Transportation Research Part E 205 (2026) 104491



Contents lists available at ScienceDirect

Transportation Research Part E

journal homepage: www.elsevier.com/locate/tre





Inventory routing with heterogeneous vehicles and hazardous material backhauling

Farzad Avishan [a,b,*], Amira Dems [c], Yossiri Adulyasak [a], Okan Arslan [a], Jean-François Cordeau [a]

[a] HEC Montréal, 3000 chemin de la Côte-Sainte-Catherine, Montréal H3T 2A7, Canada
[b] Özyeğin University, 34794 Çekmeköy, Istanbul, Türkiye
[c] Hydro-Québec Research Institute (IREQ), 1800 Bd Lionel-Boulet, Varennes J3X 1S1, Canada

ARTICLE INFO

Keywords:
Inventory routing problem
Backhauling
Branch-and-cut
Matheuristic

ABSTRACT

Efficient coordination of distribution and backhauling is a critical challenge for many industries. This paper is motivated by a real-world case study at Hydro-Québec, a large-scale utility company in North America, and introduces an inventory routing problem that integrates inventory management and vehicle routing under several operational constraints. The problem involves distributing multiple commodities to customer sites while backhauling hazardous materials to depots. The objective is to minimize delivery, collection, and inventory holding costs using a fleet of capacitated heterogeneous vehicles, while ensuring that hazardous materials are transported separately from regular delivery commodities. In each period, a customer's delivery and backhauling can be split and satisfied by multiple vehicles. We propose a mathematical formulation, introduce valid inequalities, and solve the resulting model using a branch-and-cut algorithm. To tackle large-size instances, a two-phase decomposition matheuristic is developed. To highlight the value of split delivery and backhauling, we compare the solutions from our model with those when split delivery is prohibited and when backhauling is optimized independently. In addition, we investigate the order-up-to level policy and the case when stockout is allowed. An extensive numerical study is conducted on synthetic instances to evaluate the performance of the models and solution approaches. The heuristic algorithm solves the synthetic instances in less than two hours with an average optimality gap of less than 2 %. Finally, a case study is conducted on the Hydro-Québec network to demonstrate the real-world applicability of the model and quantify the benefits to the company. Our proposed model reduces the total routing costs by 21 % compared to the case where backhauling is not integrated and split delivery is not allowed.

1. Introduction

Supply chain management is increasingly recognized as fundamental to survival and competitiveness. Business leaders acknowledge that no company can succeed in isolation, but rather depends on strong relationships with both suppliers and customers (Ross, 2010). Within supply chains, transportation and inventory operations often account for a substantial portion of total costs, and poor coordination between them can lead to inefficiencies, logistical and operational challenges, delays, and higher expenses. This

* Corresponding author.
E-mail addresses: farzad.avishan@ozu.edu.tr (F. Avishan), dems.amira@hydroquebec.com (A. Dems), yossiri.adulyasak@hec.ca (Y. Adulyasak), okan.arslan@hec.ca (O. Arslan), jean-francois.cordeau@hec.ca (J.-F. Cordeau).

https://doi.org/10.1016/j.tre.2025.104491
Received 27 November 2024; Received in revised form 1 October 2025; Accepted 2 October 2025
Available online 8 November 2025
1366-5545/© 2025 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

highlights the need for optimization models that jointly consider inventory and routing decisions while incorporating realistic operational constraints. A typical supply chain requires managing production, inventory, and distribution of finished products to multiple customers through various routes, with the primary objective often being cost minimization. When production quantities at the plant are fixed and decisions focus on inventory management and vehicle routing, the resulting challenge is known as the *inventory routing problem* (IRP) (Campbell et al., 1998; Chitsaz et al., 2019). The IRP has received considerable attention in the research community because of the economic benefits associated with integrating transportation and inventory management. At the same time, it remains a challenging optimization problem due to its inherent complexity (Coelho et al., 2014; Archetti and Ljubić, 2022). The basic IRP involves distributing a single product from a supply facility, referred to as a depot, to a group of customers using a fleet of identical vehicles. Products are stocked at the depot, and each customer has a storage capacity and a known demand. The objective of the IRP is to minimize total transportation and holding costs throughout a planning horizon while ensuring that none of the customers experience stockouts.

From a practical standpoint, the basic IRP involves numerous simplifications and limiting assumptions. These include the presence of a single supply or production site, a single product with unlimited production or supply capacity, identical vehicles and unlimited storage capacity at the production site. While these assumptions may be valid in certain cases, they are usually too restrictive in more complex practical settings and may lead to solutions that lack the flexibility and responsiveness necessary for efficient supply chain management. Moreover, the basic IRP disregards the total travel time limit, which is an important constraint in practice. More specifically, a delivery time limit is essential to ensure that products arrive on time to meet consumer demand and that the vehicles can return to their depot before the end of each period in the planning horizon. The time limit for vehicles becomes more important when products for delivery are perishable and must be kept fresh, as they cannot remain in the vehicle for too long. Split delivery instead of consolidated delivery is another characteristic of real problems that is often necessary to optimize vehicle capacity and reduce transportation costs (Dinh et al., 2023).

While formulations of the IRP primarily focus on distributing commodities and minimizing the associated costs, incorporating pickups and backhauling into these models can be beneficial in many cases. To this end, the IRP *with pickup and delivery* (IRP-PD) is defined as a problem that focuses on direct transfers between various locations and involves collecting commodities to move them between pickup and delivery points or returning them to a central warehouse (Archetti et al., 2020). Backhauling, on the other hand, takes advantage of vehicle return trips by transporting materials instead of returning empty, enhancing efficiency and reducing costs. Ignoring backhauling can lead to inefficiencies, including increased operational costs and environmental impacts, such as higher greenhouse gas emissions. Backhauling is also important for networks where the products to be delivered cannot be transported simultaneously in a vehicle with the materials to be picked up due to incompatibility and potential hazards. For example, in the food industry, products should not be transported with toxic chemicals because of the risk of contamination, and if a vehicle plans to collect spoiled foods, they cannot simultaneously be in a vehicle that is delivering the foods. It is also the same for electronic products that should not be transported with flammable liquids due to fire risk, textiles transported alongside explosives pose significant safety hazards, and household goods that should not be mixed with radioactive materials to prevent contamination. Another practical application of this concept can be seen in the grocery sector, where supermarkets and retail stores often serve as customers for delivery, while grocery suppliers function as backhaul customers (Toth and Vigo, 2002; Koç and Laporte, 2018).

Inspired by an industrial application at *Hydro-Québec* (HQ), we study a distribution network where pickups are permitted only after deliveries. This restriction arises because all backhauled materials are hazardous and cannot be transported together with commodities. In particular, HQ backhauls transformer oils for cleaning, reuse, or disposal, and these pickups must be carried out exclusively. In other words, delivery commodities should not be on the vehicle at the same time as the backhauled materials. Fig. 1 illustrates the delivery and backhauling network. Vehicles can start either delivering or backhauling after leaving a depot, but they cannot carry deliveries and backhauled materials simultaneously. Hence, routing decisions must properly sequence these operations. Additionally, a vehicle must be empty when starting backhauling, and performing deliveries after backhauling is not permitted. We aim to coordinate inventory levels and vehicle routes across multiple depots, commodities, and time periods to meet demand, while minimizing the transportation and holding costs, and ensuring operational feasibility under multiple interacting constraints.

In this study, we consider an IRP with *delivery and backhauling* (IRP-DB) while investigating the advantages of integrating delivery and backhauling. We address practical complexities and real-world challenges of the IRP, such as split delivery and heterogeneous vehicles. To this end, we propose a mathematical model for the IRP-DB that incorporates *multiple depot and multiple commodity* (MDC), referred to as the IRP-DB-MDC. Each vehicle begins its journey from a depot during a specific time period and returns to the same depot at the end of the same time period, while ensuring that the route length does not exceed the available time. Different extended models are also formulated to facilitate comprehensive comparisons of our solutions with similar approaches in the literature. These extended models consider different replenishment policies and disjoint backhauling and delivery operations while avoiding split delivery. We employ a *branch-and-cut* (B&C) algorithm that effectively solves the problem, demonstrating improved solution time and quality compared to solving the *mixed-integer programming* (MIP) model directly by a general-purpose solver. The proposed algorithms use valid inequalities to strengthen the formulation. Additionally, we develop a matheuristic approach to handle large-size instances where exact methods fail to provide efficient solutions. The effectiveness of the model and solution approaches is also tested through their application to synthetic and case study instances. These numerical experiments demonstrate the practical utility and competitive efficiency of the proposed models and solution approaches in complex and realistic scenarios, offering valuable insights. The matheuristic approach solves problems with up to 100 sites in less than two hours, achieving an average optimality gap of 3 % for instances with 100 sites and less than 2 % on average across all instances. Furthermore, numerical instances show that the proposed model can significantly reduce the total cost compared to extended models. In our case study, this improvement is particularly notable for routing costs, with a 21 % reduction by considering integrated delivery and backhauling with split delivery,

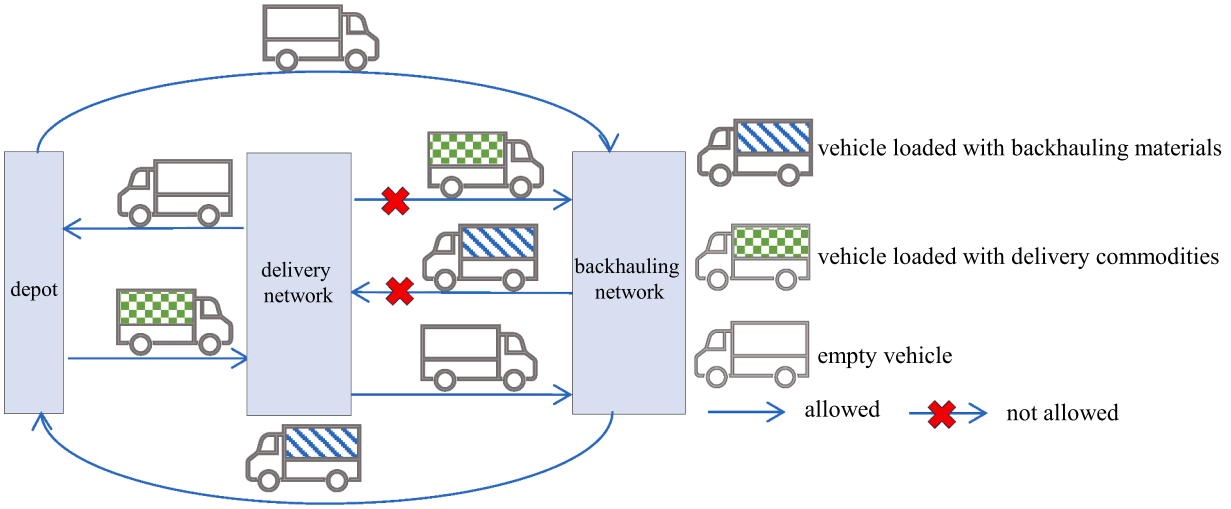

Fig. 1. The delivery and backhauling network (IRP-DB).

highlighting the model's efficiency in cost minimization. We now review the relevant literature and discuss our contributions in detail in the next section.

2. Literature review

Since the introduction of the IRP by Bell et al. (1983), many variants have been developed and formulated. These include the perishable IRP (Shaabani, 2022); stochastic IRP (Li et al., 2023; Feng et al., 2024; Ortega et al., 2024); IRP with time windows (Tiniç et al., 2021); sustainable and green IRP (Soysal et al., 2019; Rau et al., 2018); the multi-product IRP (Coelho and Laporte, 2013a; Neves-Moreira et al., 2019); and the IRP with a heterogeneous fleet (Cheng et al., 2017; Hewitt et al., 2013). The IRP has also been extended to include deliveries and pickups, where vehicles collect excess inventory from some customers and deliver it to others in need, ensuring efficient inventory management and distribution. This extension addresses various logistical challenges, but few contributions are made in this field. Archetti et al. (2018) have studied a single product IRP-PD with a single vehicle and extended the formulation to multi-vehicle cases (Archetti et al., 2020). Both of the papers have employed B&C algorithms to solve the problem. One of the main applications of IRP-PD is in maritime transportation, especially for transporting liquid and bulk materials. Due to the large size and capacity of ships, the significant quantities transported, and the long distances and travel times involved, it is more efficient to plan routes that combine pickup and delivery operations at ports rather than schedule them separately. Grønhaug et al. (2010) study maritime IRP in the liquefied natural gas business, where the natural gas is stored at designated pickup (loading) ports and is transported via tankers to storage facilities at delivery (unloading) ports. In a similar context with different applications, Van Anholt et al. (2016) studied a multiperiod IRP involving both pickups and deliveries motivated by ATM replenishment. Their approach allows for the dynamic transportation of commodities between the depot and customers, with customers also able to exchange goods as demand fluctuates. Lei et al. (2024) have further expanded the application of IRP-PD in a different domain and have applied IRP-PD to the collection, disassembly, and delivery problems, formulating a scenario where products are collected from centers, disassembled into materials, and delivered to remanufacturing plants. The IRP can also be categorized based on inventory replenishment policies that dictate delivery quantities. The most common policies are the *order up to level* (OU) and *maximum level* (ML) policies. The OU policy ensures that the inventory is fully stocked up to its *target stock level* (TSL) if visited, while the ML policy allows flexible delivery quantities if stock capacities are not exceeded (Mahmutoğulları and Yaman, 2023). Adulyasak et al. (2014) study a multi-vehicle IRP and address the problems considering the ML and OU inventory replenishment policies, and Dinh et al. (2023) examine the advantages of incorporating split deliveries into the problem under both ML and OU replenishment strategies. Agra et al. (2022) study an IRP in which each visit has specified minimum and maximum pickup or delivery quantities. In addition to these policies, Diabat et al. (2024) have studied the *zero inventory ordering* (ZIO) policy, where replenishment occurs only when the customer's inventory level reaches zero. This policy is more customer-oriented than the ML policy. The authors analyze its advantages for customers and demonstrate how the ZIO policy can reduce customers' inventory costs compared to the ML policy while discussing potential disadvantages for the system. Chiu et al. (2024) examine how incorporating safety stocks can improve the long-term performance of the rolling horizon strategy in IRP. Additionally, the study explores the impact of establishing minimum inventory levels for the final period of the planning horizon.

