Les Cahiers du GERAD ISSN: 0711–2440

Combining Benders decomposition and column generation for the petrol station replenishment problem with inventory management

A. Bani, E. M. Er Raqabi, I. El Hallaoui, A. I Corréa

G–2024–18

March 2024

La publication de ces rapports de recherche est rendue possible grâce au soutien de HEC Montréal, Polytechnique Montréal, Université McGill, Université du Québec à Montréal, ainsi que du Fonds de recherche du Québec – Nature et technologies.

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2024
– Bibliothèque et Archives Canada, 2024

The publication of these research reports is made possible thanks to the support of HEC Montréal, Polytechnique Montréal, McGill University, Université du Québec à Montréal, as well as the Fonds de recherche du Québec – Nature et technologies.

Legal deposit – Bibliothèque et Archives nationales du Québec, 2024
– Library and Archives Canada, 2024

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

Combining Benders decomposition and column generation for the petrol station replenishment problem with inventory management

Abderrahman Bani [a, b]

El Mehdi Er Raqabi [a]

Issmaïl El Hallaoui [a]

Ayoub Insa Corréa [c]

[a] *Mathematics and Industrial Engineering Department, Polytechnique Montréal & GERAD, Montréal (Qc), Canada*

[b] *Data Science and High-Performance Computing, Hydro Québec Research Institute (IREQ), Varennes (Qc), Canada*

[c] *Unit of Education and Research in Engineering Sciences, University of Thiès, Thiès, Senegal*

abderrahman.bani@polymtl.ca
el-mehdi.er-raqabi@polymtl.ca
issmail.elhallaoui@polymtl.ca
ayoub@univ-thies.sn

March 2024
Les Cahiers du GERAD
G–2024–18

Abstract : The truck loading and inventory routing problems are the two most important decisions made by companies replenishing petrol stations. This paper investigates a complex petrol station replenishment problem (PSRP) that integrates the truck loading problem (TLP) and the inventory routing problem (IRP). We introduce a compact mixed-integer linear programming formulation for the problem. Given its complexity, the compact formulation cannot be solved by general-purpose solvers, even for small instances. Driven by a *decoupling* intuition, we develop an exact two-phase solution approach that combines Benders decomposition and column generation. In the first phase, we solve the relaxed (integrality) Benders subproblems using column generation until the inventory levels stabilize. In the second phase, we solve the Benders subproblems using column generation embedded in a branch-and-bound framework. We enhance our approach with acceleration strategies, including warm-start, parallelism, a hashing technique, and a primal diving heuristic. Extensive computational results using real-world instances from a geographical zone in West Africa highlight the strength of our approach. We reach near-optimal solutions in all these instances and note that acceleration strategies significantly boost the performance of our two-phase method. We generate several managerial insights that highlight our approach's benefits.

Keywords: Petrol station replenishment, inventory routing, truck loading, Benders decomposition, column generation, branch-and-price

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 1 illustrates a petrol station replenishment process.

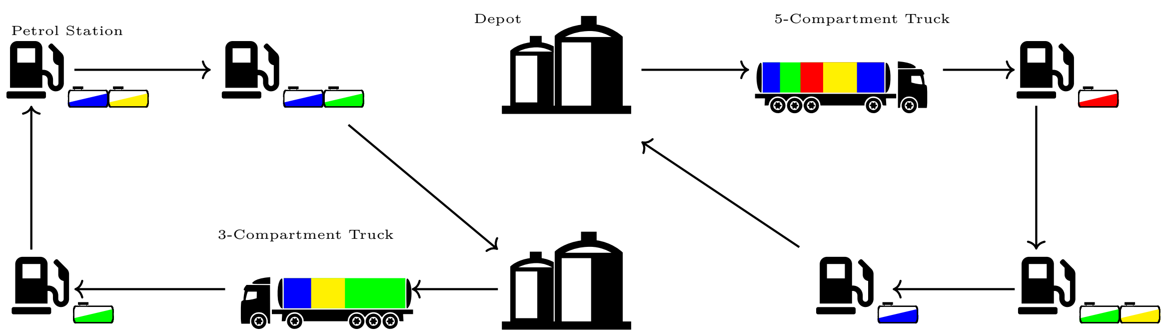


Figure 1: Illustration of a Petrol Station Replenishment Process

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 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 2. We describe the problem and formulate it as a compact MILP model in Sections 3 and 4. Our solution approach is highlighted in 5, followed by computational experiments and their interpretation in Section 6. Section 7 presents the general conclusion with some directions for future research.

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.

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 and 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 and 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).

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 and 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 and 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.

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 and Ramser, 1959; Ostermeier et al., 2021; Coelho and

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 and Graves, 1981), (2) the multi-period PSRP, referred to as MPPSRP (Triki, 2013; Archetti et al., 2015), (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 meta-heuristics 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 and 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.

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 and 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. (2001) 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 and 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. (2001), where the last phase is solved just once (which does not guarantee optimality), Zeighami and Soumis (2019) solve the last phase by generating 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.

2.5 Paper positioning

To the best of our knowledge, no paper in the literature tackles the MDMPPSRPIM exactly (See Table 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 1: Past close studies

Paper	Heterogenous Fleet	Multi-Depot	Depot Inventory	Station Inventory	Multi-Period	Multi Visits/Day	#Cients/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., 2001; Zeighami and 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).

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.

An order placed by a client consists of a specific product, its associated requested volume, and a due date. A client may order one or several types of petrol products. The weekly client orders are computed from their average daily consumptions and thus are known before the week of delivery. 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 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 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.

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 3, sets are in calligraphic style, parameters are in bold, and decision variables are in italics.

Table 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)
$\mathcal{D}^w_k$	Set of availability days $d \in \mathcal{D}^w_k$ 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 *compatible* 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	
$\hat{\boldsymbol{Q}}_{pe}$	Maximum quantity of product $p \in \mathcal{P}$ allowed at depot $e \in \mathcal{E}$
$\boldsymbol{U}^w_{pe}$	Unit price for a liter of product $p \in \mathcal{P}$ at depot $e \in \mathcal{E}$ at week $w \in \mathcal{W}$
$\boldsymbol{H}^w_{pe}$	Storage unit cost for a liter of product $p \in \mathcal{P}$ at depot $e \in \mathcal{E}$ for a week $w \in \mathcal{W}$
$\boldsymbol{C}_{lk}$	Capacity of compartment $l \in \mathcal{L}_k$ in tank truck $k \in \mathcal{K}$
$\boldsymbol{c}_{ij}$	Cost of visiting node j after node i, including distance
$\boldsymbol{\tau}_{ij}$	Travel time from node i to node j, including the waiting time and the unloading time of the products
$\boldsymbol{T}_{max}$	Length of a workday
$\hat{\boldsymbol{Q}}_n$	Tank capacity of client-product $n \in \mathcal{N}$
$\hat{\boldsymbol{O}}^d_n$	Daily consumption of client-product $n \in \mathcal{N}$
$\boldsymbol{\Psi}_k$	Penalty value of using tank truck $k \in \mathcal{K}$
$\boldsymbol{\beta}$	Penalty for leaving free space in the tank truck
$\boldsymbol{\Delta}_k$	Capacity of tank truck $k \in \mathcal{K}$, also equal to $\sum_{l \in \mathcal{L}_k} \boldsymbol{C}_{lk}$
Decision variables	
u_{kd}	$= 1$ if a tank truck $k \in \mathcal{K}$ is used on day $d \in \mathcal{D}$, 0 otherwise
x^{kd}_{ij}	$= 1$ 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^{kd}_{ln}	$= 1$ 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^{kd}_{pe}	≥ 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^w_{pe}	≥ 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^d_{pe}	≥ 0 quantity of product $p \in \mathcal{P}$ to be bought at depot $e \in \mathcal{E}$ at day $d \in \mathcal{D}$
o^{kd}_n	≥ 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^d_n	inventory level of client-product $n \in \mathcal{N}$ at the end of day $d \in \mathcal{D}$ $[0, \hat{\boldsymbol{Q}}_n]$

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 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.

