, 6, 7]	7	✓	✓
k_{17}	18	[2, 1.5, 1, 1, 1.5, 4, 4, 3]	8	□	□	k_{39}	38	[6, 4, 1, 1, 4, 2, 4, 6, 4, 3, 5]	11	✓	✓
k_{18}	19	[5, 1, 1, 5, 5, 2]	6	□	□	k_{40}	38	[8, 4, 3, 2, 2, 1, 1, 2, 5, 4, 6]	11	✓	✓
k_{19}	20	[6, 5, 5, 2, 2]	5	□	✓	k_{41}	38	[6, 4, 6, 2, 3, 5, 4, 2, 6]	9	✓	✓
k_{20}	20	[3, 3, 3, 3, 2, 2, 2, 2]	8	□	✓	k_{42}	40	[7, 5, 4, 1, 1, 4, 2, 5, 2, 3, 6]	11	□	✓
k_{21}	20	[2, 4, 3, 4, 5, 1, 1]	7	□	✓	k_{43}	40	[7, 5, 4, 1, 1, 4, 2, 5, 2, 3, 6]	11	□	✓
k_{22}	20	[2, 4, 3, 4, 5, 1, 1]	7	□	✓	k_{44}	40	[6, 4, 1, 1, 4, 2, 4, 6, 4, 3, 5]	11	□	✓

The coding language is C++, and tests are conducted using version 22.1.1 of the IBM ILOG CPLEX solver. All experiments were carried out on a 3.20GHz Intel^R CoreTM i7-700 processor, with a 64GiB system memory, running on Oracle Linux Server release 7.7. We use real time to measure runtime. In addition to the BD gap threshold (0.1% for Phase 1 and 0.05% for Phase 2), we stop the BD iterations when the value of the BD RMP improves by less than 0.1% over five iterations. We also stop the CG when the value of the CG RMP improves by less than 0.1% over five iterations.

5.6.3 Computational Results

In this section, we first present the performance of the combined BD and CG approach. Then, we highlight the impact of our acceleration strategies.

Performance

Table 5.7 shows the performance of BD. For each instance, we report the total time (T) in seconds, the time required by the BD MP (TMP) in seconds, and the number of Benders cuts (#Cuts). Furthermore, for each phase, we report the integrality gap (Gap) percentage,

the percentage of time required by the BD MP (TMP), and the number of Benders cuts (#Cuts).

We observe that, on average, the four classes require around 4 seconds, 9 seconds, 47 seconds, and 40 seconds for both phases, respectively. The Benders MP requires less than 1% of the total execution time (T). For the Benders cuts, on average, the four classes require 4, 7, 8, and 7 Benders cuts, respectively. We reach near-optimal, if not optimal, solutions in all the instances. It is worth mentioning that Phase 1 requires more iterations and consumes most of the Benders MP time. Phase 2 benefits from the Benders cuts kept from Phase 1, reaching near-optimal integer solutions more quickly. During Phase 2, the stabilized inventory levels from Phase 1 are not changed. In other words, Phase 2 confirms the results obtained in Phase 1. Among all instances, only 10 instances do not reach the 0.05% gap threshold in Phase 2 but have at most 0.70% in integrality gap.

Table 5.8 shows the performance of CG. For each instance, we report the total time (T) in seconds (same as Table 5.7), the total number of columns generated (Total #Col), and the number of CG iterations (Total #It). Furthermore, for each phase, we report the percentage of time required by the BD SP (TSP), the number of columns generated (#Col), and the number of CG iterations (#It). For Phase 2, we also report the number of branching nodes (#No).

We observe that the Benders subproblem consumes most of the total execution time, with a higher portion consumed in Phase 1, as compared to Phase 2. In Phase 1, the number of columns generated in the CG process increases significantly, from 6169 columns on average for the first class (24 clients-products) to 81353 columns on average for the fourth class (77 clients-products). The number of CG iterations also increases from 163 for the first class to 329 for the third class (47 clients-products). In Phase 2, a similar trend is observed for the number of iterations, which increases from 72 on average for the first class to 161 on average for the third class. Figure 5.5 shows the improvement of the UB (provided by the BD subproblem) and the LB (provided by BD master problem) along the number of Benders iterations for Instance **P4-C77-D4-S2**.

The compact model (MDMPPSRPIM) has a large integrality gap, especially in the presence of inventory constraints. The main insight behind the performance results originates from observing that when the inventory levels are fixed, the resulting problems have a lower integrality gap, as compared to the compact model. The Benders master problem (which is LP) is easy to solve. Then, the Benders subproblems can be efficiently solved as shown above. Furthermore, the gap between the near-optimal solution obtained in Phase 1 and the near-optimal integer solution obtained in Phase 2 is small. This observation is established

Table 5.7 Benders Decomposition Results

N	W	Inst.	T(s)	TMP(s)	#Cuts	Phase 1			Phase 2		
						Gap(%)	TMP(%)	#Cuts	Gap(%)	TMP(%)	#Cuts
24	4	P4-C24-D1-S1	1.82	0.00	4	0.10	87.71	3	0.05	12.29	1
24	4	P4-C24-D1-S2	1.62	0.00	4	0.10	88.44	3	0.05	11.56	1
24	4	P4-C24-D4-S1	1.85	0.00	4	0.10	87.56	3	0.05	12.44	1
24	4	P4-C24-D4-S2	1.63	0.00	4	0.10	88.33	3	0.05	11.67	1
24	8	P8-C24-D1-S1	3.33	0.01	4	0.02	88.93	3	0.05	11.07	1
24	8	P8-C24-D1-S2	5.56	0.01	7	0.10	74.83	3	0.09	25.17	4
24	8	P8-C24-D4-S1	3.33	0.01	4	0.02	88.77	3	0.05	11.23	1
24	8	P8-C24-D4-S2	3.26	0.01	4	0.10	90.31	3	0.05	9.69	1
24	12	P12-C24-D1-S1	5.16	0.02	4	0.10	86.58	3	0.05	13.42	1
24	12	P12-C24-D1-S2	5.46	0.02	4	0.10	87.31	3	0.05	12.69	1
24	12	P12-C24-D4-S1	5.09	0.02	4	0.10	86.73	3	0.05	13.27	1
24	12	P12-C24-D4-S2	5.31	0.02	4	0.10	86.85	3	0.05	13.15	1
Avg			3.62	0.01	4	0.09	86.86	3	0.05	13.14	1
34	4	P4-C34-D1-S1	2.86	0.01	4	0.10	89.72	3	0.05	10.28	1
34	4	P4-C34-D1-S2	8.61	0.01	11	0.05	79.11	7	0.70	20.89	4
34	4	P4-C34-D4-S1	2.92	0.01	4	0.10	89.35	3	0.05	10.65	1
34	4	P4-C34-D4-S2	9.59	0.01	12	0.10	71.86	7	0.42	28.14	5
34	8	P8-C34-D1-S1	8.78	0.03	6	0.10	90.61	5	0.05	9.39	1
34	8	P8-C34-D1-S2	15.52	0.05	10	0.02	76.63	6	0.20	23.37	4
34	8	P8-C34-D4-S1	8.73	0.03	6	0.10	89.60	5	0.05	10.40	1
34	8	P8-C34-D4-S2	13.74	0.04	9	0.10	72.97	5	0.09	27.03	4
34	12	P12-C34-D1-S1	8.49	0.04	4	0.10	77.90	3	0.05	22.10	1
34	12	P12-C34-D1-S2	12.11	0.04	5	0.10	80.90	4	0.05	19.10	1
34	12	P12-C34-D4-S1	8.42	0.04	4	0.10	77.24	3	0.05	22.76	1
34	12	P12-C34-D4-S2	9.85	0.03	4	0.01	86.24	3	0.05	13.76	1
Avg			9.14	0.03	7	0.08	81.84	5	0.15	18.16	2
47	4	P4-C47-D1-S1	39.62	0.02	11	0.10	87.02	10	0.05	12.98	1
47	4	P4-C47-D1-S2	16.02	0.01	8	0.10	94.66	7	0.05	5.34	1
47	4	P4-C47-D4-S1	39.07	0.02	11	0.10	79.46	10	0.05	20.54	1
47	4	P4-C47-D4-S2	15.95	0.01	8	0.10	94.79	7	0.05	5.21	1
47	8	P8-C47-D1-S1	30.48	0.05	9	0.02	73.54	5	0.13	26.46	4
47	8	P8-C47-D1-S2	33.26	0.06	10	0.10	94.22	9	0.05	5.78	1
47	8	P8-C47-D4-S1	15.42	0.03	4	0.10	88.84	3	0.05	11.16	1
47	8	P8-C47-D4-S2	26.23	0.05	8	0.10	92.32	7	0.00	7.68	1
47	12	P12-C47-D1-S1	108.89	0.10	8	0.04	52.47	4	0.10	47.53	4
47	12	P12-C47-D1-S2	63.70	0.10	10	0.10	71.38	6	0.06	28.62	4
47	12	P12-C47-D4-S1	132.53	0.09	7	0.01	44.21	3	0.09	55.79	4
47	12	P12-C47-D4-S2	42.09	0.08	7	0.02	86.31	6	0.03	13.69	1
Avg			46.94	0.05	8	0.07	79.93	6	0.06	20.07	2
77	4	P4-C77-D1-S1	23.59	0.03	6	0.10	89.37	5	0.05	10.63	1
77	4	P4-C77-D1-S2	17.67	0.04	7	0.10	91.74	6	0.05	8.26	1
77	4	P4-C77-D4-S1	21.81	0.03	6	0.10	90.41	5	0.03	9.59	1
77	4	P4-C77-D4-S2	17.19	0.03	7	0.10	91.46	6	0.05	8.54	1
77	8	P8-C77-D1-S1	23.94	0.08	6	0.10	84.52	5	0.05	15.48	1
77	8	P8-C77-D1-S2	41.07	0.10	8	0.03	92.62	7	0.01	7.38	1
77	8	P8-C77-D4-S1	27.26	0.09	7	0.03	86.52	6	0.01	13.48	1
77	8	P8-C77-D4-S2	85.00	0.18	16	0.06	77.40	12	0.14	22.60	4
77	12	P12-C77-D1-S1	71.21	0.11	5	0.03	80.27	4	0.05	19.73	1
77	12	P12-C77-D1-S2	44.83	0.13	6	0.10	78.21	5	0.05	21.79	1
77	12	P12-C77-D4-S1	70.57	0.11	5	0.03	80.41	4	0.05	19.59	1
77	12	P12-C77-D4-S2	35.71	0.11	5	0.01	83.01	4	0.01	16.99	1
Avg			39.99	0.09	7	0.07	85.49	6	0.05	14.51	1

Table 5.8 Column Generation Results

$\mathcal{N}$	$\mathcal{W}$	Inst.	T(s)	Total	#Col	Total	#Itr	Phase 1			Phase 2			
								TSP(%)	#Col	#Itr	TSP(%)	#Col	#Itr	#No
24	4	P4-C24-D1-S1	1.76		4601		120	73.08	3434	84	26.92	1167	36	8
24	4	P4-C24-D1-S2	1.55		3977		110	74.33	2905	82	25.67	1072	28	5
24	4	P4-C24-D4-S1	1.78		4601		120	72.66	3434	84	27.34	1167	36	8
24	4	P4-C24-D4-S2	1.57		4001		110	74.63	2920	82	25.37	1081	28	5
24	8	P8-C24-D1-S1	3.22		8401		216	74.21	6334	156	25.79	2067	60	10
24	8	P8-C24-D1-S2	5.45		13959		384	41.29	6015	168	58.71	7944	216	32
24	8	P8-C24-D4-S1	3.22		8401		216	74.36	6334	156	25.64	2067	60	10
24	8	P8-C24-D4-S2	3.14		8079		220	73.69	6053	168	26.31	2026	52	8
24	12	P12-C24-D1-S1	5.00		12243		322	73.91	9113	238	26.09	3130	84	14
24	12	P12-C24-D1-S2	5.30		12373		338	73.29	9196	250	26.71	3177	88	14
24	12	P12-C24-D4-S1	4.93		12225		322	74.32	9095	238	25.68	3130	84	14
24	12	P12-C24-D4-S2	5.15		12377		338	73.46	9196	250	26.54	3181	88	14
Avg			3.51		8770		235	71.10	6169	163	28.90	2601	72	12
34	4	P4-C34-D1-S1	2.79		6938		114	73.20	5184	80	26.80	1754	34	7
34	4	P4-C34-D1-S2	8.53		28760		298	63.16	18852	186	36.84	9908	112	20
34	4	P4-C34-D4-S1	2.84		6938		114	73.39	5184	80	26.61	1754	34	7
34	4	P4-C34-D4-S2	9.51		31596		336	55.84	18886	186	44.16	12710	150	30
34	8	P8-C34-D1-S1	8.64		19266		318	82.38	16114	260	17.62	3152	58	11
34	8	P8-C34-D1-S2	15.36		40541		560	57.53	24449	328	42.47	16092	232	40
34	8	P8-C34-D4-S1	8.59		19292		316	82.20	16116	258	17.80	3176	58	11
34	8	P8-C34-D4-S2	13.58		35663		490	54.02	20099	274	45.98	15564	216	32
34	12	P12-C34-D1-S1	8.30		18107		314	73.98	13572	236	26.02	4535	78	13
34	12	P12-C34-D1-S2	11.91		28040		422	79.07	22387	332	20.93	5653	90	16
34	12	P12-C34-D4-S1	8.23		18107		314	74.22	13572	236	25.78	4535	78	13
34	12	P12-C34-D4-S2	9.66		22626		342	73.18	16821	250	26.82	5805	92	16
Avg			8.99		22990		328	70.18	15936	226	29.82	7053	103	18
47	4	P4-C47-D1-S1	39.42		86928		302	89.41	78622	266	10.59	8306	36	9
47	4	P4-C47-D1-S2	15.93		49300		226	83.11	42850	196	16.89	6450	30	6
47	4	P4-C47-D4-S1	38.96		86928		302	89.52	78622	266	10.48	8306	36	9
47	4	P4-C47-D4-S2	15.86		49300		226	83.26	42850	196	16.74	6450	30	6
47	8	P8-C47-D1-S1	30.30		96739		584	54.41	54891	288	45.59	41848	296	64
47	8	P8-C47-D1-S2	33.07		104735		594	89.13	94668	522	10.87	10067	72	15
47	8	P8-C47-D4-S1	15.27		51794		242	72.59	37841	168	27.41	13953	74	17
47	8	P8-C47-D4-S2	26.05		85795		482	84.96	74623	410	15.04	11172	72	15
47	12	P12-C47-D1-S1	108.61		161455		730	52.77	82899	338	47.23	78556	392	80
47	12	P12-C47-D1-S2	63.41		176507		942	54.48	105379	526	45.52	71128	416	84
47	12	P12-C47-D4-S1	132.26		164464		628	33.91	62824	252	66.09	101640	376	68
47	12	P12-C47-D4-S2	41.84		127887		622	82.12	108523	520	17.88	19364	102	19
Avg			46.75		103486		490	72.47	72049	329	27.53	31437	161	33
77	4	P4-C77-D1-S1	23.47		52981		190	79.44	43315	148	20.56	9666	42	10
77	4	P4-C77-D1-S2	17.54		53244		218	84.32	45584	184	15.68	7660	34	7
77	4	P4-C77-D4-S1	21.69		52994		194	82.62	43841	148	17.38	9153	46	12
77	4	P4-C77-D4-S2	17.06		52574		208	84.55	45089	178	15.45	7485	30	5
77	8	P8-C77-D1-S1	23.71		82374		366	81.24	69146	292	18.76	13228	74	16
77	8	P8-C77-D1-S2	40.81		114884		488	86.22	100149	414	13.78	14735	74	15
77	8	P8-C77-D4-S1	27.01		92930		424	83.75	79817	346	16.25	13113	78	18
77	8	P8-C77-D4-S2	84.62		234120		988	72.11	174268	700	27.89	59852	288	60
77	12	P12-C77-D1-S1	70.89		117928		480	76.65	93482	360	23.35	24446	120	24
77	12	P12-C77-D1-S2	44.48		129412		542	78.19	105679	436	21.81	23733	106	22
77	12	P12-C77-D4-S1	70.25		117928		480	76.45	93482	360	23.55	24446	120	24
77	12	P12-C77-D4-S2	35.38		104211		458	76.49	82378	356	23.51	21833	102	21
Avg			39.74		100465		420	80.17	81353	327	19.83	19113	93	20

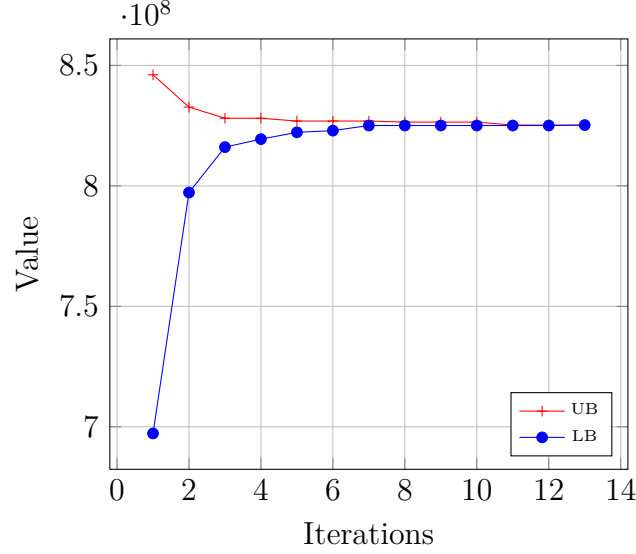


Figure 5.5 UB and LB Improvement Along With the Number of Benders Iterations for Instance **P4-C77-D4-S2**

for VRPs with fewer side constraints (Bani et al., 2023). These problems can be seen as set-partitioning problems, thus they have the quasi-integrality property. Thus, after using Phase 1 to stabilize the inventory levels, Phase 2 quickly reaches the near-optimal integer solution. For some instances, we observed that Phase 1 is enough to reach the near-optimal integer solution. The results also confirm the effectiveness of the solution approach and the decoupling intuition behind it. To further complement the performance analysis, we will now discuss the impact of the acceleration strategies.

Impact of Acceleration Strategies

To evaluate the acceleration strategies, we run additional experiments. We distinguish the strategies based on the BD master problem and the BD subproblems. Table 5.9 shows the impact of the warm-start strategy on the Benders master problem. We report the results without warm-start (No WS) and with warm-start (Our Approach).