The IRP is a mathematically challenging problem due to a combination of routing and inventory management decisions, and researchers have used different exact and heuristic algorithms to solve the problem (Archetti and Ljubić, 2022). The exact methods devised for the IRP are primarily categorized into *branch-cut-and-price* (BC&P) algorithms and B&C algorithms. Although BC&P

Table 1
Summary of literature in the field of IRP-PD and IRP-DB.

Reference paper	Multi depots	Multi products	Time limits	Split delivery	Heterogeneous vehicles	PD/DB	Replenish policy	Solution approach
Ramkumar et al. (2012)	✓	✓	✓	✓	-	PD	ML, Min.Inv.	No method
Van Anholt et al. (2016)	✓	–	–	✓	–	PD	ML	B&C, heuristic
Iassinovskaia et al. (2017)	–	–	✓	–	–	PD	ML	B&C, heuristic
Archetti et al. (2018)	–	–	–	–	–	PD	ML	B&C
Archetti et al. (2020)	–	–	–	–	–	PD	ML, Min.Inv.	B&C
Arab et al. (2020)	–	✓	–	–	✓	DB	ML	heuristics
Agra et al. (2022)	–	–	–	✓	–	PD	OU	B&C
Neves-Moreira et al. (2022)	–	✓	–	–	✓	PD	ML	B&C, heuristic
Londoño et al. (2023)	–	–	–	–	–	DB	–	–
A. Guimarães et al. (2019)	✓	–	–	–	–	PD	ML, OU	B&C, heuristic
Neves-Moreira et al. (2022)	–	✓	–	–	✓	PD	ML	B&C, heuristic
Soysal (2016)	–	✓	–	✓	–	PD	–	–
Mahjoob et al. (2021)	–	–	–	✓	✓	PD	ML	–
Golsefidi and Jokar (2020)	–	–	–	–	–	PD	ML	heuristic
Savelsbergh and Song (2008)	–	–	–	–	–	PD	ML	B&C, heuristic
Yavrucu et al. (2025)	–	✓	–	✓	–	PD	ML	heuristic
Zhang et al. (2024)	–	–	–	–	–	PD	ML	heuristic
This research	✓	✓	✓	✓	✓	DB	ML, OU	B&C, heuristic

B&C: branch-and-cut algorithm; PD: pickup and delivery, DB: delivery and backhauling; ML: maximum level, OU: order up to level Min.Inv.: minimum level inventory

algorithms are used in some studies (Desaulniers et al., 2016), B&C algorithms have been the most widely used exact methods for various IRP variants, including those with different policies, split deliveries, and product substitution (Skålnes et al., 2024). The paper of Archetti et al. (2007) is one of the first to propose an exact IRP approach for the single-vehicle case. The literature on developing solution approaches has advanced and aims to solve more complex problems with multiple vehicles. Coelho and Laporte (2013b) propose a B&C algorithm as an exact solution approach for several classes of IRPs. Adulyasak et al. (2014), Archetti et al. (2017) and Mahmutoğulları and Yaman (2023) have taken important steps to improve the B&C algorithms for different variants of problems. A few studies also used Benders decomposition to solve the IRP exactly. Adulyasak et al. (2015) applied Benders decomposition to a production planning problem, which is a generalization of IRP that incorporates production decisions. They account for demand uncertainty across a large scenario set and introduce several computational enhancements for Benders decomposition. Zhou (2024) also employed Benders decomposition to solve the IRP, though the largest instance they solved comprises only 15 nodes.

Due to the complexity of the problem, heuristic algorithms have also been developed to solve it. Le et al. (2013) have developed a heuristic algorithm for an IRP involving perishable goods based on column generation. Absi et al. (2015) and Chitsaz et al. (2019) have proposed an iterative algorithm to solve the problem, and other researchers have developed new algorithms to get more efficient solutions. Skålnes et al. (2023) developed a heuristic that employs a B&C embedded matheuristic, in which the matheuristic is invoked each time a new primal solution is obtained during the B&C. The authors have conducted a comprehensive review of well-established heuristic algorithms in the literature for the IRP and have compared the performance of their proposed algorithm with these existing approaches. Several studies proposed both exact and heuristic algorithms to solve the IRP. For instance, Bertazzi et al. (2019) focus on the multi-depot IRP and present both a B&C algorithm and a three-phase matheuristic designed specifically to handle realistic-size instances of the problem. There are studies that have applied a decomposition-based heuristic to solve IRP. Lefever et al. (2021) consider an IRP with travel time uncertainty. The problem is decomposed into a master problem, which generates a route, and a subproblem, which evaluates its robustness. If the resulting route is not robust, the subproblem produces a cut that is added to the master problem. Wang and Chu (2022) propose a decomposition-based heuristic algorithm in which the master problem determines retailers' replenishments, and the subproblem solves the routing decision for each period. For more information on heuristic algorithms for IRP, we refer the readers to Andersson et al. (2010) and to Sofianopoulou and Mitsopoulos (2021).

A summary of recent studies on IRP-PD and IRP-DB is presented in Table 1, indicating whether they consider key problem features such as multi-depot, multi-product, time limits, split deliveries, and heterogeneous vehicles. The table also details each study's replenishment policy and solution approach. Since IRP variants that consider only delivery operations fall outside the scope of this work, the table focuses exclusively on how our research differs from the IRP-DB and IRP-PD formulations. For comprehensive reviews of the IRP, we refer readers to Bertazzi and Speranza (2012), Bertazzi and Speranza (2013), and Coelho et al. (2014). Considering different features of real-world applications is crucial to developing IRP solution methods that are broadly applicable. However, most existing methods for the IRP are developed for basic problem settings, limiting their practical applicability (Song and Furman, 2013). Considering the aforementioned research studies, our paper addresses the gaps in the literature by proposing a comprehensive IRP-DB problem and solution approach that integrates multiple realistic features, including multiple depots, heterogeneous vehicles, time limits, split deliveries, backhauling, and alternative replenishment policies.

As presented in Table 1, no single model in the literature simultaneously incorporates all the key features of the IRP. Only two studies consider time limits on tours, and only three address multiple depots. In our study, we propose a model for the IRP-DB that incorporates multiple depots, multiple commodities, a maximum travel time for each route, and split deliveries. The fleet consists of

heterogeneous vehicles with varying capacities and costs, and each vehicle is allowed to perform at most one tour per time period. Another key distinction between the basic IRP and the model presented in this study is the inclusion of backhauling trips, which optimize vehicle capacity utilization after deliveries are completed. The integration of backhauling with deliveries is well-studied in the context of the *vehicle routing problem* (VRP), known as *VRP with backhauling* (VRP-B) and has been extensively explored in the literature. Deif and Bodin (1984) were the first to introduce VRP-B and extended the Clarke and Wright (1964) algorithm to address it. Despite the extensive research on VRP-B, the specific case of IRP, where vehicles perform backhauling tasks within the framework of inventory routing, remains underexplored. This is also seen in Table 1, where we observe that the IRP-PD has been commonly studied in the literature, while the IRP-DB is less studied, with only a few studies addressing this particular problem. Arab et al. (2020) approached the problem by considering a single depot without imposing any time limits on the tours, and they proposed population heuristic algorithms that could solve instances with up to 40 sites. Londoño et al. (2023) formulated a model for identical vehicles while ignoring time constraints, with their computational experiments limited to instances involving a maximum of 15 customers. Additionally, Table 1 shows that the majority of research on IRP with backhauling or pickup focuses on the ML replenishment policy. In contrast, we provide formulations and numerical experiments for both ML and OU replenishment policies. Finally, we address the problem using both exact and matheuristic algorithms, which are evaluated through extensive numerical experiments to assess their efficiency and scalability. The benchmark instances are generated for IRP-DB-MDC with heterogeneous vehicles based on the well-established IRP instances by Archetti et al. (2007). We implement our model in a real-world case study, which demonstrates its practical relevance and highlights the effectiveness of our approach in optimizing logistics management for our industrial partner by investigating the impact of split deliveries, integrated delivery and backhauling, and different replenishment policies.

The main contributions of this paper can be summarized as follows: 1) We present the IRP-DB-MDC problem that simultaneously incorporates multiple depots, multiple commodities, a maximum travel time per route, split deliveries, and a heterogeneous fleet; 2) This problem integrates backhauling trips with deliveries, addressing an underexplored case of inventory routing with backhauling within the IRP framework; 3) We extend the analysis to both ML and OU replenishment policies; 4) We solve the problem using both exact and matheuristic algorithms, validated through extensive numerical experiments and a real-world case study, demonstrating the practical relevance and effectiveness of our approach.

The subsequent sections of the paper are organized as follows. Section 3 defines the problem and presents the optimization model. Section 4 outlines the solution approach, including the exact and matheuristic algorithms used to solve the problem. Section 5 discusses the computational results and the case study. Finally, conclusions and managerial implications are presented in Section 6.

3. Problem definition and mathematical formulation

In this section, we present and model the IRP-DB-MDC. We first describe the problem in Section 3.1 and introduce a mathematical model. We then provide extended models in Section 3.2. Throughout this section, brackets refer to the mathematical model of a problem. For example, IRP-DB-MDC refers to the problem itself, while [IRP-DB-MDC] refers to its formulation.

3.1. Problem definition

The IRP-DB-MDC can be defined on a complete undirected graph $G = (M \cup N \cup N', E)$, where $M = \{d_1, d_2, \dots, d_{\mathcal{M}}\}$ represents the set of depots serving as the origin and destination points for vehicle routes, $N = \{1, 2, \dots, \mathcal{N}\}$ denotes the set of sites for product deliveries, and $N' = \{\mathcal{N}+1, \mathcal{N}+2, \dots, \mathcal{N}'\}$ denotes the set of the sites for backhauling material pickups. If a site has both delivery and backhauling demands, it is considered as two separate sites. The set of edges, denoted by E, is defined as $E = \{(i,j) \mid i \in M, j \in N \cup N'\} \cup E_{N,N'} \cup E_N \cup E_{N'}$, where $E_{N,N'} = \{(i,j) \mid i \in N, j \in N'\}$, $E_N = \{(i,j) \mid i \in N, j \in N, i < j\}$, $E_{N'} = \{(i,j) \mid i \in N', j \in N', i < j\}$. Let $E(S)$ be the set of edges $(i,j) \in E$ such that $i, j \in S$, where $S \subseteq N \cup N'$. Additionally, $\delta(i)$ is defined as the set of incident edges to site $i \in N \cup N' \cup M$, such that $\delta(i) = \{(a,b) \mid a, b \in N \cup N' \cup M, (a,i) \in E \text{ or } (i,b) \in E\}$. The set of time periods $T = \{1, 2, \dots, \mathcal{T}\}$ denotes the planning horizon and the sets of commodities to deliver and materials to backhaul are denoted by $P = \{1, 2, \dots, \mathcal{P}\}$ and $P' = \{\mathcal{P}+1, \mathcal{P}+2, \dots, \mathcal{P}'\}$, respectively. The quantity of commodity $p \in P$ to be delivered to site $i \in N$ in period $t \in T$ is represented by d_{ipt}. Similarly, the quantity of backhauled material of type $p \in P'$ collected from site $i \in N'$ during period $t \in T$ is represented by g_{ipt}.

To formulate our problem, we define a network that encompasses both the delivery process, which is responsible for transporting commodities from depots to sites and the backhauling process, which is responsible for collecting materials. A fleet of heterogeneous vehicles $K = \{1, 2, \dots, \mathcal{K}\}$ with transportation costs and times denoted by c_{ijk} and t_{ijk} for $(i,j) \in E$, respectively, and a capacity of b_k, depart from depots to deliver commodities to sites to fulfill their demand or backhaul hazardous materials from sites to depots. Additionally, we define $K_i \subseteq K$ as the set of vehicles located at depot $i \in M$. The route of each vehicle starts and ends at the predetermined depot. The interplay between the inventory level and demand at sites and the vehicle capacity determines the number of vehicles that will visit a site within a given period, i.e., more than one vehicle may visit a site for delivery and backhauling during a period. A vehicle cannot carry commodities and backhauling materials concurrently. Therefore, vehicles may only start backhauling after completing commodity deliveries, or they should immediately start collecting backhauling materials after leaving the depot without making any delivery en route. The total amount of supplied commodities and backhauled materials in each period is individually limited to the depot capacities, denoted by l_{ip} and l'_{ip}, respectively. Furthermore, inventory can be stored at sites, but the inventory level must not exceed the capacity o_i at site $i \in N \cup N'$. The unit cost of holding inventory of product p at site i is denoted by h_{ip}. The summary of notations is given in Table 2.

Table 2
Summary of all notation.

