The Two-Echelon Inventory-Routing Problem with Fleet Management

Cleder M. Schenekemberg
Cassius T. Scarpin
José E. Pécora Jr.
Thiago A.Guimarães
Leandro C. Coelho

September 2019

The Two-Echelon Inventory-Routing Problem with Fleet Management

Cleder M. Schenekemberg[1,*], Cassius T. Scarpin[1], José E. Pécora J.[1,2],
Thiago A. Guimarães[1,3] Leandro C. Coelho[2,3]

1. Research Group of Technology Applied to Optimization (GTAO), Federal University of Paraná (UFPR), Curitiba, Brazil
2. Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)
3. Federal Institute of Science and Technology of Paraná (IFPR), Curitiba, Brazil
4. Canada Research Chair in Integrated Logistics, Operations and Decision Systems Department, Faculty of Administration Sciences, Université Laval, 2325, de la Terrasse, bureau 2525, Québec, Canada, G1V 0A6

Abstract. This paper introduces the Two-Echelon Inventory-Routing Problem with Fleet Management. This problem arises under a two-echelon vendor-managed inventory system when a company must make vehicle routing and inventory management decisions, while renting a fleet subject to short-and mid-term agreements. Different chemical products are transported contaminating the vehicles that may require cleaning activities. Pickups of input take place in the first echelon, and the final product deliveries are performed in the second echelon. Based on a real-life case in the petrochemical industry, we introduce a formulation that takes into account vehicle rentals, cleanings, transportation, inventory management, and vehicle returns decisions. We design a branch-and-cut algorithm to solve it and also propose a matheuristic, in which vehicle routes are handled by an adaptive large neighborhood mechanism, while input pickups, product deliveries, and fleet planning are performed by solving several subproblems to optimality. Moreover, we introduce a hybrid parallel framework, combining our matheuristic and the branch-and-cut algorithm in order to solve very large instances exactly. We validate our methods by solving a set of instances of the two-echelon multi-depot inventory-routing problem from the literature, obtaining new best solutions for all instances. We also perform an extensive assessment of our methods, both on overall performance and on supply chain insights.

Keywords. Fleet management, two-echelon inventory-routing, matheuristic, branch-and-cut, parallel computing.

_____

* Corresponding author: cledercms@hotmail.com

1 Introduction

Modern supply chains require high level of coordination and synchronization of their decisions [9], and these can help decrease logistics costs and improve vendor-customer relationships [29]. In this context, Vendor-Managed Inventory (VMI) systems configure a collaborative practice between suppliers and customers [2]. Under the VMI paradigm, suppliers control the inventory of the customers, deciding when to serve them and how much to deliver. According to Zhao et al. [30], the VMI practice is mutually beneficial, since customers do not have to spend resources to control their inventories and to manage orders, while suppliers improve their logistics activities coordination, particularly on delivery routes composition.

To operate a VMI strategy one must solve an Inventory-Routing Problem (IRP), which integrates the inventory management and the multi-period vehicle routing problem into the same framework. Since it was introduced by Bell et al. [6], a wide range of IRP variants have been proposed, involving strategic, tactical and operational criteria (see [13]). Despite these studies, two simplifications are common in road-based logistics. First, the supply chain is usually simplified into one-echelon only and involves one plant serving multiple customers, known as a one-to-many structure [13]. Second, fleet decisions are often taken on a tactical perspective, like sizing and mix [2].

Fleet management is one of the most expensive activities in the logistics industry [27]. Regarding routing problems, the main concern relates to fleet sizing, when an unlimited number of homogeneous vehicles are considered, or to the fleet composition, when vehicles with different capacities are available to perform deliveries [8]. Few studies address short-term vehicle availability, and in the following, we review the relevant literature. Fagerholt and Lindstad [15] present an interactive optimization-based decision support system for ship routing and scheduling, where a cleaning procedure can be taken into account when needed. Hvattum et al. [18] describe a similar approach for a tank allocation problem arising in the shipping of bulk oil and chemical products by tanker ships. In road-based

transportation, the majority of studies consider multi-compartment vehicles, as in Oppen et al. [23], which deal with live animals from farms to slaughterhouses, where vehicle disinfection is performed between consecutive tours. More recently, Lahyani et al. [19] address a multi-product, multi-period routing problem arising in the collection of olive oil in Tunisia. Due to the difference among olive oil grades, a compartment may require a cleaning activity.

While fleet management in a integrated context is a complex task, outsourced fleet employment enables companies to focus their efforts on their core competence. According to Rabinovich and Corsi [26], the outsourcing of integrated logistics functions allows to the companies reduce their costs and improve the customer service level.

When more echelons are considered, the integrated supply chain management becomes more complex, and the decision-making process is more difficult. Recently, Guimarães et al. [17] introduced the two-echelon multi-depot IRP (2E-MDIRP), inspired by a VMI system implementation of a real fuel distribution problem in South America. In that case, plants in the middle layer control the gasoline inventory of a set of gas stations, besides their own inventory of input (ethanol) picked up from supplying facilities. The aim is to minimize pickups and deliveries routing costs, in addition to inventories costs. Despite the costs involved, the authors assume that a fleet of vehicles is available at each plant, omitting any cost for the use and maintenance of the vehicles.

In order to take advantage from the fleet outsourcing, to enable companies to focus on their core business, and to consider a more realistic scenario, we extend the 2E-MDIRP, incorporating fleet management decisions. We consider a two-echelon (2E) supply chain, in which the plants in the middle layer are responsible for managing inputs pickups from suppliers in the first echelon, and the product deliveries to customers in the second one. Plants operate with a non-compartmentalized outsourced fleet, which must be rented. Cleaning activities must take place whenever the vehicle exchanges the loaded product or when it is returned to the rental company. In addition to traditional transportation and inventory decisions, plants are in charge of fleet planning, including renting, cleaning,

and returning vehicles. These aspects define an unparalleled problem in the literature, which we call the 2E-IRP with fleet management (2E-IRPFM). The aim is to minimize fleet management (rent and cleaning), routing (pickups of the input and deliveries of gasoline blend) and inventories (input and final product) costs, avoiding stock-outs over a planning horizon. We also perform extensive computational experiments on a new set of 2E-IRPFM instances, inspired by a real-life case. We design a set of performance indicators and derive many managerial insights for this rich and new problem.