Table 5.9 shows that without a warm start, the total time required to reach small gaps increases significantly. Even with more time, the gap is still higher. For many instances, the gap is higher than 1%. The number of Benders cuts is also higher than with our approach. For the first class, the time increases by a factor of 40, the gap is 0.52% higher, and the number of Benders cuts increases by a factor of 4.5. For the second class, the time increases by a factor of 30, the gap is 0.48% higher, and the number of Benders cuts increases by a factor of 4.7. For the third class, the time increases by a factor of 56, the gap is 0.36%

Table 5.9 Benders Master Problem Acceleration Strategies Impact

$\mathcal{N}$	$\mathcal{W}$	Inst.	No WS			Our Approach		
			T(s)	Gap(%)	#Cuts	T(s)	Gap(%)	#Cuts
24	4	P4-C24-D1-S1	50.71	0.15	14	1.82	0.05	4
24	4	P4-C24-D4-S1	128.86	0.05	7	1.62	0.05	4
24	4	P4-C24-D1-S2	36.41	0.61	11	1.85	0.05	4
24	4	P4-C24-D4-S2	49.49	2.06	15	1.63	0.05	4
24	8	P8-C24-D1-S1	212.16	0.31	15	3.33	0.05	4
24	8	P8-C24-D4-S1	100.06	0.73	19	5.56	0.09	7
24	8	P8-C24-D1-S2	246.58	0.04	11	3.33	0.05	4
24	8	P8-C24-D4-S2	98.93	0.51	17	3.26	0.05	4
24	12	P12-C24-D1-S1	214.67	0.54	41	5.16	0.05	4
24	12	P12-C24-D4-S1	140.20	0.05	9	5.46	0.05	4
24	12	P12-C24-D1-S2	204.35	1.43	41	5.09	0.05	4
24	12	P12-C24-D4-S2	235.37	0.41	15	5.31	0.05	4
Avg			143.15	0.57	18	3.62	0.05	4
34	4	P4-C34-D1-S1	99.52	1.16	41	2.86	0.05	4
34	4	P4-C34-D4-S1	24.68	0.21	12	8.61	0.70	11
34	4	P4-C34-D1-S2	67.13	0.05	33	2.92	0.05	4
34	4	P4-C34-D4-S2	64.91	0.05	31	9.59	0.42	12
34	8	P8-C34-D1-S1	186.22	0.88	26	8.78	0.05	6
34	8	P8-C34-D4-S1	161.59	1.24	33	15.52	0.20	10
34	8	P8-C34-D1-S2	412.33	1.24	41	8.73	0.05	6
34	8	P8-C34-D4-S2	252.35	0.57	40	13.74	0.09	9
34	12	P12-C34-D1-S1	445.09	0.71	41	8.49	0.05	4
34	12	P12-C34-D4-S1	285.44	0.40	30	12.11	0.05	5
34	12	P12-C34-D1-S2	418.00	0.05	36	8.42	0.05	4
34	12	P12-C34-D4-S2	788.60	0.96	37	9.85	0.05	4
Avg			267.16	0.63	33	9.14	0.15	7
47	4	P4-C47-D1-S1	506.68	0.03	25	39.62	0.05	11
47	4	P4-C47-D4-S1	1900.41	0.05	35	16.02	0.05	8
47	4	P4-C47-D1-S2	247.78	0.05	37	39.07	0.05	11
47	4	P4-C47-D4-S2	202.94	0.05	41	15.95	0.05	8
47	8	P8-C47-D1-S1	3076.22	0.65	37	30.48	0.13	9
47	8	P8-C47-D4-S1	2099.24	0.33	41	33.26	0.05	10
47	8	P8-C47-D1-S2	2718.42	1.11	40	15.42	0.05	4
47	8	P8-C47-D4-S2	2852.87	0.54	45	26.23	0.00	8
47	12	P12-C47-D1-S1	5227.72	0.66	39	108.89	0.10	8
47	12	P12-C47-D4-S1	4949.52	0.67	37	63.70	0.06	10
47	12	P12-C47-D1-S2	2326.98	0.28	41	132.53	0.09	7
47	12	P12-C47-D4-S2	5346.13	0.74	38	42.09	0.03	7
Avg			2621.24	0.42	38	46.94	0.06	8
77	4	P4-C77-D1-S1	510.96	0.38	21	23.59	0.05	6
77	4	P4-C77-D4-S1	474.61	1.13	25	17.67	0.05	7
77	4	P4-C77-D1-S2	358.71	0.58	67	21.81	0.03	6
77	4	P4-C77-D4-S2	359.53	0.59	65	17.19	0.05	7
77	8	P8-C77-D1-S1	6309.51	0.76	46	23.94	0.05	6
77	8	P8-C77-D4-S1	6699.86	0.47	61	41.07	0.01	8
77	8	P8-C77-D1-S2	6593.56	1.02	41	27.26	0.01	7
77	8	P8-C77-D4-S2	7800.53	1.34	52	85.00	0.14	16
77	12	P12-C77-D1-S1	7142.98	0.36	44	71.21	0.05	5
77	12	P12-C77-D4-S1	7359.32	0.28	85	44.83	0.05	6
77	12	P12-C77-D1-S2	6830.86	0.69	41	70.57	0.05	5
77	12	P12-C77-D4-S2	1239.64	1.23	38	35.71	0.01	5
Avg			4306.67	0.74	49	39.99	0.05	7

higher, and the number of Benders cuts increases by a factor of 4.8. For the fourth class, the time increases by a factor of 108, the gap is 0.69% higher, and the number of Benders cuts increases by a factor of 7. Warm-start reduces the execution time from hours to minutes for several instances from the third and fourth classes.

Table 5.10 reports the impact of the Benders subproblem acceleration strategies. We show the results without parallelism (No Parallel), without the hashing technique (No Hash), and without the primal diving heuristic (No PDH). For the No PDH scenario, we implement an exact branch-and-price method that confirms the quality of the proposed PDH approach. This exact method is implemented in C++ using the Branch-and-Cut-and-Price (BCP) framework version 1.4 (Coin-Or, 2019). We use the default branching rule.

Table 5.10 Benders Subproblem Acceleration Strategies Impact

N	W	Inst.	No Parallel				No Hash				No PDH				Our Approach			
			T(s)	#Col	#Itr	#No	T(s)	#Col	#Itr	#No	T(s)	#Col	#Itr	#No	T(s)	#Col	#Itr	#No
24	4	P4-C24-D1-S1	14.81	4601	120	8	4.16	4601	120	8	13.43	5786	152	22	1.76	4601	120	8
24	4	P4-C24-D4-S1	10.90	3977	110	5	2.29	3977	110	5	12.95	4951	152	27	1.55	3977	110	5
24	4	P4-C24-D1-S2	14.45	4601	120	8	4.13	4601	120	8	13.70	5786	152	22	1.78	4601	120	8
24	4	P4-C24-D4-S2	11.38	4001	110	5	2.30	4001	110	5	13.54	4951	152	27	1.57	4001	110	5
24	8	P8-C24-D1-S1	20.90	8401	216	10	6.21	8401	216	10	27.08	10448	272	32	3.22	8401	216	10
24	8	P8-C24-D4-S1	45.50	13959	384	32	12.96	13959	384	32	27.51	12052	270	11	5.45	13959	384	32
24	8	P8-C24-D1-S2	20.96	8401	216	10	6.20	8401	216	10	25.22	10448	272	32	3.22	8401	216	10
24	8	P8-C24-D4-S2	46.10	8079	220	8	13.39	8079	220	8	27.68	12146	268	10	3.14	8079	220	8
24	12	P12-C24-D1-S1	29.70	12243	322	14	8.18	12243	322	14	38.54	15118	373	33	5.00	12243	322	14
24	12	P12-C24-D4-S1	29.59	12373	338	14	8.66	12373	338	14	42.27	18724	419	17	5.30	12373	338	14
24	12	P12-C24-D1-S2	30.67	12225	322	14	8.89	12225	322	14	37.48	15090	368	33	4.93	12225	322	14
24	12	P12-C24-D4-S2	29.44	12377	338	14	8.63	12377	338	14	42.47	18724	419	17	5.15	12377	338	14
Avg			25.37	8770	235	12	7.17	8770	235	12	26.82	11185	272	24	3.51	8770	235	12
34	4	P4-C34-D1-S1	20.53	6938	114	7	6.71	6938	114	7	17.90	8737	152	24	2.79	6938	114	7
34	4	P4-C34-D1-S2	62.58	28760	298	20	17.80	28760	298	20	28.79	21014	228	26	8.53	28760	298	20
34	4	P4-C34-D4-S1	20.52	6938	114	7	6.69	6938	114	7	17.40	8737	152	24	2.84	6938	114	7
34	4	P4-C34-D4-S2	81.04	31596	336	30	20.93	31596	336	30	29.71	21014	228	26	9.51	31596	336	30
34	8	P8-C34-D1-S1	38.20	19266	318	11	11.28	19266	318	11	45.10	22162	367	32	8.64	19266	318	11
34	8	P8-C34-D1-S2	104.97	40541	560	40	35.98	40541	560	40	132.31	70923	954	148	15.36	40541	560	40
34	8	P8-C34-D4-S1	68.75	19292	316	11	21.57	19292	316	11	45.35	22123	406	46	8.59	19292	316	11
34	8	P8-C34-D4-S2	65.56	35663	490	32	20.28	35663	490	32	129.03	66644	902	148	13.58	35663	490	32
34	12	P12-C34-D1-S1	50.22	18107	314	13	12.71	18107	314	13	50.43	22540	375	37	8.30	18107	314	13
34	12	P12-C34-D1-S2	71.74	28040	422	16	20.63	28040	422	16	66.17	32616	522	65	11.91	28040	422	16
34	12	P12-C34-D4-S1	50.45	18107	314	13	12.73	18107	314	13	50.60	22528	371	35	8.23	18107	314	13
34	12	P12-C34-D4-S2	71.74	22626	342	16	21.30	22626	342	16	64.62	32625	524	66	9.66	22626	342	16
Avg			58.86	22990	328	18	17.39	22990	328	18	56.45	29305	432	56	8.99	22990	328	18
47	4	P4-C47-D1-S1	86.76	86928	302	9	30.90	86928	302	9	123.50	96555	1062	395	39.42	86928	302	9
47	4	P4-C47-D4-S1	107.49	49300	226	6	36.94	49300	226	6	108.59	87739	820	228	15.93	49300	226	6
47	4	P4-C47-D1-S2	92.94	86928	302	9	30.82	86928	302	9	122.61	96555	1062	395	38.96	86928	302	9
47	4	P4-C47-D4-S2	112.05	49300	226	6	36.98	49300	226	6	108.34	87739	820	228	15.86	49300	226	6
47	8	P8-C47-D1-S1	196.43	96739	584	64	57.10	96739	584	64	68.82	59221	394	56	30.30	96739	584	64
47	8	P8-C47-D4-S1	258.78	104735	594	15	90.07	104735	594	15	251.28	147486	12460	5884	33.07	104735	594	15
47	8	P8-C47-D1-S2	307.70	51794	242	17	87.66	51794	242	17	180.86	127225	2044	728	15.27	51794	242	17
47	8	P8-C47-D4-S2	280.44	85795	482	15	94.16	85795	482	15	247.13	147486	12460	5884	26.05	85795	482	15
47	12	P12-C47-D1-S1	487.93	161455	730	80	132.72	161455	730	80	138.43	110556	422	49	108.61	161455	730	80
47	12	P12-C47-D4-S1	458.93	176507	942	84	169.92	176507	942	84	545.79	260293	14386	6456	63.41	176507	942	84
47	12	P12-C47-D1-S2	512.17	164464	628	68	139.29	164464	628	68	138.86	110589	390	27	132.26	164464	628	68
47	12	P12-C47-D4-S2	388.18	127887	622	19	144.93	127887	622	19	542.56	260293	14386	6456	41.84	127887	622	19
Avg			274.15	103486	490	33	87.62	103486	490	33	214.73	132645	5059	2232	46.75	103486	490	33
77	4	P4-C77-D1-S1	169.94	52981	190	10	61.23	52981	190	10	71.99	56700	388	94	23.47	52981	190	10
77	4	P4-C77-D4-S1	140.33	53244	218	7	47.76	53244	218	7	51.36	42085	216	38	17.54	53244	218	7
77	4	P4-C77-D1-S2	185.69	52994	194	12	58.60	52994	194	12	70.58	56700	388	94	21.69	52994	194	12
77	4	P4-C77-D4-S2	263.09	52574	208	5	88.83	52574	208	5	51.68	42085	216	38	17.06	52574	208	5
77	8	P8-C77-D1-S1	135.22	82374	366	16	40.77	82374	366	16	109.67	74140	1522	606	23.71	82374	366	16
77	8	P8-C77-D4-S1	582.72	114884	488	15	216.93	114884	488	15	306.01	197152	4572	1936	40.81	114884	488	15
77	8	P8-C77-D1-S2	182.88	92930	424	18	57.42	92930	424	18	330.94	174507	7376	3375	27.01	92930	424	18
77	8	P8-C77-D4-S2	230.13	234120	988	60	80.73	234120	988	60	339.89	220704	3114	1148	84.62	234120	988	60
77	12	P12-C77-D1-S1	564.89	117928	480	24	144.84	117928	480	24	634.13	154937	1352	407	70.89	117928	480	24
77	12	P12-C77-D4-S1	324.73	129412	542	22	86.30	129412	542	22	418.52	251958	5582	2296	44.48	129412	542	22
77	12	P12-C77-D1-S2	611.46	117928	480	24	171.72	117928	480	24	278.91	160245	8288	3908	70.25	117928	480	24
77	12	P12-C77-D4-S2	273.98	104211	458	21	90.90	104211	458	21	1,336.60	230824	100608	49761	35.38	104211	458	21
Avg			305.42	100465	420	20	95.50	100465	420	20	333.36	138503	11135	5308	39.74	100465	420	20

The results in Table 5.10 confirm that the acceleration strategies designed for the Benders subproblem enhance the two-phase performance. When the parallelism is deactivated, the Benders subproblems take more time. The increase factors are on average 7.2, 6.5, 5.9, and 7.7 for the four classes, respectively. When the hashing technique is deactivated, the Benders subproblems take more time as well. The increase factors are on average 2.0, 1.9, 1.9, and 2.4 for the four classes, respectively. The reason behind this is that when the solved TLPs are not memorized, they must be re-solved several times during subsequent iterations, which consumes a lot of time. We note that for No Parallel and No Hash, the number of columns and the number of iterations are not affected since the same solutions are found as with our approach. When the primal diving heuristic is deactivated, the Benders subproblem explores more branching nodes, which significantly increases the number of columns and the number of iterations, and consequently, the execution time. The increase factors are on average 7.6, 6.3, 4.6, and 8.4 for the four classes, respectively.

5.6.4 Comparison with Other Approaches

The first intuitive comparison one can think of is running the compact Model MDMPPSRPIM on CPLEX. However, given the size of the model, even small instances cannot be solved using CPLEX or any other solver. Thus, we did not conduct any runs on CPLEX.

Since most of the approaches used to tackle the MDMPPSRPIM are heuristics, we compare the results obtained using our approach with those obtained with the two following heuristics:

- Heuristic 1: A greedy heuristic that mimics the practice in real life. It delivers a quantity equivalent to 12 days of average consumption each time the client's inventory decreases below the safety stock level. Once the set of client-products that need to be delivered each day $d \in \mathcal{D}$ is defined, the routing problem is solved to find the best routes and trucks to satisfy the demand. It is worth mentioning that, in real life, this process is performed manually and requires two days of planning by the petroleum distribution company operator. To implement the heuristic, we use the DSPP to simulate the manual part.
- Heuristic 2: A decomposition heuristic based on decomposing the decision process of the petroleum distribution company. To create the planning for a single day $d \in \mathcal{D}$, the decision process is decomposed into five phases. Since the planning for a certain day $d \in \mathcal{D}$ affects the inventory levels of the next day, a dynamic programming approach is implemented to create the planning day by day. After each iteration, inventory levels are adjusted with the client-products orders. More details can be found in Boers et al. (2020).

In Table 5.11, we report the gap percentage (Gap), the number of visits(#Vis), the number of routes (#Ro), the number of trucks(#Tr), and the free space percentage (F) for the heuristics and our approach. We observe that Heuristic 1 outperforms Heuristic 2. For both of them, the integrality gap is above 5%. Compared to Heuristic 1, our approach achieves 6 fewer visits, using 3 fewer trucks, on 4 fewer routes, and with close free space percentages on average. Compared to Heuristic 2, our approach achieves 46 fewer visits, using 10 fewer trucks, on 18 fewer routes, with a close free space percentage on average.