Sets	
P	Set of delivery commodities (indexed by $p = 1, 2, \dots, \mathcal{P}$)
P'	Set of backhauling materials (indexed by $p = \mathcal{P}+1, \mathcal{P}+2, \dots, \mathcal{P}'$)
T	Set of time periods (indexed by $t = 1, 2, \dots, \mathcal{T}$)
N	Set of sites for delivery (indexed by $i, i', j = 1, 2, \dots, \mathcal{N}$)
N'	Set of sites for backhauling (indexed by $i, i', j = \mathcal{N}+1, \mathcal{N}+2, \dots, \mathcal{N}'$)
M	Set of depots (indexed by $i, j = d_1, d_2, \dots, d_{\mathcal{M}}$)
K	Set of vehicles (indexed by $k = 1, 2, \dots, \mathcal{K}$); i_k is the depot where vehicle k is located
K_i	Set of vehicles that are located at depot $i \in M, K_i \subseteq K$
E	Set of edges $\{(i,j) \mid i \in M, j \in N \cup N'\} \cup E_{N,N'} \cup E_N \cup E_{N'}$, where $E_{N,N'} = \{(i,j) \mid i \in N, j \in N'\}$, $E_N = \{(i,j) \mid i \in N, j \in N, i < j\}$, $E_{N'} = \{(i,j) \mid i \in N', j \in N', i < j\}$
$E(S)$	Set of edges $(i,j) \in E$ such that $i, j \in S$, where $S \subseteq N \cup N'$ is a given set of sites
$\delta(i)$	Set of incident edges to site $i \in N \cup N' \cup M$, $\{(a,b) \mid a, b \in N \cup N' \cup M, (a,b) \in E, a = i \text{ or } b = i\}$
Parameters	
c_{ijk}	Transportation cost of edge $(i,j) \in E$ with vehicle $k \in K$
t_{ijk}	Travel time of edge $(i,j) \in E$ with vehicle $k \in K$
$T_{\max}$	Available time for each vehicle in each period
d_{ipt}	Demand at site $i \in N$ for commodity $p \in P$ in period $t \in T$
g_{ipt}	Backhauling demand of material type $p \in P'$ at site $i \in N'$ in period $t \in T$
q_{ip}	Storage capacity of site $i \in N$ for commodity $p \in P$ ($i \in N'$ for backhauling material $p \in P'$)
q'_{ip}	TSL of site $i \in N$ for commodity $p \in P$
o_i	Global storage capacity of site $i \in N \cup N'$
a_{kp}	Capacity of vehicle $k \in K$ for commodity or backhauling material $p \in P \cup P'$
b_k	Global capacity of vehicle $k \in K$
h_{ip}	Unit holding cost of commodity $p \in P$ at site $i \in N$ ($i \in N'$ for backhauling material $p \in P'$)
e_{ip}	Unit stock out cost of commodity $p \in P$ at site $i \in N$
l_{ip}	Maximum supply of commodity $p \in P$ at depot $i \in M$ at the beginning of each period
l'_{ip}	Maximum amount of backhauling material $p \in P'$ that can be accumulated at each depot $i \in M$
f_{ik}	Binary parameter equal to 1, if vehicle $k \in K$ is located in depot $i \in M$, and 0 otherwise
s_{ip0}	Initial inventory level of commodity $p \in P$ at site $i \in N$ (backhauling material $p \in P'$ at site $i \in N'$)
D_{ikt}	Maximum commodity that can be delivered to site $i \in N$ by vehicle $k \in K$ in period $t \in T$ (maximum backhauling material that can be picked up from site $i \in N'$ by vehicle $k \in K$ in period $t \in T$) $D_{ikt} = \min\left\{o_i, b_k, \sum_{p \in P} \sum_{t'=t}^{\mathcal{T}} d_{ipt'}\right\} \quad i \in N, k \in K, t \in T$ $D'_{ikt} = \min\left\{o_i, b_k, \sum_{p \in P'} \left(s_{ip0} + \sum_{t'=1}^{t} g_{ipt'}\right)\right\} \quad i \in N', k \in K, t \in T$
Decision variables	
s_{ipt}	Inventory level of commodity $p \in P$ at site $i \in N$ in period $t \in T$ (backhauling material $p \in P'$ at site $i \in N'$)
u_{ipkt}	Quantity of commodity $p \in P$ delivered to site $i \in N$ in period $t \in T$ with vehicle $k \in K$ (backhauling material $p \in P'$ picked up from site $i \in N'$)
x_{ijkt}	Binary variable equal to 1 if vehicle $k \in K$ traverses directly between edges $i, j \in E$ in period $t \in T$, and 0 otherwise
z_{ikt}	Binary variable equal to 1 if site or depot $i \in N \cup N' \cup M$ is visited by vehicle $k \in K$ in period $t \in T$, and 0 otherwise
y_{ipt}	Binary variable equal to 1 if site $i \in N$ is visited in period $t \in T$ to deliver commodity $p \in P$ (site $i \in N'$ is visited by vehicle to backhaul material $p \in P'$)

The following decision variables are used to formulate the IRP-DB-MDC. Variable s_{ipt} represents the inventory level of delivery commodity as backhauling material p at site i. The amount of commodity or material of type p delivered or collected from site i by vehicle k in period t is denoted by u_{ipkt}. Binary decision variable x_{ijkt} equals one if vehicle k traverses edge (i,j) in period t, and 0 otherwise, while z_{ikt} equals one if site i is visited by vehicle k in period t. Fig. 1 illustrates the delivery and backhauling network. The sequence entails the delivery of commodities to sites, followed by the backhauling of materials to depots. It is worth mentioning that a vehicle can directly begin backhauling after leaving a depot, and delivery is not a prerequisite for initiating backhauling operations. All quantities in this formulation share the same units and can be expressed in pallets to facilitate calculations.

The problem can be formulated as a generalization of the formulations presented in Adulyasak et al. (2014) and in Archetti et al. (2018).

[IRP-DB-MDC] :

$$\min \sum_{t \in T} \left(\sum_{k \in K} \sum_{(i,j) \in E} c_{ijk} x_{ijkt} + \sum_{i \in N} \sum_{p \in P} h_{ip} s_{ipt} + \sum_{i \in N'} \sum_{p \in P'} h_{ip} s_{ipt} \right)$$

$$\text{s.t.} \sum_{(i,j) \in E} t_{ijk} x_{ijkt} \le T_{\max} \qquad k \in K, t \in T \qquad (1)$$

$$s_{ip(t-1)} + \sum_{k \in K} u_{ipkt} = d_{ipt} + s_{ipt} \qquad i \in N, p \in P, t \in T \qquad (2)$$

$$s_{ip(t-1)} + g_{ipt} = \sum_{k \in K} u_{ipkt} + s_{ipt} \qquad i \in N', p \in P', t \in T \qquad (3)$$

$$\sum_{p\in P}\left(s_{ip(t-1)}+\sum_{k\in K}u_{ipkt}\right)\le o_i \qquad i\in N, t\in T \qquad (4)$$

$$\sum_{p\in P'}\left(s_{ip(t-1)}+g_{ipt}\right)\le o_i \qquad i\in N', t\in T \qquad (5)$$

$$\sum_{i\in N}\sum_{p\in P}u_{ipkt}\le b_k\sum_{i\in M}z_{ikt} \qquad k\in K, t\in T \qquad (6)$$

$$\sum_{i\in N'}\sum_{p\in P'}u_{ipkt}\le b_k\sum_{i\in M}z_{ikt} \qquad k\in K, t\in T \qquad (7)$$

$$s_{ip(t-1)}+\sum_{k\in K}u_{ipkt}\le q_{ip} \qquad i\in N, p\in P, t\in T \qquad (8)$$

$$s_{ip(t-1)}+g_{ipt}\le q_{ip} \qquad i\in N', p\in P', t\in T \qquad (9)$$

$$\sum_{i\in N}u_{ipkt}\le a_{kp}\sum_{i\in M}z_{ikt} \qquad p\in P, k\in K, t\in T \qquad (10)$$

$$\sum_{i\in N'}u_{ipkt}\le a_{kp}\sum_{i\in M}z_{ikt} \qquad p\in P', k\in K, t\in T \qquad (11)$$

$$\sum_{i'\in N}\sum_{k\in K_i}u_{i'pkt}\le l_{ip} \qquad i\in M, p\in P, t\in T \qquad (12)$$

$$\sum_{i'\in N}\sum_{k\in K_i}u_{i'pkt}\le l'_{ip} \qquad i\in M, p\in P', t\in T \qquad (13)$$

$$\sum_{p\in P}u_{ipkt}\le D_{ikt}z_{ikt} \qquad i\in N, k\in K, t\in T \qquad (14)$$

$$\sum_{p\in P'}u_{ipkt}\le D'_{ikt}z_{ikt} \qquad i\in N', k\in K, t\in T \qquad (15)$$

$$z_{ikt}\le f_{ik} \qquad i\in M, k\in K, t\in T \qquad (16)$$

$$\sum_{(i,i')\in\delta(j)}x_{ii'kt}=2z_{jkt} \qquad j\in N\cup N'\cup M, k\in K, t\in T \qquad (17)$$

$$\sum_{(i,j)\in E_{N,N'}}x_{ijkt}\le 1 \qquad k\in K, t\in T \qquad (18)$$

$$\sum_{(i,j)\in E(S)}x_{ijkt}\le\sum_{i\in S}z_{ikt}-z_{i'kt} \qquad S\subseteq N\cup N', |S|\ge 2, i'\in S, k\in K, t\in T \qquad (19)$$

$$x_{ijkt}\in\{0,1\} \qquad (i,j)\in E, i\notin M, k\in K, t\in T \qquad (20)$$

$$x_{ijkt}\in\{0,1,2\} \qquad i\in M, j\in N\cup N', k\in K, t\in T \qquad (21)$$

$$z_{ikt}\in\{0,1\} \qquad i\in N\cup N'\cup M, k\in K, t\in T \qquad (22)$$

$$s_{ipt}, u_{ipkt}\ge 0 \qquad \begin{cases} i\in N, & p\in P,\\ i\in N', & p\in P', \end{cases} \quad k\in K,\ t\in T \qquad (23)$$

The objective function minimizes the total routing and inventory holding costs. Constraints (1) ensure that the total travel time for each vehicle route in each period will not exceed the available travel time in that period. Constraints (2) and (3) represent the inventory flow balance at sites for delivery commodities and backhauling materials, respectively. Constraints (4) enforce the overall capacity of sites for all commodities, while (5) impose this consideration for all backhauling materials in warehouses. The global vehicle capacity is expressed by constraints (6) for delivery commodities and by constraints (7) for backhauling materials. The industrial application additionally requires constraints regarding the capacities per commodity and material for warehouses and vehicles. Constraints (8) ensure that the total quantity of each commodity does not exceed the available storage capacity at the site, while (9) serves an analogous role for materials. Constraints (10) ensure that the total quantity of commodities on each vehicle does not surpass the capacity, and constraints (11) are formulated for the same purpose for materials. Constraints (12) and (13) guarantee that the total delivery commodity supplied in each depot and the material backhauled do not exceed the associated capacity in the depot. Constraints (14) and (15) allow the delivery of commodities to a site and the backhauling of materials by a vehicle from a site in a period only if the vehicle visits the site in that period. Constraints (16) ensure that the route of each vehicle must start and end at the predetermined depot. Constraints (17) are the degree constraints and represent the conservation of vehicle flow. These constraints ensure that if a site is visited, the number of edges incident to that site must be 2. Constraints (18) ensure that if a vehicle performs both delivery and backhauling, it can only begin the backhauling operation after completing the delivery. In other words, the vehicle cannot return for delivery once it has traversed between the networks. Constraints (19) are *subtour elimination constraints* (SECs) for each vehicle in each period. These constraints are similar to those used in Archetti et al. (2007) and in Adulyasak et al. (2014). Constraints (20)–(23) impose domain restrictions on the variables. Notice that although [IRP-DB-MDC] may initially appear decomposable over a time period; however, constraints (2) and (3) introduce dependencies between consecutive periods. These constraints link each time period to its predecessor, thereby preventing decomposition.

Table 3
Summary of extended models.

	Model	Split delivery	Integrated backhauling and delivery	Replenish policy	Stock-out	Depot inventory
Extended models	[IRP-DB-MDC-I]	No	No	ML	No	No
	[IRP-DB-MDC-II]	No	Yes	ML	No	No
	[IRP-DB-MDC-III]	No	Yes	OU	No	No
	[IRP-DB-MDC-IV]	No	Yes	ML	Yes	No
	[IRP-DB-MDC-V]	Yes	Yes	ML	No	Yes
Main model	[IRP-DB-MDC]	Yes	Yes	ML	No	No

3.2. Extended models

We introduce five extended models to evaluate the flexibility and performance of our proposed model under various operational and replenishment policies. The first extension, referred to as IRP-DB-MDC-I, models the case when a vehicle cannot be utilized for both backhauling and delivery within the same period. Furthermore, split deliveries are prohibited in this model. The second extended model, referred to as IRP-DB-MDC-II, allows a vehicle to perform backhauling after completing its delivery tasks but forbids split deliveries. In the third model, IRP-DB-MDC-III, we adopt an OU replenishment policy that restores inventory levels to a predetermined threshold at each visit. The fourth model, IRP-DB-MDC-IV, allows for stockouts by penalizing the stockout amount in the objective function. Finally, IRP-DB-MDC-V keeps track of inventory levels at the depots. Table 3 summarizes the characteristics of the main model and the extended models introduced above.

To formulate IRP-DB-MDC-I, we replace constraints (18) in [IRP-DB-MDC] with constraints (24), ensuring that a vehicle engaged in delivery cannot also be used for backhauling at the same time. Additionally, we add (25) to the model to prevent split deliveries. The [IRP-DB-MDC-I] formulation is given as follows:

[IRP-DB-MDC-I] :

$$\min \sum_{t\in T}\left(\sum_{k\in K}\sum_{(i,j)\in E} c_{ijk}x_{ijkt} + \sum_{i\in N}\sum_{p\in P} h_{ip}s_{ipt} + \sum_{i\in N'}\sum_{p\in P'} h_{ip}s_{ipt}\right)$$

s.t. (1) – (17), (19) – (23)

$$\sum_{(i,j)\in E_{N,N'}} x_{ijkt} \le 0 \qquad k\in K, t\in T \tag{24}$$

$$\sum_{k\in K} z_{ikt} \le 1 \qquad i\in N\cup N', t\in T. \tag{25}$$

To formulate IRP-DB-MDC-II, we solve the main model while including constraints (25) to forbid split delivery.

[IRP-DB-MDC-II] :

$$\min \sum_{t\in T}\left(\sum_{k\in K}\sum_{(i,j)\in E} c_{ijk}x_{ijkt} + \sum_{i\in N}\sum_{p\in P} h_{ip}s_{ipt} + \sum_{i\in N'}\sum_{p\in P'} h_{ip}s_{ipt}\right)$$

s.t. (1) – (23), (25).

