CIRRELT

CIRRELT-2024-38

Inventory Routing with Heterogeneous Vehicles and Backhauling

Farzad Avishan
Amira Dems
Yossiri Adulyasak
Okan Arslan
Jean-François Cordeau

November 2024

Bureau de Montréal
Université de Montréal
C.P. 6128, succ. Centre-Ville
Montréal (Québec) H3C 3J7
Tél : 1-514-343-7575
Télécopie : 1-514-343-7121

Bureau de Québec
Université Laval,
2325, rue de la Terrasse
Pavillon Palasis-Prince, local 2415
Québec (Québec) G1V 0A6
Tél : 1-418-656-2073
Télécopie : 1-418-656-2624

Inventory Routing with Heterogeneous Vehicles and Backhauling

Farzad Avishan[1,2,*], Amira Dems[3], Yossiri Adulyasak[2], Okan Arslan[1,2], Jean-François Cordeau[1,2]

1. Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)
2. HEC Montréal, 3000 chemin de la Côte-Sainte-Catherine, Montréal H3T 2A7, Canada
3. Hydro Québec Research Institute (IREQ), 1800 Bd Lionel-Boulet, Varennes J3X 1S1, Canada

Abstract. Motivated by a real-world problem faced by Hydro-Québec, a large Canadian utility, this study investigates an inventory routing problem involving the efficient distribution of commodities to sites and the backhauling of materials from these sites to the depots. We minimize the total cost of distributing multiple products with a fleet of capacitated heterogeneous vehicles departing from various depots to customers while also backhauling multiple materials back to the depots, all within a discrete and finite time horizon. In each period, a customer's delivery and backhauling can be split and satisfied by multiple vehicles. The total cost includes unit inventory holding costs at customer sites and transportation costs. We provide 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 that solves the assignment and routing problems iteratively. This study is the first to consider multiple depots, split delivery, heterogeneous vehicles, multiple products, and backhauling in a single model, presenting both exact and heuristic solution methods to solve the problem efficiently. An extensive numerical study is conducted on synthetic instances to evaluate the performance of the model 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 commodity distribution and hazardous material backhauling network to demonstrate the real-world applicability of the model. We show how the company may benefit from integrating the delivery and backhauling networks. 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.

Keywords: inventory routing problem, backhauling, branch-and-cut, matheuristic

Acknowledgements: 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.

---

* Corresponding author: farzad.avishan@hec.ca

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

1. Introduction

A typical supply chain involves managing the production, inventory, and distribution of finished products to multiple customers through various routes, often by minimizing costs. When the production quantities at the plant are known, and the decisions are limited to inventory and route planning, the optimization problem is often referred to as the *inventory routing problem* (IRP) (Campbell et al. 1998; Chitsaz, Cordeau, and Jans 2019). 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. This problem has drawn significant attention from the research community due to the economic advantages of integrating transportation and inventory management, and it is considered a complex optimization problem (Coelho, Cordeau, and Laporte 2014; Archetti and Ljubić 2022).

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 strict 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, Archetti, and Bertazzi 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). Motivated by an industrial application at *Hydro-Québec* (HQ), we consider a problem in networks where pickups are only permitted after completing deliveries to improve efficiency and reduce operational costs. In our case, this is required because the materials to be transported in the backhauling stage are hazardous.

In this study, we consider an IRP with *delivery and backhauling* (IRP-DB), inspired by our industrial partner, HQ. We explore the benefits of integrating delivery and backhauling while addressing real-world challenges of the IRP, such as split delivery, heterogeneous vehicles, and multiple depots. We introduce a mathematical model for the IRP-DB that incorporates *multi-depot and multi-commodity* (MDC) considerations, referred to as the IRP-DB-MDC. The problem involves multiple depots, each with varying supply capacities. 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. Additionally, we consider the possibility of split deliveries, where a single site may receive commodities from more than one vehicle. Our fleet comprises heterogeneous vehicles with varying capacities and costs, and each vehicle can make at most one tour in each time period. Note that the aforementioned assumptions in this general problem description can be relaxed, and the problem can easily be simplified to the basic IRP form. In the context of this research, the trips and activities of vehicles collecting materials from different points are referred to as backhauling since it happens after completing deliveries. We refer to the commodities being delivered as delivery commodities and the materials being backhauled as backhauling materials.

In this study, we employ a *branch-and-cut* (B&C) algorithm that effectively addresses the problem, demonstrating improved CPU time and performance compared to solving the *mixed-integer programming* (MIP) model directly by a commercial solver. The proposed algorithms use valid inequalities to strengthen the formulation. Additionally, we introduce a matheuristic approach to handle large-size instances where exact methods fail to provide efficient solutions. To evaluate the effectiveness of the proposed model and solution approaches, we test them on synthetic instances. The heuristic approach can produce good solutions for instances with up to 100 sites, $3-5$ vehicles, $15-30$ commodities, and $3-5$ periods within two hours. The effectiveness of the model and the solution approaches is further demonstrated through their application to networks that are representative of the HQ distribution and backhauling network. Benchmark models, considering disjoint backhauling and delivery operations while avoiding split delivery, are also formulated to compare solutions across all models comprehensively. Our study demonstrates the practical utility and the competitional efficiency of the proposed solution approaches in complex, realistic scenarios, offering valuable insights. These include strategies for optimizing vehicle routes, improving vehicle allocation, and minimizing transportation costs within intricate logistics networks, ultimately enhancing logistics management and ensuring efficient material transportation throughout the network. Furthermore, numerical instances show that the proposed model can significantly reduce the total cost of the network compared to benchmark models. In our case study, this improvement is particularly notable for routing costs, with a 21% reduction, highlighting the model's efficiency in cost minimization.