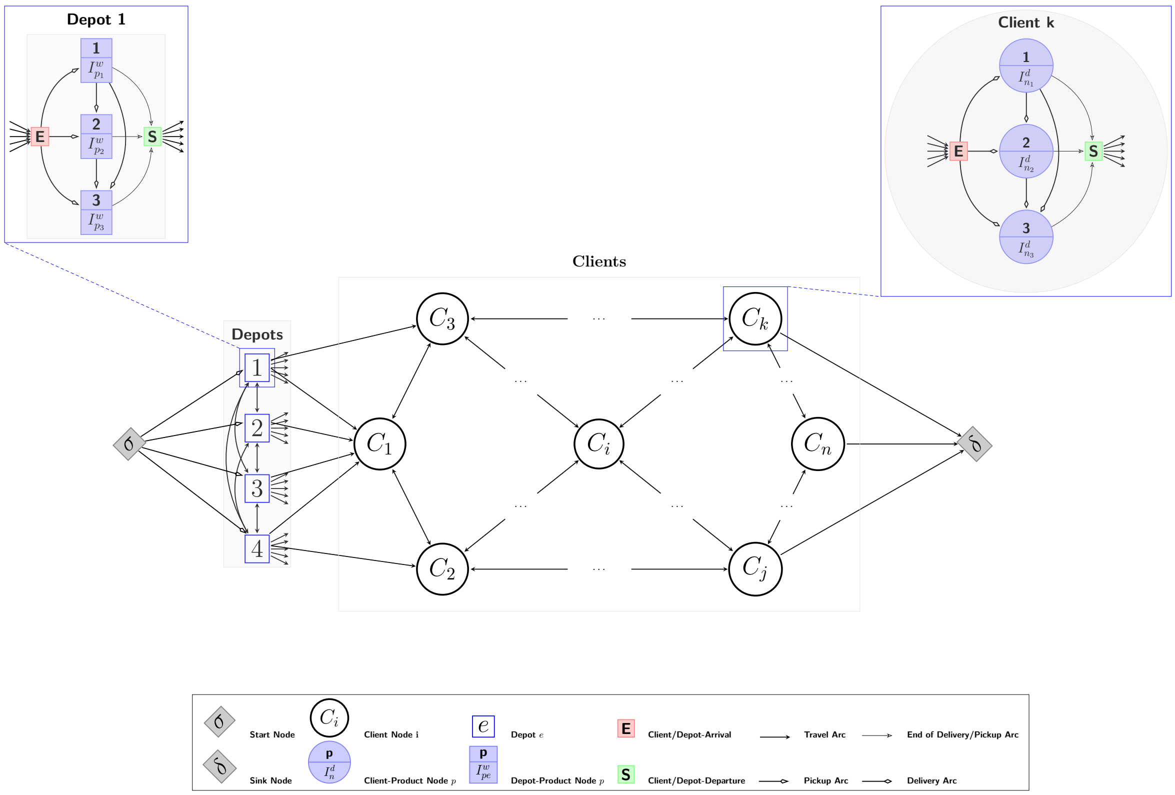


Figure 2: Part of the MDMPPSRPIM Network

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 client-product ($n \in \mathcal{N}$) node represents the inventory level for the period ($d \in \mathcal{D}$). If the quantity $I_n^d \leq \hat{\boldsymbol{O}}_n^d$, 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: $I_n^{d+1} = I_n^d - \hat{\boldsymbol{O}}_n^d + o_n^{kd}$. Otherwise, we update the client-product inventory for the next period as follows: $I_n^{d+1} = I_n^d - \hat{\boldsymbol{O}}_n^d$. 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 $w \in \mathcal{W}$.

4.2 The MDMPPSRPIM Formulation

We present a compact MILP formulation for the MDMPPSRPIM, which recalls an extension of the MDMPPSRP formulation studied by Bani et al. (2023). Using the notation in Table 3, the compact formulation integrates the TLP and the IRP as follows:

$$\min_{\substack{x,y,u,q\\I,o,z}} \sum_{k\in\mathcal{K}}\sum_{d\in\mathcal{D}_k}\left(\sum_{(i,j)\in\mathcal{A}} \boldsymbol{c}_{ij}x_{ij}^{kd} + \boldsymbol{\Psi}_k u_{kd} + \boldsymbol{\beta}\left(\boldsymbol{\Delta}_k u_{kd} - \sum_{l\in\mathcal{L}_k}\sum_{n\in\mathcal{N}} \boldsymbol{C}_{lk}y_{ln}^{kd}\right)\right) + \sum_{w\in\mathcal{W}}\sum_{p\in\mathcal{P}}\sum_{e\in\mathcal{E}}\left(\boldsymbol{U}_{pe}^{w}z_{pe}^{w} + \boldsymbol{H}_{pe}^{w}I_{pe}^{w}\right) \tag{MDMPPSRPIM}$$

s.t.:

$$I_{pe}^{w} - \sum_{k\in\mathcal{K}}\sum_{d\in\mathcal{D}_k^w} q_{pe}^{kd} + z_{pe}^{w} = I_{pe}^{w+1} \qquad \forall\, p\in\mathcal{P}, \forall\, e\in\mathcal{E}, \forall\, w\in\mathcal{W} \tag{1}$$

$$I_{n}^{d} - \hat{\boldsymbol{O}}_{n}^{d} + \sum_{k\in\mathcal{K}} o_{n}^{kd} = I_{n}^{d+1} \qquad \forall\, n\in\mathcal{N}, \forall\, d\in\mathcal{D} \tag{2}$$

$$\sum_{e\in\mathcal{E}}\left(I_{pe}^{w} - I_{pe}^{w+1} + z_{pe}^{w}\right) = \sum_{n\in\mathcal{N}_p}\sum_{d\in\mathcal{D}^w}\left(\hat{O}_{n}^{d} + I_{n}^{d+1} - I_{n}^{d}\right) \qquad \forall\, p\in\mathcal{P}, \forall\, w\in\mathcal{W} \tag{3}$$

$$o_{n}^{kd} \le \hat{\boldsymbol{Q}}_{n}^{d} \sum_{(i,n)\in\mathcal{A}^{kd}} x_{in}^{kd} \qquad \forall\, n\in\mathcal{N}, \forall\, k\in\mathcal{K}, \forall\, d\in\mathcal{D}_k \tag{4}$$

$$\sum_{k\in\mathcal{K}} \boldsymbol{C}_{lk}y_{ln}^{kd} = o_{n}^{kd} \qquad \forall\, n\in\mathcal{N}, \forall\, k\in\mathcal{K}, \forall\, d\in\mathcal{D}_k \tag{5}$$

$$\sum_{(i,n)\in\mathcal{A}}\sum_{k\in\mathcal{K}}\sum_{d\in\mathcal{D}_k^w} x_{in}^{kd} \le 1 \qquad \forall\, n\in\mathcal{N}, \forall\, w\in\mathcal{W} \tag{6}$$

$$q_{pe}^{kd} \le \hat{\boldsymbol{Q}}_{pe} \sum_{i\in\{\mathcal{E}\setminus\{e\}\}\cup\{\sigma\}} x_{ie}^{kd} \qquad \forall\, p\in\mathcal{P}, \forall\, e\in\mathcal{E}, \forall\, k\in\mathcal{K}, \forall\, d\in\mathcal{D}_k \tag{7}$$

$$x_{ij}^{kd} \le u_{kd} \qquad \forall\, (i,j)\in\mathcal{A}, \forall\, k\in\mathcal{K}, \forall\, d\in\mathcal{D}_k \tag{8}$$

$$\sum_{n\in\mathcal{N}} y_{ln}^{kd} \le 1 \qquad \forall\, k\in\mathcal{K}, \forall\, l\in\mathcal{L}_k, \forall\, d\in\mathcal{D}_k \tag{9}$$

$$\sum_{e\in\mathcal{E}} q_{pe}^{kd} = \sum_{n\in\mathcal{N}_p}\sum_{l\in\mathcal{L}_k} \boldsymbol{C}_{lk}y_{ln}^{kd} \qquad \forall\, p\in\mathcal{P}, \forall\, k\in\mathcal{K}, \forall\, d\in\mathcal{D}_k \tag{10}$$

$$\sum_{(i,j)\in\mathcal{A}} \boldsymbol{\tau}_{ij}x_{ij}^{kd} \le \boldsymbol{T}_{max} \qquad \forall\, k\in\mathcal{K}, \forall\, d\in\mathcal{D}_k \tag{11}$$

$$\sum_{i\in\mathcal{V}} x_{ij}^{kd} - \sum_{i\in\mathcal{V}} x_{ji}^{kd} = \begin{cases} -1 & \text{if } j=\sigma\\ 0 & \text{if } j\in\mathcal{V}\setminus\{\sigma,\delta\}\\ 1 & \text{if } j=\delta\end{cases} \qquad \forall\, k\in\mathcal{K}, \forall\, d\in\mathcal{D}_k \tag{12}$$

$$\sum_{i,j\in\mathcal{S}} x_{ij}^{kd} \le |\mathcal{S}| - 1 \qquad \forall\, k\in\mathcal{K}, \forall\, d\in\mathcal{D}_k\ (\mathcal{S}\subset\mathcal{V}, 2\le|\mathcal{S}|\le|\mathcal{V}|-2) \tag{13}$$