The scientific contributions of this work are:

1. we are the first to consider an IRP while managing fleet decisions, not limited to size and mix, but also taking into account rentals, cleanings, and returns;

2. we describe, model, solve, and compare different combinations of inventory policies with different fleet management costs. We analyze the performance of these configurations based on their partial costs;

3. we design a matheuristic and a hybrid parallel exact algorithm based on branch-and-cut (B&C);

4. we validate our algorithms on a special case of our problem from the literature, proving optimally for several open instances and providing best known solutions to all of them.

The remainder of the paper is organized as follows. In Section 2, we formally describe the 2E-IRPFM, and propose a mixed-integer linear programming (MILP) formulation in Section 3, where we present sets of new and existing valid inequalities. The B&C algorithm is detailed in Section 4. In Section 5, we describe the matheuristic algorithm we propose to solve the 2E-IRPFM, while Section 6 presents the hybrid parallel exact approach. In Section 7, we discuss the results of extensive computational experiments performed to assess the quality of the algorithms, and we also derive business insights based on the results. Conclusions are presented in Section 8.

2 Problem description

The 2E-IRPFM is defined over an undirected graph $G = (\mathcal{V}, \mathcal{E})$, where the vertex set $\mathcal{V}$ represents the union of the sets $\mathcal{F}$ of suppliers, $\mathcal{P}$ of plants, and $\mathcal{C}$ of customers, while $\mathcal{E}$ is the set of edges. The first echelon links suppliers and plants and it is defined by subgraph $G' = (\mathcal{V}', \mathcal{E}')$, where $\mathcal{V}' = \mathcal{F} \cup \mathcal{P}$ and $\mathcal{E}' = \{(u, v) : u, v \in V', u \in \mathcal{F} \wedge v \in \mathcal{P}\}$. The second echelon links plants and customers and is defined by subgraph $G'' = (\mathcal{V}'', \mathcal{E}'')$ with $\mathcal{V}'' = \mathcal{P} \cup \mathcal{C}$ and $\mathcal{E}'' = \{(u, v) : u, v \in \mathcal{V}'' \wedge u, v \oplus \mathcal{P}, u < v\}$, in which $\mathcal{V} = \mathcal{V}' \cup \mathcal{V}''$ and $\mathcal{E} = \mathcal{E}' \cup \mathcal{E}''$. A non-negative cost c_{uv} is associated with each edge $(u, v) \in \mathcal{E}$.

The planning horizon is defined over a set $\mathcal{T} = \{1, ..., p\}$ of periods. In each period t, each plant $j \in \mathcal{P}$ is allowed to rent up to $|\mathcal{K}|$ of homogeneous vehicles of capacity Q at rental cost f_w per vehicle per period. Each vehicle can be used to pick up a certain amount of input (α) from a supplier, and/or to deliver a certain amount of final product (β) to customers. Once a customer is visited and one must determine the quantity to be delivered, one of two policies are often applied. Under the maximum level (ML) policy, the plant is free to decide how much to deliver to a customer, as long as the inventory capacity is not exceeded. The order-up-to (OU) policy fills the customer's inventory capacity whenever a delivery occurs [4].

After performing a pickup of α (delivery of β), the vehicle remains contaminated with α (β). Before a new trip with a different product, each vehicle must undergo a chemical cleaning procedure, incurring in cleaning cost f_s. This can occur in the same period, if α is picked up at the beginning of t, and β is delivered at the end of t by the same vehicle k, or in different periods, if the vehicle remains at the plant. In this case, f_w is due for each additional period in which the vehicle remains at the plant. A vehicle contaminated with α in t can perform a new pickup in $t', t' > t$ without going through chemical cleaning, since it does not perform any delivery of β between $[t, t']$. The same idea works for β. As a vehicle remains contaminated with the last load, when a plant decides to return it, a cleaning cost f_s is due as every vehicle must be cleaned before being returned.

Each unity of β requires a certain quantity φ of α in its blend, and the demand d_l^t of each customer l in each period t is known a priori. Plant j has a maximum inventory capacity U_j for α, and its level cannot be lower than L_j. One unit of α incurs an inventory holding cost h_j per period. Likewise, U_l and L_l denote the maximum and minimum inventory levels for β at customer l, with unit holding cost h_l. The total availability of α at all suppliers is not constrained. However, each supplier i disposes of Φ_i units for the whole planning horizon, according to a pre-established contract with plants. In $t = 0$, there are no rented vehicles at the plants, and initial inventory levels I_j^0 and I_l^0 are known at each plant j and each customer l.

Regarding the timing of the activities, we assume that α is always picked up at the beginning of a period if a pickup is needed. When a delivery is also required, it must be scheduled after all the pickups had been performed. That is mandatory to enable the blending process and to produce β in the same period. A vehicle must be clean before its return to the rental company at the end of a period.

The objective of the 2E-IRPFM is to minimize the total inventory, transportation, and fleet management cost, determining, for each plant:

- when, how much and from which supplier to pickup α;

- when and how much to deliver β to a customer;

- when and how many vehicles to rent, to clean, to keep, and to return;

- how to combine customer deliveries into vehicle routes.

A customer may be visited at most by one vehicle per period. Likewise, a plant is allowed to pick up α from one supplier using one vehicle in a period. Each input pickup or delivery route must end at the same starting plant. As the fleet is non-compartmentalized, a pickup and a delivery cannot be combined in the same tour. Thus, after collecting α from a supplier, the vehicle needs to return to its original plant and will only be able to perform a delivery route if it is properly cleaned.