To formulate the model with OU replenishment policy, we define a new binary decision variable y_{ipt} that indicates if site i is visited in period t to deliver commodity p and add Constraints (26) to ensure that if a site is visited for the replenishment of commodity $p \in P$, the amount replenished is sufficient to bring the site's inventory level to the TSL of the product. Moreover, constraints (27), added to ensure that y_{ipt} takes the value of one if site $i \in N$ is visited to replenish commodity $p \in P$ in period t.
[IRP-DB-MDC-III]:

$$\min \sum_{t\in T}\left(\sum_{k\in K}\sum_{(i,j)\in E} c_{ijk}x_{ijkt} + \sum_{i\in N}\sum_{p\in P} h_{ip}s_{ipt} + \sum_{i\in N'}\sum_{p\in P'} h_{ip}s_{ipt}\right)$$

s.t. (1) – (25)

$$s_{ip(t-1)} + \sum_{k\in K} u_{ipkt} \ge q'_{ip}y_{ipt} \qquad i\in N, p\in P, t\in T \tag{26}$$

$$\sum_{k\in K} u_{ipkt} \le q'_{ip}y_{ipt} \qquad i\in N, p\in P, t\in T. \tag{27}$$

We also extend the formulation of the main model to permit stockouts. New decision variables, r_{ipt}, are introduced to represent the quantity of commodity p that is stocked out at site $i \in N$ (material p that is stocked out at site $i \in N'$) in period $t \in T$ and constraints (2) and (3) are replaced by (28) and (29), respectively.

[IRP-DB-MDC-IV]:

$$\min \sum_{t\in T}\left(\sum_{k\in K}\sum_{(i,j)\in E} c_{ijk}x_{ijkt} + \sum_{i\in N}\sum_{p\in P}\left(h_{ip}s_{ipt} + e_{ip}r_{ipt}\right)\sum_{i\in N'}\sum_{p\in P'} h_{ip}s_{ipt}\right)$$

s.t. (1), (4) – (25)

$$s_{ip(t-1)} + \sum_{k\in K} u_{ipkt} + r_{ipt} = d_{ipt} + s_{ipt} \qquad i\in N, p\in P, t\in T \tag{28}$$

$$s_{ip(t-1)} + g_{ipt} = \sum_{k\in K} u_{ipkt} + r_{ipt} + s_{ipt} \qquad i\in N', p\in P', t\in T \tag{29}$$

$$r_{ipt} \geq 0 \qquad \begin{cases} i\in N, & p\in P, \\ i\in N', & p\in P', \end{cases} t\in T. \tag{30}$$

Ultimately, the formulation is extended to track inventory levels at the depots. In the main model, we assume that depot stock capacity for commodities is sufficiently large, so inventory levels and their associated costs are not explicitly considered; instead, we limit the maximum supply of commodity l_{ip} by a given parameter. However, we also demonstrate how the model can be modified to include inventory costs when depot capacity is constrained. Following the approach of Coelho et al. (2014), we redefine l_{ip} as the amount of commodity made available at each depot at the beginning of each period and introduce an inventory balance constraint for depots. Note that inventory levels for backhauled materials are not tracked, since these materials are immediately disposed of upon arrival; the maximum disposal amount per period is enforced by constraints (13).

[IRP-DB-MDC-V]:

$$\min \sum_{t\in T}\left(\sum_{k\in K}\sum_{(i,j)\in E} c_{ijk}x_{ijkt} + \sum_{i\in N\cup M}\sum_{p\in P}\left(h_{ip}s_{ipt} + e_{ip}r_{ipt}\right)\sum_{i\in N'\cup M}\sum_{p\in P'} h_{ip}s_{ipt}\right)$$

s.t. (1) – (11), (13) – (25), (30)

$$s_{ipt} = l_{ip} - \sum_{k\in K}\sum_{i\in N} u_{ipkt} + s_{ip(t-1)} \qquad i\in M, p\in P, t\in T. \tag{31}$$

Considering l_{ip} as a decision variable or determining the disposal amount of backhauled commodities lies beyond the scope of our study and falls within the domain of production routing problems.

4. Solution approach

The model formulated in this study is a computationally expensive MIP that demands substantial memory and processing power to obtain optimal solutions for large instances. We now develop both exact and matheuristic algorithms to address this challenge. In the following, we present and discuss the valid inequalities in Section 4.1. We then present the details of exact and matheuristic algorithms and provide thorough explanations. The exact algorithm is described in Section 4.2, and the matheuristic algorithm is detailed in Section 4.3. The matheuristic algorithm provides near-optimal solutions more efficiently by simplifying and breaking down the problem into manageable subproblems. It is particularly advantageous for larger instances where the exact algorithm may be impractical.

4.1. Valid inequalities

Similar to Archetti et al. (2007) and Adulyasak et al. (2014), we use valid inequalities to strengthen our formulation. In this study, we adopt and customize a set of cuts from the referenced papers for [IRP-DB-MDC] and also derive new inequalities and adopt the routing inequalities unchanged from prior works. Specifically, inequalities (32)–(35), originally formulated for single-product models in the referenced papers, have been extended for multi-product IRP. Inequalities (36) and (37) are newly derived to capture the backhauling constraints by following the same logical structure as the prior cuts. Finally, routing inequalities (38)–(40) are adopted directly from Adulyasak et al. (2014). In the following, the details of the inequalities are provided.

Let t' be the earliest period when at least one customer must be replenished to prevent a stockout, and κ be the minimum shipping quantity in t', inequality (32) prevents stockouts:

$$\sum_{k\in K}\sum_{t=1}^{t'} z_{i_k kt} \geq \left\lceil \frac{\kappa}{b_{\max}} \right\rceil \tag{32}$$

where,

$$t'_{ip} = \text{argmin}_{1\leq t\leq \mathcal{T}}\left\{\sum_{t''=1}^{t} d_{ipt''} - s_{ip0} > 0\right\}$$

$$t' = \min_{i\in N, p\in P} t'_{ip}$$

$$\kappa = \sum_{i\in N}\sum_{p\in P} \max\{0, \sum_{t''}^{t'} d_{ipt''} - s_{ip'0}\}$$

$$b_{\max} = \max_{k \in K} b_k.$$

Inequalities (33) imply that if site i is not served in period t, then the inventory level at the site should be sufficient to meet the demand of the site for that period without resulting in a stockout. Inequalities (34) are an extension of (33) and ensure that if a site i is not visited for delivery in periods $t-e,\dots,t$, then the decision variables for visiting the site take the value of zero for all the given periods and all vehicles; therefore the inventory level at time $t-e-1$ should be sufficient to satisfy demand in those periods. By considering the demand between $t-e$ and t and accounting for the capacity of vehicles, inequalities (35) ensure that the total number of visits to a site between periods should be sufficient to prevent stockouts.

$$\left(1 - \sum_{k \in K} z_{ik,t+1}\right) d_{ip,t+1} \le s_{ipt} \qquad i \in N, p \in P, t \in [1,\dots,\mathcal{T}-1] \tag{33}$$

$$\left(\sum_{t'=0}^{e} d_{ip,t-t'}\right)\left(1 - \sum_{k \in K}\sum_{t'=0}^{e} z_{ik,t-t'}\right) \le s_{ip,t-e-1} \qquad i \in N, p \in P, t \in T, e \in [0,\dots,t-1] \tag{34}$$

$$\sum_{t'=0}^{e} d_{ip,t-t'} - s_{ip,t-e-1} \le \sum_{k \in K}\sum_{t'=0}^{e} z_{ik,t-t'} a_{kp} \qquad i \in N, p \in P, t \in T, e \in [0,\dots,t-1]. \tag{35}$$

Inequalities (36) ensure that if a site i is not visited for backhauling in periods $t-e,\dots,t$, then there should be enough capacity for stocking backhauling material accumulated in those periods. Inequalities (37) ensure that the total number of visits to a site across periods is sufficient to keep the accumulated backhauling material at each site below its capacity. Finally, we have the routing inequalities in (38)–(40):

$$s_{ip,t-1} + \sum_{t'=t}^{t+e} g_{ipt'}\left(1 - \sum_{k \in K}\sum_{t'=t}^{t+e} z_{ikt'}\right) \le q_{ip} \qquad i \in N', p \in P', t \in T, e \in [0,\dots,\mathcal{T}-t] \tag{36}$$

$$\sum_{t'=0}^{e} g_{ip,t-t'} + s_{ip,t-e-1} - q_{ip} \le \sum_{k \in K}\sum_{t'=0}^{e} z_{ik,t-t'} a_{kp} \qquad i \in N', p \in P', t \in T, e \in [0,\dots,t-1] \tag{37}$$

$$z_{ikt} \le \sum_{i' \in M} z_{i'kt} \qquad i \in N \cup N', k \in K, t \in T \tag{38}$$

$$x_{ijkt} \le z_{ikt} \qquad (i,j) \in E, k \in K, t \in T \tag{39}$$

$$x_{ijkt} \le z_{jkt} \qquad (i,j) \in E, k \in K, t \in T. \tag{40}$$

4.2. The branch-and-cut algorithm

We use the B&C algorithm for solving [IRP-DB-MDC], and it is strengthened by the valid inequalities formulated in Section 4.1, all of which are polynomial in number. Additionally, an exact separation algorithm, which solves a minimum $s-t$ cut problem, determines the violated SECs for each vehicle in each period as the vehicle tours are identified by the vehicle index (Adulyasak et al., 2014). When solving the separation problem at each node of the *branch-and-bound* (B&B) tree, we denote the current values of the variables z_{ikt} and x_{ijkt} by $\bar{z}_{ikt}$ and $\bar{x}_{ijkt}$, respectively. A graph for vehicle k in period t is generated from the set of nodes with $\bar{z}_{ikt} > 0$. The weights of the edges in the generated graph are equal to the $\bar{x}_{ijkt}$ values. Next, for each site in a given route, a minimum $s-t$ cut problem is solved where the depot of the route is set as the source node and the site is set as the sink node. If the value of the minimum cut is less than $2\bar{z}_{ikt}$, then a violated SEC is identified. This procedure is executed at the root node and in a predefined number of initially explored nodes in the B&B tree. For the subtour on a set of nodes S that is found for vehicle k in period t, we add constraint (19) with $i' = \arg\max_{i \in \bar{N}_{kt}}(\bar{z}_{ikt})$ to the formulation where $\bar{N}_{kt} = \{i \in N \mid \bar{z}_{ikt} > 0\}$. Note that we have utilized the NetworkX (Hagberg et al., 2008) library in Python to solve the minimum $s-t$ cut problem and the Gurobi callback function (Gurobi Optimization, 2024) for implementing the SECs as both lazy and user cuts.

4.3. Matheuristic algorithm

In this section, we describe a matheuristic algorithm to solve the problem by breaking it into more manageable subproblems. The algorithm operates in two phases. In Phase I, an initial solution is generated, which involves solving an assignment problem to attribute sites to vehicles in each period and to obtain an optimal route for each assignment. Phase II addresses the main optimization model by using the obtained assignment and routing variables from Phase I as a warm start for the solver. Combining these two phases allows the algorithm to obtain near-optimal solutions. Fig. 2 shows the flowchart of the proposed algorithm. The idea for the algorithm used to obtain the initial solution is inspired by the matheuristic algorithms presented by Absi et al. (2015) and by Chitsaz et al. (2019) for solving production routing problems. The mentioned algorithms consider identical vehicles and do not account for travel time limits. However, our algorithm addresses heterogeneous vehicles and incorporates travel time constraints. Additionally, instead of relying on numerous iterations, we employ the B&C algorithm to enhance the performance and reduce CPU time. In the following, we describe the algorithm in detail.

Phase I: Iterative algorithm to generate an initial solution. This phase is divided into three steps. In Step 1, a mathematical model is solved to assign sites to vehicles. In Step 2, a series of *traveling salesman problems* (TSPs) are solved to find the best route for each

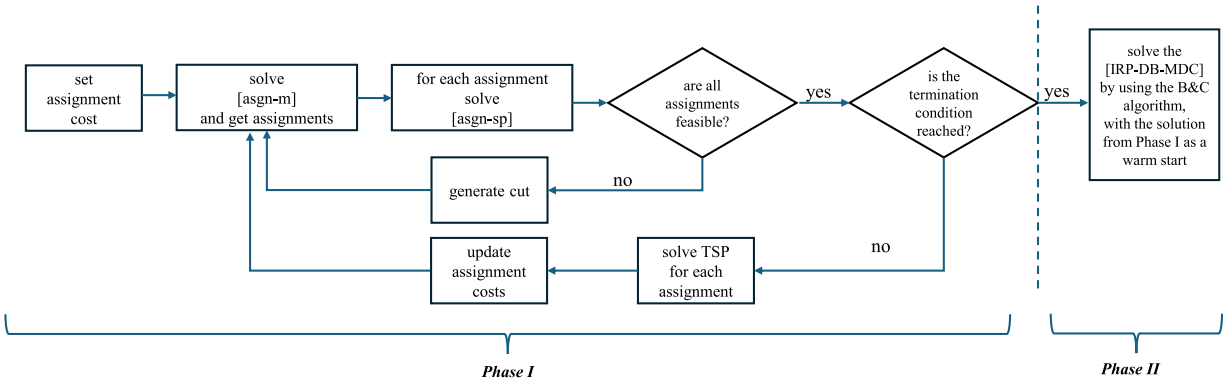

Fig. 2. The algorithm generates an initial solution in Phase I, then uses the B&C to solve the optimization model using the initial solution as a warm start.

vehicle in each period. Finally, in Step 3, the assignment costs are updated according to the obtained solutions. This procedure is repeated iteratively for a specified number of iterations to refine the on-hand solution. The algorithmic steps are provided in detail below.

[**Step 1**]. In this step, we solve an assignment model, denoted as [asgn], using the constraints of [IRP-DB-MDC]:

$$[\text{asgn}]: \min \sum_{t\in T}\left(\sum_{k\in K}\sum_{i\in N\cup N'\cup M}\sigma_{ikt}z_{ikt}+\sum_{i\in N}\sum_{p\in P}h_{ip}s_{ipt}+\sum_{i\in N'}\sum_{p\in P'}h_{ip}s_{ipt}\right)$$

$$\text{s.t. } (1)-(40).$$

The main difference between this model and [IRP-DB-MDC] lies in the objective function. In the [asgn] model, the objective function excludes routing costs and instead focuses on the dispatching cost of vehicles and the costs of assigning sites to vehicles. The cost of assigning site i to vehicle k in period t is denoted by σ_{ikt}, which is approximated as half of the travel cost for the vehicle to reach site i from the depot where the vehicle is located. Additionally, the cost of dispatching each vehicle from its depot is calculated as $\sigma_{i_k kt}=\frac{1}{\mathcal{N}}\left(\sum_{i\in N}\sigma_{ikt}\right)$, where i_k is the depot where vehicle k is located. It is important to note that all decisions regarding the quantities delivered to sites by each vehicle in every period are made within the model. These quantities adhere to the vehicle and site capacity constraints.