Table 5.11 Comparison with Heuristics 1 and 2

N	W	Inst.	Heuristic 1					Heuristic 2					Our Approach				
			Gap(%)	#Vis	#Ro	#Tr	F(%)	Gap(%)	#Vis	#Ro	#Tr	F(%)	Gap(%)	#Vis	#Ro	#Tr	F(%)
24	4	P4-C24-D1-S1	5.03	55	44	26	12.30	18.77	68	49	28	10.41	0.05	54	44	27	12.50
24	4	P4-C24-D4-S1	5.63	48	38	25	14.71	15.28	57	44	26	8.98	0.05	48	37	23	14.60
24	4	P4-C24-D1-S2	5.03	55	44	26	12.30	8.63	68	49	28	10.41	0.05	54	44	27	12.50
24	4	P4-C24-D4-S2	5.93	48	38	25	14.71	7.17	57	44	26	8.98	0.05	48	37	23	13.59
24	8	P8-C24-D1-S1	5.13	112	91	59	10.55	20.97	140	101	63	9.50	0.05	109	91	58	9.70
24	8	P8-C24-D4-S1	8.97	106	84	54	10.24	17.25	128	96	61	9.45	0.09	104	84	54	10.57
24	8	P8-C24-D1-S2	5.13	112	91	59	10.55	19.86	140	101	63	9.50	0.05	109	91	58	9.70
24	8	P8-C24-D4-S2	5.35	106	83	56	10.37	16.42	128	96	61	9.45	0.05	104	83	55	9.06
24	12	P12-C24-D1-S1	5.00	165	137	84	8.98	12.25	210	153	95	10.16	0.05	166	135	87	9.09
24	12	P12-C24-D4-S1	5.00	161	130	85	8.32	13.18	199	143	91	8.71	0.05	159	131	85	8.63
24	12	P12-C24-D1-S2	5.10	165	137	84	9.01	13.41	210	153	95	10.16	0.05	166	135	88	8.79
24	12	P12-C24-D4-S2	5.00	161	130	84	8.32	11.51	199	143	91	8.71	0.05	159	131	85	8.59
Avg			5.52	108	87	56	10.86	14.56	134	98	61	9.53	0.05	107	87	56	10.61
34	4	P4-C34-D1-S1	5.28	74	60	35	14.03	13.45	96	71	44	11.91	0.05	72	58	38	13.65
34	4	P4-C34-D1-S2	8.27	79	60	36	18.79	17.10	75	62	36	11.63	0.70	70	55	35	18.39
34	4	P4-C34-D4-S1	5.31	74	60	35	14.03	17.93	96	71	44	11.91	0.05	72	58	38	13.65
34	4	P4-C34-D4-S2	5.02	79	60	36	18.79	10.21	75	62	36	11.63	0.42	71	53	34	17.72
34	8	P8-C34-D1-S1	5.41	151	125	85	11.56	21.13	195	146	93	11.48	0.05	146	121	84	10.37
34	8	P8-C34-D1-S2	11.73	160	129	84	13.83	16.21	175	140	85	11.92	0.20	149	121	75	11.93
34	8	P8-C34-D4-S1	5.24	151	125	83	11.55	15.58	195	146	93	11.48	0.05	147	122	85	11.06
34	8	P8-C34-D4-S2	11.00	160	129	84	13.73	15.40	175	140	85	11.92	0.09	148	117	74	11.02
34	12	P12-C34-D1-S1	4.93	219	185	122	9.47	11.70	291	223	137	12.59	0.05	219	186	125	9.48
34	12	P12-C34-D1-S2	6.07	233	194	124	11.91	13.51	272	210	128	11.41	0.05	221	183	119	9.54
34	12	P12-C34-D4-S1	4.93	219	185	125	9.48	10.54	291	223	137	12.59	0.05	219	186	125	9.48
34	12	P12-C34-D4-S2	5.92	233	195	121	11.88	12.99	272	210	128	11.41	0.05	223	187	123	10.10
Avg			6.59	153	126	81	13.25	14.65	184	142	87	11.82	0.15	146	121	80	12.20
47	4	P4-C47-D1-S1	5.55	94	57	42	9.44	31.87	109	70	50	7.75	0.05	94	55	42	8.06
47	4	P4-C47-D4-S1	11.66	103	63	46	12.53	29.14	96	64	46	7.53	0.05	87	51	40	6.84
47	4	P4-C47-D1-S2	5.54	94	57	42	9.44	29.91	109	70	50	7.75	0.05	94	55	42	8.06
47	4	P4-C47-D4-S2	11.37	103	63	46	12.53	23.48	96	64	46	7.53	0.05	87	51	40	6.84
47	8	P8-C47-D1-S1	13.30	197	137	93	9.09	12.63	246	149	104	6.79	0.13	195	139	93	8.21
47	8	P8-C47-D4-S1	6.54	210	141	98	8.79	15.17	234	146	103	7.19	0.05	193	132	91	7.02
47	8	P8-C47-D1-S2	5.30	197	137	96	9.06	16.12	246	149	104	6.79	0.05	205	138	93	8.95
47	8	P8-C47-D4-S2	1.49	210	141	98	8.79	13.25	234	146	103	7.19	0.00	195	132	87	7.07
47	12	P12-C47-D1-S1	10.08	293	211	146	6.80	7.58	376	226	159	6.69	0.10	293	206	142	6.53
47	12	P12-C47-D4-S1	6.71	316	214	138	7.34	7.64	364	222	159	6.88	0.06	299	204	147	5.78
47	12	P12-C47-D1-S2	9.21	293	212	144	7.01	9.36	376	226	159	6.69	0.09	295	205	148	6.28
47	12	P12-C47-D4-S2	3.64	316	214	138	7.34	7.47	364	222	159	6.88	0.03	303	201	143	6.02
Avg			7.53	202	137	94	9.01	16.97	238	146	104	7.14	0.06	195	131	92	7.14
77	4	P4-C77-D1-S1	5.45	161	107	63	10.23	12.78	191	127	71	8.88	0.05	161	103	64	9.29
77	4	P4-C77-D4-S1	8.42	164	105	63	12.85	14.53	163	114	65	8.23	0.05	146	98	57	11.64
77	4	P4-C77-D1-S2	3.34	161	107	63	10.23	16.60	191	127	71	8.88	0.03	164	105	65	9.48
77	4	P4-C77-D4-S2	8.52	164	105	63	12.85	17.40	163	114	65	8.23	0.05	144	97	55	11.56
77	8	P8-C77-D1-S1	5.25	331	245	137	9.68	17.56	418	268	150	7.97	0.05	327	241	125	8.97
77	8	P8-C77-D4-S1	1.98	338	237	132	9.86	15.03	388	256	144	8.16	0.01	321	228	126	8.04
77	8	P8-C77-D1-S2	1.22	331	245	139	9.56	18.14	418	268	150	7.97	0.01	327	242	131	9.30
77	8	P8-C77-D4-S2	14.76	338	239	133	9.78	14.90	388	256	144	8.16	0.14	323	231	130	8.53
77	12	P12-C77-D1-S1	8.57	493	369	208	7.93	9.61	633	403	227	7.96	0.05	495	372	202	7.90
77	12	P12-C77-D4-S1	5.44	503	364	214	8.03	9.07	606	390	216	7.80	0.05	495	350	198	6.91
77	12	P12-C77-D1-S2	10.03	493	369	210	7.96	9.08	633	403	227	7.96	0.05	495	372	202	7.90
77	12	P12-C77-D4-S2	11.07	503	363	214	7.99	9.37	606	390	216	7.80	0.01	494	350	195	6.75
Avg			7.00	332	238	137	9.75	13.67	400	260	146	8.17	0.05	324	232	129	8.86

5.6.5 Managerial Insights

The main insight is that, from a systemic point of view, the proposed approach is very interesting for two reasons. First, it leverages the usage of the tool developed in Bani et al. (2023), which can be either used as a standalone for route planning or integrated to incorporate the inventory management component. This tool has been shown to be very effective when tested independently. Second, given the short execution time, the approach allows decision-makers and users to check several what-if scenarios and run re-optimization in case of disruptions. We further support the analysis above with additional tests and statistics highlighting several managerial gains.

Figure 5.6 reports the number of routes and trucks used per week for the **P12-C77-D4-S1** instance. On average, the number of trucks used represents around 60% of the number of routes. This implies that specific trucks are preferred over others. This might be explained by the compartment numbers and sizes of each truck. For clients requiring small quantities, trucks with several, smaller-capacity compartments are preferred since several clients can be grouped on a single route. For clients requiring large quantities, trucks with fewer, larger-capacity compartments are preferred.

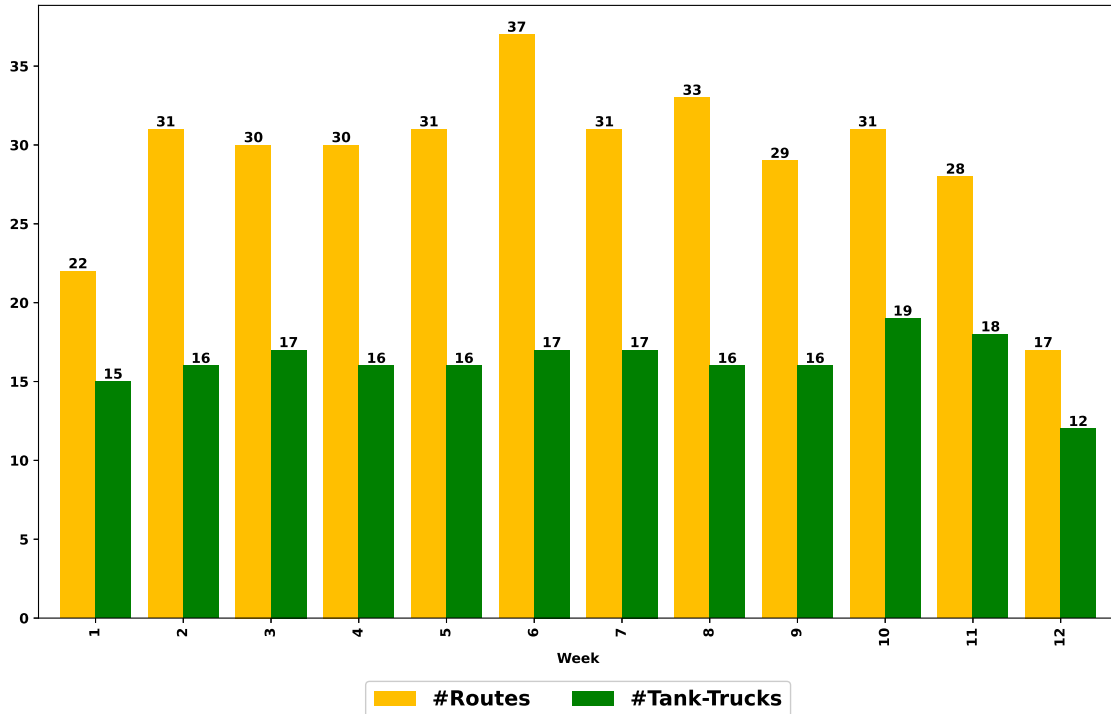


Figure 5.6 Number of Routes and Trucks Used per Week **P12-C77-D4-S1**

Results also highlight that, on average, the number of clients visited by a given truck (on a single route) is 1.5. This can be explained by the clients' large tank capacities, which implies that the trucks will serve few clients. This is similar to the American context, where direct billing is the main trend.

We report in Figure 5.7 the average free space in trucks per week for all 48 instances. We observe that 32 instances have an average free space of less than 10%, 14 instances have free space between 10% and 15%, and only 2 instances have free space higher than 15%. This shows that the two-phase approach tends to maximize the utilization of the trucks. This also depends on the significance of the cost given to penalize free space within trucks. The higher this cost, the less empty the trucks are in the optimal solutions.

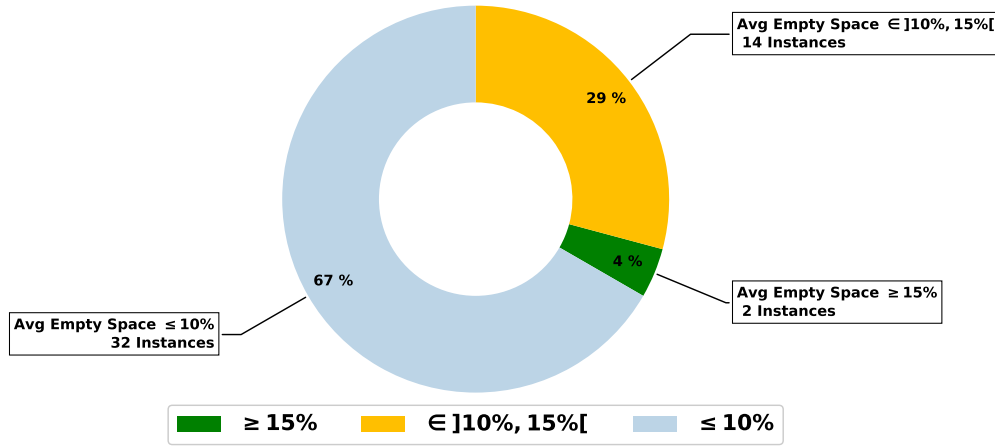


Figure 5.7 Average Free Space Percentage Within Trucks per Week for All 48 Instances

Figure 5.8 highlights the number of visits per client-products for instance **P12-C77-D4-S1**. We compare the minimum number of visits required and the number of visits performed (obtained from the optimal solution). The minimum number of visits required is approximated as the ratio of weekly capacity to demand $\frac{C}{D}$ for each client-product $n \in \mathcal{N}_p$.

The results highlight that, at most, the number of visits performed is larger than the minimum threshold by 3. For most instances, the near-optimal solution performs either the minimum number of visits required or 1 additional visit. This can be explained by the delivery strategy in the near-optimal solution. When a route contains several clients, a client might not receive the order within the minimum number of visits. When the strategy is direct billing, the number of visits performed tends to be equal to the minimum number of visits required. This also depends on the type of truck used for delivery.

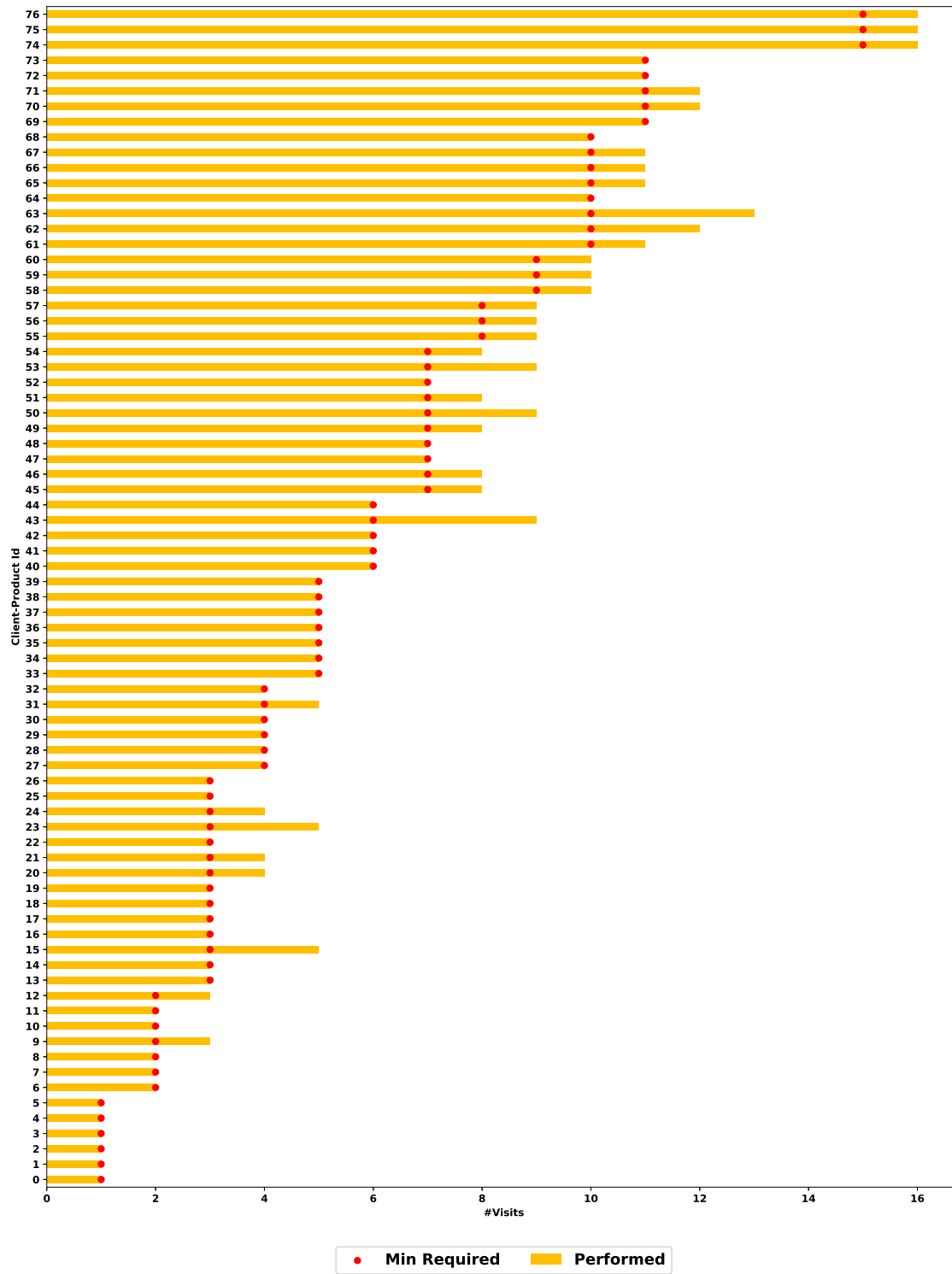


Figure 5.8 Number of Visits Required and Performed for Each Client-Product for Instance P12-C77-D4-S1

We lastly discuss an interesting aspect for oil companies, namely, when to buy and when to store. Figure 5.9 shows the inventory (variation) levels and the quantity purchased over 12 weeks for instance **P12-C77-D4-S1**. We observe that the optimal solutions follow a *buy to sell* strategy, i.e., the petroleum distribution company buys the quantity required for a given week and sells it in the same week, without keeping any stock, except for weeks 9, 10, and 11, where the company buys a large quantity in week 9 and consumes it over weeks 9, 10, and 11. This can be explained by the variations in the purchasing and inventory costs. When the former is lower, the company buys and delivers. When the latter is lower, the company buys when the purchasing cost is low and keeps the stock.

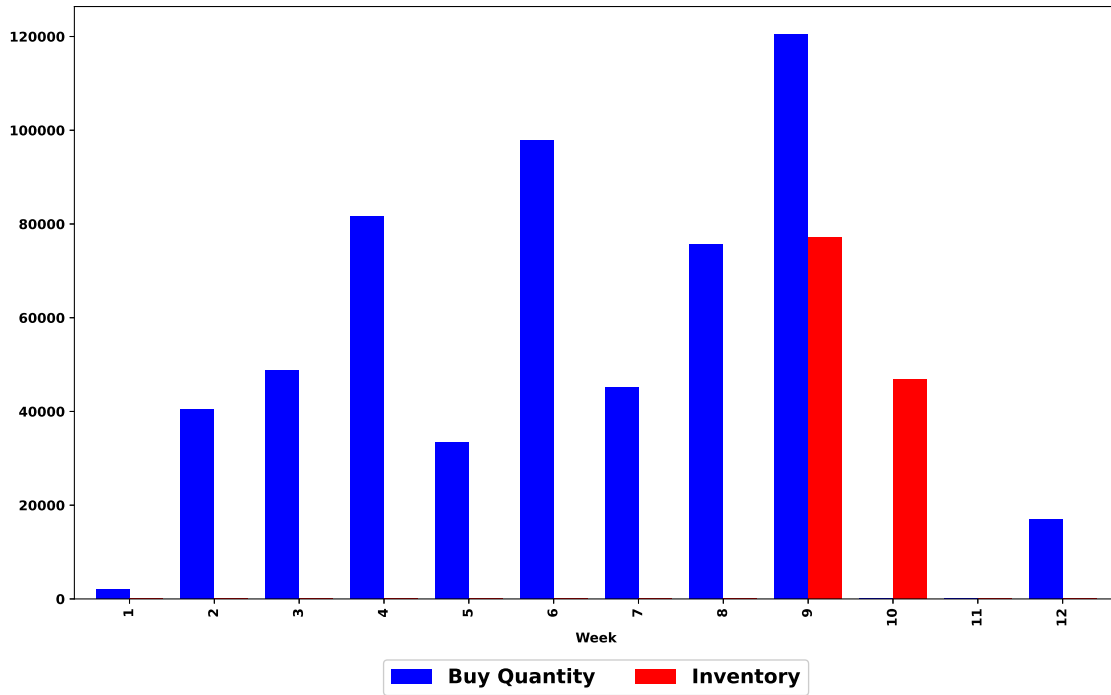


Figure 5.9 Inventory Levels (Variation) and Buying Pulses for Instance **P12-C77-D4-S1**

A similar observation holds when the transport costs are higher than the inventory costs. In such a case, we observe that the number of visits is lower. The inverse behavior is observed when the inventory costs are higher. When the petrol price increases significantly, the delivery is restricted to the quantities needed by the clients.

5.7 Conclusion

The petrol station replenishment problem is a famous and well-studied operations research problem. For many practical applications and large-scale cases, the size of the underlying

MILP is extremely large, making the direct use of a MILP solver inefficient. Driven by a decoupling intuition, our motivation was to provide a generic exact framework that could help efficiently solve the problem as well as its complex variants.