3   Mathematical formulation for the 2E-IRPFM

We now present the mathematical formulation for the 2E-IRPFM. For each plant j one must determine the quantity of product q_{jl}^{kt} delivered to customer l and the total amount of input r_{ij}^{kt} picked up from supplier i, using vehicle k in period t. At the end of each period, the inventory level of β at customer l is given by I_l^t, while the inventory level of α at plant j is I_j^t. The remaining variables used in our model are:

- $W_j^{kt} = 1$ if vehicle k is rented by plant j in period t, 0 otherwise;

- $R_j^{kt} = 1$ if vehicle k rented by plant j is returned in period t, 0 otherwise;

- $X_{ij}^{kt} = 1$ if vehicle k rented by plant j picks up α from supplier i in period t, 0 otherwise;

- $Y_{jl}^{kt} = 1$ if vehicle k rented by plant j delivers β to customer l in period t, 0 otherwise;

- $y_{uv}^{kjt} = 1$ if vehicle k rented by plant j travels between customers u and v, $u < v$, in period t, 0 otherwise;

- $y_{jl}^{kjt} \in \{0, 1, 2\}$. When $y_{jl}^{kjt} = 1$, vehicle k rented by plant j travels directly from plant j to customer l in period t. If $y_{jl}^{kjt} = 2$, a round trip is defined, 0 otherwise;

- $Z_j^{\alpha,kt} = 1$ if vehicle k rented by plant j ends period t contaminated with α, 0 otherwise;

- $Z_j^{\beta,kt} = 1$ if vehicle k rented by plant j ends period t contaminated with β, 0 otherwise;

- $S_j^{\alpha,kt} = 1$ if vehicle k rented by plant j contaminated with β is cleaned to pickup α in period t, 0 otherwise;

- $S_j^{\beta,kt} = 1$ if vehicle k rented by plant j contaminated with α is cleaned to deliver β in period t, 0 otherwise;

- $S_j^{R,kt} = 1$ if vehicle k rented by plant j, contaminated with α or β, is cleaned to be returned in period t, 0 otherwise;

- S_j^{kt}: total number of cleanings of vehicle k rented by plant j in period t.

In addition, we also define variable $X_{jj}^{kt} = 1$ to indicate that vehicle k, rented by plant j, performs a pickup in t, 0 otherwise. Equivalently, $Y_{jj}^{kt} = 1$ works for a delivery, 0 otherwise. Without loss of generality, we assume that no vehicle is housed at plants in $t = 0$, i.e., $W_j^{k0}, R_j^{k0}, Z_j^{\alpha,k0}$ and $Z_j^{\beta,k0}$ are set to zero.

A suitable representation of the 2E-IRPFM is shown on Figure 1. Considering two consecutive periods, t and $t+1$, we depict the sequence of decisions for a given vehicle k, rented by plant j. In order to simplify the example, indices k and j are omitted on the variables. First, the vehicle is rented in period t, which obviously implies in $W^t = 1$. Then, this vehicle performs a pickup, becoming contaminated with α. Before it can be assigned to a delivery, a cleaning is carried out, i.e., $S^{\beta,t} = 1$. The vehicle then leaves to the plant contaminated with β, and remains there in the following period, yielding $Z^{\beta,t} = 1$ and an additional lease with $W^{t+1} = 1$. At the beginning of $t+1$, the vehicle is cleaned to perform a new input pickup ($S^{\alpha,t+1} = 1$). After returning to the plant, one last cleaning is performed ($S^{R,t+1} = 1$), and the vehicle is returned at the end of $t+1$.

The 2E-IRPFM is formulated by $(1)-(42)$.

$$\min \sum_{t \in \mathcal{T}} \left(\sum_{j \in \mathcal{P}} \sum_{k \in \mathcal{K}} f_w W_j^{kt} + \sum_{j \in \mathcal{P}} \sum_{k \in \mathcal{K}} f_s S_j^{kt} + \sum_{j \in \mathcal{P}} h_j I_j^t + \sum_{l \in \mathcal{C}} h_l I_l^t + \sum_{k \in \mathcal{K}} \sum_{(i,j) \in \mathcal{E}'} 2c_{ij} X_{ij}^{kt} + \sum_{j \in \mathcal{P}} \sum_{k \in \mathcal{K}} \sum_{(u,v) \in \mathcal{E}''} c_{uv} y_{uv}^{kjt} \right) \tag{1}$$

subject to

$$\sum_{t \in \mathcal{T}} \sum_{j \in \mathcal{P}} \sum_{k \in \mathcal{K}} r_{ij}^{kt} \le \Phi_i \quad i \in \mathcal{F} \tag{2}$$