We use a logic-based decomposition method to solve the [asgn] problem by dividing it into a master problem and subproblems. The master problem involves assigning sites to vehicles, while the subproblems focus on routing decisions. The subproblems check whether the assignment from the master problem has a feasible solution, i.e., if the vehicle can cover the sites within the given time. If an assignment is found to be infeasible due to travel time constraints, the following cut is generated to exclude this assignment from consideration. Here, z^*_{ikt} represents the values in the current solution. The sets $\overline{N}$ and $\overline{N'}$ correspond to the nodes visited by vehicle k^* in the infeasible route for delivery and backhauling, respectively. The parameter α takes the value 1 if $\overline{N}$ and $\overline{N'}$ share a common node and backhauling begins immediately from that node, and 0 otherwise:

$$\sum_{i\in\overline{N}\cup\overline{N'}} z_{ik^*t}\le\sum_{i\in\overline{N}\cup\overline{N'}} z^*_{ik^*t}-1-\alpha. \tag{41}$$

To this end, the master problem is solved first, followed by the subproblems for each assignment in each period. The master problem and the subproblem are given below as [asgn-m] and [asgn-sp] respectively:

$$[\text{asgn-m}]: \min \sum_{t\in T}\left(\sum_{k\in K}\sum_{i\in N\cup N'\cup M}\sigma_{ikt}z_{ikt}+\sum_{i\in N}\sum_{p\in P}h_{ip}s_{ipt}+\sum_{i\in N'}\sum_{p\in P'}h_{ip}s_{ipt}\right)$$

$$\text{s.t. } (2)-(16),(22)-(38)$$

$$[\text{asgn-sp}]: \min 0$$

$$\text{s.t. } (1),(17)-(21),(39),(40).$$

[**Step 2**]. In this step, we determine the optimal routes for the sites assigned to each vehicle in each period. Specifically, we solve a series of TSPs for the sites allocated to each vehicle during each period and record the resulting optimal routes.

[**Step 3**]. In this step, we update the assigning cost of sites to the vehicle using the solution obtained in the first iteration. Let N_{kt} represent the set of sites assigned to the vehicle k in period t, obtained from Step 1, and let R_{kt} be the route of vehicle k in period t. For vehicle k in period t and $i\in N_{kt}$, let i^p and i^s denote the predecessor and successor of site i in route R_{kt}, respectively. For any site not in N_{kt}, we update σ_{ikt} to the cheapest insertion cost for inserting it into R_{kt}. For sites in N_{kt}, we update the visiting cost σ_{ikt} to $(c_{i^p i}+c_{ii^s}-c_{i^s i^p})$. We repeat this cost update mechanism for all sites across all vehicles and periods.

Phase II: Improving initial solution. In this phase, we use the solution generated in Phase I as a warm start for [IRP-DB-MDC] and start the solving procedure with a given time limit. Finally, we report the optimized solution as the final solution of the algorithm.

To customize the algorithm to solve the [IRP-DB-MDC-I] model, we add constraints (25),(42)–(45) to [asgn] and [asgn-m] to forbid split delivery and ensure that each assignment involves either delivery or backhauling, but not both. In this context, A and A' are defined as auxiliary binary decision variables. Constraints (42) set the value of A_{kt} to 1 if vehicle k in period t does the delivery, and constraints (43) set the value of A'_{kt} to one if vehicle k in period t does the backhauling. Constraints (44) prevent a vehicle from delivering and backhauling in the same period:

$$\sum_{i\in N} z_{ikt} \leq \mathcal{N} A_{kt} \qquad k \in K, t \in T \tag{42}$$

$$\sum_{i\in N'} z_{ikt} \leq \mathcal{N}' A'_{kt} \qquad k \in K, t \in T \tag{43}$$

$$A_{kt} + A'_{kt} \leq 1 \qquad k \in K, t \in T \tag{44}$$

$$A_{kt}, A'_{kt} \in \{0,1\} \qquad k \in K, t \in T. \tag{45}$$

To solve [IRP-DB-MDC-II] with the algorithm, we only add constraints (24) to the [asgn] and [asgn-m] models to forbid split delivery. Similarly, for [IRP-DB-MDC-III], we include constraints (26) and (27) into [asgn] and [asgn-m] to enforce the OU replenishment policy. To solve [IRP-DB-MDC-IV], we update the objective functions of [asgn] and [asgn-m] to include stockout costs, and replace inventory balance constraints (2) and (3) with (28) and (29) to accommodate stockouts. Notice that the inequalities (32)–(37) are not valid for the model with stockout and should be excluded from [asgn] and [asgn-m]. Finally, for [IRP-DB-MDC-V], we follow the same procedure by updating the objective function to include inventory costs in depots, removing constraint (12) from both [asgn] and [asgn-m], and adding (31) as the inventory balance constraint for commodities at the depots in both problems.

In the algorithm, there may be instances where a strict travel time constraint for vehicle tours prevents the proposed metaheuristic algorithm from generating an initial solution on the first iteration or causes it to take too long. This issue occurs when the time limit is short and the distances between locations are large, since [asgn-m] assigns sites regardless of their distances. Consequently, the algorithm may need to solve the subproblem multiple times to avoid infeasible assignments and to add numerous cuts to restore feasibility. To address this, we initially solve the problem with a larger time limit than the actual one and then iteratively reduce the time limit in steps until the desired travel time constraint is met. This approach allows us to obtain feasible solutions more efficiently by gradually tightening the constraints. We applied this method to our case study, and further details are provided in Section 5.4.2.

5. Numerical study

In this section, we present the numerical experiments designed to evaluate the performance of the proposed algorithms. In our numerical experiments, we use two distinct datasets. The first one is used to generate synthetic instances, while the second one is specifically generated to meet the requirements of our industrial partner, HQ. The synthetic instances are designed to evaluate the model and the performance of the solution approaches. We avoid using the case study data for this purpose to ensure that the approaches are general and not biased toward the possible specific characteristics of the case study data. We describe the data and the synthetic instances in Section 5.1. We then evaluate the performance of the B&C and matheuristic algorithms in Sections 5.2 and 5.3, respectively. Finally, the details of the data for HQ are presented, and we discuss the case study in Section 5.4. The computational experiments were conducted on a 64-bit Windows server equipped with an Intel Xeon Gold 6148 CPU @ 2.40GHz. The algorithms were implemented using Python and Gurobi 10.0.1.

5.1. Generation of synthetic instances

To evaluate the efficacy of proposed B&C and matheuristic algorithms, we adapt the data generation instruction proposed by Archetti et al. (2007) to create synthetic instances for multi-commodity IRP-DB-MDC with heterogeneous vehicles. The generation process is customized to address the specific requirements of our problem. Detailed information on the generation process is provided in Tables 4 and 5. Each instance in our study varies in scale, with the number of sites $\mathcal{N} = 12\alpha$ and $\mathcal{N}' = 8\alpha$, where $\alpha \in \{1,3,5\}$. We solve the problem for different numbers of periods, $\mathcal{T} \in \{3,5\}$, and allocate a varying number of vehicles, $\mathcal{K} \in \{3,5\}$. The number of commodities and materials are set to $\mathcal{P} = 10\beta$ and $\mathcal{P}' = 5\beta$, with $\beta \in \{1,2\}$. As summarized in Table 6, 12 unique combinations are generated. In all instances, we assume two depots, with vehicles of odd indices departing from one depot and vehicles of even indices departing from the other. All vehicles return to their starting depot at the end of their routes. We designed eight *data generation frameworks* (DGFs) for each instance based on the varying parameter values listed in Tables 4 and 5 by considering different intervals for holding costs, coordinate ranges, and vehicle capacities. This process yielded $12 \times 8 = 96$ distinct instances. In other words, for each DGF, the intervals of holding costs, coordinates and capacities are different; however, parameters in Table 4 have the same intervals. To account for variability, we generated five data sets for each of the 96 instances by randomly generating parameters and reporting the average performance. To report the results, we employ various aggregation methods and group the instances based on DGF or instance scales. Out of the 480 synthetic instances generated, 450 were found to have feasible solutions, while the remaining instances were determined to be infeasible. All subsequent analyses in this paper focus on the 450 instances with verified feasible solutions. The parameter ω_k in Tables 4 and 5 represents a coefficient specific to each vehicle k, used to scale its base capacity, thereby introducing heterogeneity among the vehicles in terms of their capacities.

Table 4
Sampling distributions of parameters.

Parameter		Value	Parameter		Value
	d^{*}_{ipt}	$[5, 25]$		g^{*}_{ipt}	$[5, 25]$
	a_{kp}	$\max_{i\in N}(q_{ip} + d_{ipt}) \times \omega_k$		a_{kp}	$\max_{i\in N'}(q_{ip} + g_{ipt}) \times \omega_k$
$(p \in P)$	q_{ip}	$d_{ip1} \times rnd_i$	$(p \in P')$	q_{ip}	$g_{ip1} \times rnd_i$
$(i \in N)$	l_{ip}	$\max_{t\in\mathcal{T}} \left(\sum_{j\in\mathcal{N}} d_{jpt}\right) \times rnd_i$	$(i \in N')$	l_{ip}	$\max_{t\in\mathcal{T}} \left(\sum_{j\in\mathcal{N}'} g_{jpt}\right) \times rnd_i$
	s_{ip0}	$(q_{ip} - d_{ip1})$		s_{ip0}	$(q_{ip} - g_{ip1})$
	o_i	$\sum_{p\in P} q_{ip}$		o_i	$\sum_{p\in P'} q_{ip}$
	b_k	$\sum_{p\in P} a_{kp}$		$T_{\max}$	$\text{avg}(t_{ijk}) \times (\mathcal{N} + \mathcal{N}') \times \mathcal{V}^{-1}$

rnd_i: a random number selected from set {2,3,4}
avg: average
* assumed to be the same for all periods

Table 5
Characteristic of different data generation frameworks.

DGF	h_{ip}	$h_{ip'}$	Coordinates range	ω_k
1	[6,10]	[11,15]	[0,500]	[1, 1.5]
2	[6,10]	[11,15]	[0,500]	[1.5, 2]
3	[1,5]	[6,10]	[0,500]	[1, 1.5]
4	[1,5]	[6,10]	[0,500]	[1.5, 2]
5	[6,10]	[11,15]	[0,1000]	[1, 1.5]
6	[6,10]	[11,15]	[0,1000]	[1.5, 2]
7	[1,5]	[6,10]	[0,1000]	[1, 1.5]
8	[1,5]	[6,10]	[0,1000]	[1.5, 2]

$[a, b]$ denotes a continuous uniform distribution between a and b
h_{ip}: holding cost of commodity
$h_{ip'}$: holding cost of backhauling material

Table 6
Synthetic instances.

Instance size $(\mathcal{N} \times \mathcal{N}' \times \mathcal{P} \times \mathcal{P}' \times \mathcal{T} \times \mathcal{K})$	Instance size $(\mathcal{N} \times \mathcal{N}' \times \mathcal{P} \times \mathcal{P}' \times \mathcal{T} \times \mathcal{K})$
$(12 \times 8 \times 10 \times 5 \times 3 \times 3)$	$(12 \times 8 \times 10 \times 5 \times 5 \times 5)$
$(12 \times 8 \times 20 \times 10 \times 3 \times 3)$	$(12 \times 8 \times 20 \times 10 \times 5 \times 5)$
$(36 \times 24 \times 10 \times 5 \times 3 \times 3)$	$(36 \times 24 \times 10 \times 5 \times 5 \times 5)$
$(36 \times 24 \times 20 \times 10 \times 3 \times 3)$	$(36 \times 24 \times 20 \times 10 \times 5 \times 5)$
$(60 \times 40 \times 10 \times 5 \times 3 \times 3)$	$(60 \times 40 \times 10 \times 5 \times 5 \times 5)$
$(60 \times 40 \times 20 \times 10 \times 3 \times 3)$	$(60 \times 40 \times 20 \times 10 \times 5 \times 5)$

5.2. Branch-and-cut algorithm performance

We now show the impact of including valid inequalities on the performance of the algorithm. We first show the improvement made by valid inequalities and SECs on the lower bound obtained at the root node of the B&B tree. We then show the performance of the proposed B&C algorithm over generated synthetic instances.

Table 7 presents the improvement achieved in lower bounds by incorporating valid inequalities. These improvements are all calculated compared to the *linear programming* (LP) relaxation of the [IRP-DB-MDC]. We present the improvement achieved by the default cuts of the commercial solver and demonstrate that incorporating additional valid inequalities and cuts further enhances this performance. The total improvement reaches 18.5 %, compared to the 16.2 % achieved by the commercial solver's default cuts alone. The table also provides detailed insights into the performance across different DGFs and scales. Adding inequalities shows improvement across instance sizes. This consistency indicates that the proposed method is versatile and can be effectively applied to a wide range of problem instances. However, analyzing the results based on different DGF values and scales shows the sensitivity of the approach to key parameters. We observe greater improvements in DGFs where the routing cost is relatively higher (DGF = 3, 4, 7, 8). This effect is most pronounced in DGFs 7 and 8, where routing costs constitute the largest portion of the objective function while holding costs have the smallest share. This is because the added valid inequalities primarily strengthen the routing component of the model, leading to more significant improvements when routing costs dominate. Overall, the results presented in Table 7 show the relevance of integrating inequalities in improving the lower bound at the root node.

Table 7
Improvement of the lower bound at the root node compared to LP relaxation.