We therefore investigated an exact MILP approach based on combining Benders decomposition and column generation to tackle a complex variant of the PSRP. The proposed two-phase approach stabilizes the inventory levels in the first phase, before finding a near-optimal integer solution in the second phase. The proposed approach is enhanced through several acceleration strategies, including warm-start, parallelism, a hashing technique, and a primal diving heuristic. The computational results on realistic instances from a geographical zone in West Africa show that the novel approach reaches near-optimal solutions and significantly outperforms existing techniques in the literature.

While this work was motivated by practices in Africa, it makes several fundamental and methodological contributions and can be extended in several ways. First, although we assume deterministic parameters, several parameters, such as client demands, are uncertain in real life. Hence, a promising extension direction is to design solution methods for the MDMPPSRPIM under demand uncertainty. Second, the inventory forecast can be done using more sophisticated deep learning models and incorporated into the optimization model.

CHAPITRE 6 ARTICLE 3: THE PETROL STATION REPLENISHMENT PROBLEM : A CASE STUDY FROM WEST AFRICA

A. Bani, E. M. Er Raqabi, I. El Hallaoui et A. I. Corr  a ont   crit cet article et soumis le 15 mars 2024    la revue African Transport Studies.

6.1 Introduction

Despite the increasing adoption of electric vehicles (Collett et al., 2021; Er Raqabi & Li, 2023), Africa still relies heavily on petrol products for transport, even with the emergence of other energies (e.g., electric, hydrogen). For petrol distribution companies, this comes with a need to invest heavily in the infrastructure, the petrol tank truck fleet, and the business analytics tools that can ensure better performance for all stakeholders in the petroleum distribution sectors (e.g., distribution companies, petrol stations, governments, and customers).

The petrol station replenishment process has three parts: the loading, the routing, and the unloading. The trip starts from a depot, where a multi-compartment tank truck is loaded with petrol products. Following the loading, the tank truck visits some petrol stations. These petrol stations usually have underground tanks where the petrol products are unloaded. Once the trip is completed, the tank truck returns to the depot. This generic problem, with many variants in the literature, is referred to as the petrol station replenishment problem (PSRP). Mathematically, it integrates the tank truck loading problem (TTLP) and the inventory routing problem (IRP). Traditionally, the TTLP and IRP have been tackled sequentially, with tank truck loading being an input for the routing problem. The sequential procedure leads to suboptimal and sometimes infeasible solutions. Furthermore, when considering inventory management at the depots and petrol stations, attempts to tackle the PSRP variant obtained have been mainly based on heuristics rather than exact methods.

6.1.1 Contributions

Using operations research (OR) techniques, Bani et al. (2023) tackled exactly a complex and realistic variant of the PSRP, referred to as the multi-depot multi-period petrol station replenishment problem (MDMPPSRP). Furthermore, Bani et al. (2024) solve exactly another complex and realistic variant of the PSRP, referred to as the multi-depot multi-period petrol station replenishment problem with inventory management (MDMPPSRPIM). Our study has two main contributions:

- Leveraging insights from Bani et al. (2023) and Bani et al. (2024), we highlight a gradual approach to tackle a real-life MDMPPSRPIM from West Africa. In the first step of the approach, we tackle the weekly PSRP using Column Generation (CG). In the second step of the approach, we tackle the MDMPPSRPIM by combining Benders decomposition (BD) and CG.
- We use this gradual approach to conduct extensive experiments, which highlight the following: (1) Among two popular billing policies, we highlight that the direct billing policy is more costly than the standard billing policy, (2) The oil price fluctuations can inform buying and stocking decisions. In particular, if the oil price fluctuates significantly, stocking becomes profitable for the Petroleum Company (PC), (3) At petrol stations, there are optimal tank capacities. In our experiments, we found that increasing the tank capacities by 15% ensures a significant decrease in total cost, and (4) Among all available types of tank trucks (more than 40), less than 14 types are used frequently.

The remainder of this paper is organized as follows. We discuss the relevant studies in Section 6.2. We describe the problem and the methods in Sections 6.3 and 6.4, respectively. The results and the discussion are highlighted in Section 6.5, followed by conclusions in Section 6.6.

6.2 Literature Review

Considered as the most studied variant of the multi-compartment vehicle routing problem (Derigs et al., 2011), the PSRP has received substantial attention in the literature over the last decades. Several variants have been studied after the problem was formulated for the first time by Brown & Graves (1981). For a comprehensive overview of the history of the PSRP, the papers published by Cornillier et al. (2012), Benantar et al. (2016), and Lima et al. (2016) provide additional background on the PSRP. Many researchers created models to address real-world problems to help organizations reduce their logistic costs (Ng et al., 2008; Triki, 2013; Benantar et al., 2016; Al-Hinai & Triki, 2020).

6.2.1 Tank Truck Loading Problem

In the context of the PSRP, tank trucks are used for the petrol products distribution. However, due to compatibility constraints, it is only possible to load one petrol product into each compartment Kaabi & Jabeur (2015). Coelho & Laporte (2015) have introduced several mixed integer linear programming (MILP) models to address this issue, categorizing them as

fractional and non-fractional delivery. Fractional delivery allows for the distribution of compartment contents among multiple petrol stations, while non-fractional delivery necessitates the complete delivery of compartment contents to a single petrol station. Furthermore, it is worth noting that previous studies often assumed that the tank trucks don't have flow meters, resulting in the requirement to empty the compartment at each petrol station (Macedo et al., 2011; Cornillier et al., 2008a,b; Benantar et al., 2016; Popović et al., 2012; Vidović et al., 2014). This absence of flow meters does not allow accurately measuring the quantity of product delivered to each petrol station. In such a case, a constraint is needed to ensure at most one petrol product from one petrol station can be loaded in a tank truck compartment. This type of constraint is hard to solve because we must solve a series of branching nodes.

In their study, Wang et al. (2020) propose a novel approach for the PSRP, which involves using a different TTLP with similar compartments. These compartments are equipped with flow meters, enabling the assignment of multiple orders of the same petrol product to each compartment. This feature simplifies the problem by transforming the TTLP into assigning petrol quantities to the compartments rather than petrol station orders' quantities. In this case, the TTLP can be formulated as a pure linear program, and given its small size, it solves very quickly.

Recently, in Bani et al. (2023), even if the tank trucks are equipped with flow meters, for management simplification and other administrative reasons, a compartment is assigned in its entirety at most to a single petrol station. They solve the MDMPPSRP using CG. The latter involves solving an elementary shortest path problem with resource constraints in the CG subproblems, where hundreds of millions of extensions are performed. Each extension includes the resolution of a TTLP that ensures the feasibility of loading the petrol station's orders into the tank truck at each extension. To solve the TTLP, Bani et al. (2023) use a hashing technique that they employ in each iteration of their solution for the TTLP. This technique allows the memorization of solved TTLPs to leverage them again in subsequent iterations. By reducing the number of MILP problems that need to be solved in each iteration, the hashing technique significantly decreases the execution time of the CG subproblem labeling algorithm. Additionally, the authors demonstrate that examining the total petrol stations' orders allows the optimization of various resources, such as the tank truck capacity and the compartment number.

6.2.2 Inventory Routing Problem

In recent studies, researchers have examined the introduction of IRP in PSRP to reduce both routing and inventory costs. Popović et al. (2011, 2012); Vidović et al. (2014) have all

explored this approach.

The underlying assumption in the problems above is the vendor-managed inventory (VMI) concept. Under this concept, the PC has access to information regarding inventory levels and is responsible for making decisions regarding replenishment. Decisions to take simultaneously include the delivery time and quantities for each petrol station, and the most efficient routes for petrol station service. This arrangement benefits both the PC and vendors, as it helps minimize distribution costs and eliminates the need for vendors to allocate resources towards inventory management (Coelho et al., 2014).

There are various inventory policies to determine the quantities to deliver. The two inventory policies are the *Maximum-Level* policy and the *Order-Up-to-Level* policy. Under the *Maximum-Level* policy, the order quantity is flexible but lower than the maximum capacity. On the other hand, the *Order-Up-to-Level* policy utilizes a fixed order quantity, which ensures that the inventory is maintained beyond specific level (Coelho et al., 2014). The most commonly used policy for the PSRP is the *Maximum-Level* policy. This policy has been applied by Cornillier et al. (2008a) and Popović et al. (2011), who set limits on the delivery quantities based on the petrol station tank capacity. Li et al. (2014), on the other hand, implemented an *Order-Up-to-Level* policy where the delivery quantity is set to the remaining tank capacity at the time of delivery.

6.2.3 Petrol Station Replenishment Problem

Many variants of the problem have emerged, differing in whether planning takes place over one or multiple periods, whether tank trucks are homogeneous, and whether there are time windows to model delivery times.

PSRP Variants: i) The standard PSRP consists of one depot, a single period, and no time windows (Brown & Graves, 1981). ii) The MPPSRP, a multi-period PSRP, has been studied by Malépart et al. (2003); Triki (2013); Archetti et al. (2015b). iii) The PSRPTW is a PSRP with time windows, as discussed by Cornillier et al. (2009). iv) The MDPSRPTW is a multi-depot PSRPTW, which has been addressed by Cornillier et al. (2012) using heuristics. v) The MDMPPSRP, a multi-depot MPPSRP, has been tackled by Carotenuto et al. (2018) and Bani et al. (2023).

Heuristic/Matheuristic Approaches. Most surveyed papers have proposed heuristic or metaheuristic approaches to address the PSRP variants. This preference stems from the challenges associated with implementing exact methods, which are often more complex, and their limited suitability for solving large-scale instances. As pointed out by Laporte (1992), heuristic and metaheuristic approaches are considered more practical in such cases. The

frequently utilized metaheuristics include cluster first (which considers geographic limitations and expert considerations) followed by route determination, adapted VNS introduced by Hansen & Mladenović (2001), load first (which involves assigning orders to tank trucks) followed by route determination and tailored heuristics. It is important to note that for the problems addressed, neither lower bounds nor optimal solutions are currently known, making it challenging to assess the quality of the obtained solutions.

Exact Models/Approaches. Only a few papers have attempted to develop exact models or solution methods for solving small instances of the PSRP. For instance, Cornillier et al. (2008a) decomposed the standard PSRP into two subproblems, namely the TTLP and the routing problem (RP), using a CG scheme. The TTLP assigns orders to compartments to maximize profit, while the RP focuses on selecting routes that minimize overall transportation costs. On the other hand, Avella et al. (2004) proposed a *Branch-and-Price* algorithm for the standard PSRP, allowing for multiple petrol stations per route. They use a heuristic approach to generate an initial set of columns for the algorithm and test real-world data with specific characteristics. Additionally, Benantar et al. (2016) solved small instances of the MPPSRP using CPLEX. In another study, Cornillier et al. (2008a) presented an exact model for the MDPSRPTW, which aimed to select a subset of feasible trips that satisfy demand while maximizing overall daily net revenue. However, due to the large number of possible trips, the authors proposed an alternative heuristic approach. Bani et al. (2023) solve the integrated TTLP and RP simultaneously using an exact *Branch-and-Price* algorithm with multiple petrol stations per route.

6.2.4 State of Practice

America. One of the pioneering studies in the field of petroleum distribution is the study of Brown & Graves (1981) in the United States. Their research focused on a real-life case for the PSRP and aimed to automate the dispatching of tank trucks. This study laid the foundation for future advancements in optimizing the distribution process. In a subsequent study, Brown et al. (1987) explored a planning optimization approach for a PC in the United States. The objective of this research was to enhance personnel productivity and reduce costs. By implementing their proposed optimization model, the PC improved significantly efficiency and cost savings. Kazemi & Szmerekovsky (2015) examined petrol distribution in the United States provinces as an IRP. Cornillier et al. (2008a) further expanded on petroleum distribution network optimization. Their study proposed a heuristic approach to solve the problem at hand. The authors successfully demonstrated the benefits of optimization to the PC, leading to the integration of the proposed algorithm into the planning systems of the

distribution company. This integration increased awareness about optimization advantages and facilitated the practical implementation of the algorithm in real-world scenarios.

Europe. A notable contribution to the field was made by Van der Bruggen et al. (1995) through their consultancy study conducted in the Netherlands. Their research focused on optimizing the distribution network of a large PC. Specifically, they aimed to improve the efficiency of transporting petrol products from the depot to petrol stations. The findings of this study provided valuable insights for the management of the PC, aiding them in restructuring their distribution network.

Asia. Several researchers have developed various models to analyze real-life scenarios. In one study, Ng et al. (2008) focused on enhancing the petrol distribution within a network in Hong Kong. The authors implement the VMI concept. The proposed approach proved beneficial for the PC as it resulted in an increase in delivery volume and a reduction in driver costs. In another investigation, Triki (2013) utilizes real-life cases to evaluate the effectiveness of solution methods proposed by the authors for solving the PSRP. The most efficient method yielded a remarkable 17.7% savings compared to planning conducted by a human operator. Additionally, they explore the PSRP in the context of a PC in Oman. They present a MILP model for bidding in transportation procurement auctions. Li et al. (2014) focused on a large PC in China and modeled the distribution of petrol products across provinces as an IRP. They developed a heuristic approach that demonstrated near-optimal solutions. They employed a MILP model to determine the optimal supply chain design, incorporating multi-modal transportation methods and facility locations.

Africa. The case study conducted by Benantar et al. (2016) evaluated the distribution process for an Algerian PC. The authors proposed a model for the PSRP that incorporated compartmented tank trucks and time windows. Their method outperformed the company's existing solution in the number of tank trucks required and the total travel distance.

Globally. Petrol product prices have been fluctuating significantly over recent years. For instance, during the COVID pandemic, the world witnessed historic petrol product price fluctuation (Le et al., 2021). To face fluctuations, governments implement and reform subsidy systems to ensure the smooth supply of petrol products. For instance, major reforms to the subsidy system in Morocco began on September 16th, 2013, with the decision to reactivate the price indexation mechanism for petrol products, including *Gasoline*, *Diesel*, and *Fuel oil* (Verme & Araar, 2017). The highest petrol barrel price was \$147.27 in July 2008. As of March 1st, 2024, the petrol barrel price is \$79.81 per barrel. The petrol barrel prices significantly affect the petrol products purchase and transport costs. The purchase cost is directly linked to the raw material price, and the transport cost is also directly linked since

Gasoline is the main energy used for transport. Both *Gasoline* and *Diesel* are stored in large cylindrical underground tanks. Each tank has a capacity of 21,000 to 22,000 liters. The number of tanks depends on the number of dispensing machines on the ground. A typical petrol station has a tank capacity of 110,000 to 150,000 liters (Evans, 2009). For billing policies, we distinguish direct billing and standard billing. The direct billing policy is billing the petrol station based on the kilometer (km) distance from the depot. The standard billing scenario is billing the petrol station based on the distance in km from the previous stop (e.g., another petrol station served just before).

6.3 Problem

The problem under investigation centers on the PC's decision-making process regarding the types of petrol products to be delivered and the petrol product volumes transported by contracted distributors over a specified planning horizon. These private transport companies operate a diverse fleet of multi-compartment tank trucks, each with limited capacity, to replenish two distinct groups of petrol products for three different classes of petrol stations. The objective is to schedule efficiently the weekly replenishment for these petrol stations. During each planning horizon, the PC faces constraints on the available quantity of petrol products (stored in multiple depots).

Planning Horizon. The planning horizon is up to four months. We decompose the planning horizon into weeks, each having seven days.

Petrol Products. There are a total of four petrol products: Two unmarked (*Super Gasoline* and *Gasoline*) and two marked (*Marine Zoom* and *Diesel*). These petrol products are chemically marked to detect fraud because the state subsidizes them. Additional customs escort fees apply to marked petrol products delivered to marine stations, and the amounts may vary between destinations. Here, we are considering the case where the total supply of each petrol product type is greater than or equal to the total demand for that type.

Depots. The petrol products depots are in the capital and have a limited capacity. Some of them specialize in specific types of petrol products they store. We want to determine the distribution of petrol product volumes that each transport company must distribute to the petrol stations. The goal is to minimize the distribution cost.

Petrol Stations. The network of petrol stations, comprising 12 regular petrol stations, 37 bakeries, and 11 marine stations, is distributed throughout Senegal (see Figures 6.1). The PC owns some petrol stations, while others are private. The regular petrol stations are equipped with two *Super Gasoline* tanks and 1 *Gasoline* tank. The Bakeries generally have a tank

capacity of 1000, 2000, or 5000 liters for *Diesel*. Maritime stations typically have a single tank with a capacity ranging from 14000 to 50000 liters, except some have two tanks. The average demand for each product in a petrol station is constant and deterministic, providing known daily consumption quantities before optimization. A discount of 40% on custom escort fees is applicable when two maritime stations are visited successively. Multiple tank trucks can visit the petrol station during the same week for supplies of different petrol products. Some petrol stations are close to the petrol products depots, while others are very far away, some in cities with regulated access, others in rural areas with less easy access, making the routing problem more challenging and richer.

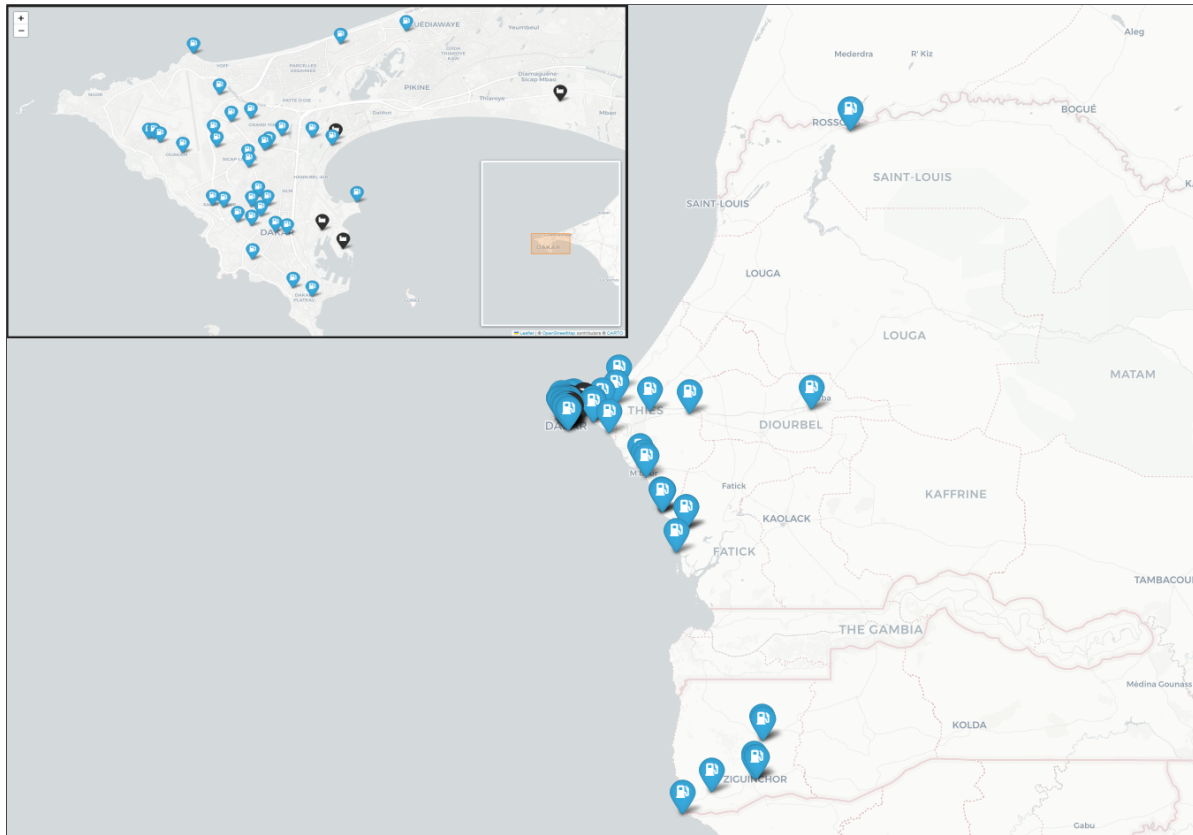


Figure 6.1 PC Petrol Stations (Blue) and Depots (Black)

Tank Trucks. There are four private transport companies, each maintaining fleets composed of heterogeneous multi-compartment tank trucks located at the transport company depots. The number of compartments varies from one tank truck to another, ranging from 4 to 15. The total capacity of a tank truck is the sum of the capacities of its compartments. There are 44 tank trucks, but not all may be available simultaneously, as other PCs allocate them based on petrol station demands for tasks in different locations. All tank trucks are assumed to travel at the same average speed, resulting in equal travel times for the same

distance. A tank truck is active as soon as at least one compartment is in use, even though it may travel with empty compartments, which should be minimized. Only tank trucks available within the planning horizon are considered, and once a delivery is completed, the private transport company incurs fixed rental costs for the tank truck. Some tank trucks are exclusively dedicated to transporting marked petrol products. These features should be considered in the search strategy for feasible solutions. Transportation costs per kilometer are fixed due to government price equalization for petrol products. Some tank trucks may receive preferential treatment if painted in PC colors (mobile advertising), agree to pay partially for petrol products, or are close to high daily consumption petrol stations.