$$I_j^t = I_j^{t-1} + \sum_{i \in \mathcal{F}} \sum_{k \in \mathcal{K}} r_{ij}^{kt} - \sum_{l \in \mathcal{C}} \sum_{k \in \mathcal{K}} \varphi q_{jl}^{kt} \quad j \in \mathcal{P}, t \in \mathcal{T} \tag{3}$$
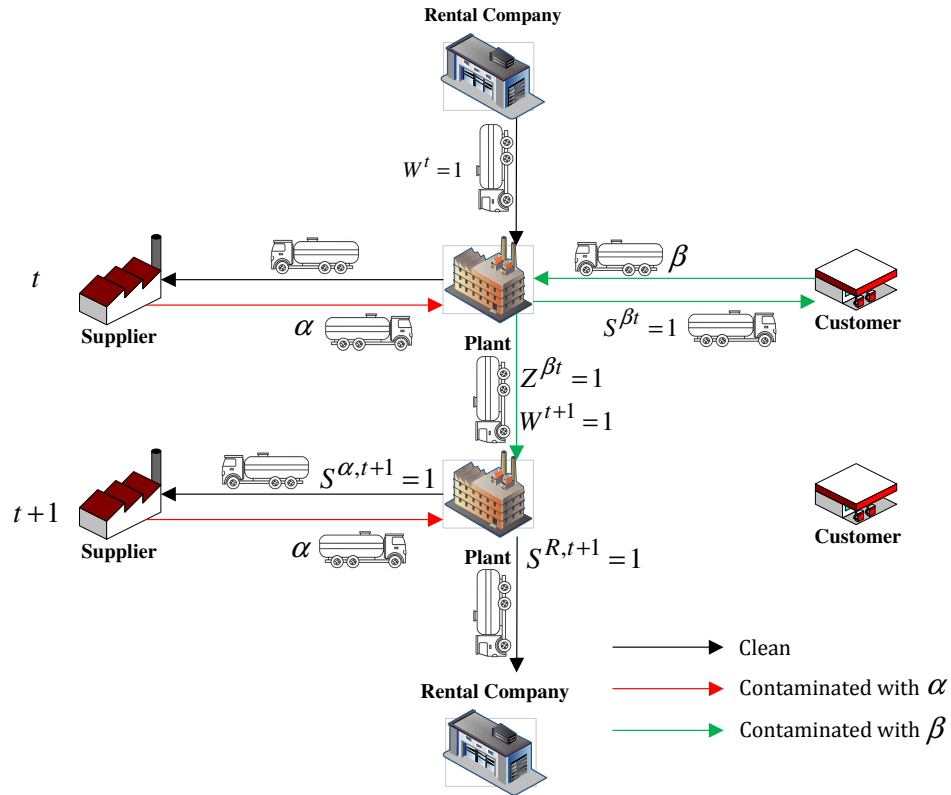
Figure 1: Graphical representation of the 2E-IRPFM

$$I_l^t = I_l^{t-1} + \sum_{j \in \mathcal{P}} \sum_{k \in \mathcal{K}} q_{jl}^{kt} - d_l^t \quad l \in \mathcal{C}, t \in \mathcal{T} \tag{4}$$

$$L_l \leq I_l^t \leq U_l \quad l \in \mathcal{V}'', t \in \mathcal{T} \tag{5}$$

$$\sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{F}} X_{ij}^{kt} \leq 1 \quad j \in \mathcal{P}, t \in \mathcal{T} \tag{6}$$

$$\sum_{i \in \mathcal{F}} \sum_{k \in \mathcal{K}} r_{ij}^{kt} \leq U_j - I_j^{t-1} \quad j \in \mathcal{P}, t \in \mathcal{T} \tag{7}$$

$$r_{ij}^{kt} \leq Q X_{ij}^{kt} \quad j \in \mathcal{P}, i \in \mathcal{F}, k \in \mathcal{K}, t \in \mathcal{T} \tag{8}$$

$$\sum_{j \in \mathcal{P}} \sum_{k \in \mathcal{K}} q_{jl}^{kt} \leq U_l - I_l^{t-1} \quad l \in \mathcal{C}, t \in \mathcal{T} \tag{9}$$

$$q_{jl}^{kt} \leq U_l Y_{jl}^{kt} \quad l \in \mathcal{C}, j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T} \tag{10}$$

$$\sum_{l \in \mathcal{C}} q_{jl}^{kt} \leq Q Y_{jj}^{kt} \quad j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T} \tag{11}$$

$$\sum_{j \in \mathcal{P}} \sum_{k \in \mathcal{K}} Y_{jl}^{kt} \leq 1 \quad l \in \mathcal{C}, t \in \mathcal{T} \tag{12}$$

$$\sum_{\substack{u \in \mathcal{V}'' \\ u < l}} y_{ul}^{kjt} + \sum_{\substack{u \in \mathcal{V}'' \\ l < u}} y_{lu}^{kjt} = 2 Y_{jl}^{kt} \quad l \in \mathcal{V}'', j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T} \tag{13}$$

$$\sum_{l \in S} \sum_{\substack{u \in S \\ l < u}} y_{lu}^{kjt} \leq \sum_{l \in S} Y_{jl}^{kt} - Y_{jm}^{kt} \quad S \subseteq \mathcal{C}, |S| \geq 2, m \in S, j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T} \tag{14}$$

$$\sum_{i \in \mathcal{F}} X_{ij}^{kt} = X_{jj}^{kt} \quad j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T} \tag{15}$$

$$X_{jj}^{kt} \leq W_j^{kt} \quad j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T} \tag{16}$$

$$Y_{jj}^{kt} \leq W_j^{kt} \quad j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T} \tag{17}$$

$$R_j^{kt} \leq W_j^{kt} \quad j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T} \tag{18}$$

$$Z_j^{\alpha,k,t-1} \leq W_j^{kt} \quad j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T} \tag{19}$$

$$Z_j^{\beta,k,t-1} \leq W_j^{kt} \quad j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T} \tag{20}$$

$$Z_j^{\alpha,kt} + Z_j^{\beta,kt} + R_j^{kt} = W_j^{kt} \quad j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T} \tag{21}$$

$$X_{jj}^{kt} - Y_{jj}^{kt} - R_j^{kt} \leq Z_j^{\alpha,kt} \quad j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T} \tag{22}$$

$$Z_j^{\alpha,k,t-1} - Z_j^{\beta,kt} - R_j^{kt} \leq Z_j^{\alpha,kt} \quad j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T} \tag{23}$$

$$Y_{jj}^{kt} - R_j^{kt} \leq Z_j^{\beta,kt} \quad j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T} \tag{24}$$

$$Z_j^{\beta,k,t-1} - Z_j^{\alpha,kt} - R_j^{kt} \leq Z_j^{\beta,kt} \quad j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T} \tag{25}$$

$$W_j^{k,t-1} - R_j^{k,t-1} \leq W_j^{kt} \quad j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T} \tag{26}$$

$$X_{jj}^{kt} + Z_j^{\beta,k,t-1} - 1 \leq S_j^{\alpha,kt} \quad j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T} \tag{27}$$

$$Z_j^{\beta,k,t-1} + Z_j^{\alpha,kt} - 1 \leq S_j^{\alpha,kt} \quad j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T} \tag{28}$$