The subsequent sections are organized as follows. Section 2 reviews the relevant literature and discusses our contributions. 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.

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, Che, and Tian 2024; Ortega et al. 2024); IRP with time windows (Tiniç, Koca, and Yaman 2021); sustainable and green IRP (Soysal et al. 2019; Rau, Budiman, and Widyadana 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, known as IRP *with delivery and pickup* (IRP-DP), addresses various logistical challenges, but few contributions are made in this field. Archetti, Christiansen, and Speranza (2018) have studied a single product IRP-DP 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-DP 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, Che, and Van Woensel (2024) have further expanded the application of IRP-DP in a different domain and have applied IRP-DP to the collection, disassembly, and delivery problems, formulating a scenario where products are collected from centres, 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 if visited, while the ML policy allows flexible delivery quantities if stock capacities are not exceeded (Mahmutoğulları and Yaman 2023). Adulyasak, Cordeau, and Jans (2014) study a multi-vehicle IRP and address the problems considering the ML and OU inventory replenishment policies and Dinh, Archetti, and Bertazzi (2023) examine the advantages of incorporating split deliveries into the problem under both ML and OU replenishment strategies. In addition to these policies, Diabat, Bianchessi, and Archetti (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, Angulo, and Larrain (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 algorithms are used in some studies (Desaulniers, Rakke, and Coelho 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, Cordeau, and Jans (2014), Archetti, Boland, and Grazia Speranza (2017) and Mahmutoğulları and Yaman (2023) have taken important steps to improve the B&C algorithms for different variants of problems. 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, Cordeau, and Jans (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. For more information on heuristic algorithms for IRP, we refer the readers to Andersson et al. (2010) and to Sofianopoulou and Mitsopoulos (2021).

A 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. Furthermore, the formulated model considers a multi-depot network with heterogeneous vehicles that deliver and backhaul multiple products. 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. For a comprehensive review of VRP-B, readers are referred to Santos et al. (2020) and Koç and Laporte (2018). Despite the extensive research on VRP-B, the specific case of IRP, where vehicles perform backhauling tasks within the framework of inventory routing, remains under-researched. Only a few studies have addressed this particular problem. Arab, Ghaderi, and Tavakkoli-Moghaddam (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. On the other hand, Londoño et al. (2023) formulated a model for identical vehicles while ignoring time constraints, with their solution limited to instances involving a maximum of 15 customers. A summary of recent studies on IRP-DP and IRP-DB is presented in Table 1. The table highlights how our research differs from the other studies in the literature.

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

Reference paper	Multiple depots	Multiple products	Time limits	Split delivery	Heterogeneous vehicles	PD/DB	Solution approach
Ramkumar et al. (2012)	✓	✓	✓	✓	-	PD	No method
Van Anholt et al. (2016)	✓	-	-	✓	-	PD	B&C, heuristic
Iassinovskaia, Limbourg, and Riane (2017)	-	-	-	✓	-	PD	B&C
Archetti, Christiansen, and Speranza (2018)	-	-	-	-	-	PD	B&C
Archetti et al. (2020)	-	-	-	-	-	PD	B&C
Arab, Ghaderi, and Tavakkoli-Moghaddam (2020)	-	✓	-	-	✓	DB	Meta-heuristics
Agra, Christiansen, and Wolsey (2022)	-	-	-	✓	-	PD	B&C
Neves-Moreira et al. (2022)	-	✓	-	-	✓	PD	B&C, heuristic
Londoño et al. (2023)	-	-	-	-	-	DB	No method
This research	✓	✓	✓	✓	✓	DB	B&C, heuristic

B&C: branch-and-cut algorithm; PD: pickup and delivery, DB: delivery and backhauling

In this study, we propose a model for the IRP-DB that incorporates multiple depots, multiple commodities, heterogeneous vehicles, a maximum travel time for each route, and split deliveries. Most solution approaches for the IRP are developed to exploit conventional basic problem characteristics, which may limit their practical applicability (Song and Furman 2013). Our study offers a generalized solution approach that can be applied to a wide range of real-world IRPs of similar complexity. We tackle this problem using both exact and matheuristic algorithms. These methods are tested through extensive numerical experiments, providing insights into the efficiency and scalability of our methods. Our matheuristic approach efficiently solves problems with up to 100 sites in less than two hours, achieving an average optimality gap of 3% for the 100-site instances and less than 2% on average across all instances. We generate benchmark instances for IRP-DB-MDC with heterogeneous vehicles to ensure comparability based on the well-established IRP instances provided by Archetti et al. (2007). We also implement our model in a real-world case study to demonstrate its practical relevance. This application highlights the real-world applicability of our methods, particularly in optimizing the logistics management of our industrial partner. Finally, we formulate benchmark models to facilitate comprehensive comparisons with other approaches in the literature.

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 benchmark 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 $M = \{d_1, d_2, \ldots, d_{\mathcal{M}}\}$ represents the set of depots serving as the origin and destination points for vehicle routes, $N = \{1, 2, \ldots, \mathcal{N}\}$ denotes the set of sites for product deliveries, and $N' = \{\mathcal{N}+1, \mathcal{N}+2, \ldots, \mathcal{N}'\}$ 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. 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.

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. Figure 1 illustrates the delivery and backhauling network. The sequence entails the delivery of commodities to sites, succeeded 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, Cordeau, and Jans (2014) and in Archetti, Christiansen, and Speranza (2018).