Routes. The starting point for each route is the petrol products depot, with the endpoint being the transport company's depot. Petrol station routes are categorized into two families: routes for marked petrol products ordered by maritime stations and bakeries and routes for unmarked petrol products ordered by general petrol stations. Some routes for unmarked petrol products may consist of only one service point and are termed singleton routes. For each singleton petrol station, the tank truck supplying it must return directly to the depot immediately after the supply. All empty return trips to the transport company's depot are not invoiced to the PC. Only the travel time is considered to ensure the feasibility of the routes. Once a tank truck is parked in the loading area, the loading time for petrol products at the depot is assumed to be 1 hour. The unloading time at a petrol station depends on the diameter of the hoses used. Still, an average time can be established: 1 hour and 30 minutes for bakeries and 30 minutes for general petrol stations.

Planning Decisions. Allocation of distribution requests to the transport companies involves ensuring that all tank trucks used to supply each petrol station are accurately identified. Then, petrol volumes are assigned to specific tank trucks and their compartments, resulting in a comprehensive truck-loading plan. Details of the supplied petrol stations, including the tanks that have been replenished and the corresponding quantities received, are recorded. Additionally, the routes taken by the tank trucks, specifying the order of visits to the petrol stations, are documented for tracking and logistical analysis.

Petrol Station Constraints. We distinguish the following petrol station constraints: i) Meeting the demand of each petrol station for every type of petrol product (i.e., preventing any petrol station from running out of stock for each period). ii) Adhering to the tank capacity of each petrol station. iii) Ensuring the petrol station is not visited more than once per period to deliver the same petrol product type. iv) Imposing a constraint that determines the inventory for each petrol station at the end of each period for each type of petrol product. v) As the visited petrol stations are supplied with at least one petrol product, it is possible to

introduce an additional constraint to ensure that the petrol station visited during the same route does not exceed the number of compartments in a tank truck.

Tank truck constraints. We distinguish the following tank truck constraints: i) Adhere to the capacities of tank truck compartments. ii) Each compartment is restricted to containing only one petrol product from one petrol station for each tank truck route. iii) A tank truck cannot simultaneously transport marked and unmarked petrol products on the same route. iv) Some tank trucks are exclusively designated for transporting the marked petrol products. v) A tank truck transporting marked petrol products can visit at most two maritime stations during the same route. vi) Ensure that a compartment is assigned to deliver a petrol product to a petrol station only if it needs to be visited. vii) Guarantee that if a petrol station is visited during the period, it will be served at least one petrol product.

Depots constraints. We distinguish the following depot constraints: i) Ensure the availability of petrol products in each depot for each period. ii) Impose a constraint that determines the inventory for each depot and petrol product at the end of each period for each petrol product type.

The presented problem is a rich PSRP variant, which results in a complex model with millions of constraints and variables, among which several are integers. The mathematical formulation without inventory management is available in Bani et al. (2023) while the mathematical formulation with inventory management is available in Bani et al. (2024). Both problems have a large size. Solving them with generic commercial solvers is untractable, even for small instances. Thus, we design a two-step method highlighted next.

6.4 Method

In this section, we present the framework. Then, we describe each step of the two-step method.

6.4.1 Framework

We highlight the two-step method in Figure 6.2. In the first step, we tackle the MDMPPSRP. In the second step, we tackle the MDMPPSRPIM. The insight behind the two-step method is solving first the weekly MDMPPSRP using CG. Then, to solve the MDMPPSRPIM, we use BD to decompose the MDMPPSRPIM into several weekly MDMPPSRP that we solve efficiently using *Step 1*.

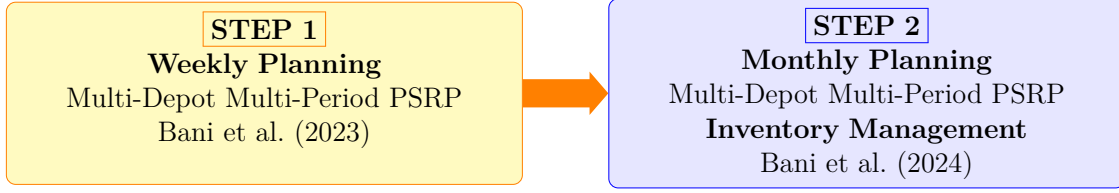


Figure 6.2 Two-Step Approach

6.4.2 First Step

The first step addresses the MDMPPSRP. In this step, we solve the complex PSRP variant with an exact *Branch-and-Price* approach and some derived heuristics. An important feature is that we can model this problem as a set partitioning type problem with low to moderate density (the number of ones per column, i.e., petrol stations to visit, is not large). The set partitioning type problems have some interesting polyhedral properties to consider for favoring integrality. Furthermore, some case-specific complex handling rules apply. A natural way is to address them in the CG subproblem as an elementary shortest path problem with resource constraints, which constitutes the major bottleneck. To succeed in tackling the latter, we need to design some sophisticated techniques: i) for branching to profit from the polyhedral properties and ii) for solving the CG subproblem. The direct use of on-the-shelf algorithms does not work, unfortunately. More details about the problem and the mathematical formulation can be found in Bani et al. (2023).

6.4.3 Second Step

Driven by a *decoupling* intuition, we develop an exact two-phase solution approach that combines BD and CG for the MDMPPSRPIM. This intuition comes from the fact that the inventory management constraints couple the planning horizon weeks because they link all buying, storage, and transport decisions in a given week with subsequent weeks. Furthermore, keeping these constraints in the model makes solving very difficult since the model is enormous. The BD (Rahmaniani et al., 2017; Er Raqabi et al., 2023b) allows us to achieve the *decoupling* intuition. Once the inventory variables (and the corresponding linking constraints) are fixed using the solution of the BD master problem, the MDMPPSRPIM becomes decoupled. The BD allows decomposing the MDMPPSRPIM into $|\mathcal{W}|$ MDMPPSRPs, which are solved using the first step insights. In the first phase, we solve the relaxed (integrality) BD subproblems using CG until the inventory levels stabilize. In the second phase, we solve the BD subproblems using CG embedded in a branch-and-bound framework. We enhance the two-phase solution approach with acceleration strategies, including warm-start, parallelism,

a hashing technique, and a primal diving heuristic. More details about the problem and the mathematical formulation can be found in Bani et al. (2024).

6.5 Results

In this section, we run computational experiments on realistic instances. We first describe the experimental design. After that, we highlight the computational results.

6.5.1 Experimental Design

For our tests, we consider 48 realistic instances introduced by Bani et al. (2024). These instances belong to two seasons (2016 and 2017). The features of these instances presented in Table 6.1 are: the number of client-products ($|\mathcal{N}|$), where a client-product represents a petrol product $p \in \mathcal{P}$ requested by a petrol station, the number of petrol stations ($\#C$), the number of petrol products ($\mathcal{P}$), the number of weeks/days ($|\mathcal{W}|/|\mathcal{D}|$), the number of depots ($|\mathcal{E}|$), and the instance name. In total, we have four types of petrol products. The instance name includes the number of weeks, the number of client-products, the number of depots, and the season. For example, instance **P4-C47-D1-S1** has four weeks, 47 client-products, one depot, and belongs to season 1 (2016). We classify instances into two classes based on $|\mathcal{N}|$.

Table 6.1 Instances Properties

$ \mathcal{N} $	$\#C$	$\mathcal{P}$	$ \mathcal{W} / \mathcal{D} $	$ \mathcal{E} $	Inst.	$ \mathcal{N} $	$\#C$	$\mathcal{P}$	$ \mathcal{W} / \mathcal{D} $	$ \mathcal{E} $	Inst.
47	47	$[2, 3]$	4/28	1	P4-C47-D1-S1	77	65	$[0, 1, 2, 3]$	4/28	1	P4-C77-D1-S1
					P4-C47-D1-S2						P4-C77-D1-S2
				4	P4-C47-D4-S1					4	P4-C77-D4-S1
					P4-C47-D4-S2						P4-C77-D4-S2
			8/56	1	P8-C47-D1-S1				8/56	1	P8-C77-D1-S1
					P8-C47-D1-S2						P8-C77-D1-S2
				4	P8-C47-D4-S1					4	P8-C77-D4-S1
					P8-C47-D4-S2						P8-C77-D4-S2
	47	$[2, 3]$	12/84	1	P12-C47-D1-S1		65	$[0, 1, 2, 3]$	12/84	1	P12-C77-D1-S1
					P12-C47-D1-S2						P12-C77-D1-S2
				4	P12-C47-D4-S1					4	P12-C77-D4-S1
					P12-C47-D4-S2						P12-C77-D4-S2

The tank trucks' features, including the capacity (Ca.) in m^3 , the compartment configuration (Config.), the number of compartments (Co.), if the tank truck is suitable for marked and unmarked petrol products (F.), and if the tank truck is jumbo or not (J.) are highlighted in Table 6.2.

Table 6.2 Tank Trucks Properties

Id	Ca.	Config.	Co.	F.	J.	Id	Ca.	Config.	Co.	F.	J.
k_1	13	[2, 1, 4, 6]	4	✓	□	k_{23}	21	[2, 5, 5, 5, 4]	5	✓	✓
k_2	13	[1, 2, 4, 6]	4	✓	□	k_{24}	21	[2, 3, 1, 1, 2, 3, 5, 4]	8	□	✓
k_3	13	[3, 1, 1, 1, 1, 2, 2, 2]	8	✓	□	k_{25}	21	[6, 4, 6, 2, 3, 5, 4, 2, 6]	9	□	✓
k_4	13	[2, 1, 1, 1, 1, 5, 2]	7	✓	□	k_{26}	22	[2, 2, 1, 3, 2, 1, 2, 3, 2, 2, 2]	11	✓	✓
k_5	13	[3, 2, 2, 3, 5, 3]	5	✓	□	k_{27}	30	[7, 7, 2, 7, 7]	5	□	✓
k_6	14	[2, 1, 1, 4, 4, 2]	6	□	□	k_{28}	33	[5, 3, 4, 1.5, 1.5, 1, 1, 2, 3, 2, 4, 3, 2]	13	✓	✓
k_7	14	[2, 2, 3, 1, 2, 2, 2]	7	✓	□	k_{29}	33	[5, 4, 3, 2, 1, 1, 2, 2, 3, 4, 6]	11	□	✓
k_8	14	[4, 2, 3, 2, 3]	5	✓	□	k_{30}	33	[5, 4, 3, 2, 1, 1, 2, 2, 3, 4, 6]	11	□	✓
k_9	14	[2, 1, 1, 1, 1, 1, 1, 1, 2, 1, 2]	11	□	□	k_{31}	33	[4, 5, 4, 3, 2, 3, 2, 5, 5]	9	□	✓
k_{10}	14	[2, 1, 1, 1, 1, 2, 3, 3]	8	✓	□	k_{32}	35	[6, 4, 5, 2, 2, 2, 3, 8, 3]	9	✓	✓
k_{11}	14	[4, 2, 1, 1, 4, 2]	6	✓	□	k_{33}	35	[4, 6, 2, 4, 3, 4, 3, 6, 3]	9	□	✓
k_{12}	18	[3, 2, 2, 3, 2, 6]	6	□	□	k_{34}	35	[6, 4, 3, 1, 1, 2, 5, 6, 7]	9	□	✓
k_{13}	18	[5, 2, 2, 2, 3, 4]	6	✓	□	k_{35}	36	[4, 2, 6, 2, 4, 2, 4, 3, 6, 3]	10	□	✓
k_{14}	18	[4, 2, 5, 4, 3]	5	□	□	k_{36}	37	[5, 2, 5, 2, 2, 1, 1, 2, 2, 2, 2, 5, 2, 2, 2]	15	✓	✓
k_{15}	18	[2, 2, 2, 2, 1, 1, 1, 1, 3, 3]	10	□	□	k_{37}	37	[6, 3, 6, 4, 6, 6, 6]	7	✓	✓
k_{16}	18	[2, 1.5, 1, 1, 1.5, 4, 4, 3]	8	□	□	k_{38}	37	[10, 3, 4, 2, 5, 6, 7]	7	✓	✓
k_{17}	18	[2, 1.5, 1, 1, 1.5, 4, 4, 3]	8	□	□	k_{39}	38	[6, 4, 1, 1, 4, 2, 4, 6, 4, 3, 5]	11	✓	✓
k_{18}	19	[5, 1, 1, 5, 5, 2]	6	□	□	k_{40}	38	[8, 4, 3, 2, 2, 1, 1, 2, 5, 4, 6]	11	✓	✓
k_{19}	20	[6, 5, 5, 2, 2]	5	□	✓	k_{41}	38	[6, 4, 6, 2, 3, 5, 4, 2, 6]	9	✓	✓
k_{20}	20	[3, 3, 3, 3, 2, 2, 2, 2]	8	□	✓	k_{42}	40	[7, 5, 4, 1, 1, 4, 2, 5, 2, 3, 6]	11	□	✓
k_{21}	20	[2, 4, 3, 4, 5, 1, 1]	7	□	✓	k_{43}	40	[7, 5, 4, 1, 1, 4, 2, 5, 2, 3, 6]	11	□	✓
k_{22}	20	[2, 4, 3, 4, 5, 1, 1]	7	□	✓	k_{44}	40	[6, 4, 1, 1, 4, 2, 4, 6, 4, 3, 5]	11	□	✓

The coding language is C++, and tests are conducted using version 22.1.1 of the IBM ILOG CPLEX solver. All experiments were carried out on a 3.20GHz Intel(R) Core(TM) i7-700 processor, with 64GiB System memory, running on Oracle Linux Server release 7.7. We use real-time to measure runtime.

6.5.2 Computational Results

In this section, we compare between direct billing and standard billing. Then, we check the impact of price fluctuations. After that, we evaluate the effect of capacities before discussing minimal configurations.

Comparison between Direct Billing and Standard Billing

We compare two popular billing scenarios: direct billing and standard billing. We recall that the direct billing scenario is billing the petrol station based on the number of kilometers (km) from the depot, while the standard billing scenario is billing the petrol station based on the number of kilometers (km) from the previous petrol stop (e.g., another petrol station served just before).

Table 6.3 compares the number of visits (#Vis), the number of routes (#Ro), the number of tank trucks (#Tr), and the tank trucks free space percentage (F) for the two scenarios. Furthermore, we report the transport cost (local currency) for the direct billing and the

Table 6.3 Direct Billing vs Standard Billing

$ \mathcal{N} $	$ \mathcal{W} $	Inst.	Direct Billing					Standard Billing				
			#Vis	#Ro	#Tr	F(%)	Transport Cost	#Vis	#Ro	#Tr	F(%)	R(%)
47	4	P4-C47-D1-S1	95	55	44	8.48	21339956	94	55	42	8.06	1.36
47	4	P4-C47-D4-S1	88	54	41	7.49	18009887	87	51	40	6.84	2.52
47	4	P4-C47-D1-S2	95	55	44	8.48	21339956	94	55	42	8.06	1.36
47	4	P4-C47-D4-S2	88	54	41	7.23	18009887	87	51	40	6.84	4.38
47	8	P8-C47-D1-S1	197	140	96	9.27	43422166	195	139	93	8.21	7.76
47	8	P8-C47-D4-S1	197	135	92	7.35	43091682	193	132	91	7.02	3.42
47	8	P8-C47-D1-S2	207	139	96	10.01	43422166	205	138	93	8.95	7.76
47	8	P8-C47-D4-S2	194	135	91	7.89	42366889	195	132	87	7.07	4.25
47	12	P12-C47-D1-S1	297	212	143	7.72	64696920	293	206	142	6.53	1.48
47	12	P12-C47-D4-S1	302	208	148	6.09	65239896	299	204	147	5.78	1.45
47	12	P12-C47-D1-S2	299	211	149	6.43	62336922	295	205	148	6.28	5.85
47	12	P12-C47-D4-S2	306	203	138	6.31	65239896	303	201	143	6.02	2.14
Avg			197	133	94	7.73	42376352	195	131	92	7.14	3.65
77	4	P4-C77-D1-S1	158	102	57	10.48	38356235	161	103	64	9.29	3.18
77	4	P4-C77-D4-S1	156	105	56	14.30	35173229	146	98	57	11.64	1.89
77	4	P4-C77-D1-S2	158	102	57	10.48	37796242	164	105	65	9.48	1.67
77	4	P4-C77-D4-S2	157	106	63	14.14	34523208	144	97	55	11.56	0.36
77	8	P8-C77-D1-S1	325	244	135	9.51	86125302	327	241	125	8.97	0.74
77	8	P8-C77-D4-S1	335	230	133	9.43	76229624	321	228	126	8.04	0.92
77	8	P8-C77-D1-S2	324	244	131	9.80	85905135	327	242	131	9.30	3.68
77	8	P8-C77-D4-S2	332	227	135	9.36	78249658	323	231	130	8.53	2.79
77	12	P12-C77-D1-S1	490	367	203	7.81	121436837	495	372	202	7.90	0.50
77	12	P12-C77-D4-S1	511	354	206	8.12	112664792	495	350	198	6.91	1.12
77	12	P12-C77-D1-S2	490	368	201	7.84	120206621	495	372	202	7.90	0.61
77	12	P12-C77-D4-S2	517	356	200	7.70	114175404	494	350	195	6.75	0.91
Avg			329	234	131	9.91	78403524	324	232	129	8.86	1.53

percentage of savings in transportation costs (R) under the standard billing compared to the direct billing. Direct billing requires more visits than standard billing, uses more routes and tank trucks on average, and maintains a higher free space percentage. This is because each tank truck serves a single petrol station more frequently under this scenario. Compared to direct billing, standard billing ensures around 3.7% and 1.5% reduction in transportation costs for the first class ($|\mathcal{N}| = 47$) and second class ($|\mathcal{N}| = 77$), respectively. Under the standard billing scenario, the PC charges and serves each petrol station often separately.