DGF	#ins	LP + Gurobi	LP + Gurobi + VI	Instance size	#ins	LP + Gurobi	LP + Gurobi + VI
1	50	8.9 %	10.2 %	$\mathcal{N} = 12, \mathcal{N}' = 8$	160	25.9 %	28.1 %
2	60	6.8 %	8.7 %	$\mathcal{N} = 36, \mathcal{N}' = 24$	150	15.1 %	17.5 %
3	55	14.3 %	16.7 %	$\mathcal{N} = 60, \mathcal{N}' = 40$	140	7.6 %	9.5 %
4	60	20.3 %	23.1 %	$\mathcal{P} = 10, \mathcal{P}' = 5$	220	19.6 %	21.9 %
5	50	10.7 %	12.2 %	$\mathcal{P} = 20, \mathcal{P}' = 10$	230	12.8 %	15.2 %
6	60	13.3 %	15.5 %	$\mathcal{T} = 3, \mathcal{T} = 3$	210	9.9 %	11.1 %
7	55	23.0 %	25.3 %	$\mathcal{T} = 5, \mathcal{K} = 5$	240	22.5 %	25.8 %
8	60	33.0 %	36.8 %				
Total/Average	**450**	16.2 %	18.5 %				

VI: valid inequalities, DGF: data generation framework, #ins: number of instances

Table 8
Performance comparison of MIP vs B&C.

Scale	#ins	[IRP-DB-MDC]				B&C				
		#opt	#nf	avg.gap[1]	CPU time	#opt	#nf	avg.gap[1]	avg.gap[2]	CPU time
$\mathcal{N} = 12, \mathcal{N}' = 8$	160	160	0	0	1774	160	0	0	0 %	450
$\mathcal{N} = 36, \mathcal{N}' = 24$	140	0	29	3.1 %	7200	0	14	2.0 %	2.0 %	7200
$\mathcal{N} = 60, \mathcal{N}' = 40$	150	0	150	-	7200	0	150	-	-	7200
$\mathcal{P} = 10, \mathcal{P}' = 5$	220	78	69	1.8 %	4814	80	60	1.3 %	1.3 %	4808
$\mathcal{P} = 20, \mathcal{P}' = 10$	230	82	110	0.6 %	5968	80	104	0.2 %	0.7 %	5091
$\mathcal{T} = 3, \mathcal{K}' = 3$	210	80	53	1.1 %	5755	80	47	0.7 %	0.7 %	4976
$\mathcal{T} = 5, \mathcal{K}' = 5$	240	80	126	1.5 %	5027	80	117	1.0 %	1.2 %	4923
DGF										
1	50	20	21	0.2 %	5019	20	19	0.2 %	0.2 %	4412
2	60	20	21	1.2 %	5407	20	20	0.4 %	0.5 %	4882
3	55	20	25	0.5 %	5150	20	22	0.6 %	0.6 %	4750
4	60	20	25	1.2 %	5498	20	22	1.0 %	1.0 %	5050
5	50	20	20	0.8 %	5406	20	19	0.4 %	0.7 %	4727
6	60	20	20	1.0 %	5480	20	19	0.6 %	0.8 %	4950
7	55	20	24	1.5 %	5293	20	22	1.0 %	1.0 %	4768
8	60	20	23	1.5 %	5033	20	21	1.3 %	1.6 %	4982
Total/Average	**450**	**160**	**179**	1.0 %	5391	**160**	**164**	0.7 %	0.8 %	4950

#opt: number of instances solved to optimality
#nf: number of instances with no feasible solution
avg.gap[1]: average gap for instances that both MIP and B&C can generate a feasible solution
avg.gap[2]: average gap for instances that B&C can generate a feasible solution
gaps are calculated compared to the lower bound

Next, we assess the performance of B&C compared to solving the [IRP-DB-MDC] by Gurobi with a time limit of 7200 s. This evaluation aims to determine how valid inequalities can efficiently facilitate the convergence of the problem to optimality. The results of this assessment are summarized in Table 8. The results show that both algorithms successfully solve all 160 instances with 20 sites (12 for delivery and 8 for backhauling) to optimality. However, when applied to instances with 60 sites (36 for delivery and 24 for backhauling), Gurobi fails to find feasible solutions for 29 instances when the [IRP-DB-MDC] is solved without adding inequalities, whereas the B&C algorithm reduces this number to 14 instances. As shown in the table, no feasible solution is generated for any instance with 100 sites by any of the algorithms. This is where the matheuristic algorithm is more useful. The results indicate that larger instances may decrease the efficiency of both approaches, though B&C demonstrates superior performance as the instance size increases. Generally, the B&C algorithm achieves an average optimality gap of 0.7 %, which is smaller than the 1 % gap observed when solving the [IRP-DB-MDC] with Gurobi without adding cuts (see the last line of Table 8). This highlights the enhanced efficiency of the B&C algorithm. Additionally, checking the CPU time shows that adding valid inequalities leads to faster convergence to optimality. The results also show that when DGF 7 and 8 are used to generate instances, the largest optimality gaps are observed. This occurs because the travel cost constitutes a larger proportion of the objective function in these cases. However, the increase in the gap is not substantial, and we conclude that the overall performance of the B&C algorithm is not significantly sensitive to these key parameters. Overall, the B&C algorithm outperforms the approach without B&C in finding efficient and feasible solutions and reducing the optimality gap across various scales and parameters, as shown by the detailed results in Tables 7 and 8.

Table 9
Performance of matheuristic compared to the best lower bound.

Instance size	#ins	Phase I				Phase II			
		Avg. gap[1]	Avg. gap[2]	Imp	CPU time	Avg. gap[1]	Avg. gap[2]	Imp	CPU time
$\mathcal{N} = 12, \mathcal{N}' = 8$	160	1.8 %	–	-1.8 %	135	0 %	–	0.0 %	466
$\mathcal{N} = 36, \mathcal{N}' = 24$	160	4.6 %	4.9 %	-2.5 %	539	2.4 %	4.1 %	0.2 %	3600
$\mathcal{N} = 60, \mathcal{N}' = 40$	160	3.4 %	3.4 %	–	1689	3.0 %	3.0 %	–	3600
$\mathcal{P} = 10, \mathcal{P}' = 5$	160	3.8 %	4.7 %	-2.6 %	531	2.3 %	3.7 %	-0.1 %	2550
$\mathcal{P} = 20, \mathcal{P}' = 10$	160	2.7 %	3.8 %	-1.7 %	1061	1.5 %	2.6 %	0.2 %	2564
$\mathcal{T} = 3, \mathcal{K}' = 3$	160	1.9 %	2.1 %	-1.1 %	309	1.0 %	1.4 %	-0.1 %	2407
$\mathcal{T} = 5, \mathcal{K}' = 5$	160	4.6 %	5.3 %	-3.2 %	1283	2.8 %	4.2 %	0.2 %	2703
DGF									
1	50	1.7 %	2.6 %	-0.8 %	1283	0.8 %	2.3 %	0.1 %	2317
2	60	1.6 %	1.9 %	-1.0 %	1283	0.8 %	1.4 %	0.1 %	2511
3	55	3.4 %	4.9 %	-1.6 %	1283	2.9 %	3.8 %	0.0 %	2564
4	60	3.6 %	4.6 %	-2.6 %	1283	1.7 %	3.0 %	0.1 %	2583
5	50	3.0 %	4.8 %	-1.5 %	1283	2.1 %	4.6 %	-0.1 %	2395
6	60	2.8 %	3.3 %	-1.8 %	1283	1.5 %	2.7 %	0.0 %	2506
7	55	4.6 %	6.3 %	-2.3 %	1283	3.1 %	5.7 %	0.0 %	2403
8	60	5.7 %	6.8 %	-4.2 %	1283	3.0 %	4.8 %	0.1 %	2560
Average		3.3 %	4.4 %	-2.0 %	825	1.9 %	3.6 %	0.0 %	2480

Avg. gap[1]: average gap for all instances
Avg. gap[2]: average gap for instances that B&C cannote generate a feasible solution
gaps are calculated compared to the lower bound
Imp: improvement made compared to B&C (if a feasible solution is obtained)

5.3. Matheuristic algorithm performance

We now discuss the performance of the matheuristic algorithm and compare it with the B&C algorithm in terms of objective function value and CPU time. The numerical results are summarized in Table 9 and demonstrate that the algorithm produces optimal solutions for instances with 20 sites. However, as the instance sizes increase, the optimality gap increases, though the average gap compared to the reported best lower bound remains below 2 %. The column labeled “Imp” in the table reports the improvement of the matheuristic algorithm over the objective function value obtained by the B&C algorithm within the given time limit. Although there are instances where the B&C algorithm solves the problem efficiently, on average, the algorithm outperforms the B&C, albeit with a marginal advantage. The proposed algorithm demonstrates significant efficiency, for large-size instances where the B&C algorithm fails to generate feasible solutions. Specifically, for the instances with 100 sites, we observe an average optimality gap of 3 % for the heuristic solutions, as reported in the column labeled “Avg. gap[2]”. The matheuristic algorithm also offers the advantage of shorter CPU times, enhancing its practicality and efficiency for large-scale problems by solving all instances in under 7200 s. In particular, Phase I solves the instances on average in 825 s with an optimality gap of 3.3 %. Phase II, which takes approximately 2480 s, further improves the solution, achieving an average optimality gap of 1.9 %. These findings suggest that although Phase I does not always yield the best objective functions, it excels in terms of CPU efficiency. This advantage becomes more significant with 100 sites, where Phase I, on average, requires 1689 s to achieve solutions with an optimality gap of 3.4 % where B&C can not even generate a feasible solution. Notice that the procedure of solving each subproblem for each vehicle in the matheuristic algorithm can be implemented in Gurobi using a callback function, which adds cuts at each integer solution of the master problem in the B&B tree. It is worthwhile to discuss the sensitivity of the heuristic algorithm to different parameter settings, summarized in the second half of the table. The CPU time remains nearly constant across all DGFs, while the solution quality shows marginal variation. Specifically, when holding costs comprise the largest proportion of total cost (DGF 1 and DGF 2), the algorithm achieves superior performance, with average optimality gaps below 1 %. In contrast, instances with higher routing costs (DGF 7 and DGF 8) see the gap increase to 3.1 % and 3.0 %, respectively.

In conclusion, when instance sizes are small and we have less than 20 sites, the B&C algorithm can find an optimal solution in less than ten minutes. As instance size increases, the CPU time of the B&C algorithm grows, i.e., for 60-site instances it produces solutions with an average optimality gap of 2 % in two hours. The two-phase matheuristic, however, obtains slightly more efficient solutions with a marginally better gap in less than two hours, suggesting that both methods perform similarly on medium-sized instances. The main advantage of the algorithm is highlighted when we solve large-sized instances where the B&C algorithm fails to produce feasible solutions, but the matheuristic succeeds. While the two-phase approach yields the best results, Phase I alone is highly efficient in terms of CPU time. Therefore, if a fast solution is required, it is conceivable to accept a marginal loss in the objective function value and run only Phase I.

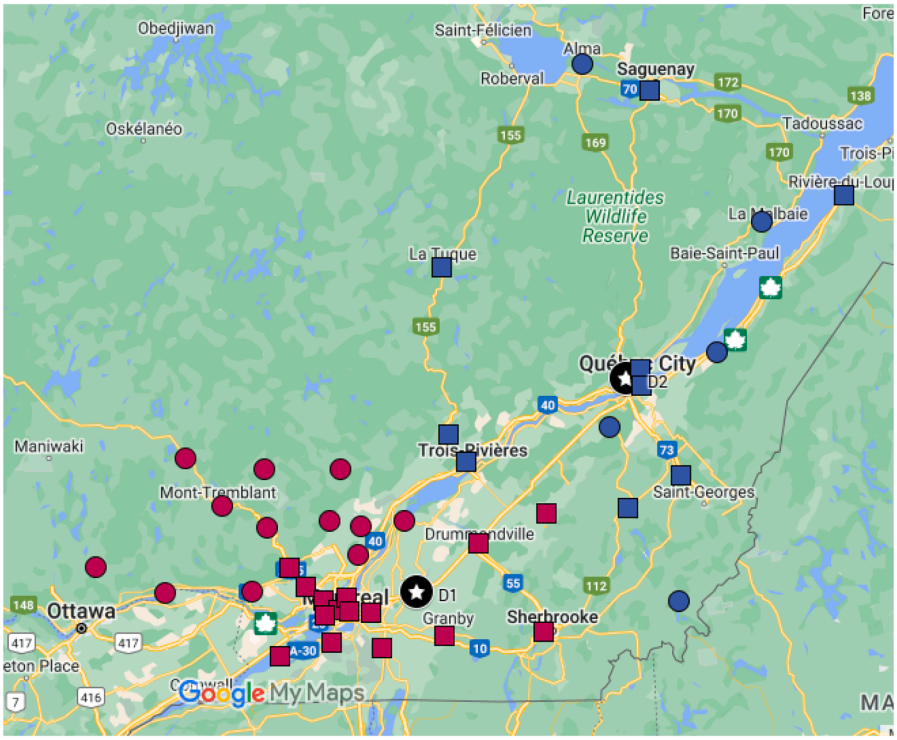

Fig. 3. Depots and satellite points: stars represent the depots (left D1, right D2), squares indicate satellite points with only delivery demand, and circles denote satellite points with both delivery and backhauling.

5.4. Case study

In this section, we analyze the case study and the associated challenges faced by HQ. We first describe the data structure relevant to our industrial partner's problem in Section 5.4.1. We next solve the problem, present the solution for the case study, and evaluate the efficiency of our proposed approach for the case study in Section 5.4.2.

5.4.1. Case study overview

The HQ network has two depots distributing various commodities to 42 satellite points and backhauling hazardous materials from 17 satellite points back to the depots. The depots (origins) and satellite points are depicted in Fig. 3. Two nodes marked with a star represent the depots, squares indicate satellite points with delivery demand only, and circles denote satellite points with both delivery and backhauling requirements. According to HQ's current policy, the red sites are serviced by Depot number 1, located on the left side of the figure, while the blue sites are only served by Depot number 2, situated on the right side. The commodities distributed to satellite points include fasteners and connectors (e.g., various types of bolts, compression punches, cable lugs), insulators and sealing components (e.g., composite insulators, insulating supports), mechanical components (e.g., cross-arms, suspension clamps), sealing components, and other equipment essential for the operation of electrical power systems. The hazardous materials to backhaul that should be handled by HQ are primarily used transformer oils, which are collected for cleaning, reuse, or disposal. The problem is solved for 5 periods, corresponding to the 5 days of the week. HQ has a diverse fleet of vehicles, including pickups, vans, and large trailers, which can be grouped into three main categories: *trucks*, which are the fastest with an average speed of 90 km/h; *vans*, suitable for larger transportation needs with an average speed of 80 km/h; and *trailers*, which have the largest capacity with an average speed of 70 km/h. In this network, four vehicles of each type are available, and each vehicle is limited to completing its tour within 8 h.