$$x_{ij}^{kd}\in\{0,1\} \qquad \forall\, (i,j,k,d)\in\mathcal{C} \tag{14}$$

$$u_{kd}\in\{0,1\} \qquad \forall\, k\in\mathcal{K}, \forall\, d\in\mathcal{D}_k \tag{15}$$

$$y_{ln}^{kd}\in\{0,1\} \qquad \forall\, n\in\mathcal{N}, \forall\, k\in\mathcal{K}, \forall\, l\in\mathcal{L}_k, \forall\, d\in\mathcal{D}_k \tag{16}$$

$$q_{pe}^{kd}\ge 0 \qquad \forall\, k\in\mathcal{K}, \forall\, p\in\mathcal{P}, \forall\, e\in\mathcal{E}, \forall\, d\in\mathcal{D}_k \tag{17}$$

$$0\le z_{pe}^{w}\le \hat{\boldsymbol{Q}}_{pe} \qquad \forall\, p\in\mathcal{P}, \forall\, e\in\mathcal{E}, \forall\, w\in\mathcal{W} \tag{18}$$

$$0\le I_{pe}^{w}\le \hat{\boldsymbol{Q}}_{pe} \qquad \forall\, p\in\mathcal{P}, \forall\, e\in\mathcal{E}, \forall\, w\in\mathcal{W} \tag{19}$$

$$0\le I_{n}^{d}\le \hat{\boldsymbol{Q}}_{n} \qquad \forall\, n\in\mathcal{N}, \forall\, d\in\mathcal{D} \tag{20}$$

$$o_{n}^{kd}\ge 0 \qquad \forall\, n\in\mathcal{N}, \forall\, k\in\mathcal{K}, \forall\, d\in\mathcal{D}_k \tag{21}$$

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 (1) manage the inventory at the depot while satisfying Rule $\mathcal{R}_3$ on available quantities. Constraints (2) manage the inventory at the clients' sites. Constraints (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 (4) ensure that the delivery can take place only when the client $n\in\mathcal{N}$ is visited. Constraints (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 (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 (7) ensure that a tank truck can load products from the depot $e \in \mathcal{E}$ only after visiting it. Constraints (8) ensure that a tank truck can only be used if it is available on day $d \in \mathcal{D}$. Constraints (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 (10) specify that the quantities of each product $p \in \mathcal{P}$ delivered to clients must be loaded from depots. Constraints (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 (12) and (13) are the flow conservation constraints and sub-tour elimination constraints, respectively. Finally, the decision variable conditions are given by (14)–(21).

We observe that the second and third terms of the objective function with constraints (5)–(10), (15), and (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 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 4, 5, and 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.

5.1 The Route Formulation

We reformulate the MDMPPSRPIM using routes. For this purpose, let $\Omega = \{1, \ldots, |\Omega|\}$ 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 $\boldsymbol{c_\varrho}$ corresponds to the sum of all the costs associated with route $\varrho \in \Omega$.

We introduce the following additional notation. The coefficient $\tilde{\boldsymbol{s}}_{\boldsymbol{\varrho}}^{\boldsymbol{nw}}$ 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 $\tilde{\boldsymbol{u}}_{\boldsymbol{\varrho}}^{\boldsymbol{kwd}}$ 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 $\tilde{\boldsymbol{q}}_{\boldsymbol{\varrho}}^{\boldsymbol{pe}}$ 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 $\tilde{\boldsymbol{o}}_{\boldsymbol{\varrho}}^{\boldsymbol{nd}}$ 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:

$$\min_{\rho,z,I} \sum_{\varrho\in\Omega} \boldsymbol{c_\varrho}\rho_\varrho + \sum_{w\in\mathcal{W}}\sum_{p\in\mathcal{P}}\sum_{e\in\mathcal{E}} \left(\boldsymbol{U_{pe}^w} z_{pe}^w + \boldsymbol{H_{pe}^w} I_{pe}^w\right) \tag{MDMPPSRPIM$_\rho$}$$

$$\text{s.t.:} \quad I_{pe}^w - \sum_{\varrho\in\Omega} \tilde{\boldsymbol{q}}_{\boldsymbol{\varrho}}^{\boldsymbol{pe}}\rho_\varrho + z_{pe}^w = I_{pe}^{w+1} \qquad \forall\, p \in \mathcal{P}, \forall\, e \in \mathcal{E}, \forall\, w \in \mathcal{W} \tag{22}$$

$$I_n^d - \hat{\boldsymbol{O}}_{\boldsymbol{n}}^{\boldsymbol{d}} + \sum_{\varrho\in\Omega} \tilde{\boldsymbol{o}}_{\boldsymbol{\varrho}}^{\boldsymbol{nd}}\rho_\varrho = I_n^{d+1} \qquad \forall\, n \in \mathcal{N}, \forall\, d \in \mathcal{D} \tag{23}$$

$$\sum_{e\in\mathcal{E}} \left(I_{pe}^w - I_{pe}^{w+1} + z_{pe}^w\right) = \sum_{n\in\mathcal{N}_p}\sum_{d\in\mathcal{D}^w} \left(\hat{O}_n^d + I_n^{d+1} - I_n^d\right) \qquad \forall\, p \in \mathcal{P}, \forall\, w \in \mathcal{W} \tag{24}$$

$$\sum_{\varrho\in\Omega} \tilde{\boldsymbol{s}}_{\boldsymbol{\varrho}}^{\boldsymbol{nw}} \rho_\varrho \leq 1 \qquad \forall\, n \in \mathcal{N}, \forall\, w \in \mathcal{W} \tag{25}$$

$$\sum_{\varrho\in\Omega} \tilde{\boldsymbol{u}}_{\boldsymbol{\varrho}}^{\boldsymbol{kwd}} \rho_\varrho \leq 1 \qquad \forall\, k \in \mathcal{K}, \forall\, w \in \mathcal{W}, \forall\, d \in \mathcal{D}_k^w \tag{26}$$

$$0 \leq z_{pe}^w \leq \hat{\boldsymbol{Q}}_{pe} \qquad \forall\, p \in \mathcal{P}, \forall\, e \in \mathcal{E}, \forall\, w \in \mathcal{W} \tag{27}$$

$$0 \leq I_{pe}^w \leq \hat{\boldsymbol{Q}}_{\boldsymbol{pe}} \qquad \forall\, p \in \mathcal{P}, \forall\, e \in \mathcal{E}, \forall\, w \in \mathcal{W} \tag{28}$$

$$0 \leq I_n^d \leq \hat{\boldsymbol{Q}}_{\boldsymbol{n}} \qquad \forall\, n \in \mathcal{N}, \forall\, d \in \mathcal{D} \tag{29}$$

$$\rho_\varrho \in \{0, 1\} \qquad \forall\, \varrho \in \Omega \tag{30}$$

The objective function minimizes the total cost. Constraints (22) manage the inventory at the depot for each product. Constraints (23) manage the inventory at the client. Constraints (24) equals the quantities that go out of the depots with the quantities received by the clients. Constraints (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 (26) ensure that we can use tank truck $k \in \mathcal{K}$ at most once per day $d \in \mathcal{D}$. Constraints (28) ensure that the quantities loaded from each depot do not exceed what is available. Constraints (29) ensure that the daily quantities delivered to each client-product n respect their maximum capacity limit. Finally, integrality constraints (30) are imposed on the route variables ρ_ϱ for each route $\varrho \in \Omega$.

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 (22–23) 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 (22–23) 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.