Impact of Price Fluctuations

We evaluate the impact of price fluctuations. Figures 6.3 and 6.4 highlight how the quantities bought and stocked change when the prices/costs are constant or variable, respectively.

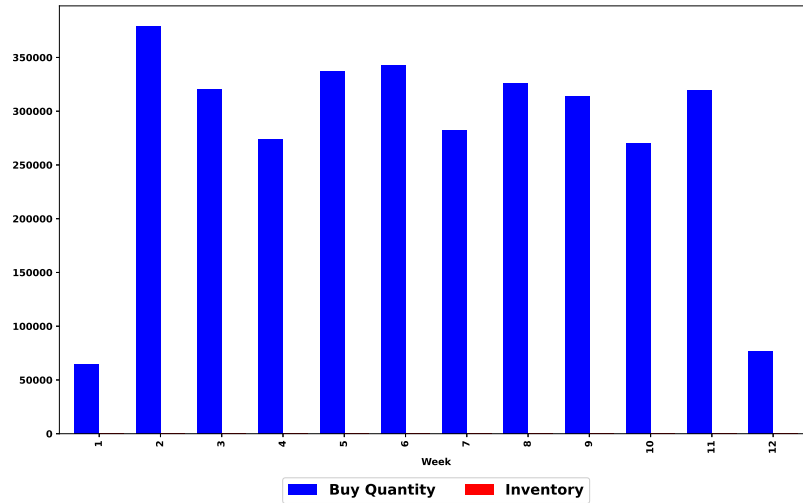


Figure 6.3 Constant Prices/Costs

When the prices/costs are constant, the PC purchases and delivers without keeping inventory at the depots. It is due to eliminating inventory costs. These costs add to the buying costs. We highlight this case in Figure 6.3. When the prices/costs are variable, buying and stocking occur. The PC will buy when the prices are low and stock when the prices are high. We highlight the second case in Figure 6.4. The PC buys two times, with the best prices, in two weeks during the horizon demand and stocks throughout the four-month horizon.

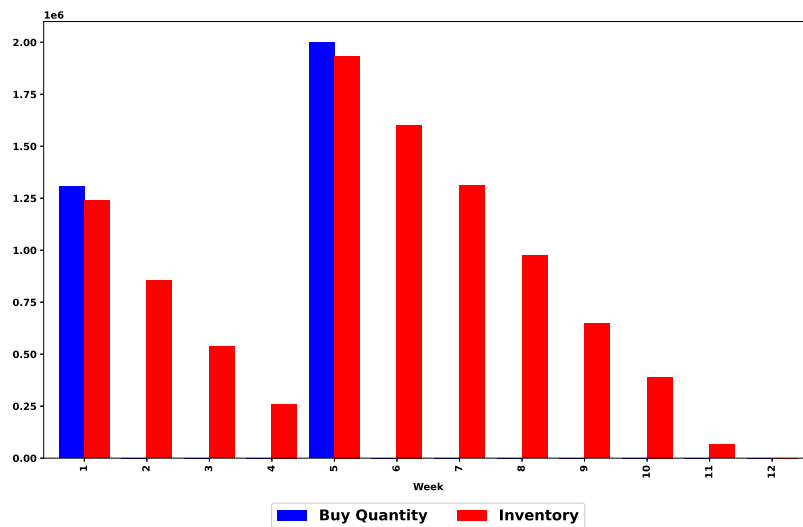


Figure 6.4 Variable Prices/Costs

Impact of Capacities

We check the impact of adjusting petrol stations tank capacities. Table 6.4 shows the impact of increasing by 15% the tank capacity for frequently visited petrol stations under the standard billing scenario. We refer to this policy as the adjusted standard billing and compare it to the direct billing scenario. We report the number of visits, the number of routes, the number of tank trucks, the tank trucks free space percentage, and the saving percentage in transportation costs compared to direct billing.

Table 6.4 Increase Petrol Stations' Tanks for Frequently Visited Petrol Stations

$ \mathcal{N} $	$ \mathcal{W} $	Inst.	Direct Billing					Adjusted Standard Billing				
			#Vis	#Ro	#Tr	F(%)	Total Cost	#Vis	#Ro	#Tr	F(%)	R(%)
47	4	P4-C47-D1-S1	95	55	44	8.48	528596571	90	56	44	8.78	10.08
47	4	P4-C47-D4-S1	88	54	41	7.49	492292241	86	54	41	9.27	4.32
47	4	P4-C47-D1-S2	95	55	44	8.48	536612958	90	56	44	8.78	7.77
47	4	P4-C47-D4-S2	88	54	41	7.23	515343240	86	54	41	9.27	6.21
47	8	P8-C47-D1-S1	197	140	96	9.27	1339204257	189	132	93	8.23	7.64
47	8	P8-C47-D4-S1	197	135	92	7.35	1260730999	189	127	89	8.37	9.79
47	8	P8-C47-D1-S2	207	139	96	10.01	1297574207	188	133	90	8.12	7.64
47	8	P8-C47-D4-S2	194	135	91	7.89	1280828575	188	128	91	8.17	8.81
47	12	P12-C47-D1-S1	297	212	143	7.72	2088190180	286	208	150	6.18	9.16
47	12	P12-C47-D4-S1	302	208	148	6.09	2037352514	294	201	135	6.37	11.19
47	12	P12-C47-D1-S2	299	211	149	6.43	2056481848	285	208	145	5.88	8.82
47	12	P12-C47-D4-S2	306	203	138	6.31	2038004458	293	198	140	6.35	13.66
Avg			197	133	94	7.73	1289267671	189	130	92	7.81	8.76
77	4	P4-C77-D1-S1	158	102	57	10.48	881749096	155	106	61	10.16	6.65
77	4	P4-C77-D4-S1	156	105	56	14.30	855580632	140	96	54	10.98	11.97
77	4	P4-C77-D1-S2	158	102	57	10.48	924302869	155	106	62	10.00	5.09
77	4	P4-C77-D4-S2	157	106	63	14.14	832848269	140	96	54	10.98	10.02
77	8	P8-C77-D1-S1	325	244	135	9.51	2162855951	320	241	133	9.00	10.85
77	8	P8-C77-D4-S1	335	230	133	9.43	2068869342	315	238	127	10.15	2.92
77	8	P8-C77-D1-S2	324	244	131	9.80	2147779341	320	243	135	9.15	10.78
77	8	P8-C77-D4-S2	332	227	135	9.36	2066854673	315	237	125	9.85	0.44
77	12	P12-C77-D1-S1	490	367	203	7.81	3417761012	482	371	212	7.55	6.92
77	12	P12-C77-D4-S1	511	354	206	8.12	3323517723	481	360	205	7.64	2.96
77	12	P12-C77-D1-S2	490	368	201	7.84	3435103033	482	368	209	7.43	7.99
77	12	P12-C77-D4-S2	517	356	200	7.70	3315566988	481	358	203	7.54	4.76
Avg			329	234	131	9.91	2119399077	316	235	132	9.20	6.78

Under the adjusted standard billing, the number of visits decreases, the number of routes and tank trucks decreases, and the free space percentage increases for the first class. For the second class, the number of visits decreases, the number of routes and tank trucks increases slightly, and the free space percentage decreases. The main difference is the number of visits, which decreases significantly for both classes. This is due to the increased tank capacity. When the tank capacity at the petrol stations increases, the transport company delivers higher quantities since the tank capacity is higher, thus reducing the number of visits to the

petrol stations. Furthermore, compared to direct billing, the adjusted standard billing ensures a reduction of around 8.7% and 6.7% in total cost for the 47 and 77 classes, respectively.

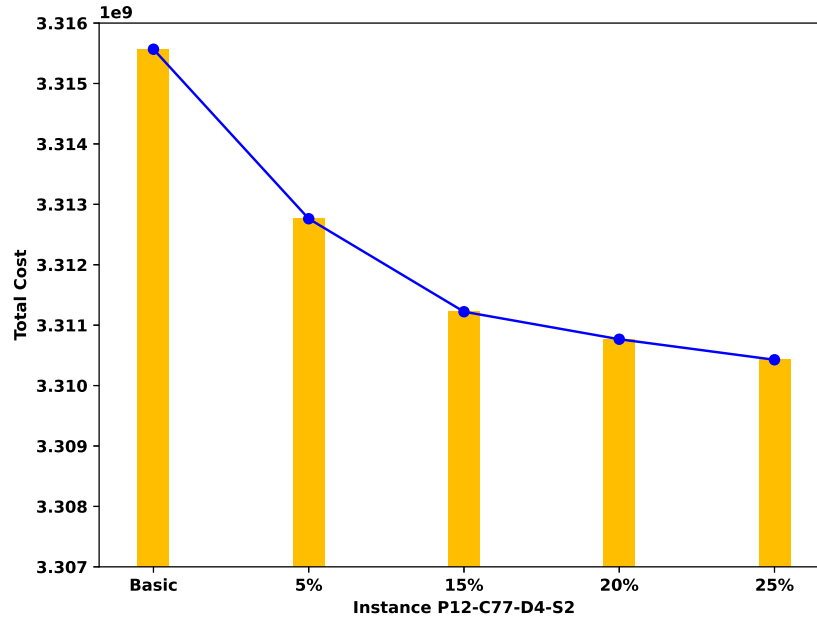


Figure 6.5 Total Cost when Changing Petrol Stations' Tank Capacities

We push the analysis further and conduct additional tests where we vary the petrol station tank capacity increase percentage. Figure 6.5 shows the total cost variation when the petrol stations' capacities are increased by 5%, 15%, 20%, and 25%. We observe that the total cost decreases strictly and significantly from 0% (basic) to 5% and from 5% to 15%. After that, while the decrease remains strict, it is no longer significant. It might suggest that increasing tank capacities by 15% is optimal.

Minimal Tank Truck Configurations

We also check the tank trucks used for delivery among all available ones. Figure 6.6 shows the tank trucks used in the optimal solution (lowest cost).

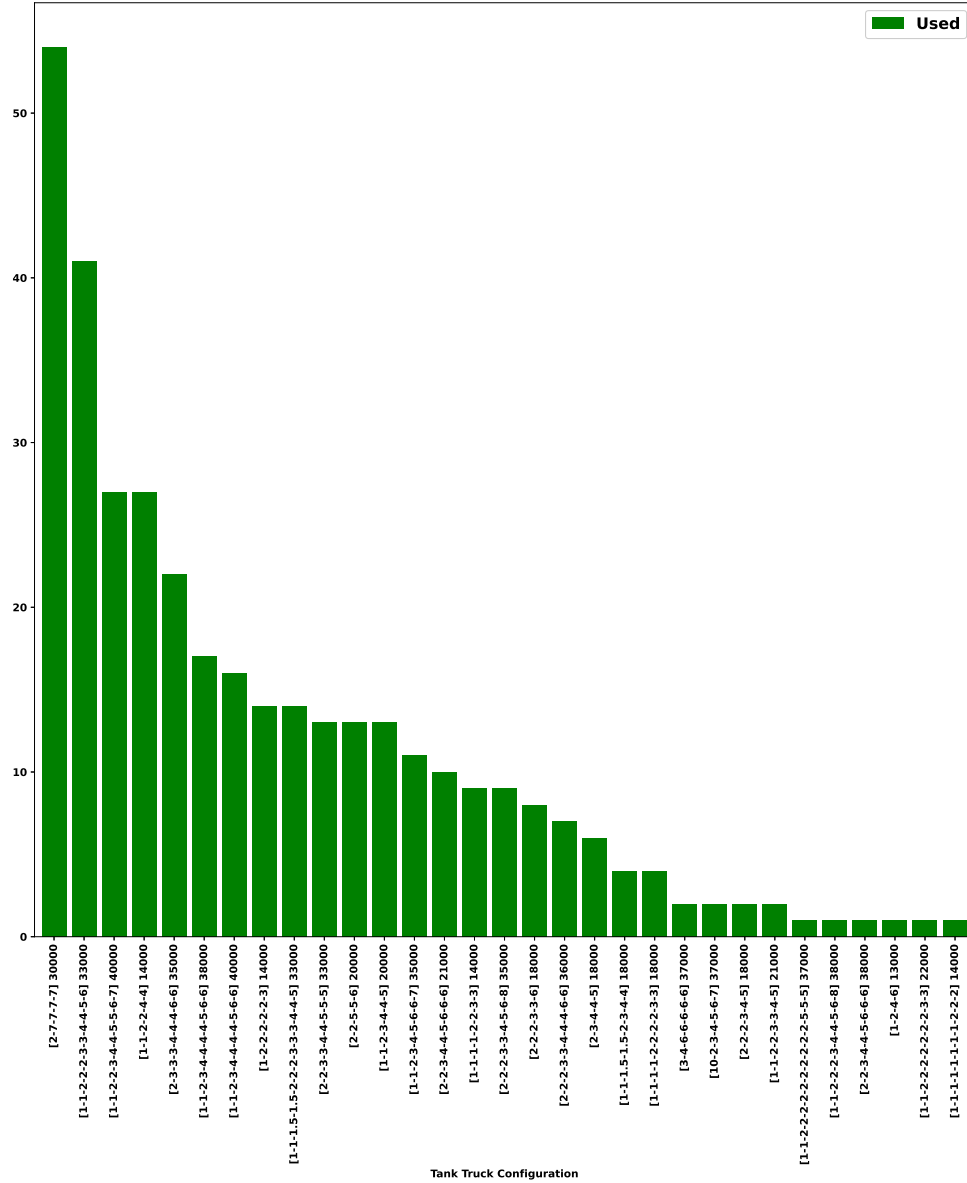


Figure 6.6 Minimal Number of Tank Truck Configurations for the Optimal Cost

Among all the available tank truck configurations, we observe that 14 tank truck configurations are used more than ten times, and five are used between 5 and 10 times. All remaining tank trucks (configurations) are used less than three times. When the number of tank trucks (configurations) is large, the optimization problem is very complex. If it is possible to reduce the tank truck fleet, the optimization becomes quick, and the management in the real world as well.

Given that not all tank truck configurations are used, we measure the impact of reducing the fleet size on the total cost. Table 6.5 shows the impact of reducing the fleet size and the

impact of the total cost (Δ Total Cost).

Table 6.5 Impact of Reducing Fleet Size on Total Cost

$ \mathcal{N} $	$ \mathcal{W} $	Inst.	Standard Billing				Fleet Size	Δ Total Cost(%)
			#Vis	#Ro	#Tr	F(%)		
77	12	P12-C77-D4-S2	494	350	195	6.75	44	Basic
77	12	P12-C77-D4-S2	501	362	189	8.82	37	+0.05
77	12	P12-C77-D4-S2	490	362	172	9.56	26	+0.09
77	12	P12-C77-D4-S2	486	367	178	9.48	23	+0.10
77	12	P12-C77-D4-S2	487	365	176	9.88	20	+0.13

We observe that when we decrease the fleet size, the total cost slightly increases. To enhance management capabilities with the PC, the latter might accept a slight cost increase to have a smaller fleet to manage. In our case, the PC might reduce its fleet size by up to 24 while accepting an increase of 0.13% in total cost.

6.5.3 Discussion

The results above highlight that optimization can help reduce costs significantly without any investment in the infrastructure. With the quick optimization capability, optimization becomes an efficient decision-making tool to check, control, simulate, and re-optimize various what-if scenarios, especially when the manual optimization is intractable, as in our case. This makes the supply chain more resilient to unexpected events and risks.

The optimization tool is interesting and practical. Furthermore, it is useful in various ways. Given price forecasts, petrol distribution companies can use the optimization tool to inform various strategic, tactical, and operational decisions. For strategic decisions, the tool can orient future design and investments by petroleum distribution companies. Examples include the petrol stations tank capacities and the preferred tank truck fleet. The optimal routes obtained from the optimization can inform potential depot locations and potential tank truck sizes with the number of compartments in each. For tactical and operational decisions, the tool can be used to inform buying and stocking decisions based on price fluctuations in the market. States can also use the optimization tool to inform sponsorship incentives, evaluate petroleum distribution policies, and check various strategies. For instance, direct billing increases costs significantly and leads distribution companies to achieve higher profits. States can then enforce a standard billing policy that will balance profits achieved by the petroleum distribution companies with the costs incurred by petrol stations.

This work with other OR success stories in Africa (e.g., Becker et al., 2022; Gibson et al., 2023; Er Raqabi et al., 2023a) highlights the benefits of using OR in Africa, especially that in several cases, organizations can achieve important gains without investing a single dollar.

6.6 Conclusion

This paper presents a complex variant of the petrol station replenishment problem. Using a two-step approach, we can tackle it to near-optimality in a few minutes. This paper adds to the practice of operations research in Africa, specifically in the context of transport. The quick optimization allows for testing several what-if scenarios, re-optimizing quickly, boosting decision-making capabilities, and comparing various policies. In particular, we compare two popular policies: direct and standard billing. We observe that the direct billing is more costly and less efficient than the standard billing. We also highlight the impact of various cost and capacity fluctuations on the optimal solution. The insights presented in this paper could incentivize African states to sponsor some policies over others. These insights can also be used to criticize and evaluate existing approaches in practice and invest more in infrastructure and analytics tools to boost operations management.

CHAPITRE 7 DISCUSSION GÉNÉRALE

Cette thèse explore une variante complexe d'un problème bien étudié dans la littérature sur la recherche opérationnelle : le problème complexe étudié ici est le PRP multi-dépôts et multi-périodes avec gestion des stocks (MDMPPRPIM). Ce problème intègre le problème de chargement des camions-citernes (TTLP) et le problème des tournées des véhicules avec gestion d'inventaires IRP, le tout dans le contexte de camions à plusieurs compartiments. Le TTLP vise à déterminer la quantité de chaque type de produit à charger dans les compartiments du camion, sans dépasser leurs capacités, tout en satisfaisant simultanément la demande de chaque produit et de chaque station-service. Le problème IRP cherche à optimiser les tournées de livraison, tout en garantissant que chaque station-service dispose d'une quantité suffisante pour sa demande quotidienne.