$$Y_{jj}^{kt} + X_{jj}^{kt} - 1 \leq S_j^{\beta,kt} \quad j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T} \tag{29}$$

$$Y_{jj}^{kt} + Z_j^{\alpha,k,t-1} - 1 \leq S_j^{\beta,kt} \quad j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T} \tag{30}$$

$$Z_j^{\alpha,k,t-1} + Z_j^{\beta,kt} - 1 \leq S_j^{\beta,kt} \quad j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T} \tag{31}$$

$$R_j^{kt} \leq S_j^{R,kt} \quad j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T} \tag{32}$$

$$S_j^{\alpha,kt} + S_j^{\beta,kt} + S_j^{R,kt} = S_j^{kt} \quad j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T} \tag{33}$$

$$I_j^t, r_{ij}^{kt} \geq 0 \quad i \in \mathcal{F}, j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T} \tag{34}$$

$$I_l^t, q_{jl}^{kt} \geq 0 \quad l \in \mathcal{C}, j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T} \tag{35}$$

$$Y_{jj}^{kt}, S_j^{R,kt}, S_j^{\beta,kt}, S_j^{\alpha,kt}, R_j^{kt}, Z_j^{\beta,kt}, Z_j^{\alpha,kt}, W_j^{kt} \in \{0,1\} \quad j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T} \tag{36}$$

$$S_j^{kt} \in \{0,1,2,3\} \quad j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T} \tag{37}$$

$$X_{ij}^{kt} \in \{0,1\} \quad i \in \mathcal{F}, j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T} \tag{38}$$

$$X_{jj}^{kt} \in \{0,1\} \quad j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T} \tag{39}$$

$$Y_{jl}^{kt} \in \{0,1\} \quad l \in \mathcal{C}, j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T} \tag{40}$$

$$y_{jl}^{kjt} \in \{0,1,2\} \quad l \in \mathcal{C}, j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T} \tag{41}$$

$$y_{uv}^{kjt} \in \{0,1\} \quad u, v \in \mathcal{C}, u < v, j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T}. \tag{42}$$

The objective function (1) minimizes the total cost, given by six terms: fleet rental, fleet cleaning, input inventory at the plants, final product inventory at the customers, pickup transportation, and delivery transportation costs. Constraints (2) bound the input availability, according to the contract between suppliers and plants. Constraints (3)−(5) balance the flow and impose inventory bounds. Constraints (6) allow at most one pickup per plant per period, while constraints (7) ensure the ML inventory policy for α. Constraints (8) guarantee that the vehicle capacity is not exceed. The ML inventory policy for β is formulated by (9). Constraints (10) link assignment variables with the quantity delivered. Constraints (11) guarantee that total delivery quantity does not exceed the vehicle capacity, while (12) avoid split deliveries. Constraints (13) and (14) impose linking and subtour elimination conditions, while constraints (15) link a pickup with the assigned vehicle. Constraints (16)–(33) formulate the fleet rental and return. In particular, constraints (16) and (17) require a vehicle to be rented if it is used to either

pickup α or deliver β. Constraints (18) impose that only rented vehicles can be returned. Constraints (19) and (20) impose that if a vehicle remains rented its contamination is kept. Constraints (21) define mutually exclusive conditions for a vehicle at the end of t: contaminated with α, contaminated with β, or returned to the rental company. In the last case, the vehicle must be cleaned. Constraints (22) and (23) balance the contamination flow of α, while (24) and (25) impose it for β. Likewise, constraints (26) ensure vehicle flow conservation. Constraints (27) and (28) encompass all the requirements to clean up the vehicle, in order to allow a pickup. Similarly, constraints (29)–(31) do the same to deliveries, while constraints (32) ensure that the vehicle is clean when returned. Constraints (33) compute the total number of cleanings. Lastly, constraints (34)–(42) define the variables domain.

The order-up-to level inventory policy (OU) links the decision of *when* and *how much* to serve a customer. Thereby, whenever a plant performs a delivery, the total quantity delivered must be equal to the customer inventory availability:

$$\sum_{j \in \mathcal{P}} \sum_{k \in \mathcal{K}} q_{jl}^{kt} \geq U_l \sum_{j \in \mathcal{P}} \sum_{k \in \mathcal{K}} Y_{jl}^{kt} - I_l^{t-1} \qquad l \in \mathcal{C}, t \in \mathcal{T}. \qquad (43)$$

In order to strengthen the formulation and improve the quality of its dual bound, we also present a set of well-known valid inequalities from basic IRPs [4, 7, 10]. These are described next.

$$y_{jl}^{kjt} \leq 2Y_{jl}^{kt} \qquad l \in \mathcal{C}, j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T} \qquad (44)$$

$$y_{ul}^{kjt} \leq Y_{jl}^{kt} \qquad u, l \in \mathcal{C}, u < l, j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T} \qquad (45)$$

$$y_{lu}^{kjt} \leq Y_{jl}^{kt} \qquad u, l \in \mathcal{C}, l < u, j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T} \qquad (46)$$

$$Y_{jl}^{kt} \leq Y_{jj}^{kt} \qquad l \in \mathcal{C}, j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T}. \qquad (47)$$

Inequalities (44) strengthen the case if customer l is served in a direct delivery by vehicle

k rented by plant j in period t. Similarly, inequalities (45) and (46) work for customer l if it is preceded or succeeded by another customer u, respectively. Inequalities (47) ensure that customer l is served by vehicle k starting from plant j in period t, only if the vehicle is assigned for that plant.