5.2 The Benders reformulation

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

5.2.1 The Benders subproblem

After fixing the complicating variables I^w_{pe}, I^d_n, and z^w_{pe} (a fixed variable has a bar on it), the Benders primal subproblem is the following:

$$\min_{\rho} \sum_{\varrho\in\Omega} c_\varrho \rho_\varrho \tag{BPSP}$$

$$\text{s.t.:} \sum_{\varrho\in\Omega} \tilde{\boldsymbol{q}}^{\boldsymbol{pe}}_{\boldsymbol{\varrho}} \rho_\varrho = \bar{I}^w_{pe} - \bar{I}^{w+1}_{pe} + \bar{z}^w_{pe} \quad \forall\, p\in\mathcal{P}, \forall\, e\in\mathcal{E}, \forall\, w\in\mathcal{W} \tag{31}$$

$$\sum_{\varrho\in\Omega} \tilde{\boldsymbol{o}}^{\boldsymbol{nd}}_{\boldsymbol{\varrho}} \rho_\varrho = \hat{O}^d_n + \bar{I}^{d+1}_n - \bar{I}^d_n \quad \forall\, n\in\mathcal{N}, \forall\, d\in\mathcal{D} \tag{32}$$

$$\sum_{\varrho\in\Omega} \tilde{\boldsymbol{s}}^{\boldsymbol{nw}}_{\boldsymbol{\varrho}} \rho_\varrho \le 1 \quad \forall\, n\in\mathcal{N}, \forall\, w\in\mathcal{W} \tag{33}$$

$$\sum_{\varrho\in\Omega} \tilde{\boldsymbol{u}}^{\boldsymbol{kwd}}_{\boldsymbol{\varrho}} \rho_\varrho \le 1 \quad \forall\, k\in\mathcal{K}, \forall\, w\in\mathcal{W}, \forall\, d\in\mathcal{D}^w_k \tag{34}$$

$$\rho_\varrho \in \{0,1\} \quad \forall\, \varrho\in\Omega \tag{35}$$

At each iteration, we are supposed to solve the dual of BPSP to obtain either an optimality or feasibility cut. Let $\boldsymbol{\pi}^{(31)} = (\pi^{(31)}_{pew} \mid p\in\mathcal{P}, e\in\mathcal{E}, w\in\mathcal{W})$, $\boldsymbol{\pi}^{(32)} = (\pi^{(32)}_{nd} \mid n\in\mathcal{N}, d\in\mathcal{D})$, $\boldsymbol{\pi}^{(33)} = (\pi^{(33)}_{nw} \le 0 \mid n\in\mathcal{N}, w\in\mathcal{W})$ and $\boldsymbol{\pi}^{(34)} = (\pi^{(34)}_{kwd} \le 0 \mid k\in\mathcal{K}, w\in\mathcal{W}, d\in\mathcal{D}^w_k)$ be the dual variables associated with constraints (31)–(34), respectively. The dual of the relaxed BPSP is then

$$\min_{\pi} \sum_{p\in\mathcal{P}}\sum_{e\in\mathcal{E}}\sum_{w\in\mathcal{W}} \left(\bar{I}^w_{pe} - \bar{I}^{w+1}_{pe} + \bar{z}^w_{pe}\right)\pi^{(31)}_{pew} + \sum_{n\in\mathcal{N}}\sum_{d\in\mathcal{D}} \left(\hat{O}^d_n + \bar{I}^{d+1}_n - \bar{I}^d_n\right)\pi^{(32)}_{nd} + \sum_{n\in\mathcal{N}}\sum_{w\in\mathcal{W}} \pi^{(33)}_{nw} + \sum_{k\in\mathcal{K}}\sum_{w\in\mathcal{W}}\sum_{d\in\mathcal{D}^w_k} \pi^{(34)}_{kwd} \tag{BDSP}$$

$$\text{s.t.:} \sum_{p\in\mathcal{P}}\sum_{e\in\mathcal{E}}\sum_{w\in\mathcal{W}} \tilde{\boldsymbol{q}}^{\boldsymbol{pe}}_{\boldsymbol{\varrho}} \pi^{(31)}_{pew} + \sum_{n\in\mathcal{N}}\sum_{d\in\mathcal{D}} \tilde{\boldsymbol{o}}^{\boldsymbol{nd}}_{\boldsymbol{\varrho}} \pi^{(32)}_{nd} + \sum_{n\in\mathcal{N}}\sum_{w\in\mathcal{W}} \tilde{\boldsymbol{s}}^{\boldsymbol{nw}}_{\boldsymbol{\varrho}} \pi^{(33)}_{nw} + \sum_{k\in\mathcal{K}}\sum_{w\in\mathcal{W}}\sum_{d\in\mathcal{D}^w_k} \tilde{\boldsymbol{u}}^{\boldsymbol{kwd}}_{\boldsymbol{\varrho}} \pi^{(34)}_{kwd} \le \boldsymbol{c_\varrho} \quad \forall\, \varrho\in\Omega \tag{36}$$

$$\pi^{(33)}_{nw} \le 0 \quad \forall\, n\in\mathcal{N}, \forall\, w\in\mathcal{W} \tag{37}$$

$$\pi^{(34)}_{kwd} \le 0 \quad \forall\, k\in\mathcal{K}, \forall\, w\in\mathcal{W}, \forall\, d\in\mathcal{D}_k \tag{38}$$

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 $\bar{o}^d_n = \hat{\boldsymbol{O}}^d_n + \bar{I}^{d+1}_n - \bar{I}^d_n$. We also know the quantities of each product p that must be taken from the depot e each week w, which we denote $\bar{q}^w_{pe} = \bar{I}^w_{pe} - \bar{I}^{w+1}_{pe} + \bar{z}^w_{pe}$. For a given week $w\in\mathcal{W}$, the weekly problem is this:

$$\max_{\pi,w} \sum_{p\in\mathcal{P}}\sum_{e\in\mathcal{E}} \bar{\boldsymbol{q}}^{\boldsymbol{w}}_{\boldsymbol{pe}} \pi^{(31)}_{pew} + \sum_{n\in\mathcal{N}}\sum_{d\in\mathcal{D}^w} \bar{\boldsymbol{o}}^{\boldsymbol{d}}_{\boldsymbol{n}} \pi^{(32)}_{nd} + \sum_{n\in\mathcal{N}} \pi^{(33)}_{nw} + \sum_{k\in\mathcal{K}}\sum_{d\in\mathcal{D}^w_k} \pi^{(34)}_{kwd} \tag{WBDSP}$$

$$\text{s.t.:} \sum_{p\in\mathcal{P}}\sum_{e\in\mathcal{E}} \tilde{\boldsymbol{q}}^{\boldsymbol{pe}}_{\boldsymbol{\varrho}} \pi^{(31)}_{pew} + \sum_{n\in\mathcal{N}}\sum_{d\in\mathcal{D}^w} \tilde{\boldsymbol{o}}^{\boldsymbol{nd}}_{\boldsymbol{\varrho}} \pi^{(32)}_{nd} + \sum_{n\in\mathcal{N}} \tilde{\boldsymbol{s}}^{\boldsymbol{nw}}_{\boldsymbol{\varrho}} \pi^{(33)}_{nw} + \sum_{k\in\mathcal{K}}\sum_{d\in\mathcal{D}^w_k} \tilde{\boldsymbol{u}}^{\boldsymbol{kwd}}_{\boldsymbol{\varrho}} \pi^{(34)}_{kwd} \le \boldsymbol{c_\varrho} \quad \forall\, \varrho\in\Omega \tag{39}$$

$$\pi^{(33)}_{nw} \le 0 \quad \forall\, n\in\mathcal{N} \tag{40}$$

$$\pi^{(34)}_{kwd} \le 0 \quad \forall\, k\in\mathcal{K}, \forall\, d\in\mathcal{D}^w_k \tag{41}$$

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

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

$$\text{s.t.:} \sum_{p\in\mathcal{P}}\sum_{e\in\mathcal{E}} \tilde{\boldsymbol{q}}_{\boldsymbol{\varrho}}^{\boldsymbol{pe}}\alpha_{pe}^{(1)} + \sum_{n\in\mathcal{N}} \tilde{\boldsymbol{s}}_{\boldsymbol{\varrho}}^{\boldsymbol{n}}\alpha_n^{(2)} + \sum_{k\in\mathcal{K}}\sum_{d\in\mathcal{D}_k} \tilde{\boldsymbol{u}}_{\boldsymbol{\varrho}}^{\boldsymbol{kd}}\alpha_{kd}^{(3)} \le \boldsymbol{c}_{\boldsymbol{\varrho}} \qquad \forall\ \varrho\in\Omega \tag{42}$$

$$\alpha_{pe}^{(1)} \le 0 \qquad \forall\ n\in\mathcal{N} \tag{43}$$

$$\alpha_{kd}^{(3)} \le 0 \qquad \forall\ k\in\mathcal{K}, \forall\ d\in\mathcal{D}_k \tag{44}$$

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

$$\alpha_{pe}^{(1)} = \pi_{pew}^{(31)} \qquad \forall\ p\in\mathcal{P}, \forall\ e\in\mathcal{E} \tag{Dual Equalities}$$

$$\alpha_n^{(2)} = \sum_{d\in\mathcal{D}^w} \bar{\boldsymbol{o}}_{\boldsymbol{n}}^{\boldsymbol{d}}\pi_{nd}^{(32)} + \pi_{nw}^{(33)} \qquad \forall\ n\in\mathcal{N}$$

$$\tilde{s}_{\varrho}^{n}\alpha_n^{(2)} = \sum_{d\in\mathcal{D}^w} \tilde{\boldsymbol{o}}_{\boldsymbol{\varrho}}^{\boldsymbol{nd}}\pi_{nd}^{(32)} + \tilde{\boldsymbol{s}}_{\boldsymbol{\varrho}}^{\boldsymbol{nw}}\pi_{nw}^{(33)} \qquad \forall\ n\in\mathcal{N}$$

$$\alpha_{kd}^{(3)} = \pi_{kwd}^{(34)} \qquad \forall\ k\in\mathcal{K}, \forall\ d\in\mathcal{D}_k^w$$

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

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).