In this study, in accordance with the guidance obtained from the team at HQ, we use we use the $q'_{ip} = \mu\mathcal{L} + \mathcal{Z}\sigma\sqrt{\mathcal{L}}$ formulation to calculate the TSL for each site in the model solved under OU replenishment policy, where $\mu = \frac{\sum_{t\in\mathcal{T}} d_{ipt}}{\mathcal{T}}$ represents the average demand of a product at site i over the planning horizon, and $\mathcal{T}$ is the total number of periods in the planning horizon. The lead time, $\mathcal{L}$, is set to 2 periods to ensure each site replenishes at least twice its average demand in each visit. The value of $\mathcal{Z}$, set to 1.96, corresponds to the number of standard deviations for a 97.5 % service level. The term σ denotes the standard deviation of demand. Due to the confidentiality of the data, we cannot disclose or use the original demand data. Instead, we have obtained approximate data, which is summarized in Table 10 below. The demand and capacity are measured in pallets, each weighing approximately 20 kg.

In the following, we first solve [IRP-DB-MDC-I], which assumes that a vehicle cannot be used for both backhauling and delivery within the same period, and split deliveries are not permitted. We next address the second extension, [IRP-DB-MDC-II], which allows

Table 10
Case study parameters.

Parameter		Value	Parameter		Value
	d_{ipt}	[1, 5]		g_{ipt}	[10, 20]
$(i \in N)$	l_{ip}	1000	$(i \in N')$	l_{ip}	250
$(p \in P)$	s_{ip0}	5	$(p \in P')$	s_{ip0}	5
	h_{ip}	[60, 100] CAD		h_{ip}	[110, 150] CAD
	a_{1p}, a_{2p}, a_{3p}	30, 50, 100		b_1, b_2, b_3	100, 200, 400
	o_i	1000		q_{ip}	50
	$T_{\max}$	8 h		$c_{ij1}, c_{ij2}, c_{ij3}$	1, 1.15, 1.25 CAD per km
	e_{ip}	$1.5(h_{ip} + 2(\frac{\sum_{j\in\mathcal{M}}\sum_{k\in\mathcal{K}} c_{jik}}{\mathcal{K}+\mathcal{M}}))$			
# depots			2		
# periods			2		
# satellite points (delivery, backhauling)			(42, 17)		
# vehicles (trucks, commercial vehicles,trailers)			(4, 4, 4)		

Table 11
Performance of the algorithms for the case study instances.

	B&C			Matheuristic			CPU time (P-I, P-II)
	LB	OFV	CPU time	LB	OFV PI	OFV PII	
[IRP-DB-MDC-I]^1	88816	NF	21600	84878	91813	91270	(1205, 18000)
[IRP-DB-MDC-I]^2	84619	NF	21600	84552	91818	89432	(9273, 18000)
[IRP-DB-MDC-II]	81973	NF	21600	81877	88077	85417	(2095, 18000)
[IRP-DB-MDC-III]	154001	NF	21600	153095	159602	155602	(1244, 18000)
[IRP-DB-MDC]	81062	NF	21600	80810	87661	84977	(1646, 18000)

OFV: objective function value, PI: phase I, PII: Phase II, NF: no feasible solution obtained

a vehicle to backhaul after completing the delivery without splitting the deliveries. We then solve [IRP-DB-MDC-III], which applies the OU policy for the replenishment. Finally, [IRP-DB-MDC], as the primary model proposed in this paper, is solved. Although a comparison with HQ's current exact plan might be insightful, it is not possible due to the sensitive nature of HQ's operational data and the confidentiality agreement. Furthermore, HQ does not currently utilize a dedicated tool for optimizing delivery and backhauling strategies, which limits direct comparison with their present processes. Nevertheless, we have attempted to assess the performance of our model in relation to the closest scenario to HQ's operations. To this end, we evaluate HQ's current policy by solving the first extension, [IRP-DB-MDC-I], assuming that the red sites are serviced by Depot number 1 and the blue sites by Depot number 2, as per HQ's existing strategy. We then relax this assumption to allow our approach to determine the assignment. In the following sections and Tables 11 and 12, the results for the first scenario are denoted as [IRP-DB-MDC-I]^1, while the results for the scenario with relaxed assumptions are denoted as [IRP-DB-MDC-I]^2.

5.4.2. Case study solution

We solve the case study instances using the B&C and the matheuristic algorithms developed in this paper. The performance of these algorithms is summarized in Table 11. The results indicate that the B&C algorithm fails to produce a feasible solution within 6 h for any of the instances. However, the matheuristic algorithm effectively solves the instances with an optimality gap of less than 5 %. It is important to note that a time limit of 8 h ($T_{\max} = 8$) for each vehicle to complete each tour is quite restrictive for the problem. Therefore, the first iteration of the algorithm for the case study requires considerable time to find a feasible solution. We employ the technique introduced in the last paragraph of Section 4.3 to accelerate the algorithm and start the algorithm with an initial time limit of 14 h ($T_{\max} = 14$). Next, we reduce the time limit by one hour in each subsequent iteration until the seventh iteration, when the time limit reaches 8 h. The algorithm operates under this 8 h constraint for the remaining iterations, leading to the final solution accordingly. The results demonstrate that [IRP-DB-MDC] is solved in Phase I in less than an hour, and Phase II further improves the solution from Phase I by about 3 % within 5 h. This highlights the efficiency and effectiveness of the matheuristic approach in tackling the problem under the various scenarios considered. Next, we examine the objective function values of the solved instances. According to the results in Table 11, allowing backhauling after delivery improves the objective function by 4 %. Furthermore, permitting split deliveries results in an additional improvement, with the objective function enhancement reaching up to 5 %. Comparing the objective function values for [IRP-DB-MDC-I]^1 and [IRP-DB-MDC-I]^2 reveals that relaxing the constraint of assigning red sites to Depot 1 and blue sites to Depot 2 and allowing the optimization process to determine assignments yields a 2 % improvement in the objective function. The results also indicate that the solution for [IRP-DB-MDC] in Phase I is achieved in 1646 s. However, when split deliveries are avoided in [IRP-DB-MDC-II], the CPU time of the first phase is increased to 2095 s. Avoiding backhauling after delivery resulted in a more significant rise in CPU time, with Phase I taking 9273 s to solve [IRP-DB-MDC-I]^2. The algorithm obtains the solution within acceptable limits, as the total CPU time to solve [IRP-DB-MDC] is less than 7 h, aligning with HQ's standard practice of running this algorithm overnight through the batch process.

Table 12
Case study detailed solutions.

		[IRP-DB-MDC-I]^1	[IRP-DB-MDC-I]^2	[IRP-DB-MDC-II]	[IRP-DB-MDC-III]	[IRP-DB-MDC]
Routing cost		22971	21880	17539	13430	17227
Inventory cost		68299	67551	67877	142172	67750
#tours by each	pickup	16 (6667)	13 (5879)	13 (5929)	11 (3338)	14 (6139)
vehicle type	van	18 (6045)	16 (6966)	15 (5886)	12 (5075)	13 (5039)
(distance in km)	trailer	18 (7481)	17 (6391)	12 (3873)	9 (3404)	10 (4234)
	total	52	46	40	32	38
tour started	depot #1	39	27	24	19	23
from	depot #2	13	19	16	13	15

Further analysis of the detailed solutions provided in Table 12 highlights the benefits of our proposed model. In the [IRP-DB-MDC-I]^2 solution, where vehicles are restricted to either delivery or backhauling and split deliveries are not permitted, the routing cost is 21,880 CAD, and the inventory cost is 67,551 CAD. This represents an improvement over the [IRP-DB-MDC-I]^1 solution; however, the costs remain higher than those in other scenarios and require a total of 46 tours. The distribution of tours among vehicle types is as follows: pickups perform 13 tours, vans perform 16 tours, and trailers perform 17 tours. Out of the total 46 tours, 23 are delivery tours and 23 are backhauling tours. In contrast, the [IRP-DC-MDC-II] model, which allows backhauling to be started after delivery is finished, demonstrates significant improvements in total costs. The routing cost decreases to 17,539 CAD, although the inventory cost slightly increases to 67,877 CAD. More importantly, the total number of tours required reduces to 40, reflecting a more optimized vehicle fleet use. The final model, [IRP-DB-MDC], further enhances these benefits by reducing the routing cost to 17,227 CAD and decreasing the total number of tours to 38, where 30 tours have both delivery and backhauling. These findings highlight the effectiveness of the proposed model. A closer examination of the traveled distances by each vehicle type reveals another key advantage of our model. In the [IRP-DB-MDC-I]^2 model, pickups travel 5,879 km, vans travel 6,966 km, and trailers travel 6,391 km (a total of 19,236 km). The [IRP-DC-MDC-II] model shows improvement, with pickups traveling 5,929 km, vans 5,886 km, and trailers only 3,873 km, reflecting better route optimization (a total of 15,688 km). However, the most notable reduction in traveled distance is seen in the [IRP-DB-MDC] model, where pickups travel 6,139 km, vans 5,039 km, and trailers only 4,234 km (a total of 15,412 km). This reduction in traveled distances is particularly significant for trailers, which typically have higher fuel consumption than smaller vehicles like pickups and vans. By decreasing the distance covered by trailers, our model lowers fuel costs and reduces greenhouse gas emissions and environmental impact. The reduced travel distance for trailers thus contributes to economic and environmental sustainability, enhancing the overall benefits of the model. These results clearly demonstrate that allowing integrated delivery and backhauling, and split delivery not only reduces the total number of tours required but also minimizes the total cost and distance traveled by the fleet. The analysis strongly supports the practical benefits of adopting this model. Investigating the results of the [IRP-DB-MDC-III] that solves the model under the OU replenishment policy reveals that total costs are significantly higher than those of the other extensions. This increase is caused by the substantial rise in inventory level and the associated holding costs. Table 12 provides a detailed cost breakdown showing that routing costs under OU are 22 % lower than the main model that is solved under ML policy, while the holding costs are almost doubled. Under the OU policy, sites are replenished up to their TSL whenever they are visited, which leads to higher inventory levels and fewer vehicle trips, ultimately driving the substantial increase in holding costs.

Finally, we investigate the model's sensitivity to depot capacity, exploring how limits on delivery commodities and backhauling materials in depots affect performance and how permitting stockouts can prevent infeasibility. We follow the approach outlined by Bertazzi et al. (2013) to approximate the penalty cost of the stockout commodities and materials (e_{ip}) in the objective function (See Table 10). The results of this experiment are summarized in Table 13. In the main instance, each depot has a capacity of 250 units per period. Reducing capacity to 125, increases the main model's total holding cost by 70 %. Under the same settings, the [IRP-DB-MDC-IV] model, which allows stockouts at a penalty, lowers total holding costs by imposing a stockout cost. Further decreasing capacity to 100 increases holding costs and reduces the number of site visits, thereby decreasing routing costs. At a capacity of 50, the main model becomes infeasible; however, [IRP-DB-MDC-IV] still finds a solution, albeit with significant stockout penalties. Importantly, when depot capacity is constrained, initial inventory for delivery products can satisfy delivery demands; however, this is not applicable for backhauled materials, leading to higher delivery commodity holding costs and backhaul materials stockout costs. At lower capacity

Table 13
Case study for limited depot capacity.

Depot capacity	[IRP-DB-MDC]			[IRP-DB-MDC-IV]		
	Routing	Holding	Stockout	Routing	Holding	Stockout
125	17076	114528	–	18723	68941	19913
100	15324	202528	–	16587	78326	67818
50	infeasible			18220	139169	167423

levels, the stockout model leverages initial inventory to mitigate holding costs, yet the depot's limited backhaul capacity continues to generate substantial stockout penalties.

5.5. Managerial implications

Our study yields managerial implications, particularly for industries managing complex IRPs such as Hydro-Québec. The formulated problem can address various IRP challenges, such as heterogeneous vehicles, multiple depots, and backhauling trips. Moreover, we demonstrate how the model can be adapted for different scenarios by modifying assumptions about depot and sites' replenishment policy, integrated delivery and backhauling, and split deliveries to ensure practical applicability. We propose a solution approach that scales well to large instances. It operates in two phases, with the first phase offering a quick solution that is fast and easy to implement. This aspect is particularly valuable when decision-makers need a rapid solution, ensuring managers balance solution quality and implementation speed. We also demonstrated how utilizing empty vehicles for backhauling on their return trips to the depot can benefit our industrial partner. Over five periods, this strategy resulted in a 21 % reduction in total routing costs and a 19 % reduction in the total distance traveled by all vehicles compared to the case where split deliveries were not allowed, and delivery and backhauling trips could not be combined on the same route. The decrease in the total number of tours and traveled distances contributes to lower emissions and improved fleet utilization. Thus, our study addresses the theoretical challenges and provides practical, real-world applicability that can significantly improve operational efficiency. Finally, an extensive sensitivity analysis was conducted on various parameters and instance sizes. The results show that the solution approach is not significantly affected by varying parameters; however, the balance between holding and traveling costs can create marginal differences. In instances where routing costs comprise the largest portion of the objective function and holding costs the smallest, we observe greater improvement caused by adding valid inequalities. This is mainly because the added valid inequalities primarily strengthen the routing decisions of the model, leading to more significant gains when routing costs dominate. On the other hand, the average optimality gap of the heuristic algorithms is lower when the holding costs are increased, and the algorithm shows better performance. Overall, our framework offers a versatile tool for managers seeking to solve the IRP under realistic operational constraints.