[IRP-DB-MDC] :

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

Table 2: Summary of all notation

Sets	
P	Set of delivery commodities (indexed by $p = 1, 2, \ldots, \mathcal{P}$)
P'	Set of backhauling materials (indexed by $p = \mathcal{P}+1, \mathcal{P}+2, \ldots, \mathcal{P}'$)
T	Set of time periods (indexed by $t = 1, 2, \ldots, \mathcal{T}$)
N	Set of sites for delivery (indexed by $i, i', j = 1, 2, \ldots, \mathcal{N}$)
N'	Set of sites for backhauling (indexed by $i, i', j = \mathcal{N}+1, \mathcal{N}+2, \ldots, \mathcal{N}'$)
M	Set of depots (indexed by $i, j = d_1, d_2, \ldots, d_{\mathcal{M}}$); d_k is the depot where vehicle k should start and end its trip
K	Set of vehicles (indexed by $k = 1, 2, \ldots, \mathcal{K}$); i_k is the depot where vehicle i 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 $E_{N,N'} = \{(i,j) \mid i \in N, j \in N'\}$, $E_{N,N'} = \{(i,j) \mid i \in N, j \in N'\}$, $E_N = \{(i,j) \mid i \in N, j \in N, i < j\}$, $E_{N'} = \{(i,j) \mid i \in N', j \in N', i < j\}$
$E(S)$	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}	Quantity of backhauling material of type $p \in P'$ at site $i \in N'$ accumulated 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 $p \in P'$)
o_i	Global storage capacity of site $i \in N \cup N'$
a_{kp}	Capacity of vehicle $k \in K$ for commodiy 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 $p \in P'$)
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 $t \in T$) $D_{ikt} = \min\left\{o_i, b_k, \sum_{p \in P} \sum_{t'=t}^{\mathcal{T}} d_{ipt'}\right\}$ $\quad i \in N, k \in K, t \in T$ $D'_{ikt} = \min\left\{o_i, b_k, \sum_{p \in P'} \left(s_{ip0} + \sum_{t'=1}^{t} d_{ipt'}\right)\right\}$ $\quad i \in N', k \in K, t \in T$
Decision variables	
s_{ipt}	Inventory level of commodity $p \in P$ or backhauling material $p \in P'$ at depot $i \in M$ in period $t \in T$
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 equals 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 equals 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

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

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

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

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

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

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

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

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

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

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

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

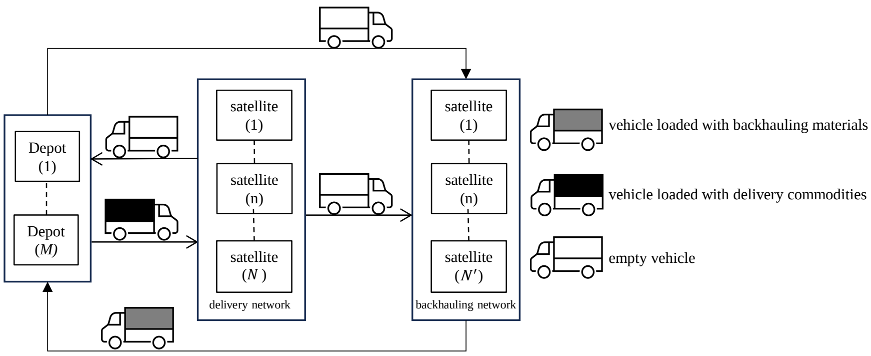

Figure 1: The delivery and backhauling network. Vehicles can start backhauling after delivery, but the reverse delivery sequence after backhauling is prohibited.

$$\sum_{i' \in N} \sum_{K_i} u_{i'pkt} \leq l_{ip} \qquad i \in M, p \in P, t \in T \tag{12}$$

$$\sum_{i' \in N} \sum_{K_i} u_{i'pkt} \leq l'_{ip} \qquad i \in M, p \in P', t \in T \tag{13}$$

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

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

$$z_{ikt} \leq f_{ik} \qquad i \in M, k \in K, t \in T \tag{16}$$

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

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

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

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

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

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

$$s_{ipt}, u_{ipkt} \geq 0 \qquad i \in N, N', M, p \in P, P', k \in K, t \in T. \tag{23}$$

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 does not exceed the associated capacity in the depot. Constraints (14) and (15) allow the delivery of commodities to a site and 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, Cordeau, and Jans (2014). Constraints (20)−(23) impose domain restriction of the variables. Notice that although [IRP-DB-MDC] may initially appear decomposable over time period, constraints (2) and (3) introduce dependencies between consecutive periods. These constraints link each time period to its predecessor, thereby preventing decomposition.

3.2. Benchmark Models

We formulate two benchmark models to assess the solution of the proposed model. The first benchmark, 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 scenario. The second benchmark model, referred to as [IRP-DB-MDC-II], allows a vehicle to perform backhauling after completing its delivery tasks but forbids split deliveries. Table 3 summarizes the characteristics of the main model and the benchmark models.

Table 3: Summary of benchmark models

	Model	Split delivery	Integrated backhauling and delivery
Benchmark models	[IRP-DB-MDC-I]	No	No
	[IRP-DB-MDC-II]	No	Yes
Main model	[IRP-DB-MDC]	Yes	Yes

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] :

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

$$\text{s.t. } (1) - (17), (19) - (23)$$

$$\sum_{(i,j)\in E_{N-N'}} x_{ijkt} \leq 1 \qquad k \in K, t \in T \tag{24}$$

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

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

[IRP-DB-MDC-II] :

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

$$\text{s.t. } (1) - (23), (25)$$

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. (2020) and Adulyasak, Cordeau, and Jans (2014), we use valid inequalities to strengthen our formulation. In this section, we derive a set of valid inequalities specifically designed for our multi-commodity network with heterogeneous vehicles and backhauling. The inequalities introduced in the referenced studies have been adapted to suit our problem.