We also consider inequalities (48)–(50) adapted by Guimarães et al. [17] for the multi-depot multi-vehicle IRP. It is useful to set the minimum number of deliveries, in order to avoid a stock-out at customer l on the interval $[t_1, t_2]$. The same authors also proposed inequalities (51), which identify the smallest range $[t_1, t_2]$ for which plant j must perform a pickup.

$$\sum_{j \in \mathcal{P}} \sum_{k \in \mathcal{K}} \sum_{t=t_1}^{t_2} Y_{jl}^{kt} \geq \left\lceil \frac{\sum_{t=t_1}^{t_2} d_l^t - U_l}{min\{Q, U_l\}} \right\rceil \qquad l \in \mathcal{C}, t_1, t_2 \in \mathcal{T}, t_2 > t_1 \qquad (48)$$

$$\sum_{j \in \mathcal{P}} \sum_{k \in \mathcal{K}} \sum_{t=t_1}^{t_2} Y_{jl}^{kt} \geq \frac{\sum_{t=t_1}^{t_2} d_l^t - I_l^{t_1-1}}{min\{Q, U_l\}} \qquad l \in \mathcal{C}, t_1, t_2 \in \mathcal{T}, t_2 > t_1 \qquad (49)$$

$$\sum_{j \in \mathcal{P}} \sum_{k \in \mathcal{K}} \sum_{t=t_1}^{t_2} Y_{jl}^{kt} \geq \frac{\sum_{t=t_1}^{t_2} d_l^t - I_l^{t_1-1}}{\sum_{t=t_1}^{t_2} d_l^t} \qquad l \in \mathcal{C}, t_1, t_2 \in \mathcal{T}, t_2 > t_1 \qquad (50)$$

$$\sum_{i \in \mathcal{F}} \sum_{k \in \mathcal{K}} \sum_{t=t_1}^{t_2} X_{ij}^{kt} \geq \frac{\sum_{l \in \mathcal{C}} \sum_{k \in \mathcal{K}} \sum_{t=t_1}^{t_2} (\varphi) q_{jl}^{kt} - I_j^{t_1-1}}{min\{Q, U_j\}} \qquad j \in \mathcal{P}, t_1, t_2 \in \mathcal{T}, t_2 > t_1. \qquad (51)$$

Based on the features of the 2E-IRPFM, we introduce the following new inequalities.

$$\frac{\sum_{l \in \mathcal{C}} \sum_{k \in \mathcal{K}} q_{jl}^{kt}}{Q} \leq \sum_{k \in \mathcal{K}} W_j^{kt} \qquad j \in \mathcal{P}, t \in \mathcal{T} \qquad (52)$$

$$\sum_{k \in \mathcal{K}} Y_{jj}^{kt} \leq \sum_{k \in \mathcal{K}} W_j^{kt} \qquad j \in \mathcal{P}, t \in \mathcal{T} \qquad (53)$$

$$Y_{jj}^{kt} + X_{jj}^{kt} \leq W_j^{kt} + S_j^{\beta, kt} \qquad j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T}. \qquad (54)$$

The left hand side of (52) computes the minimum number of vehicles required, based on the total delivery amount scheduled in t. Thus, the right hand side sets the lower bound for the number of rented vehicles per plant. Equivalently, this lower bound is also

obtained according to the number of vehicles scheduled for deliveries, departing from each plant, as presented by inequalities (53). Finally, inequalities (54) ensure that if vehicle k, housed at plant j, is assigned to a pick up and a delivery in period t, then this vehicle must be rented and cleaned.

It is known that the first-in, first-out rule can be associated with an optimal solution of the IRP. Desaulniers et al. [14] then introduce the following notation. Let $I_l^{0,s} = max\left\{0, I_l^0 - \sum_{t=1}^{s} d_l^t\right\}$ be the remaining quantity of initial inventory for customer l at the end of period $s \in \mathcal{T}$. The residual demands $(\bar{d}_l^s)$ define the demand not met by the initial inventory:

$$\bar{d}_l^s = \begin{cases} \max\left\{0, d_l^1 - I_l^0\right\} & \text{if } s = 1 \\ \max\left\{0, d_l^s - I_l^{0,s-1}\right\} & \text{otherwise} \end{cases} \quad \forall s \in \mathcal{T}. \tag{55}$$

Besides that, the same authors combine the maximum inventory level (U_l), the demand (d_l^s) and the residual demand $(\bar{d}_l^s)$ in $s \in \mathcal{T}$ for each customer l, defining the following set P_{lt}^+, containing all periods in which a sub-delivery of β for a customer l in period t can be used, either to satisfy the current demand or to be held in inventory for future periods.

$$P_{lt}^+ = \left\{t|\bar{d}_l^t > 0\right\} \cup \left\{s > t|\bar{d}_l^s > 0 \text{ and } \sum_{t'=t}^{s-1} d_l^{t'} < U_l\right\} \cup \left\{p+1|\sum_{t'=t}^{p} d_l^{t'} < U_l\right\}. \tag{56}$$

Finally, let $P_{ls}^- = \left\{t \in \mathcal{T}|s \in P_{lt}^+\right\}$ be the set of periods for which a delivery can be scheduled to satisfy the demand of l in $s \in \mathcal{T}$. Desaulniers et al. [14] propose the following valid inequalities for the IRP, derived from the minimum number of sub-deliveries per demand. We adapt them to the 2E-IRPFM, as follows:

$$\sum_{j \in \mathcal{P}} \sum_{k \in \mathcal{K}} \sum_{t \in P_{ls}^-} Y_{jl}^{kt} \geq 1, \qquad l \in \mathcal{C}, s \in \mathcal{T}, \text{ with } P_{ls}^- \neq \emptyset. \tag{57}$$

Lefever [20] propose a set of valid inequalities to the IRP with transshipment (IRPT), useful to bound the minimum number of delivery routes along the planning horizon $\mathcal{T}$. We adapt these inequalities to the 2E-IRPFM, as:

$$\sum_{j \in \mathcal{P}} \sum_{k \in \mathcal{K}} \sum_{t=1}^{s} Y_{jj}^{kt} \geq \left\lceil \frac{\sum_{l \in \mathcal{C}} \sum_{t=1}^{s} \bar{d}_l^t}{Q} \right\rceil, \qquad s \in \mathcal{T}. \qquad (58)$$