6. Conclusion

We have presented a mixed-integer linear program to model an IRP focusing on real-world requirements. The problem involves the delivery of various commodities to satellite points and the backhauling of materials using heterogeneous vehicles. Our proposed model comprehensively incorporates several real-world complexities such as multi-commodity handling, multiple depots, travel time limitations, split deliveries, heterogeneous vehicles, and backhauling. By considering all these features of real-world problems, we provide a general model that can be readily applied to various real-world scenarios. Furthermore, recognizing that real instances often employ different replenishment policies, we demonstrate how the model can be modified to accommodate those variations. We have developed a branch-and-cut algorithm and a logic-based decomposition method to tackle the complexity of large instances that commercial solvers cannot solve efficiently. The efficiency of our algorithms has been evaluated using both synthetic instances and a real-world case study provided by our industrial partner, Hydro-Québec. Our proposed algorithms have demonstrated the capability to find optimal solutions for small-sized instances (up to 20 sites) and near-optimal solutions for medium to large-size instances (60–100 sites), outperforming both approaches that solve the problem with and without the B&C algorithm in a given CPU time limit. Notably, for large-size instances (100 sites), the matheuristic approach achieved a 3 % optimality gap in less than two hours, while the commercial solver failed to generate a feasible solution within the same time limit. These results highlighted the practical applicability of our models and solution approaches for addressing complex and large-scale IRPs in real-world settings.

CRediT authorship contribution statement

Farzad Avishan: Writing – review & editing, Writing – original draft, Validation, Software, Methodology, Investigation, Data curation, Conceptualization; **Amira Dems:** Writing – review & editing, Validation, Funding acquisition, Conceptualization; **Yossiri Adulyasak:** Writing – review & editing, Supervision, Funding acquisition, Conceptualization; **Okan Arslan:** Writing – review & editing, Supervision, Funding acquisition, Conceptualization; **Jean-François Cordeau:** Writing – review & editing, Supervision, Funding acquisition, Funding acquisition.

Data availability

The data is explained in the text of the manuscript.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

The authors acknowledge the support of the Digital Research Alliance of Canada for providing the computational resources used for running the computational experiments. This work was supported by Mitacs through the Mitacs Accelerate program IT23679. We are grateful to the anonymous reviewers for their valuable feedback, which helped improve the quality of the paper.

References

A. Guimarães, T., C. Coelho, L., M. Schenekemberg, C., T. Scarpin, C., 2019. The two-echelon multi-depot inventory-routing problem. Comput. Oper. Res. 101, 220–233.
Absi, N., Archetti, C., Dauzère-Pérès, S., Feillet, D., 2015. A two-phase iterative heuristic approach for the production routing problem. Transp. Sci. 49 (4), 784–795.
Adulyasak, Y., Cordeau, J.F., Jans, R., 2014. Formulations and branch-and-cut algorithms for multivehicle production and inventory routing problems. INFORMS J. Comput. 26 (1), 103–120.
Adulyasak, Y., Cordeau, J.F., Jans, R., 2015. Benders decomposition for production routing under demand uncertainty. Oper. Res. 63 (4), 851–867.
Agra, A., Christiansen, M., Wolsey, L., 2022. Improved models for a single vehicle continuous-time inventory routing problem with pickups and deliveries. Eur. J. Oper. Res. 297 (1), 164–179.
Andersson, H., Hoff, A., Christiansen, M., Hasle, G., Løkketangen, A., 2010. Industrial aspects and literature survey: combined inventory management and routing. Comput. Oper. Res. 37 (9), 1515–1536.
Arab, R., Ghaderi, S.F., Tavakkoli-Moghaddam, R., 2020. Bi-objective inventory routing problem with backhauls under transportation risks: two meta-heuristics. Transp. Lett. 12 (2), 113–129.
Archetti, C., Bertazzi, L., Laporte, G., Speranza, M.G., 2007. A branch-and-cut algorithm for a vendor-managed inventory-routing problem. Transp. Sci. 41 (3), 382–391.
Archetti, C., Boland, N., Grazia Speranza, M., 2017. A matheuristic for the multivehicle inventory routing problem. INFORMS J. Comput. 29 (3), 377–387.
Archetti, C., Christiansen, M., Speranza, M.G., 2018. Inventory routing with pickups and deliveries. Eur. J. Oper. Res. 268 (1), 314–324.
Archetti, C., Ljubić, I., 2022. Comparison of formulations for the inventory routing problem. Eur. J. Oper. Res. 303 (3), 997–1008.
Archetti, C., Speranza, M.G., Boccia, M., Sforza, A., Sterle, C., 2020. A branch-and-cut algorithm for the inventory routing problem with pickups and deliveries. Eur. J. Oper. Res. 282 (3), 886–895.
Bell, W.J., Dalberto, L.M., Fisher, M.L., Greenfield, A.J., Jaikumar, R., Kedia, P., Mack, R.G., Prutzman, P.J., 1983. Improving the distribution of industrial gases with an on-line computerized routing and scheduling optimizer. Interfaces 13 (6), 4–23.
Bertazzi, L., Bosco, A., Guerriero, F., Lagana, D., 2013. A stochastic inventory routing problem with stock-out. Transp. Res. Part C 27, 89–107.
Bertazzi, L., Coelho, L.C., De Maio, A., Laganà, D., 2019. A matheuristic algorithm for the multi-depot inventory routing problem. Transp. Res. Part E 122, 524–544.
Bertazzi, L., Speranza, M.G., 2012. Inventory routing problems: an introduction. EURO J. Transp. Logist. 1, 307–326.
Bertazzi, L., Speranza, M.G., 2013. Inventory routing problems with multiple customers. EURO J. Transp. Logist. 2 (3), 255–275.
Campbell, A., Clarke, L., Kleywegt, A., Savelsbergh, M., 1998. The inventory routing problem. In: Fleet Management and Logistics. Springer, pp. 95–113.
Cheng, C., Yang, P., Qi, M., Rousseau, L.M., 2017. Modeling a green inventory routing problem with a heterogeneous fleet. Transp. Res. Part E 97, 97–112.
Chitsaz, M., Cordeau, J.F., Jans, R., 2019. A unified decomposition matheuristic for assembly, production, and inventory routing. INFORMS J. Comput. 31 (1), 134–152.
Chiu, A., Angulo, G., Larrain, H., 2024. Optimizing the long-term costs of an inventory routing problem using linear relaxation. Transp. Res. Part E 183, 103447.
Clarke, G., Wright, J.W., 1964. Scheduling of vehicles from a central depot to a number of delivery points. Oper. Res. 12 (4), 568–581.
Coelho, L.C., Cordeau, J.F., Laporte, G., 2014. Thirty years of inventory routing. Transp. Sci. 48 (1), 1–19.
Coelho, L.C., Laporte, G., 2013a. A branch-and-cut algorithm for the multi-product multi-vehicle inventory-routing problem. Int. J. Prod. Res. 51 (23–24), 7156–7169.
Coelho, L.C., Laporte, G., 2013b. The exact solution of several classes of inventory-routing problems. Comput. Oper. Res. 40 (2), 558–565.
Deif, I., Bodin, L., 1984. Extension of the clarke and wright algorithm for solving the vehicle routing problem with backhauling. In: Proceedings of the Babson Conference on Software Uses in Transportation and Logistics Management. Babson Park, MA, pp. 75–96.
Desaulniers, G., Rakke, J.G., Coelho, L.C., 2016. A branch-price-and-cut algorithm for the inventory-routing problem. Transp. Sci. 50 (3), 1060–1076.
Diabat, A., Bianchessi, N., Archetti, C., 2024. On the zero-inventory-ordering policy in the inventory routing problem. Eur. J. Oper. Res. 312 (3), 1024–1038.
Dinh, N.M., Archetti, C., Bertazzi, L., 2023. The inventory routing problem with split deliveries. Networks 82 (4), 400–413.
Feng, Y., Che, A., Tian, N., 2024. Robust inventory routing problem under uncertain demand and risk-averse criterion. Omega 127, 103082. https://doi.org/10.1016/j.omega.2024.103082
Golsefidi, A.H., Jokar, M.R.A., 2020. A robust optimization approach for the production-inventory-routing problem with simultaneous pickup and delivery. Comput. Ind. Eng. 143, 106388.
Grønhaug, R., Christiansen, M., Desaulniers, G., Desrosiers, J., 2010. A branch-and-price method for a liquefied natural gas inventory routing problem. Transp. Sci. 44 (3), 400–415.
Gurobi Optimization, 2024. Callback Functions in Python. Accessed: 2024-10-15. https://www.gurobi.com/documentation/current/refman/py_cb_s.html.
Hagberg, A., Swart, P.J., Schult, D.A., 2008. Exploring Network Structure, Dynamics, and Function using NetworkX. Technical Report. Los Alamos National Laboratory (LANL), Los Alamos, NM (United States).
Hewitt, M., Nemhauser, G., Savelsbergh, M., Song, J.H., 2013. A branch-and-price guided search approach to maritime inventory routing. Comput. Oper. Res. 40 (5), 1410–1419.
Iassinovskaia, G., Limbourg, S., Riane, F., 2017. The inventory-routing problem of returnable transport items with time windows and simultaneous pickup and delivery in closed-loop supply chains. Int. J. Prod. Econ. 183, 570–582.
Koç, Ç., Laporte, G., 2018. Vehicle routing with backhauls: review and research perspectives. Comput. Oper. Res. 91, 79–91.
Le, T., Diabat, A., Richard, J.P., Yih, Y., 2013. A column generation-based heuristic algorithm for an inventory routing problem with perishable goods. Optim. Lett. 7, 1481–1502.
Lefever, W., Touzout, F.A., Hadj-Hamou, K., Aghezzaf, E.H., 2021. Benders' Decomposition for robust travel time-constrained inventory routing problem. Int. J. Prod. Res. 59 (2), 342–366.
Lei, J., Che, A., Van Woensel, T., 2024. Collection-disassembly-delivery problem of disassembly centers in a reverse logistics network. Eur. J. Oper. Res. 313 (2), 478–493.
Li, R., Cui, Z., Kuo, Y.H., Zhang, L., 2023. Scenario-based distributionally robust optimization for the stochastic inventory routing problem. Transp. Res. Part E 176, 103193.
Londoño, J., Bastidas, J., González, P., Escobar, J., 2023. Inventory routing problem with backhaul considering returnable transport items collection. Int. J. Ind. Eng. Comput. 14 (4), 837–862.
Mahjoob, M., Fazeli, S.S., Tavassoli, L.S., Mirmozaffari, M., Milanlouei, S., 2021. A green multi-period inventory routing problem with pickup and split delivery: a case study in flour industry. Sustain. Oper. Comput. 2, 64–70.
Mahmutoğulları, Ö., Yaman, H., 2023. A branch-and-cut algorithm for the inventory routing problem with product substitution. Omega 115, 102752.
Neves-Moreira, F., Almada-Lobo, B., Cordeau, J.F., Guimarães, L., Jans, R., 2019. Solving a large multi-product production-routing problem with delivery time windows. Omega 86, 154–172.
Neves-Moreira, F., Almada-Lobo, B., Guimarães, L., Amorim, P., 2022. The multi-product inventory-routing problem with pickups and deliveries: mitigating fluctuating demand via rolling horizon heuristics. Transp. Res. Part E 164, 102791.
Ortega, E.J.A., Malicki, S., Doerner, K.F., Minner, S., 2024. Stochastic inventory routing with dynamic demands and intra-day depletion. Comput. Oper. Res. 163, 106503.

Ramkumar, N., Subramanian, P., Narendran, T.T., Ganesh, K., 2012. Mixed integer linear programming model for multi-commodity multi-depot inventory routing problem. Opsearch 49, 413–429.
Rau, H., Budiman, S.D., Widyadana, G.A., 2018. Optimization of the multi-objective green cyclical inventory routing problem using discrete multi-swarm PSO method. Transp. Res. Part E 120, 51–75.
Ross, D.F., 2010. Introduction to supply chain management technologies.
Savelsbergh, M., Song, J.H., 2008. An optimization algorithm for the inventory routing problem with continuous moves. Comput. Oper. Res. 35 (7), 2266–2282.
Shaabani, H., 2022. A literature review of the perishable inventory routing problem. Asian J. Ship. Logist. 38 (3), 143–161.
Skålnes, J., Ahmed, M.B., Hvattum, L.M., Stålhane, M., 2024. New benchmark instances for the inventory routing problem. Eur. J. Oper. Res. 313 (3), 992–1014.
Skålnes, J., Vadseth, S.T., Andersson, H., Stålhane, M., 2023. A branch-and-cut embedded matheuristic for the inventory routing problem. Comput. Oper. Res. 159, 106353.
Sofianopoulou, S., Mitsopoulos, I., 2021. A review and classification of heuristic algorithms for the inventory routing problem. Int. J. Oper. Res. 41 (2), 282–298.
Song, J.H., Furman, K.C., 2013. A maritime inventory routing problem: practical approach. Comput. Oper. Res. 40 (3), 657–665.
Soysal, M., 2016. Closed-loop inventory routing problem for returnable transport items. Transp. Res. Part D 48, 31–45.
Soysal, M., Çimen, M., Belbağ, S., Toğrul, E., 2019. A review on sustainable inventory routing. Comput. Ind. Eng. 132, 395–411.
Tiniç, G.Ö., Koca, E., Yaman, H., 2021. An exact solution approach for the inventory routing problem with time windows. Comput. Oper. Res. 134, 105371.
Toth, P., Vigo, D., 2002. The Vehicle Routing Problem. SIAM.
Van Anholt, R.G., Coelho, L.C., Laporte, G., Vis, I.F.A., 2016. An inventory-routing problem with pickups and deliveries arising in the replenishment of automated teller machines. Transp. Sci. 50 (3), 1077–1091.
Wang, S., Chu, F., 2022. A decomposition-based heuristic method for inventory routing problem. IEEE Trans. Intell. Transp. Syst. 23 (10), 18352–18360.
Yavrucu, E., Soysal, M., Sel, Ç., Çimen, M., Hamzadayı, A., 2025. Modeling a closed-loop inventory routing problem for returnable transport items under horizontal logistics collaborations and dynamic demand structure. Transp. Lett. 17 (3), 470–490.
Zhang, Y., Chu, F., Che, A., Li, Y., 2024. Closed-loop inventory routing problem for perishable food with returnable transport items selection. Int. J. Prod. Res. 62 (1–2), 501–521.
Zhou, K., 2024. Benders decomposition for inventory routing problem under milk-run. In: 2024 7th International Conference on Advanced Algorithms and Control Engineering (ICAACE). IEEE, pp. 1509–1512.