5.2.2 The Benders master problem

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

$$\min_{z,I} \sum_{w\in\mathcal{W}}\sum_{p\in\mathcal{P}}\sum_{e\in\mathcal{E}} \left(\boldsymbol{U}_{pe}^{w} z_{pe}^{w} + \boldsymbol{H}_{pe}^{w} I_{pe}^{w}\right) + \mu \tag{BMP}$$

$$\begin{aligned}\text{s.t.:}\quad \mu \ge & \sum_{p\in\mathcal{P}}\sum_{e\in\mathcal{E}}\sum_{w\in\mathcal{W}} \left(I_{pe}^{w} - I_{pe}^{w+1} + z_{pe}^{w}\right)\pi_{pew}^{(31)} \\ & + \sum_{n\in\mathcal{N}}\sum_{d\in\mathcal{D}} \left(\hat{\boldsymbol{O}}_{n}^{d} + I_n^{d+1} - I_n^{d}\right)\pi_{nd}^{(32)} \\ & + \sum_{n}\sum_{w}\pi_{nw}^{(33)} + \sum_{k\in\mathcal{K}}\sum_{w\in\mathcal{W}}\sum_{d\in\mathcal{D}_k^w}\pi_{kwd}^{(34)} \qquad (\boldsymbol{\pi}^{(31)},\boldsymbol{\pi}^{(32)},\boldsymbol{\pi}^{(33)},\boldsymbol{\pi}^{(34)})\in\boldsymbol{\Upsilon}_{\mathcal{F}}\end{aligned} \tag{45}$$

$$\sum_{e\in\mathcal{E}} \left(I_{pe}^{w} - I_{pe}^{w+1} + z_{pe}^{w}\right) = \sum_{n\in\mathcal{N}_p}\sum_{d\in\mathcal{D}^w} \left(\hat{O}_n^d + I_n^{d+1} - I_n^d\right) \qquad \forall\ p\in\mathcal{P}, \forall\ w\in\mathcal{W} \tag{46}$$

$$0 \le z_{pe}^{w} \le \hat{\boldsymbol{Q}}_{pe} \qquad \forall\ p\in\mathcal{P}, \forall\ e\in\mathcal{E}, \forall\ w\in\mathcal{W} \tag{47}$$

$$0 \le I_{pe}^{w} \le \hat{\boldsymbol{Q}}_{pe} \qquad \forall\ p\in\mathcal{P}, \forall\ e\in\mathcal{E}, \forall\ w\in\mathcal{W} \tag{48}$$

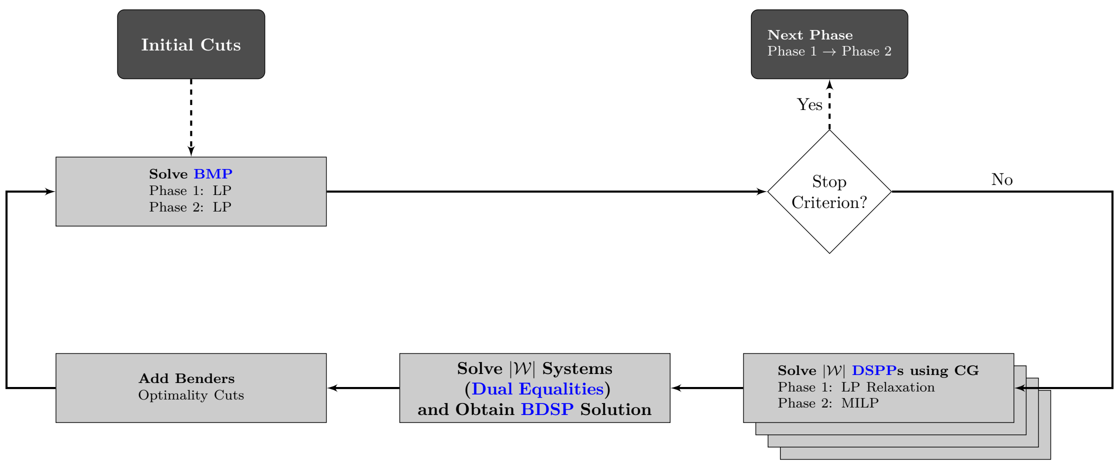
$$0 \le I_n^d \le \hat{\boldsymbol{Q}}_n \qquad \forall\ n\in\mathcal{N}, \forall\ d\in\mathcal{D} \tag{49}$$

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_{ρ}, 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.

5.3 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. (2001). We adapt it to our context where we solve $|\mathcal{W}|$ DSPPs using CG. We highlight the algorithm in Figure 3.

Figure 3: Flowchart of the Two-Phase Approach



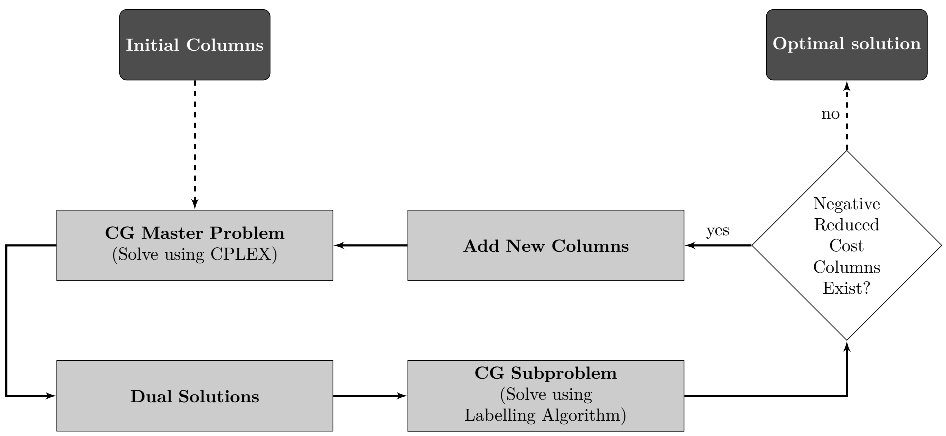
The algorithm consists of two phases: *Phase 1* and *Phase 2*. In Phase 1, the integrality conditions are relaxed, and at each iteration of the BD, the LP relaxation of the BMP and the $|\mathcal{W}|$ SPPs are solved. Each SPP is solved using CG. All the Benders optimality cuts generated in the first 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 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. (2023).

We also note that instead of solving Phase 2 once, as in Cordeau et al. (2001), we solve it until the optimality criterion is reached, as in Zeighami and Soumis (2019) (see Section 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.

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 B. We then add these routes to the RMP. We iterate until no negative reduced-cost route(s) are identified, as highlighted in Figure 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.

Figure 4: Flowchart of the Colum Generation Procedure



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 and Lodi, 2003) around primal feasible integer solution(s) and a diving heuristic (Joncour et al., 2010). 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.

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.

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 3, some additional specific business rules are summarized in Table 4 below.

Table 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:

$$\sum_{(i,j)\in\mathcal{N}\times\mathcal{N}_f} x_{ij}^{kd} \leq 2 \qquad \forall\, k \in \mathcal{K}, \forall\, d \in \mathcal{D}_k \tag{50}$$

where $\mathcal{N}_f$ is the subset of client-products related to marine stations. Constraints (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 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.

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, 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: Instances Properties

$\lvert\mathcal{N}\rvert$	#C	$\mathcal{P}$	$\lvert\mathcal{W}\rvert/\lvert\mathcal{D}\rvert$	$\lvert\mathcal{E}\rvert$	Instance	$\lvert\mathcal{N}\rvert$	#C	$\mathcal{P}$	$\lvert\mathcal{W}\rvert/\lvert\mathcal{D}\rvert$	$\lvert\mathcal{E}\rvert$	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.), whether the truck is suitable for marked and unmarked products (F.), and whether or not the truck is jumbo (J.), are highlighted in Table 6.

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 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.

Table 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	$[2, 1, 1, 1, 1, 1, 1, 1, 2, 1, 2]$	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	☐	☑

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.