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 (26) prevents stockouts:

$$\sum_{k\in K} \sum_{t=1}^{t'} z_{0kt} \geq \left\lceil \frac{\kappa}{b_{\max}} \right\rceil \tag{26}$$

where,

$$
\begin{aligned}
t'_{ip} &= \text{argmin}_{1\le t\le \mathcal{T}} \left\{ \sum_{t''=1}^{t} d_{ipt''} - s_{ip0} > 0 \right\} \\
t' &= \min_{i\in N, p\in P} t'_{ip} \\
\kappa &= \sum_{i\in N}\sum_{p\in P} \max\{0, \sum_{t''}^{t'} d_{ipt''} - s_{ip'0}\} \\
b_{\max} &= \max_{k\in K} b_k.
\end{aligned}
$$

Inequalities (27) 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 (28) are an extension of (27) 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 (29) ensure that the total number of visits to a site between periods should be sufficient to prevent stockouts:

$$\left(1-\sum_{k\in K} z_{ik,t+1}\right) d_{ip,t+1} \le s_{ipt} \qquad i\in N, p\in P, t\in T \tag{27}$$

$$\left(\sum_{t'=0}^{e} d_{ip,t-t'}\right)\left(1-\sum_{k\in K}\sum_{t'=0}^{e} z_{ik,t-t'}\right) \le s_{ip,t-e-1} \qquad i\in N, p\in P, t\in T, e\in[0,1,\dots,t-1] \tag{28}$$

$$\sum_{t'=0}^{e} d_{ip,t-t'} - s_{ip,t-e-1} \le \sum_{k\in K}\sum_{t'=0}^{e} z_{ik,t-t'} a_{kp} \qquad i\in N, p\in P, t\in T, e\in[0,1,\dots,t-1]. \tag{29}$$

Inequalities (30) 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 (31) 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 (32)−(34):

$$s_{ipt-1} + \sum_{t'=t}^{t+e} g_{ipt'} \left(1 - \sum_{k \in K} \sum_{t'=t}^{t+e} z_{ikt'}\right) \leq q_{ip} \qquad i \in N', p \in P', t \in T, e \in [0, 1, \dots, t-1] \tag{30}$$

$$\sum_{t'=0}^{e} g_{ip,t-t'} + s_{ip,t-e-1} - q_{ip} \leq \sum_{k \in K} \sum_{t'=0}^{e} z_{ik,t-t'} a_{kp} \quad i \in N', p \in P', t \in T, e \in [0, 1, \dots, t-1] \tag{31}$$

$$z_{ikt} \leq \sum_{i \in M} z_{ikt} \qquad i \in N \cup N', k \in K, t \in T \tag{32}$$

$$x_{ijkt} \leq z_{ikt} \qquad (i,j) \in E, k \in K, t \in T \tag{33}$$

$$x_{ijkt} \leq z_{jkt} \qquad (i,j) \in E, k \in K, t \in T. \tag{34}$$

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, Cordeau, and Jans 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 $\overline{z}_{ikt}$ and $\overline{x}_{ijkt}$, respectively. A graph for vehicle k in period t is generated from the set of nodes with $\overline{z}_{ikt} > 0$. The weights of the edges in the generated graph are equal to the $\overline{x}_{ijkt}$ 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 $2\overline{z}_{ikt}$, 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 $i' = \arg\max_{i \in \bar{N}_{kt}}(\overline{z}_{ikt})$ to the formulation where $\bar{N}_{kt} = \{i \in N \mid \overline{z}_{ikt} > 0\}$. Note that we have utilized the NetworkX (Hagberg, Swart, and Schult 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. Figure 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, Cordeau, and Jans (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,

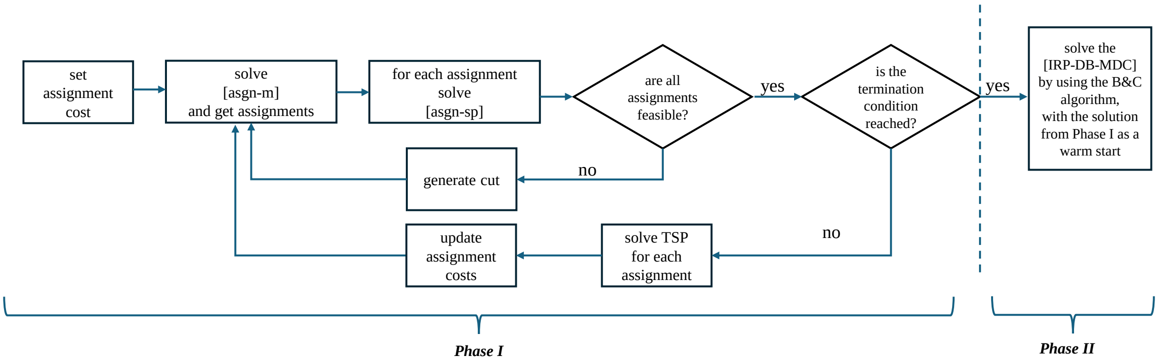

Figure 2: The algorithm generates an initial solution in Phase I, then uses the B&C to solve the optimization model using the initial solution as a warm start

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 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]:

$$[\text{asgn}] : \min \sum_{t\in T}\left(\sum_{k\in K}\sum_{i\in\{N+N'+M\}}\sigma_{vit}z_{ikt} + \sum_{i\in N}\sum_{p\in P}h_{ip}s_{ipt} + \sum_{i\in N'}\sum_{p\in P'}h_{ip}s_{ipt}\right)$$
$$\text{s.t. } (1) - (34).$$

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_{kit}, 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 σ_{vit} for $i \in M$ is calculated as $\frac{1}{N}\left(\sum_{i\in N}\sigma_{vit}\right)$. 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:

$$\sum_{i \in \overline{N}+\overline{N'}} z_{ik^*t} \leq \sum_{i \in \overline{N}+\overline{N'}} z^*_{ik^*t} - 1 - \alpha.$$