Finally, to strengthen the IRP inventory management component formulation, Lefever et al. [21] employ a remaining quantity to restrict the range of continuous variables I_l^t and q_{jl}^{kt}. These improved bounds can be tightened as follows.

$$I_l^t \geq I_l^{0,t}, \qquad\qquad l \in \mathcal{C}, t \in \mathcal{T} \qquad (59)$$

$$q_{jl}^{kt} \leq U_l - I_l^{0,t}, \qquad\qquad j \in \mathcal{P}, l \in \mathcal{C}, k \in \mathcal{K}, t \in \mathcal{T}. \qquad (60)$$

4 Branch-and-cut algorithm

Due to their combinatorial features, the number of subtour elimination constraints (SEC) (14) is too large, and their full enumeration is impracticable. To overcome this limitation, these constraints must be dynamically generated along the search process. We use an exact approach to solve the model presented in Section 3, where SEC are added to the search tree whenever subtours are found at the current solution. This technique is known as a branch-and-cut method and works as follows.

At the beginning of the search, all valid inequalities are generated and added in the root node. Whenever a node of the search tree is solved by a MIP solver, a search for violated SEC is performed. We have used the CVRPSEP package of Lysgaard et al. [22] to generate

SEC. When subtours are identified by CVRPSEP, their corresponding SEC are added to the search tree. This process is repeated until a feasible or dominated solution is found, or until there are no more cuts to be added. At this point, a new subproblem is generated by branching on a fractional variable, and the model is reoptimized in a new node.

We provide a scheme of our branch-and-cut algorithm on **Algorithm 1**.

Algorithm 1 Pseudocode of the proposed B&C algorithm

1: At the root node of the search tree, generate (1)–(13), (15)–(42) and all valid inequalities (44)–(54), (58)–(60).
2: Subproblem solution, Solve the LP relaxation of the node.
3: Termination check:
4: **if** there are no more nodes to evaluate **then**
5:    Stop.
6: **else**
7:    Select one node from the B&C tree.
8: **end if**
9: **while** the solution of the current LP relaxation contains subtours **do**
10:    Add violated subtour elimination constraints.
11:    Subproblem solution. Solve the LP relaxation of the node.
12: **end while**
13: **if** the solution of the current LP relaxation is integer **then**
14:    Go to the termination check.
15: **else**
16:    Branching: branch on one of the fractional variables.
17:    Go to the termination check.
18: **end if**

---

5 Matheuristic-based adaptive large neighborhood search algorithm

Since the 2E-IRPFM generalizes the vehicle routing problem (VRP), it is $\mathcal{NP}$-hard and traditional exact methods, like B&C algorithms, can solve only small size instances. To handle large instances, we propose a matheuristic approach, combining mathematical programming techniques with heuristic search procedures. Matheuristics are widely used

to solve routing problems (see Archetti and Speranza [3]). In particular, Archetti et al. [5] combine integer programming with tabu search to solve a multi-vehicle IRP. More recently, Bertazzi et al. [7] propose a matheuristic to solve a multi-depot IRP.

In the context of the 2E-IRPFM we make use of an adaptive large neighborhood search (ALNS) mechanism, responsible for handling the delivery routes, while pickup, delivery quantities, fleet management, and some improvements are determined exactly by mixed integer programming (MIP) subproblems.

5.1   ALNS mechanism

The ALNS was proposed by Pisinger and Ropke [25] to solve the VRP and its extensions. In the inventory-routing context, it has since been successfully applied to several variants (Aksen et al. [1], Coelho et al. [11, 12], Guimarães et al. [17]). Our ALNS deals with the delivery routes, while delivery quantities, pickups and fleet decisions are determined exactly by MIP subproblems. The search procedure follows the general scheme proposed by Pisinger and Ropke [25] and is divided into Δ segments. Given a solution, some customers are either inserted, removed, or swapped among routes at each iteration, by specific operators. Each operator i contains three attributes. The first one is the weight, given by ω_i, whose value depends on past performance. The second attribute is the score, given by π_i, which quantifies the effect on the solution when the operator is applied. It increases by σ_1 if the operator leads to a new best solution, by σ_2 if it finds a solution better than the current one, and by σ_3 if the solution is worse but is still accepted according to a simulated annealing criteria; the third and last attribute is ς_{ij}, which measures the number of times the operator i has been applied in the last segment $j \in \Delta$.

By the general ALNS framework, we adopt a simulated annealing acceptance criterion. Let s be a solution and s' a neighbor solution, obtained from s. The acceptance probability of s' is $e^{(z(s)-z(s'))/\tau}$, where $z(\cdot)$ is the solution cost and $\tau > 0$ is the current temperature. The initial temperature is τ_{start}, decreasing at each iteration by a cooling rate factor ϕ,

with $0 < \phi < 1$.

In the first iteration, all scores are equal to zero, and all weights are equal to one. A roulette wheel mechanism controls the choice of operators. Given h operators, operator i is chosen with probability $\omega_i / \sum_{j=1}^{h} \omega_j$. In each segment Δ, if the operator was chosen, a reaction factor $\eta \in [0, 1]$ is applied to balance its weight between the past and present performance, according to (61). After this step, all scores are reset to zero.

$$\omega_i := \begin{cases} \omega_i & \text{if } \varsigma_{ij} = 0 \\ (1 - \eta)\omega_i + \eta\pi_i/\varsigma_{ij} & \text{if } \varsigma_{ij} \neq 0. \end{cases} \tag{61}$$

A neighbor solution s' is obtained when an operator is selected and applied on s. In the list below, we present the operators developed for our ALNS.

1. ***Randomly remove ρ***: It randomly selects a period t, a plant j, a vehicle k rented by this plant, and a customer l served from it and removes this customer. It is repeated ρ times.

2. ***Remove worst ρ***: This operator computes the transportation saving of not visiting each client served, according to the triangle inequality. It is applied ρ times, removing, at each time, the customer yielding the highest saving.