6.3.1 Performance

Table 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 8 shows the performance of CG. For each instance, we report the total time (T) in seconds (same as Table 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

Table 7: Benders Decomposition Results

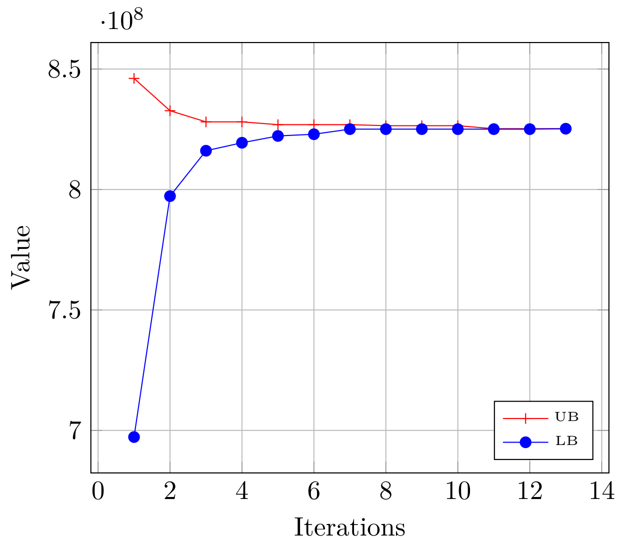
						Phase 1			Phase 2		
$\|\mathcal{N}\|$	$\|\mathcal{W}\|$	**Inst.**	**T(s)**	**TMP(s)**	**#Cuts**	**Gap(%)**	**TMP(%)**	**#Cuts**	**Gap(%)**	**TMP(%)**	**#Cuts**
24	4	**P4-C24-D1-S1**	1.82	0.00	4	0.10	87.71	3	0.05	12.29	1
24	4	**P4-C24-D1-S2**	1.62	0.00	4	0.10	88.44	3	0.05	11.56	1
24	4	**P4-C24-D4-S1**	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 8: Column Generation Results

$\lvert\mathcal{N}\rvert$	$\lvert\mathcal{W}\rvert$	Inst.	T(s)	Total #Col	Total #Itr	Phase 1 TSP(%)	Phase 1 #Col	Phase 1 #Itr	Phase 2 TSP(%)	Phase 2 #Col	Phase 2 #Itr	Phase 2 #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**

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 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**.

Figure 5: UB and LB Improvement Along With the Number of Benders Iterations for Instance P4-C77-D4-S2



The compact model 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 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.

6.3.2 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 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 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

Table 9: Benders Master Problem Acceleration Strategies Impact

			No WS			Our Approach		
$\|\mathcal{N}\|$	$\|\mathcal{W}\|$	**Inst.**	**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**

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% 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 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, 2023). We use the default branching rule.

Table 10: Benders Subproblem Acceleration Strategies Impact

			No Parallel				**No Hash**				**No PDH**				**Our Aapproach**			
$\|\mathcal{N}\|$	$\|\mathcal{W}\|$	**Inst.**	**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 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.

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 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.

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 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.

Table 11: Comparison with Heuristics 1 and 2

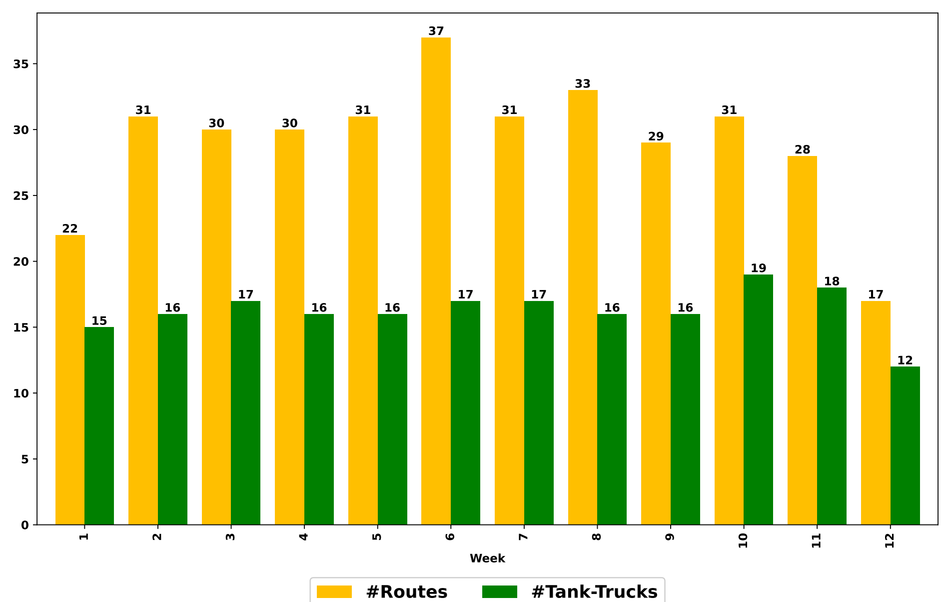
			Heuristic 1					**Heuristic 2**					**Our Approach**				
$\lvert\mathcal{N}\rvert$	$\lvert\mathcal{W}\rvert$	**Inst.**	**Gap(%)**	**#Vis**	**#Ro**	**#Tr**	**F(%)**	**Gap(%)**	**#Vis**	**#Ro**	**#Tr**	**F(%)**	**Gap(%)**	**#Vis**	**#Ro**	**#Tr**	**F(%)**
24	4	**P4-C24-D1-S1**	5.03	55	44	26	12.30	18.77	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**

6.6 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 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 6: Number of Routes and Trucks Used per Week P12-C77-D4-S1



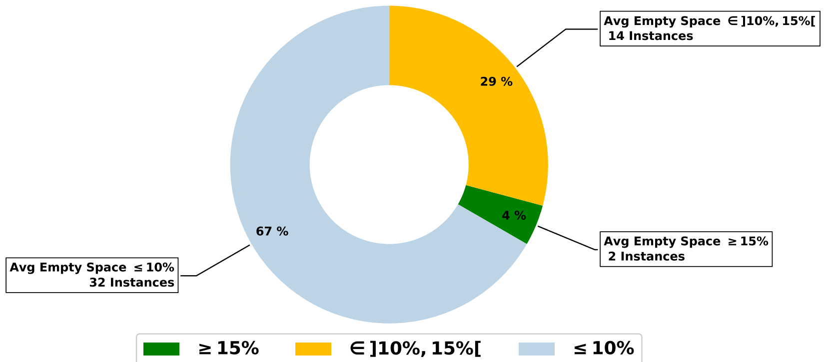
Results also highlight that, on average, the number of clients visited by a given truck (on a single route) is 1.5. This can be explained by the clients' large tank capacities, which implies that the trucks will serve few clients. This is similar to the American context, where direct billing is the main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

We report in Figure 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 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 7: Average Free Space Percentage Within Trucks per Week for All 48 Instances



We lastly discuss an interesting aspect for oil companies, namely, when to buy and when to store. Figure 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.

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.

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.

Figure 8: Number of Visits Required and Performed for Each Client-Product for Instance P12-C77-D4-S1