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:

$$\begin{aligned}
[\text{asgn-m}] : \min \sum_{t \in T} & \left(\sum_{k \in K} \sum_{i \in \{N+N'+M\}} \sigma_{vit} z_{ikt} + \sum_{i \in N} \sum_{p \in P} h_{ip} s_{ipt} + \sum_{i \in N'} \sum_{p \in P'} h_{ip} s_{ipt} \right) \\
& \text{s.t. } (2)-(16), (22)-(32) \\
[\text{asgn-sp}] : \min 0 & \\
& \text{s.t. } (1), (17)-(21), (33), (34).
\end{aligned}$$

[**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_{vit} to the cheapest insertion cost for inserting it into R_{kt}. For sites in N_{kt}, we update the visiting cost σ_{vit} to $(c_{i^p i} + c_{i i^s} - c_{i^s i^p})$. 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),(35)−(38) 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, y and y' are defined as auxiliary binary decision variables. Constraints (35) set the value of y_{kt} to 1 if vehicle k in period t does the delivery and constraints (36) set the value of y'_{kt} to one if vehicle k in period t does the backhauling. Constraints (37) prevent a vehicle from delivering and backhauling in the same period:

$$\sum_{i \in N} z_{ikt} \leq \mathcal{N} y_{kt} \qquad k \in K, t \in T \tag{35}$$

$$\sum_{i \in N'} z_{ikt} \leq \mathcal{N}' y'_{kt} \qquad k \in K, t \in T \tag{36}$$

$$y_{kt} + y'_{kt} \leq 1 \qquad k \in K, t \in T \tag{37}$$

$$y_{kt}, y'_{kt} \in \{0, 1\} \qquad k \in K, t \in T. \tag{38}$$

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. In the algorithm, there may be instances where a strict travel time constraint for vehicle tours prevents the proposed matheuristic 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}	$\max_{i \in N}(q_{ip} + d_{ipt}) \times \omega_k$	$a_{kp'}$	$\max_{i \in N'}(q_{ip'} + g_{ip't}) \times \omega_k$
q_{ip}	$d_{ip1} \times rnd_i$	q_{ip}	$g_{ip1} \times ran_i$
o_i	$\sum_{p \in P} q_{ip}, i \in N$	o_i	$\sum_{p \in P'} q_{ip}, i \in N'$
l_{ip}	$\max_{t \in \mathcal{T}} \left(\sum_{j \in \mathcal{N}} d_{jpt}\right) \times rnd_i$	$l_{ip'}$	$\max_{t \in \mathcal{T}} \left(\sum_{j \in \mathcal{N}'} g_{jpt}\right) \times rnd_i$
s_{ip0}	$(q_{ip} - d_{ip1})$	$s_{ip'0}$	$(q_{ip'} - g_{ip'1})$
b_k	$\sum_{p \in P} a_{kp}$	$T_{\max}$	$\text{avg}(c_{ijk}) \times (\mathcal{N} + \mathcal{N}') \times \mathcal{V}^{-1}$

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

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 VIs 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 reveals that selecting parameters from various intervals can impact the improvement amount. 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. Overall, the results presented in Table 7 show the relevance of integrating valid 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 frame work, #ins: number of instances

Next, we assess the performance of B&C compared to solving the [IRP-DB-MDC] by Gurobi with a time limit of 7200 seconds. 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 44 instances when the [IRP-DB-MDC] is solved without adding inequalities, whereas the B&C algorithm reduces this number to 24 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. 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. When DGF 7 and 8 are used to generate instances, we face the largest optimality gap. This occurs because the cost of traveling in the objective function has a higher proportion. 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.

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

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	130	-	7200	0	125	-	-	7200
$\mathcal{P}=10, \mathcal{P}'=5$	220	77	79	1.8%	4814	80	66	1.3%	1.3%	4808
$\mathcal{P}=20, \mathcal{P}'=10$	230	82	110	0.6%	5968	80	73	0.2%	0.7%	5091
$\mathcal{T}=3, \mathcal{K}'=3$	210	80	53	1.1%	5755	80	48	0.7%	0.7%	4976
$\mathcal{T}=5, \mathcal{K}'=5$	240	80	106	1.5%	5027	80	91	1.0%	1.2%	4923
DGF										
1	50	20	19	0.2%	5019	20	14	0.2%	0.2%	4412
2	60	20	21	1.2%	5407	20	20	0.4 %	0.5 %	4882
3	55	20	23	0.5%	5150	20	18	0.6%	0.6 %	4750
4	60	20	22	1.2%	5498	20	20	1.0 %	1.0%	5050
5	50	20	12	0.8%	5406	20	10	0.4 %	0.7%	4727
6	60	20	20	1.0%	5480	20	18	0.6%	0.8 %	4950
7	55	20	22	1.5%	5293	20	19	1.0%	1.0%	4768
8	60	20	20	1.5%	5033	20	20	1.3%	1.6%	4982
Total/ Average	**480**	**160**	**159**	1.0%	5391	**160**	**139**	0.7%	0.8%	4950

(): the number in parentheses indicates the number of infeasible instances.
#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

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. 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 main advantage of the algorithm is highlighted when we study instances where the B&C algorithm fails to produce feasible solutions, but the matheuristic succeeds. For these instances, we observe an average optimality gap of 3.5% for the heuristic solutions, as reported in the column labeled “Avg. gap^2”. The matheuristic algorithm offers the advantage of shorter CPU times, enhancing its practicality and efficiency for large-scale problems by solving all instances in under 7,200 seconds. In particular, Phase I solves the instances on averages in 825 seconds with an optimality gap of 3.3%. Phase II, which takes approximately 2,480 seconds, 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 1,689 seconds to achieve solutions with an optimality gap of 3.4% where B&C can not even generate a feasible solution.