La contribution de cette thèse réside dans le développement d'une approche hautement évolutive pour résoudre des problèmes d'optimisation de grande envergure liés au PRP. La méthodologie adoptée repose sur le principe du *découplage* des décisions d'inventaire et ceux des tournées des véhicules. Cette approche est présentée comme une avancée significative pour trois raisons majeures : premièrement, nous démontrons l'efficacité du *découplage* des modèles classiques de recherche opérationnelle grâce aux technologies actuelles ; deuxièmement, nous proposons d'utiliser des solutions heuristiques comme entrées pour le modèle mathématique, permettant ainsi d'atteindre rapidement des solutions optimales ; troisièmement, les intuitions présentées sont facilement implantables, évolutives et suffisamment génériques pour être appliqués à d'autres problèmes impliquant la synchronisation d'inventaires et des tournées des véhicules pour plusieurs ressources. Grâce à l'approche proposée, nous avons pu résoudre des modèles en moins d'une minute, ce qui serait impossible avec des solveurs commerciaux, même pour de petites instances.

La méthodologie adoptée comprend l'utilisation de la BD et de la CG dans une approche en deux phases (*Phase 1* et *Phase 2*). La BD est utilisée pour mettre en œuvre le *découplage* en gérant l'inventaire dans le problème maître de BD et le RP dans le ou les sous-problèmes de la BD. La CG est utilisée pour résoudre ces sous-problèmes. Diverses stratégies d'accélération, telles que le *warm-start*, le parallélisme, les techniques de hachage et les heuristiques primales de branchement en profondeur, sont également proposées. Les résultats présentés sont issus d'expérimentations numériques réalisées avec des données réelles provenant d'une zone géographique située en Afrique de l'Ouest. Ces résultats démontrent l'efficacité de notre approche et sa capacité à obtenir des solutions quasi optimales pour des instances du monde

réel.

Dans cette thèse, nous avons présenté une méthode en deux étapes qui consiste d'abord à résoudre le MDMPPRP hebdomadaire en utilisant la CG. Ensuite, pour résoudre le MDMPPRPIM, nous utilisons la BD pour décomposer le MDMPPRPIM en plusieurs MDMPPRP hebdomadaires que nous résolvons efficacement en utilisant la *Phase 1*.

La première Étape. La première étape aborde le MDMPPRP. Dans cette étape, nous résolvons la variante complexe du PRP avec une approche exacte de BP et quelques heuristiques dérivées. Une caractéristique importante est que ce problème peut être modélisé comme un problème de partitionnement d'ensemble avec une densité faible à modérée (le nombre de "un" par colonne, c'est-à-dire les stations-service à visiter, n'est pas élevé). Les problèmes de type partitionnement d'ensemble présentent des propriétés polyédriques intéressantes à prendre en compte pour favoriser l'intégralité. De plus, certaines règles de traitement complexes spécifiques au cas s'appliquent. Une manière naturelle est de les aborder dans le sous-problème CG sous la forme d'un problème de plus court chemin élémentaire avec des contraintes de ressources, ce qui constitue le principal goulot d'étranglement. Pour réussir à le résoudre, nous avons conçu des techniques sophistiquées : i) pour le branchement afin de tirer profit des propriétés polyédriques et ii) pour résoudre le sous-problème CG. L'utilisation directe d'algorithmes disponibles ne fonctionne malheureusement pas.

La deuxième Étape. Guidés par l'intuition du *découplage*, nous créons une approche de solution exacte en deux phases qui combine la décomposition de BD et la CG pour le MDMPPRPIM. Cette intuition vient du fait que les contraintes de gestion des stocks couplent les semaines de l'horizon de planification, car elles lient toutes les décisions d'achat, de stockage et de transport dans une semaine donnée avec les semaines suivantes. De plus, conserver ces contraintes dans le modèle rend la résolution très difficile parce que le modèle est grand. La BD nous permet de concrétiser l'intuition du *découplage*. Une fois que les variables d'inventaire (et les contraintes de liaison correspondantes) sont fixées en utilisant la solution du problème maître de BD, le MDMPPRPIM devient découplé. La BD permet de décomposer le MDMPPRPIM en $|\mathcal{W}|$ MDMPPRPs, qui sont résolus en utilisant les idées de la première étape. Dans la première phase, nous résolvons les sous-problèmes de BD relaxés (sans les contraintes d'intégralité) à l'aide de la CG, jusqu'à ce que les niveaux d'inventaire se stabilisent. Dans la deuxième phase, nous résolvons les sous-problèmes de la BD à l'aide de la CG intégrée dans un cadre de BB. Nous améliorons l'approche de solution à deux phases par des stratégies d'accélération, y compris le *warm-start*, le parallélisme, une technique de hachage et une heuristique primale de branchement en profondeur.

CHAPITRE 8 CONCLUSION

Le problème de réapprovisionnement des stations-service (PRP) forme l'un des problèmes de recherche opérationnelle les plus étudiés du MCVRP. Pour de nombreuses applications pratiques et des cas à grande échelle, la taille du MILP sous-jacent est extrêmement grande, rendant ainsi l'utilisation directe d'un solveur MILP inefficace. Guidée par l'intuition du *découplage* des décisions d'inventaire et ceux des tournées des véhicules, notre motivation consistait à fournir un cadre générique exact pour résoudre efficacement le problème ainsi que ses variantes complexes. Dans ce qui suit, nous passerons en revue la synthèse de nos recherches (Section 8.1), leurs limitations, en plus de leurs futures pistes d'amélioration, qui seront mises en perspective (Section 8.2).

8.1 Synthèse des travaux

Dans ce document, nous avons présenté l'exemple concret d'un problème de réapprovisionnement en produits pétroliers, avec plusieurs dépôts et de multiples périodes d'une entreprise pétrolière basée au Sénégal. La complexité de ce problème réside dans plusieurs aspects, notamment dans les contraintes spécifiques liées au chargement des camions-citernes fortement compartimentés (de 3 à 15 compartiments), les aspects multi-produits, multi-dépôts et les multiples trajets, les restrictions régionales entre certaines stations-service desservis par le même camion-citerne, les exclusions entre des types spécifiques de produits pétroliers chargés dans le même camion-citerne, la diversité des commandes des stations-service et la possibilité pour les stations-service de commander de petites quantités, augmentant ainsi le nombre de stations-service sur un seul itinéraire.

En premier temps, nous avons développé une approche de BP pour obtenir des solutions (presque) optimales à ce problème MDMPPRP hebdomadaire. Dans cette méthode, nous avons résolu le problème intégré du chargement des camions-citernes et du problème de routage. Une technique de hachage et d'autres méthodes ont été mises au point pour accélérer la génération de routes potentielles. Ainsi, nous avons identifié et corrigé un goulot d'étranglement majeur qui entravait considérablement le processus de résolution. Cette approche marque une avancée importante dans ce domaine, étant donné que les métaheuristiques ou la CG a priori sont généralement utilisées pour résoudre de tels problèmes.

En outre, nous avons exploré une approche exacte s'appuyant sur la combinaison de la BD et de la CG pour traiter une variante complexe du MDMPPRPIM s'étendant sur un horizon

de quatre mois (12 semaines). L'approche en deux phases proposée stabilise les niveaux d'inventaire dans la première phase, avant de trouver une solution entière quasi optimale dans la deuxième phase. Cette approche a été améliorée grâce à plusieurs stratégies d'accélération, notamment le démarrage rapide, le parallélisme, une technique de hachage, et une méthode primale de branchement en profondeur (PDH).

Les résultats sur des instances réalistes d'une grande entreprise de distribution de produits pétroliers en Afrique de l'Ouest démontrent que la nouvelle approche atteint des solutions quasi optimales et surpasse significativement les techniques existantes dans la littérature. Ces résultats témoignent de l'efficacité et du fort potentiel de l'approche proposée.

8.2 Limitations de la solution proposée et Améliorations futures

L'utilisation de la programmation dynamique pour résoudre le sous-problème de CG n'a pas été testée sur un nombre élevé de stations-service (1000 stations-service par exemple), le fait que les routes doivent être élémentaires peut aussi augmenter rapidement les étiquettes générées et ralentir la convergence de l'algorithme, malgré les accélérations développées. La dominance exacte des étiquettes devient plus coûteuse en temps, limitant ainsi l'algorithme. Un autre aspect qui limite l'approche utilisée pour résoudre le MDMPRPIM est de considérer des décisions déterministes, au niveau de la demande des stations-service, les prix des produits pétroliers et leur disponibilité dans les dépôts.

Ce travail peut être étendu de plusieurs manières. D'abord, bien que nous supposons que le modèle est déterministe, il existe en pratique de nombreux paramètres incertains, tels que les demandes des stations-service, le prix des produits pétroliers et leur disponibilité. Par conséquent, une direction d'extension prometteuse consiste à concevoir des méthodes de résolution pour le MDMPRPIM avec des demandes stochastiques ainsi que le cas où l'offre totale est inférieure à la demande totale (c'est-à-dire des pénuries causées par la durée prolongée de maintenance de la raffinerie combinée à des perturbations des importations de produits pétroliers). Ensuite, la prévision des stocks peut être réalisée à l'aide de modèles d'apprentissage profond plus sophistiqués et incorporée au modèle d'optimisation. Les développements futurs pourraient aborder aussi une généralisation de la technique de hachage au MCVRP général. Nous sommes convaincus que cette méthode sera utile pour accélérer la résolution des sous-problèmes de génération de routes, en utilisant la programmation dynamique. Nous avons l'intention de tester notre approche sur des réseaux réels plus importants. Ces améliorations devraient aboutir à une méthode pouvant être appliquée par un large éventail d'entreprises de distribution de produits pétroliers.

RÉFÉRENCES

- Abdelaziz, F. B., Roucairol, C., & Bacha, C. (2002). Deliveries of liquid fuels to sndp gas stations using vehicles with multiple compartments. In *IEEE international conference on systems, man and cybernetics*, volume 1, pages 478–483. IEEE.
- Adulyasak, Y., Cordeau, J.-F., & Jans, R. (2014). Formulations and branch-and-cut algorithms for multivehicle production and inventory routing problems. *INFORMS Journal on Computing*, 26(1) :103–120.
- Adulyasak, Y., Cordeau, J.-F., & Jans, R. (2015). Benders decomposition for production routing under demand uncertainty. *Operations Research*, 63(4) :851–867.
- Al-Hinai, N. & Triki, C. (2020). A two-level evolutionary algorithm for solving the petrol station replenishment problem with periodicity constraints and service choice. *Annals of Operations Research*, 286(1-2) :325–350.
- Anbil, R., Forrest, J. J., & Pulleyblank, W. R. (1998). Column generation and the airline crew pairing problem. *Documenta Mathematica*, 3(1) :677.
- Androutsopoulos, K. N. & Karouti, E. (2022). A safety-driven truck loading problem. *Operational Research*, 22(5) :4931–4963.
- Archetti, C., Bertazzi, L., Hertz, A., & Speranza, M. G. (2012). A hybrid heuristic for an inventory routing problem. *INFORMS Journal on Computing*, 24(1) :101–116.
- Archetti, C., Bertazzi, L., Laporte, G., & Speranza, M. G. (2007). A branch-and-cut algorithm for a vendor-managed inventory-routing problem. *Transportation science*, 41(3) :382–391.
- Archetti, C., Boland, N., & Grazia Speranza, M. (2017). A matheuristic for the multivehicle inventory routing problem. *INFORMS Journal on Computing*, 29(3) :377–387.
- Archetti, C., Jabali, O., & Speranza, M. G. (2015a). Multi-period vehicle routing problem with due dates. *Computers & Operations Research*, 61 :122–134.
- Archetti, C., Jabali, O., & Speranza, M. G. (2015b). Multi-period vehicle routing problem with due dates. *Computers & Operations Research*, 61 :122–134.
- Archetti, C. & Ljubić, I. (2022). Comparison of formulations for the inventory routing problem. *European Journal of Operational Research*, 303(3) :997–1008.
- Avella, P., Boccia, M., & Sforza, A. (2004). Solving a fuel delivery problem by heuristic and exact approaches. *European Journal of Operational Research*, 152(1) :170–179.
- Avella, P., Boccia, M., & Wolsey, L. A. (2018). Single-period cutting planes for inventory routing problems. *Transportation Science*, 52(3) :497–508.

- Azi, N., Gendreau, M., & Potvin, J.-Y. (2010). An exact algorithm for a vehicle routing problem with time windows and multiple use of vehicles. *European Journal of Operational Research*, 202(3) :756–763.
- Bani, A., El Hallaoui, I., Corr  a, A. I., & Tahir, A. (2023). Solving a real-world multi-depot multi-period petrol replenishment problem with complex loading constraints. *European Journal of Operational Research*, 311(1) :154–172.
- Bani, A., Er Raqabi, E. M., El Hallaoui, I., & Corr  a, A. I. (2024). Combining Benders decomposition and column generation for the petrol station replenishment problem with inventory management. *Les Cahiers du GERAD*, G-2024-18 :1–31.
- Barnhart, C., Johnson, E. L., Nemhauser, G. L., Savelsbergh, M. W., & Vance, P. H. (1998). Branch-and-price : Column generation for solving huge integer programs. *Operations research*, 46(3) :316–329.
- Becker, A., Benabderrazik, K., Bertsimas, D., Chtinna, N., El Majdoub, N., Mahboubi, E. M., Kitane, D. L., Kokkotos, S., Mourtzinou, G., & Rakhis, I. (2022). Toward global food security : Transforming ocp through analytics. *INFORMS Journal on Applied Analytics*, 52(1) :90–107.
- Benantar, A., Ouafi, R., & Boukachour, J. (2016). A petrol station replenishment problem : new variant and formulation. *Logistics Research*, 9(1) :6.
- Benders, J. (1962). Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(1) :238 – 252. Cited by : 2625.
- Bertsimas, D., Kallus, N., & Hussain, A. (2016). Inventory management in the era of big data. *Production and Operations Management*, 25(12) :2006–2009.
- Bloom, J. A. (1983). Solving an electricity generating capacity expansion planning problem by generalized benders’ decomposition. *Operations Research*, 31(1) :84–100.
- Boers, L. (2020). Optimisation methods for the multi-period petrol station replenishment problem : A case study at amcs. *Delft University of Technology*.
- Boers, L., Atasoy, B., Correia, G., & Negenborn, R. R. (2020). The multi-period petrol station replenishment problem : Formulation and solution methods. In *Computational Logistics : 11th International Conference, ICCL 2020, Enschede, The Netherlands, September 28–30, 2020, Proceedings 11*, pages 600–615. Springer.
- Boost (2019). Boost C++ Libraries. <http://www.boost.org/>. Last accessed 2019-03-15.
- Bortfeldt, A. & W  scher, G. (2013). Constraints in container loading–a state-of-the-art review. *European Journal of Operational Research*, 229(1) :1–20.

- Boschetti, M. A., Maniezzo, V., Roffilli, M., & Bolufé Röhler, A. (2009). Matheuristics : Optimization, simulation and control. In *Hybrid Metaheuristics : 6th International Workshop, HM 2009, Udine, Italy, October 16-17, 2009. Proceedings 6*, pages 171–177. Springer.
- Brown, G. G., Ellis, C. J., Graves, G. W., & Ronen, D. (1987). Real-time, wide area dispatch of mobil tank trucks. *Interfaces*, 17(1) :107–120.
- Brown, G. G. & Graves, G. W. (1981). Real-time dispatch of petroleum tank trucks. *Management science*, 27(1) :19–32.
- Cai, X., McKinney, D. C., Lasdon, L. S., & Watkins Jr, D. W. (2001). Solving large nonconvex water resources management models using generalized benders decomposition. *Operations Research*, 49(2) :235–245.
- Canto, S. P. (2008). Application of benders’ decomposition to power plant preventive maintenance scheduling. *European Journal of Operational Research*, 184(2) :759–777.
- Carotenuto, P., Giordani, S., Massari, S., & Vagaggini, F. (2016). A multi-depot periodic vehicle routing model for petrol station replenishment. In *Advanced Concepts, Methodologies and Technologies for Transportation and Logistics*, pages 421–437. Springer.
- Carotenuto, P., Giordani, S., Massari, S., & Vagaggini, F. (2018). A multi-depot periodic vehicle routing model for petrol station replenishment. In *Advanced Concepts, Methodologies and Technologies for Transportation and Logistics*, pages 421–437. Springer.
- Charusakwong, N. & Lohatepanont, M. (2016). Optimization approach for multi-period fuel replenishment. *Engineering Journal*, 20(5) :239–261.
- Chitsaz, M., Cordeau, J.-F., & Jans, R. (2019). A unified decomposition matheuristic for assembly, production, and inventory routing. *INFORMS Journal on Computing*, 31(1) :134–152.
- Coelho, L. C., Cordeau, J.-F., & Laporte, G. (2012). Consistency in multi-vehicle inventory-routing. *Transportation Research Part C : Emerging Technologies*, 24 :270–287.
- Coelho, L. C., Cordeau, J.-F., & Laporte, G. (2014). Thirty years of inventory routing. *Transportation Science*, 48(1) :1–19.
- Coelho, L. C. & Laporte, G. (2013). The exact solution of several classes of inventory-routing problems. *Computers & Operations Research*, 40(2) :558–565.
- Coelho, L. C. & Laporte, G. (2015). Classification, models and exact algorithms for multi-compartment delivery problems. *European Journal of Operational Research*, 242(3) :854–864.
- Coin-Or (2019). Branch-Cut-Price Framework. <https://projects.coin-or.org/Bcp>. Last accessed 2019-03-15.

- Collett, K. A., Hirmer, S. A., Dalkmann, H., Crozier, C., Mulugetta, Y., & McCulloch, M. D. (2021). Can electric vehicles be good for sub-saharan africa ? *Energy Strategy Reviews*, 38 :100722.
- Contreras, I., Cordeau, J.-F., & Laporte, G. (2011). Benders decomposition for large-scale uncapacitated hub location. *Operations research*, 59(6) :1477–1490.
- Cordeau, J.-F., Soumis, F., & Desrosiers, J. (2001a). Simultaneous assignment of locomotives and cars to passenger trains. *Operations research*, 49(4) :531–548.
- Cordeau, J.-F., Stojković, G., Soumis, F., & Desrosiers, J. (2001b). Benders decomposition for simultaneous aircraft routing and crew scheduling. *Transportation science*, 35(4) :375–388.
- Cornillier, F., Boctor, F., & Renaud, J. (2012). Heuristics for the multi-depot petrol station replenishment problem with time windows. *European Journal of Operational Research*, 220(2) :361–369.
- Cornillier, F., Boctor, F. F., Laporte, G., & Renaud, J. (2008a). An exact algorithm for the petrol station replenishment problem. *Journal of the Operational Research Society*, 59(5) :607–615.
- Cornillier, F., Boctor, F. F., Laporte, G., & Renaud, J. (2008b). A heuristic for the multi-period petrol station replenishment problem. *European Journal of Operational Research*, 191(2) :295–305.
- Cornillier, F., Laporte, G., Boctor, F. F., & Renaud, J. (2009). The petrol station replenishment problem with time windows. *Computers & Operations Research*, 36(3) :919–935.
- Cui, Z., Long, D. Z., Qi, J., & Zhang, L. (2023). The inventory routing problem under uncertainty. *Operations Research*, 71(1) :378–395.
- Dantzig, G. B. & Ramser, J. H. (1959). The truck dispatching problem. *Management science*, 6(1) :80–91.
- Dantzig, G. B. & Wolfe, P. (1960). Decomposition principle for linear programs. *Operations research*, 8(1) :101–111.
- Derigs, U., Gottlieb, J., Kalkoff, J., Piesche, M., Rothlauf, F., & Vogel, U. (2011). Vehicle routing with compartments : applications, modelling and heuristics. *OR spectrum*, 33(4) :885–914.
- Desaulniers, G., Desrosiers, J., & Solomon, M. M. (2006). *Column generation*, volume 5. Springer Science & Business Media.
- Desaulniers, G., Rakke, J. G., & Coelho, L. C. (2016). A branch-price-and-cut algorithm for the inventory-routing problem. *Transportation Science*, 50(3) :1060–1076.