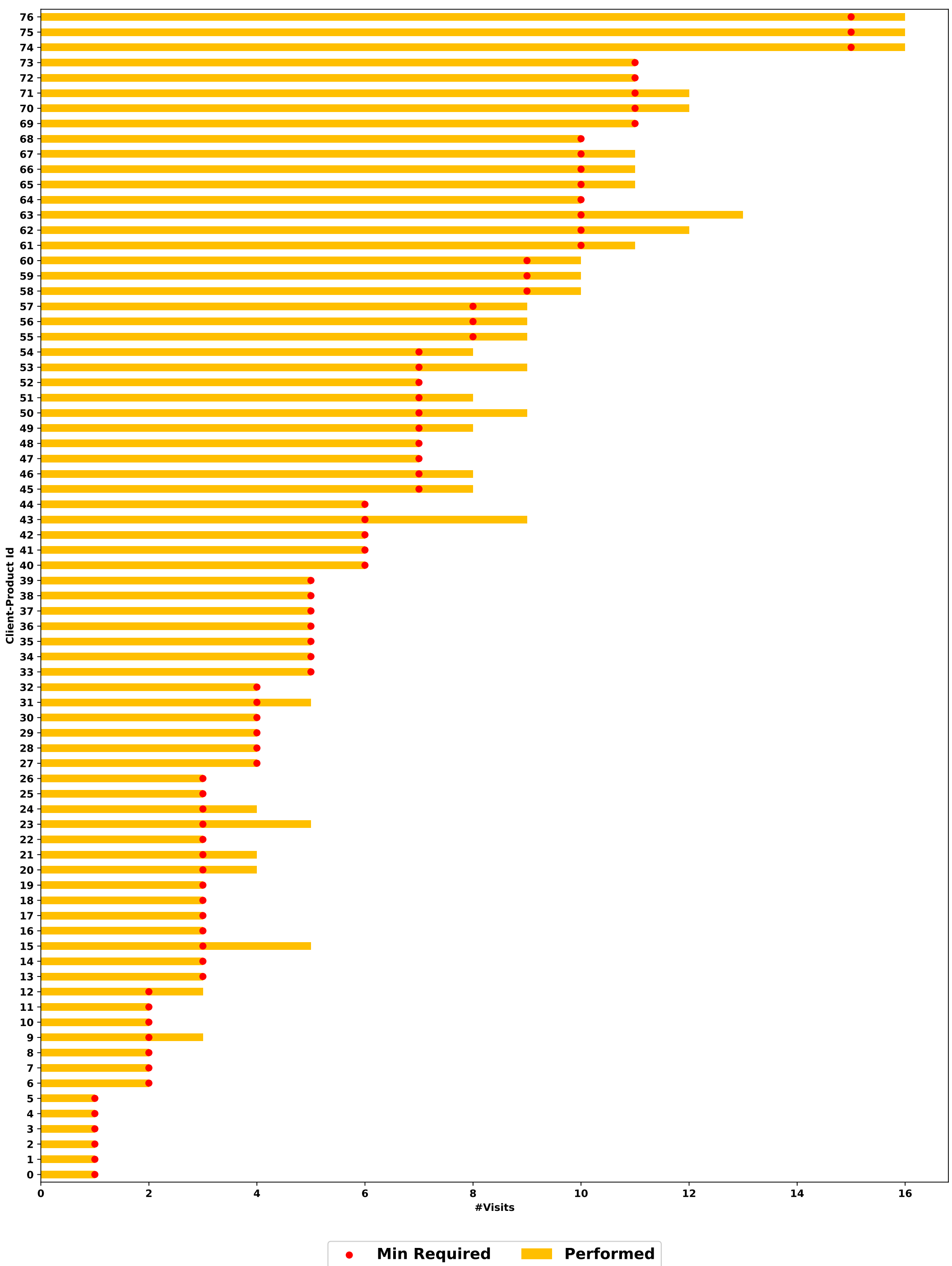
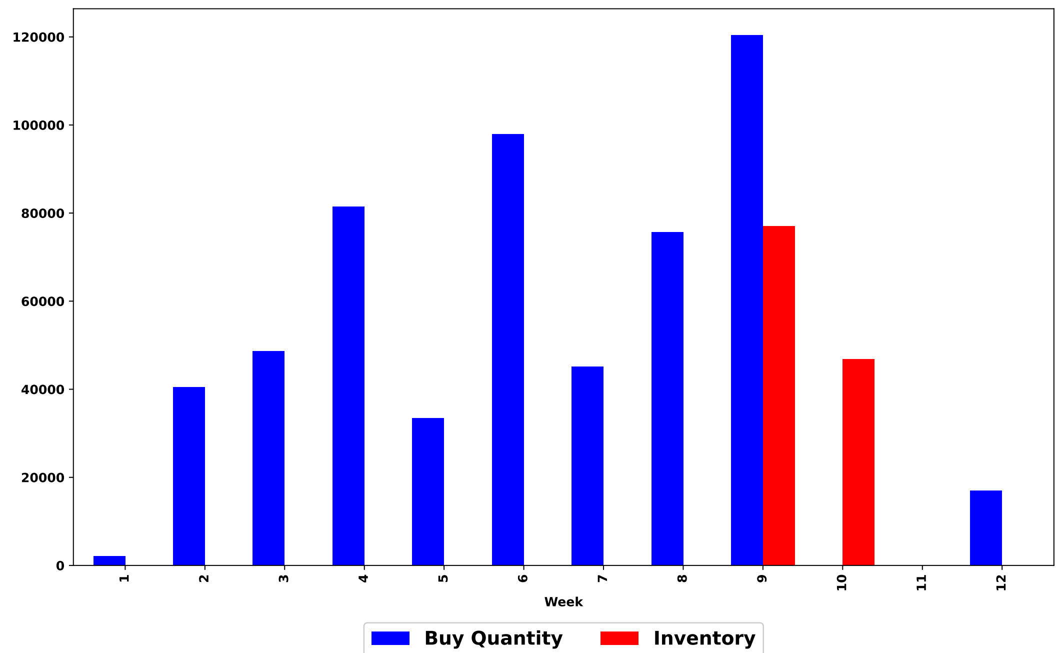


Figure 9: Inventory Levels (Variation) and Buying Pulses for Instance P12-C77-D4-S1



Appendix A The MDMPPSRP set-partitioning model

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

$$\min_{\rho} \quad \sum_{\varrho \in \Omega} c_{\varrho} \rho_{\varrho} \tag{SPP}$$

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

$$\sum_{\varrho \in \Omega} \tilde{s}_{\varrho}^{n} \rho_{\varrho} = 1 \qquad \forall\, n \in \mathcal{N} \tag{2}$$

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

$$\rho_{\varrho} \in \{0, 1\} \qquad \forall\, \varrho \in \Omega \tag{4}$$

Appendix B 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 $\boldsymbol{A^{\varrho}}$ 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:

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

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

$$\bar{c}_\varrho = c_\varrho - \sum_{n\in\mathcal{N}} \tilde{\boldsymbol{s}}^{\boldsymbol{nw}}_{\boldsymbol{\varrho}} \pi^{(33)}_{nw} - \sum_{n\in\mathcal{N}} \sum_{d\in\mathcal{D}^w_k} \tilde{\boldsymbol{o}}^{\boldsymbol{nd}}_{\boldsymbol{\varrho}} \pi^{(32)}_{nd} - \sum_{k\in\mathcal{K}} \sum_{d\in\mathcal{D}^w_k} \tilde{\boldsymbol{u}}^{\boldsymbol{kd}}_{\boldsymbol{\varrho}} \pi^{(34)}_{kwd} - \sum_{p\in\mathcal{P}} \sum_{e\in\mathcal{E}} \tilde{\boldsymbol{q}}^{\boldsymbol{pe}}_{\boldsymbol{\varrho}} \pi^{(31)}_{pew}$$

$$= \sum_{(i,j)\in \boldsymbol{A}^{\boldsymbol{\varrho}}} \bar{c}_{ij} \tag{51}$$

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

$$\bar{c}_{ij} = \begin{cases} c_{ij} - \tilde{\boldsymbol{o}}^{\boldsymbol{nd}}_{\boldsymbol{\varrho}} \pi^{(32)}_{nd} - \pi^{(33)}_{nw} & \text{if } (i,j) \in \boldsymbol{A}^{\boldsymbol{u}}_{\boldsymbol{n}} \\ c_{ij} - \tilde{\boldsymbol{q}}^{\boldsymbol{pe}}_{\boldsymbol{\varrho}} \pi^{(31)}_{pew} & \text{if } (i,j) \in \boldsymbol{A}^{\boldsymbol{l}}_{\boldsymbol{pe}} \\ c_{ij} & \text{if } (i,j) \in \boldsymbol{A}^{\boldsymbol{o}} \end{cases} \tag{52}$$

The pricing subproblem is formulated as follows:

$$\min_{x,y,u} \sum_{k\in\mathcal{K}} \sum_{d\in\mathcal{D}_k} \left(\sum_{(i,j)\in\mathcal{A}} \bar{c}_{ij} x^{kd}_{ij} + \boldsymbol{\Psi}_k u_{kd} + \boldsymbol{\beta} \left(\boldsymbol{\Delta}_k u_{kd} - \sum_{l\in\mathcal{L}_k} \sum_{n\in\mathcal{N}} \boldsymbol{C}_{lk} y^{kd}_{ln} \right) \right) \tag{Pricing}$$

$$\text{s.t.:} \quad o^{kd}_n \le \hat{\boldsymbol{Q}}^d_n \sum_{(i,n)\in\mathcal{A}^{kd}} x^{kd}_{in} \qquad \forall\, n\in\mathcal{N}, \forall\, k\in\mathcal{K}, \forall\, d\in\mathcal{D}_k \tag{53}$$

$$\sum_{k\in\mathcal{K}} \boldsymbol{C}_{lk} y^{kd}_{ln} = o^{kd}_n \qquad \forall\, n\in\mathcal{N}, \forall\, k\in\mathcal{K}, \forall\, d\in\mathcal{D}_k \tag{54}$$

$$\sum_{(i,n)\in\mathcal{A}} \sum_{k\in\mathcal{K}} \sum_{d\in\mathcal{D}^w_k} x^{kd}_{in} \le 1 \qquad \forall\, n\in\mathcal{N}, \forall\, w\in\mathcal{W} \tag{55}$$

$$q^{kd}_{pe} \le \hat{\boldsymbol{Q}}_{pe} \sum_{i\in\{\mathcal{E}\setminus\{e\}\}\cup\{\sigma\}} x^{kd}_{ie} \qquad \forall\, p\in\mathcal{P}, \forall\, e\in\mathcal{E}, \forall\, k\in\mathcal{K}, \forall\, d\in\mathcal{D}_k \tag{56}$$