In conclusion, the proposed algorithm demonstrates significant efficiency, particularly for large-size instances where the B&C algorithm fails to generate feasible solutions. While the two-phase approach yields the best results, Phase I alone is highly efficient regarding CPU time. Therefore, if a fast solution is required, it is conceivable to accept a slight loss in the objective function value and run only Phase I. 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.

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%	466
$\mathcal{N}=36, \mathcal{N}'=24$	160	4.6%	4.9%	-2.5%	539	2.4%	4.1%	2.0%	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%	-0.9%	2550
$\mathcal{P}=20, \mathcal{P}'=10$	160	2.7%	3.8%	-1.7%	1061	1.5%	2.6%	1.1%	2564
$\mathcal{T}=3, \mathcal{K}'=3$	160	1.9%	2.1%	-1.1%	309	1.0%	1.4%	-0.9%	2407
$\mathcal{T}=5, \mathcal{K}'=5$	160	4.6%	5.3%	-3.2%	1283	2.8%	4.2%	1.1%	2703
Average	480	3.3%	4.4%	-2.0%	825	1.9%	3.5%	0.1%	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)

5.4. *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 backhaul hazardous materials from 17 satellite points back to the depots. The depots (origins) and satellite points are depicted in Figure 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. 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; *commercial vehicles*, suitable for larger transportation needs with an average speed of 80 kmh; and *trailers*, which have the largest capacity with an average speed of 70 kmh. Each vehicle is limited to completing its tour within 8 hours. 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. The case study data is summarized in Table 10.

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 benchmark model, [IRP-DB-MDC-II], which allows a vehicle to backhaul after completing the delivery without splitting the deliveries. Finally, we solve [IRP-DB-MDC], the primary model proposed in this paper. 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,

Table 10: Case study parameters

Parameter	Value	Parameter	Value
d_{ipt}	$[1, 5]$	g_{ipt}	$[10, 20]$
a_{1p}, a_{2p}, a_{3p}	30, 50, 100	b_1, b_2, b_3	100, 200, 400
o_i	1000	q_{ip}	50
l_{ip}	1000	$l_{ip'}$	1000
s_{ip0}	5	$s_{ip'0}$	5
h_{ip}	$[60, 100]$ CAD	$h_{ip'}$	$[110, 150]$ CAD
$T_{\max}$	8 hours	$c_{ij1}, c_{ij2}, c_{ij3}$	1, 1.15, 1.25 CAD per km

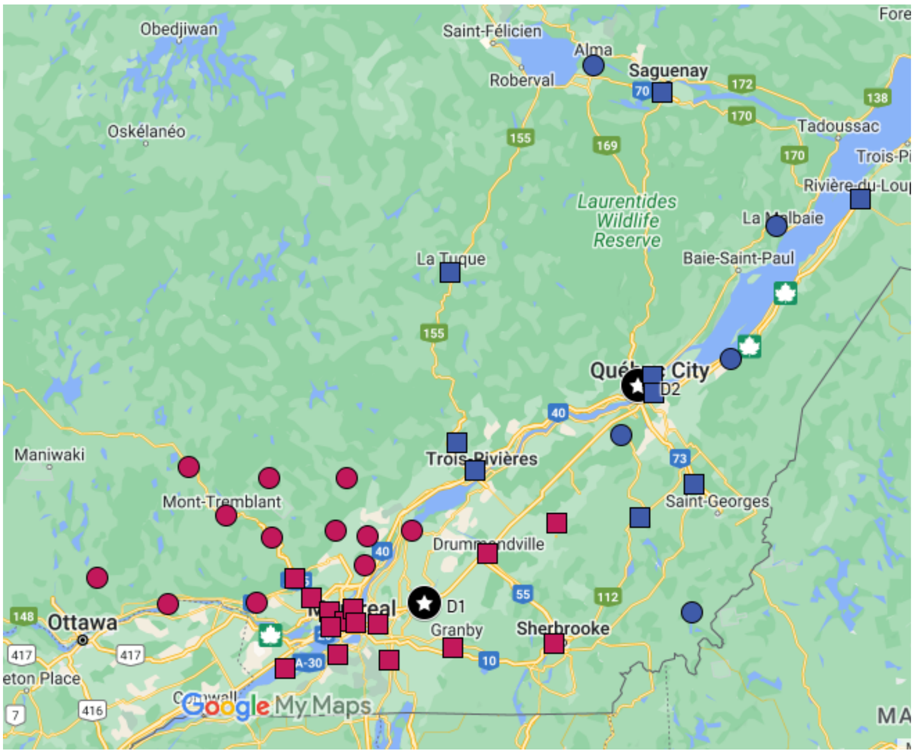

Figure 3: Depots and satellite points: stars represent the depots (left D1, right D2), squares indicate satellite points with only delivery demand, and circles denote satellite points with both delivery and backhauling

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 benchmark, [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 hours 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 hours ($T_{\max} = 8$) 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 hours ($T_{\max} = 14$). Next, we reduce the time limit by one hour in each subsequent iteration until the seventh iteration, when the time limit reaches 8 hours. The algorithm operates under this 8-hour 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 hours. 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]^1 and [IRP-DB-MDC]^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 1,646 seconds. However, when split deliveries are avoided in [IRP-DB-MDC-II], the CPU time of the first phase is increased to 2,095 minutes. Avoiding backhauling after delivery resulted in a more significant rise in CPU time, with Phase I taking 9,273 seconds 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 hours, aligning with HQ's standard practice of running this algorithm overnight through the batch process.