- Desrochers, M., Desrosiers, J., & Solomon, M. (1992). A new optimization algorithm for the vehicle routing problem with time windows. *Operations research*, 40(2) :342–354.
- Desrochers, M. & Soumis, F. (1988). A generalized permanent labelling algorithm for the shortest path problem with time windows. *INFOR : Information Systems and Operational Research*, 26(3) :191–212.
- Dror, M. (1994). Note on the complexity of the shortest path models for column generation in vrptw. *Operations Research*, 42(5) :977–978.
- Er Raqabi, E. M., Beljadid, A., Bennouna, M. A., Bennouna, R., Boussaadi, L., El Hachemi, N., El Hallaoui, I., Fender, M., Jamali, A., Si Hammou, N., & Soumis, F. (2023a). Ocp optimizes its supply chain for africa. *Les Cahiers du GERAD*, G-2023-29.
- Er Raqabi, E. M., El Hallaoui, I., & Soumis, F. (2023b). The Primal Benders Decomposition. *Les Cahiers du GERAD*, G-2023-27.
- Er Raqabi, E. M. & Li, W. (2023). An electric vehicle transitioning framework for public fleet planning. *Transportation Research Part D : Transport and Environment*, 118 :103732.
- Evans, D. (2009). A review of underground fuel storage events and putting risk into perspective with other areas of the energy supply chain. *Geological Society, London, Special Publications*, 313(1) :173–216.
- Feillet, D., Dejax, P., Gendreau, M., & Gueguen, C. (2004). An exact algorithm for the elementary shortest path problem with resource constraints : Application to some vehicle routing problems. *Networks : An International Journal*, 44(3) :216–229.
- Fischetti, M., Ljubić, I., & Sinnl, M. (2017). Redesigning benders decomposition for large-scale facility location. *Management Science*, 63(7) :2146–2162.
- Fischetti, M. & Lodi, A. (2003). Local branching. *Mathematical programming*, 98 :23–47.
- Gibson, E., Deo, S., Jónasson, J. O., Kachule, M., & Palamountain, K. (2023). Redesigning sample transportation in malawi through improved data sharing and daily route optimization. *Manufacturing & Service Operations Management*, 25(4) :1209–1226.
- Hanczar, P. (2012). A fuel distribution problem—application of new multi-item inventory routing formulation. *Procedia-Social and Behavioral Sciences*, 54 :726–735.
- Hansen, P. & Mladenović, N. (2001). Variable neighborhood search : Principles and applications. *European Journal of Operational Research*, 130(3) :449–467.
- ILOG, C. I. (2021). V20.1 : User’s Manual for CPLEX. www.ibm.com/docs/fr/icos/20.1.0. Last accessed 2023-03-15.
- Irnich, S. & Desaulniers, G. (2005). Shortest path problems with resource constraints. In *Column generation*, pages 33–65. Springer.

- Joncour, C., Michel, S., Sadykov, R., Sverdlov, D., & Vanderbeck, F. (2010a). Column generation based primal heuristics. *Electronic Notes in Discrete Mathematics*, 36 :695–702.
- Joncour, C., Michel, S., Sadykov, R., Sverdlov, D., & Vanderbeck, F. (2010b). Column generation based primal heuristics. *Electronic Notes in Discrete Mathematics*, 36 :695–702.
- Kaabi, H. & Jabeur, K. (2015). Hybrid algorithm for solving the multi-compartment vehicle routing problem with time windows and profit. In *2015 12th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, volume 1, pages 324–329. IEEE.
- Kazemi, Y. & Szmerekovsky, J. (2015). Modeling downstream petroleum supply chain : The importance of multi-mode transportation to strategic planning. *Transportation Research Part E : Logistics and Transportation Review*, 83 :111–125.
- Laporte, G. (1992). The vehicle routing problem : An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59(3) :345–358.
- Laporte, G., Louveaux, F. V., & van Hamme, L. (1994). Exact solution to a location problem with stochastic demands. *Transportation science*, 28(2) :95–103.
- Le, T.-H., Le, A. T., & Le, H.-C. (2021). The historic oil price fluctuation during the covid-19 pandemic : What are the causes ? *Research in International Business and Finance*, 58 :101489.
- Li, K., Chen, B., Sivakumar, A. I., & Wu, Y. (2014). An inventory–routing problem with the objective of travel time minimization. *European Journal of Operational Research*, 236(3) :936–945.
- Lima, C., Relvas, S., & Barbosa-Póvoa, A. P. F. (2016). Downstream oil supply chain management : A critical review and future directions. *Computers & Chemical Engineering*, 92 :78–92.
- Lin, S., Lim, G. J., & Bard, J. F. (2016). Benders decomposition and an ip-based heuristic for selecting imrt treatment beam angles. *European Journal of Operational Research*, 251(3) :715–726.
- Lübbecke, M. E. & Desrosiers, J. (2005). Selected topics in column generation. *Operations research*, 53(6) :1007–1023.
- Macedo, R., Alves, C., de Carvalho, J. V., Clautiaux, F., & Hanafi, S. (2011). Solving the vehicle routing problem with time windows and multiple routes exactly using a pseudo-polynomial model. *European Journal of Operational Research*, 214(3) :536–545.
- Malépart, V., Boctor, F., Renaud, J., & Labillois, S. (2003). Nouvelles approches pour l’approvisionnement des stations d’essence. *Revue Française de Gestion Industrielle*, 22(2) :15–31.

- Manousakis, E., Repoussis, P., Zachariadis, E., & Tarantilis, C. (2021). Improved branch-and-cut for the inventory routing problem based on a two-commodity flow formulation. *European Journal of Operational Research*, 290(3) :870–885.
- Morrison, D. R., Jacobson, S. H., Sauppe, J. J., & Sewell, E. C. (2016). Branch-and-bound algorithms : A survey of recent advances in searching, branching, and pruning. *Discrete Optimization*, 19 :79–102.
- Ng, W. L., Leung, S., Lam, J., & Pan, S. (2008). Petrol delivery tanker assignment and routing : a case study in hong kong. *Journal of the Operational Research Society*, 59(9) :1191–1200.
- Ostermeier, M., Henke, T., Hübner, A., & Wäscher, G. (2021). Multi-compartment vehicle routing problems : State-of-the-art, modeling framework and future directions. *European Journal of Operational Research*, 292(3) :799–817.
- Pisinger, D. (2002). Heuristics for the container loading problem. *European Journal of Operational Research*, 141(2) :382–392.
- Popović, D., Bjelić, N., & Radivojević, G. (2011). Simulation approach to analyse deterministic irp solution of the stochastic fuel delivery problem. *Procedia-Social and Behavioral Sciences*, 20 :273–282.
- Popović, D., Vidović, M., & Radivojević, G. (2012). Variable neighborhood search heuristic for the inventory routing problem in fuel delivery. *Expert Systems with Applications*, 39(18) :13390–13398.
- Rahmaniani, R., Crainic, T. G., Gendreau, M., & Rei, W. (2017). The benders decomposition algorithm : A literature review. *European Journal of Operational Research*, 259(3) :801–817.
- Reed, S., Campbell, A. M., & Thomas, B. W. (2022). The value of autonomous vehicles for last-mile deliveries in urban environments. *Management Science*, 68(1) :280–299.
- Ronen, D. (1995). Dispatching petroleum products. *Operations Research*, 43(3) :379–387.
- Taq'a'Allah, D., Renaud, J., & Bector, F. (2000). Le problème d'approvisionnement des stations d'essence. *Journal européen des systemes automatisés*, 34(1) :11–33.
- Triki, C. (2013). Solution methods for the periodic petrol station replenishment problem. *The Journal of Engineering Research [TJER]*, 10(2) :69–77.
- Triki, C. & Al-Hinai, N. (2016). Optimisation techniques for planning the petrol replenishment to retail stations over a multi-period horizon. *International Journal of Operational Research*, 27(1-2) :341–355.

- Triki, C., Al-Hinai, N., Kaabachi, I., & Krichenc, S. (2016). An optimization framework for combining the petroleum replenishment problem with the optimal bidding in combinatorial auctions. *International Journal of Supply and Operations Management*, 3(2) :1318–1331.
- Van der Bruggen, L., Gruson, R., & Salomon, M. (1995). Reconsidering the distribution structure of gasoline products for a large oil company. *European Journal of Operational Research*, 81(3) :460–473.
- Vanderbeck, F. (2000). On dantzig-wolfe decomposition in integer programming and ways to perform branching in a branch-and-price algorithm. *Operations research*, 48(1) :111–128.
- Verme, P. & Araar, A. (2017). The quest for subsidy reforms in the middle east and north africa region. *Washington DC : World Bank*.
- Vidović, M., Popović, D., & Ratković, B. (2014). Mixed integer and heuristics model for the inventory routing problem in fuel delivery. *International Journal of Production Economics*, 147 :593–604.
- Wang, L., Kinable, J., & Van Woensel, T. (2020). The fuel replenishment problem : A split-delivery multi-compartment vehicle routing problem with multiple trips. *Computers & Operations Research*, 118 :104904.
- Wu, L., Adulyasak, Y., Cordeau, J.-F., & Wang, S. (2022). Vessel service planning in seaports. *Operations Research*, 70(4) :2032–2053.
- Zeighami, V. & Soumis, F. (2019). Combining benders’ decomposition and column generation for integrated crew pairing and personalized crew assignment problems. *Transportation Science*, 53(5) :1479–1499.

ANNEXE A

Cette annexe est associée à l'article présenté dans le chapitre 5.

The MDMPPSRP Set-Partitioning Model

The MDMPPSRP set-partitioning formulation (SPP) of Bani et al. (2023) is the following :

$$\min_{\rho} \sum_{\varrho \in \Omega} c_{\varrho} \rho_{\varrho} \quad (\text{SPP})$$

$$\text{s.t. : } \sum_{\varrho \in \Omega} \tilde{q}_{\varrho}^{pe} \rho_{\varrho} \leq \bar{q}_{pe} \quad \forall p \in \mathcal{P}, \forall e \in \mathcal{E} \quad (1)$$

$$\sum_{\varrho \in \Omega} \tilde{s}_{\varrho}^n \rho_{\varrho} = 1 \quad \forall n \in \mathcal{N} \quad (2)$$

$$\sum_{\varrho \in \Omega} \tilde{u}_{\varrho}^{kd} \rho_{\varrho} \leq 1 \quad \forall k \in \mathcal{K}, \forall d \in \mathcal{D}_k \quad (3)$$

$$\rho_{\varrho} \in \{0, 1\} \quad \forall \varrho \in \Omega \quad (4)$$

The MDMPPSRP Pricing Subproblem

The MDMPPSRP pricing subproblem is reformulated and adapted from Bani et al. (2023). Every resource-feasible route corresponds to a feasible route $\varrho \in \Omega$. The cost of this route is the sum of the costs of its arcs $\mathbf{A}^{\varrho}$ and is equal to the corresponding route cost c_{ϱ} . However, in the pricing subproblem, the arc costs need to be modified because we seek to find negative reduced-cost routes, if at least one exists. Hence, the cost of a route should be changed to the reduced cost of the corresponding route variable. We must define the appropriate arc reduced cost $\bar{c}_{ij}$ for each arc $(i, j) \in \mathcal{A}$. To this end, we define four sets of arcs :

- $\mathbf{A}^{\varrho}$: Set of arcs of route ϱ .
- $\mathbf{A}_n^u$: Set of unloading arcs that correspond to serving client-product n .
- $\mathbf{A}_{pe}^l$: Set of loading arcs corresponding to picking up quantity q_{pe} of product p at depot e .
- $\mathbf{A}^o$: Set of other arcs.

The reduced cost $\bar{c}_{\varrho}$ of a variable ρ_{ϱ} is then given by

$$\begin{aligned} \bar{c}_{\varrho} &= c_{\varrho} - \sum_{n \in \mathcal{N}} \tilde{s}_{\varrho}^{nw} \pi_{nw}^{(5.33)} - \sum_{n \in \mathcal{N}} \sum_{d \in \mathcal{D}_k^w} \tilde{o}_{\varrho}^{nd} \pi_{nd}^{(5.32)} - \sum_{k \in \mathcal{K}} \sum_{d \in \mathcal{D}_k^w} \tilde{u}_{\varrho}^{kd} \pi_{kwd}^{(5.34)} - \sum_{p \in \mathcal{P}} \sum_{e \in \mathcal{E}} \tilde{q}_{\varrho}^{pe} \pi_{pew}^{(5.31)} \\ &= \sum_{(i,j) \in \mathbf{A}^{\varrho}} \bar{c}_{ij} \end{aligned} \quad (\text{A.1})$$

In this case, the (reduced) cost $\bar{c}_{ij}$ of arc $(i, j) \in \mathbf{A}^e$ is given by

$$\bar{c}_{ij} = \begin{cases} c_{ij} - \bar{\sigma}_{\mathbf{e}}^{nd} \pi_{nd}^{(5.32)} - \pi_{nw}^{(5.33)} & \text{if } (i, j) \in \mathbf{A}_{\mathbf{n}}^u \\ c_{ij} - \bar{q}_{\mathbf{e}}^{pe} \pi_{pew}^{(5.31)} & \text{if } (i, j) \in \mathbf{A}_{pe}^l \\ c_{ij} & \text{if } (i, j) \in \mathbf{A}^o \end{cases} \quad (\text{A.2})$$

The pricing subproblem is formulated as follows :

$$\min_{x, y, u} \sum_{k \in \mathcal{K}} \sum_{d \in \mathcal{D}_k} \left(\sum_{(i, j) \in \mathcal{A}} \bar{c}_{ij} x_{ij}^{kd} + \Psi_k u_{kd} + \beta \left(\Delta_k u_{kd} - \sum_{l \in \mathcal{L}_k} \sum_{n \in \mathcal{N}} C_{lk} y_{ln}^{kd} \right) \right) \quad (\text{Pricing})$$

$$\text{s.t. : } o_n^{kd} \leq \hat{Q}_n^d \sum_{(i, n) \in \mathcal{A}^{kd}} x_{in}^{kd} \quad \forall n \in \mathcal{N}, \forall k \in \mathcal{K}, \forall d \in \mathcal{D}_k \quad (\text{A.3})$$

$$\sum_{k \in \mathcal{K}} C_{lk} y_{ln}^{kd} = o_n^{kd} \quad \forall n \in \mathcal{N}, \forall k \in \mathcal{K}, \forall d \in \mathcal{D}_k \quad (\text{A.4})$$

$$\sum_{(i, n) \in \mathcal{A}} \sum_{k \in \mathcal{K}} \sum_{d \in \mathcal{D}_k^w} x_{in}^{kd} \leq 1 \quad \forall n \in \mathcal{N}, \forall w \in \mathcal{W} \quad (\text{A.5})$$

$$q_{pe}^{kd} \leq \hat{Q}_{pe} \sum_{i \in \{\mathcal{E} \setminus \{e\}\} \cup \{\sigma\}} x_{ie}^{kd} \quad \forall p \in \mathcal{P}, \forall e \in \mathcal{E}, \forall k \in \mathcal{K}, \forall d \in \mathcal{D}_k \quad (\text{A.6})$$

$$x_{ij}^{kd} \leq u_{kd} \quad \forall (i, j) \in \mathcal{A}, \forall k \in \mathcal{K}, \forall d \in \mathcal{D}_k \quad (\text{A.7})$$

$$\sum_{n \in \mathcal{N}} y_{ln}^{kd} \leq 1 \quad \forall k \in \mathcal{K}, \forall l \in \mathcal{L}_k, \forall d \in \mathcal{D}_k \quad (\text{A.8})$$

$$\sum_{e \in \mathcal{E}} q_{pe}^{kd} = \sum_{n \in \mathcal{N}_p} \sum_{l \in \mathcal{L}_k} C_{lk} y_{ln}^{kd} \quad \forall p \in \mathcal{P}, \forall k \in \mathcal{K}, \forall d \in \mathcal{D}_k \quad (\text{A.9})$$

$$\sum_{(i, j) \in \mathcal{A}} \tau_{ij} x_{ij}^{kd} \leq T_{max} \quad \forall k \in \mathcal{K}, \forall d \in \mathcal{D}_k \quad (\text{A.10})$$

$$\sum_{i \in \mathcal{V}} x_{ij}^{kd} - \sum_{i \in \mathcal{V}} x_{ji}^{kd} = \begin{cases} -1 & \text{if } j = \sigma \\ 0 & \text{if } j \in \mathcal{V} \setminus \{\sigma, \delta\} \\ 1 & \text{if } j = \delta \end{cases} \quad \forall k \in \mathcal{K}, \forall d \in \mathcal{D}_k \quad (\text{A.11})$$

$$\sum_{i, j \in \mathcal{S}} x_{ij}^{kd} \leq |\mathcal{S}| - 1 \quad \forall k \in \mathcal{K}, \forall d \in \mathcal{D}_k \quad (\mathcal{S} \subset \mathcal{V}, 2 \leq |\mathcal{S}| \leq |\mathcal{V}| - 2) \quad (\text{A.12})$$

$$x_{ij}^{kd} \in \{0, 1\} \quad \forall (i, j, k, d) \in \mathcal{C} \quad (\text{A.13})$$

$$u_{kd} \in \{0, 1\} \quad \forall k \in \mathcal{K}, \forall d \in \mathcal{D}_k \quad (\text{A.14})$$

$$y_{ln}^{kd} \in \{0, 1\} \quad \forall n \in \mathcal{N}, \forall k \in \mathcal{K}, \forall l \in \mathcal{L}_k, \forall d \in \mathcal{D}_k \quad (\text{A.15})$$

$$q_{pe}^{kd} \geq 0 \quad \forall k \in \mathcal{K}, \forall p \in \mathcal{P}, \forall e \in \mathcal{E}, \forall d \in \mathcal{D}_k \quad (\text{A.16})$$

$$o_n^{kd} \geq 0 \quad \forall n \in \mathcal{N}, \forall k \in \mathcal{K}, \forall d \in \mathcal{D}_k \quad (\text{A.17})$$