$$x^{kd}_{ij} \le u_{kd} \qquad \forall\, (i,j)\in\mathcal{A}, \forall\, k\in\mathcal{K}, \forall\, d\in\mathcal{D}_k \tag{57}$$

$$\sum_{n\in\mathcal{N}} y^{kd}_{ln} \le 1 \qquad \forall\, k\in\mathcal{K}, \forall\, l\in\mathcal{L}_k, \forall\, d\in\mathcal{D}_k \tag{58}$$

$$\sum_{e\in\mathcal{E}} q^{kd}_{pe} = \sum_{n\in\mathcal{N}_p} \sum_{l\in\mathcal{L}_k} \boldsymbol{C}_{lk} y^{kd}_{ln} \qquad \forall\, p\in\mathcal{P}, \forall\, k\in\mathcal{K}, \forall\, d\in\mathcal{D}_k \tag{59}$$

$$\sum_{(i,j)\in\mathcal{A}} \boldsymbol{\tau}_{ij} x^{kd}_{ij} \le \boldsymbol{T}_{max} \qquad \forall\, k\in\mathcal{K}, \forall\, d\in\mathcal{D}_k \tag{60}$$

$$\sum_{i\in\mathcal{V}} x^{kd}_{ij} - \sum_{i\in\mathcal{V}} x^{kd}_{ji} = \begin{cases} -1 & \text{if } j=\sigma \\ 0 & \text{if } j\in\mathcal{V}\setminus\{\sigma,\delta\} \\ 1 & \text{if } j=\delta \end{cases} \qquad \forall\, k\in\mathcal{K}, \forall\, d\in\mathcal{D}_k \tag{61}$$

$$\sum_{i,j\in\mathcal{S}} x^{kd}_{ij} \le |\mathcal{S}| - 1 \qquad \forall\, k\in\mathcal{K}, \forall\, d\in\mathcal{D}_k \ (\mathcal{S}\subset\mathcal{V}, 2\le|\mathcal{S}|\le|\mathcal{V}|-2) \tag{62}$$

$$x^{kd}_{ij} \in \{0,1\} \qquad \forall\, (i,j,k,d)\in\mathcal{C} \tag{63}$$

$$u_{kd} \in \{0,1\} \qquad \forall\, k\in\mathcal{K}, \forall\, d\in\mathcal{D}_k \tag{64}$$

$$y^{kd}_{ln} \in \{0,1\} \qquad \forall\, n\in\mathcal{N}, \forall\, k\in\mathcal{K}, \forall\, l\in\mathcal{L}_k, \forall\, d\in\mathcal{D}_k \tag{65}$$

$$q^{kd}_{pe} \ge 0 \qquad \forall\, k\in\mathcal{K}, \forall\, p\in\mathcal{P}, \forall\, e\in\mathcal{E}, \forall\, d\in\mathcal{D}_k \tag{66}$$

$$o^{kd}_n \ge 0 \qquad \forall\, n\in\mathcal{N}, \forall\, k\in\mathcal{K}, \forall\, d\in\mathcal{D}_k \tag{67}$$

References

Adulyasak, Y., Cordeau, J.-F., and Jans, R. (2014). Formulations and branch-and-cut algorithms for multivehicle production and inventory routing problems. INFORMS Journal on Computing, 26(1):103–120.

Androutsopoulos, K. N. and Karouti, E. (2022). A safety-driven truck loading problem. Operational Research, 22(5):4931–4963.

Archetti, C., Bertazzi, L., Hertz, A., and 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., and 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., and Grazia Speranza, M. (2017). A matheuristic for the multivehicle inventory routing problem. INFORMS Journal on Computing, 29(3):377–387.

Archetti, C., Jabali, O., and Speranza, M. G. (2015). Multi-period vehicle routing problem with due dates. Computers & Operations Research, 61:122–134.

Archetti, C. and Ljubić, I. (2022). Comparison of formulations for the inventory routing problem. European Journal of Operational Research, 303(3):997–1008.

Avella, P., Boccia, M., and 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., and Wolsey, L. A. (2018). Single-period cutting planes for inventory routing problems. Transportation Science, 52(3):497–508.

Bani, A., El Hallaoui, I., Corréa, A. I., and 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.

Benantar, A., Ouafi, R., and Boukachour, J. (2016). A petrol station replenishment problem: new variant and formulation. Logistics Research, 9(1):6.

Bertsimas, D., Kallus, N., and Hussain, A. (2016). Inventory management in the era of big data. Production and Operations Management, 25(12):2006–2009.

Boers, L., Atasoy, B., Correia, G., and 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.

Bortfeldt, A. and Wäscher, G. (2013). Constraints in container loading–a state-of-the-art review. European Journal of Operational Research, 229(1):1–20.

Brown, G. G. and Graves, G. W. (1981). Real-time dispatch of petroleum tank trucks. Management science, 27(1):19–32.

Carotenuto, P., Giordani, S., Massari, S., and 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.

Chitsaz, M., Cordeau, J.-F., and 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., and Laporte, G. (2012). Consistency in multi-vehicle inventory-routing. Transportation Research Part C: Emerging Technologies, 24:270–287.

Coelho, L. C., Cordeau, J.-F., and Laporte, G. (2014). Thirty years of inventory routing. Transportation Science, 48(1):1–19.

Coelho, L. C. and Laporte, G. (2013). The exact solution of several classes of inventory-routing problems. Computers & Operations Research, 40(2):558–565.

Coelho, L. C. and Laporte, G. (2015). Classification, models and exact algorithms for multi-compartment delivery problems. European Journal of Operational Research, 242(3):854–864.

Coin-Or (2023). Branch-Cut-Price Framework. https://projects.coin-or.org/Bcp. Last accessed 2023-11-15.

Cordeau, J.-F., Stojković, G., Soumis, F., and Desrosiers, J. (2001). Benders decomposition for simultaneous aircraft routing and crew scheduling. Transportation Science, 35(4):375–388.

Cornillier, F., Boctor, F., and 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., and 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., and 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., and 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., and Zhang, L. (2023). The inventory routing problem under uncertainty. Operations Research, 71(1):378–395.

Dantzig, G. B. and Ramser, J. H. (1959). The truck dispatching problem. Management science, 6(1):80–91.

Desaulniers, G., Desrosiers, J., and Solomon, M. M. (2006). Column generation, volume 5. Springer Science & Business Media.

Desaulniers, G., Rakke, J. G., and Coelho, L. C. (2016). A branch-price-and-cut algorithm for the inventory-routing problem. Transportation Science, 50(3):1060–1076.

Er Raqabi, E. M., El Hallaoui, I., and Soumis, F. (2023). The Primal Benders Decomposition. Les Cahiers du GERAD, G–2023–27.

Fischetti, M., Ljubić, I., and Sinnl, M. (2017). Redesigning benders decomposition for large-scale facility location. Management Science, 63(7):2146–2162.

Fischetti, M. and Lodi, A. (2003). Local branching. Mathematical Programming, 98:23–47.

Hansen, P. and Mladenović, N. (2001). Variable neighborhood search: Principles and applications. European Journal of Operational Research, 130(3):449–467.

Joncour, C., Michel, S., Sadykov, R., Sverdlov, D., and Vanderbeck, F. (2010). Column generation based primal heuristics. Electronic Notes in Discrete Mathematics, 36:695–702.

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

Manousakis, E., Repoussis, P., Zachariadis, E., and 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.

Ostermeier, M., Henke, T., Hübner, A., and 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., and Radivojević, G. (2011). Simulation approach to analyse deterministic irp solution of the stochastic fuel delivery problem. Procedia-Social and Behavioral Sciences, 20:273–282.

Rahmaniani, R., Crainic, T. G., Gendreau, M., and Rei, W. (2017). The benders decomposition algorithm: A literature review. European Journal of Operational Research, 259(3):801–817.

Reed, S., Campbell, A. M., and Thomas, B. W. (2022). The value of autonomous vehicles for last-mile deliveries in urban environments. Management Science, 68(1):280–299.

Triki, C. (2013). Solution methods for the periodic petrol station replenishment problem. The Journal of Engineering Research [TJER], 10(2):69–77.

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.

Vidović, M., Popović, D., and 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., and 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., and Wang, S. (2022). Vessel service planning in seaports. Operations Research, 70(4):2032–2053.

Zeighami, V. and Soumis, F. (2019). Combining benders' decomposition and column generation for integrated crew pairing and personalized crew assignment problems. Transportation Science, 53(5):1479–1499.