Table 11: Performance of the algorithms for 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]	81062	NF	21600	80810	87661	84977	(1646, 18000)

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

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] scenario, 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 15,658 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). The traveled distance by each vehicle type in each solution is illustrated in Figure 4. 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.

Table 12: Case study detailed solution

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

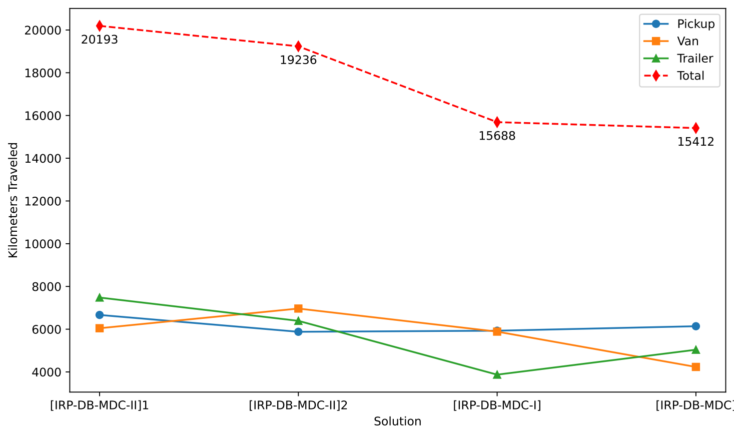

Figure 4: Traveled distance by each vehicle type in each solution

6. Conclusion

We have presented a mixed-integer linear program to model an IRP focusing on real-world requirements faced by our industrial partner, Hydro-Québec. The problem involves the delivery of various commodities to satellite points and the backhauling of materials using heterogeneous vehicles. Different aspects of the IRP, such as multiple depots, split deliveries, and the optimization of empty vehicle return trips, have been incorporated to ensure practical applicability. 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. 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.

Our study presents *managerial implications*, particularly for industries dealing with complex IRPs such as Hydro-Québec. Firstly, 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. 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 24% reduction in the total distance traveled by all vehicles compared to the scenario 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. Secondly, our proposed algorithm scales well to large instances. It operates in two phases, with the first phase offering a quick solution that, while potentially losing some optimality, 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. Thus, our study addresses the theoretical challenges and provides practical, real-world applicability that can significantly improve operational efficiency. The HQ's current operational strategy is to forbid split deliveries and not combine delivery and backhauling trips.

References

Absi, N., C. Archetti, S. Dauzère-Pérès, and D. Feillet. 2015. "A two-phase iterative heuristic approach for the production routing problem." *Transportation Science* 49 (4): 784–795.

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

Agra, A., M. Christiansen, and L. Wolsey. 2022. "Improved models for a single vehicle continuous-time inventory routing problem with pickups and deliveries." *European Journal of Operational Research* 297 (1): 164–179.

Andersson, H., A. Hoff, M. Christiansen, G. Hasle, and A. Løkketangen. 2010. "Industrial aspects and literature survey: Combined inventory management and routing." *Computers & Operations Research* 37 (9): 1515–1536.

Arab, R., S. Ghaderi, and R. Tavakkoli-Moghaddam. 2020. "Bi-objective inventory routing problem with backhauls under transportation risks: two meta-heuristics." *Transportation Letters* 12 (2): 113–129.

Archetti, C., L. Bertazzi, G. Laporte, and M. G. Speranza. 2007. "A branch-and-cut algorithm for a vendor-managed inventory-routing problem." *Transportation Science* 41 (3): 382–391.

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

Archetti, C., M. Christiansen, and M. G. Speranza. 2018. "Inventory routing with pickups and deliveries." *European Journal of Operational Research* 268 (1): 314–324.

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

Archetti, C., M. G. Speranza, M. Boccia, A. Sforza, and C. Sterle. 2020. "A branch-and-cut algorithm for the inventory routing problem with pickups and deliveries." *European Journal of Operational Research* 282 (3): 886–895.

Bell, W. J., L. M. Dalberto, M. L. Fisher, A. J. Greenfield, R. Jaikumar, P. Kedia, R. G. Mack, and P. J. Prutzman. 1983. "Improving the distribution of industrial gases with an on-line computerized routing and scheduling optimizer." *Interfaces* 13 (6): 4–23.

Bertazzi, L., L. C. Coelho, A. De Maio, and D. Laganà. 2019. "A matheuristic algorithm for the multi-depot inventory routing problem." *Transportation Research Part E: Logistics and Transportation Review* 122: 524–544.

Campbell, A., L. Clarke, A. Kleywegt, and M. Savelsbergh. 1998. "The inventory routing problem." In *Fleet Management and Logistics*, 95–113. Springer.

Cheng, C., P. Yang, M. Qi, and L.-M. Rousseau. 2017. "Modeling a green inventory routing problem with a heterogeneous fleet." *Transportation Research Part E: Logistics and Transportation Review* 97: 97–112.

Chitsaz, M., J. Cordeau, and R. Jans. 2019. "A unified decomposition matheuristic for assembly, production, and inventory routing." *INFORMS Journal on Computing* 31 (1): 134–152.

Chiu, A., G. Angulo, and H. Larrain. 2024. "Optimizing the long-term costs of an Inventory Routing Problem using linear relaxation." *Transportation Research Part E: Logistics and Transportation Review* 183: 103447.

Clarke, G., and J. W. Wright. 1964. "Scheduling of vehicles from a central depot to a number of delivery points." *Operations Research* 12 (4): 568–581.

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

Coelho, L. C., and G. Laporte. 2013a. "A branch-and-cut algorithm for the multi-product multi-vehicle inventory-routing problem." *International Journal of Production Research* 51 (23-24): 7156–7169.

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

Deif, I., and L. Bodin. 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*, 75–96. Babson Park, MA.

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

Diabat, A., N. Bianchessi, and C. Archetti. 2024. "On the zero-inventory-ordering policy in the inventory routing problem." *European Journal of Operational Research* 312 (3): 1024–1038.

Dinh, N. M., C. Archetti, and L. Bertazzi. 2023. "The inventory routing problem with split deliveries." *Networks* 82 (4): 400–413.

Feng, Y., A. Che, and N. Tian. 2024. “Robust inventory routing problem under uncertain demand and risk-averse criterion.” *Omega* 127: 103082.

Grønhaug, R., M. Christiansen, G. Desaulniers, and J. Desrosiers. 2010. “A branch-and-price method for a liquefied natural gas inventory routing problem.” *Transportation Science* 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., P. J. Swart, and D. A. Schult. 2008. *Exploring network structure, dynamics, and function using NetworkX.* Technical Report. Los Alamos National Laboratory (LANL), Los Alamos, NM (United States).

Hewitt, M., G. Nemhauser, M. Savelsbergh, and J.-H. Song. 2013. “A branch-and-price guided search approach to maritime inventory routing.” *Computers & Operations Research* 40 (5): 1410–1419.

Iassinovskaia, G., S. Limbourg, and F. Riane. 2017. “The inventory-routing problem of returnable transport items with time windows and simultaneous pickup and delivery in closed-loop supply chains.” *International Journal of Production Economics* 183: 570–582.

Koç, Ç., and G. Laporte. 2018. “Vehicle routing with backhauls: Review and research perspectives.” *Computers & Operations Research* 91: 79–91.

Le, T., A. Diabat, J.-P. Richard, and Y. Yih. 2013. “A column generation-based heuristic algorithm for an inventory routing problem with perishable goods.” *Optimization Letters* 7: 1481–1502.

Lei, J., A. Che, and T. Van Woensel. 2024. “Collection-disassembly-delivery problem of disassembly centers in a reverse logistics network.” *European Journal of Operational Research* 313 (2): 478–493.

Li, R., Z. Cui, Y.-H. Kuo, and L. Zhang. 2023. “Scenario-based distributionally robust optimization for the stochastic inventory routing problem.” *Transportation Research Part E: Logistics and Transportation Review* 176: 103193.

Londoño, J., J. Bastidas, P. González, and J. Escobar. 2023. “Inventory routing problem with backhaul considering returnable transport items collection.” *International Journal of Industrial Engineering Computations* 14 (4): 837–862.

Mahmutoğulları, Ö., and H. Yaman. 2023. “A Branch-and-Cut Algorithm for the Inventory Routing Problem with Product Substitution.” *Omega* 115: 102752.

Neves-Moreira, F., B. Almada-Lobo, J. Cordeau, L. Guimarães, and R. Jans. 2019. “Solving a large multi-product production-routing problem with delivery time windows.” *Omega* 86: 154–172.

Neves-Moreira, F., B. Almada-Lobo, L. Guimarães, and P. Amorim. 2022. “The multi-product inventory-routing problem with pickups and deliveries: Mitigating fluctuating demand via rolling horizon heuristics.” *Transportation Research Part E: Logistics and Transportation Review* 164: 102791.

Ortega, E. J. A., S. Malicki, K. F. Doerner, and S. Minner. 2024. “Stochastic inventory routing with dynamic demands and intra-day depletion.” *Computers & Operations Research* 163: 106503.

Ramkumar, N., P. Subramanian, T. Narendran, and K. Ganesh. 2012. “Mixed integer linear programming model for multi-commodity multi-depot inventory routing problem.” *Opsearch* 49: 413–429.

Rau, H., S. D. Budiman, and G. A. Widyadana. 2018. “Optimization of the multi-objective green cyclical inventory routing problem using discrete multi-swarm PSO method.” *Transportation Research Part E: Logistics and Transportation Review* 120: 51–75.

Santos, M. J., P. Amorim, A. Marques, A. Carvalho, and A. Póvoa. 2020. "The vehicle routing problem with backhauls towards a sustainability perspective: A review." *Top* 28 (2): 358–401.

Shaabani, H. 2022. "A literature review of the perishable inventory routing problem." *The Asian Journal of Shipping and Logistics* 38 (3): 143–161.

Skålnes, J., M. B. Ahmed, L. M. Hvattum, and M. Stålhane. 2024. "New benchmark instances for the inventory routing problem." *European Journal of Operational Research* 313 (3): 992–1014.

Skålnes, J., S. T. Vadseth, H. Andersson, and M. Stålhane. 2023. "A branch-and-cut embedded matheuristic for the inventory routing problem." *Computers & Operations Research* 159: 106353.

Sofianopoulou, S., and I. Mitsopoulos. 2021. "A review and classification of heuristic algorithms for the inventory routing problem." *International Journal of Operational Research* 41 (2): 282–298.

Song, J.-H., and K. C. Furman. 2013. "A maritime inventory routing problem: Practical approach." *Computers & Operations Research* 40 (3): 657–665.

Soysal, M., M. Çimen, S. Belbağ, and E. Toğrul. 2019. "A review on sustainable inventory routing." *Computers & Industrial Engineering* 132: 395–411.

Tiniç, G. Ö., E. Koca, and H. Yaman. 2021. "An exact solution approach for the inventory routing problem with time windows." *Computers & Operations Research* 134: 105371.

Toth, P., and D. Vigo. 2002. *The Vehicle Routing Problem.* SIAM.

Van Anholt, R. G., L. C. Coelho, G. Laporte, and I. F. Vis. 2016. "An inventory-routing problem with pickups and deliveries arising in the replenishment of automated teller machines." *Transportation Science* 50 (3): 1077–1091.