3. ***Shaw removal route based***: It randomly selects a period t, a plant j, a vehicle k rented by this plant, and a customer l served by this vehicle. It then computes the distance $min(c_{lu})$ to the closest customer u also being served by the same vehicle, and removes all customers within $2min(c_{lu})$ from customer l.

4. ***Empty one period***: Randomly selects a period t and removes all its deliveries.

5. ***Empty one vehicle***: Randomly selects a rented vehicle k and removes all deliveries from all plants in all periods performed by this vehicle.

6. ***Empty one plant***: Randomly selects a plant j and removes all deliveries of all vehicles from this plant over the planning horizon.

7. ***Farthest customer***: This operator randomly selects a period t, a plant j, and a vehicle rented by this plant, and removes its farthest customer, measured by the direct distance from the depot. It is repeated ρ times.

8. ***Avoid consecutive visits***: Starting from $t = 1$, it removes the second visit of all customers receiving deliveries in two consecutive periods. The delivery quantities are updated at each new period analyzed.

9. ***Remove ρ minimum residual deliveries***: This operator calculates the residual delivery for all served customers, given by $\frac{q_{jl}^{kt}}{U_l - I_l^{t-1}}$, with $q_{jl}^{kt} > 0$, for each $l \in \mathcal{C}$, and removes the customer with the minimum residual delivery. It is repeated ρ times.

10. ***Remove ρ most visited customers***: This operator removes the most visited customer over the planning horizon. Ties are broken randomly. It is repeated ρ times.

11. ***Randomly insert ρ***: Randomly selects a period t, a plant j and a vehicle k rented by this plant, and inserts a randomly selected customer l unserved in t, following the cheapest insertion rule. The operator is repeated ρ times.

12. ***Insert best ρ***: Similar to ***Remove worst ρ***, this operator inserts the customer with the smallest increase in transportation cost. It is repeated ρ times.

13. ***Assignment to the nearest plant***: This operator randomly selects a period t and a customer l not served in that period, and inserts it on the route of its nearest plant, following the cheapest insertion rule. It is repeated ρ times.

14. ***Shaw insertion***: This operator randomly selects a period t and a customer l not served in t. Then it is computes $min(c_{lv})$, with $v \in V''$. All customers not yet served in that period, distant up to $2min(c_{lv})$ are inserted in a rented vehicle k in t, following cheapest insertion rule. It is repeated ρ times.

15. ***Swap ρ customers***: Randomly selects two customers served by different vehicles and swaps their assignments. It is repeated ρ times.

16. ***Swap ρ customers inter-routes***: Randomly selects two customers served by the same plant in different vehicle routes in a given period t, and swaps their assignments. The insertion follows the cheapest rule. It is repeated ρ times.

17. ***Swap ρ customers intra-plants***: Randomly selects two customers served by different plants in a given period, and swaps their assignments, following the cheapest insertion rule. It is repeated ρ times.

5.2 MIP subproblems

Our matheuristic scheme exploits the solutions obtained from three MIP subproblems. These MIPs provide an initial solution for the 2E-IRPFM, which is polished and improved by our ALNS operators, by a solution improvement procedure, and finally by a general route improvement procedure.

5.2.1 Initial solution procedure

The initial solution procedure (ISP) simplifies the 2E-IRPFM model, disregarding route decisions. To this end, all deliveries are scheduled based on direct links between plants and customers, while minimizing pickups, inventory holding, and fleet management costs. For each delivery plan, the visiting sequence is solved exactly as a traveling salesman problem (TSP) as explained next. We employ the branch-and-cut technique proposed by Padberg and Rinaldi [24]. The decision variables are precisely the same of the 2E-IRPFM formulation, except for route variables y. The ISP model is formulated by:

$$\min \sum_{t\in\mathcal{T}} \left(\sum_{j\in\mathcal{P}}\sum_{k\in\mathcal{K}} f_w W_j^{kt} + \sum_{j\in\mathcal{P}}\sum_{k\in\mathcal{K}} f_s S_j^{kt} + \sum_{j\in\mathcal{P}} h_j I_j^t + \sum_{l\in\mathcal{C}} h_l I_l^t + \sum_{k\in\mathcal{K}}\sum_{(i,j)\in\mathcal{E}'} 2c_{ij} X_{ij}^{kt} + \sum_{j\in\mathcal{P}}\sum_{k\in\mathcal{K}}\sum_{l\in\mathcal{C}} c_{jl} Y_{jl}^{kt} \right)$$

$$(62)$$

subject to $(2)-(12)$, $(15)-(40)$ and to:

$$Y_{jj}^{kt} \leq \sum_{l \in \mathcal{C}} Y_{jl}^{kt} \qquad\qquad j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T}. \qquad (63)$$

The objective function (62) minimizes the fleet management (rental and cleaning), inventory holding, pickup and approximated delivery transportation costs, calculated as a direct link cost c_{jl}, between customer l and plant j. Constraints (63) ensure that each vehicle rented by a given plant performs a delivery route if at least one customer is served in this period. Valid inequalities $(47)-(54)$ and $(57)-(60)$ also apply and the OU policy is handled through constraints (43). After solving the ISP, the TSP B&C [24] is solved for each vehicle used in the solution, which already respects capacity due to constraints (11).

5.2.2 Solution improvement procedure

The solution improvement procedure (SIP) plays a central role in the optimization process. It first completes an 2E-IRPFM solution, after each ALNS iteration, determining fleet assignment, pickups, and delivery quantities. It can also slightly improve a solution by removing or inserting delivery customers, scheduling pickups, and swapping customers among routes. Since a destroy ALNS operator may yield an infeasible solution, the insertion mechanism embedded in our SIP can recover feasibility. The parameters used are presented as follows.

- a_{jl}^{kt}: routing reduction cost if customer l is removed from vehicle k rented by plant j in period t, where $Y_{jl}^{kt} = 1$. This cost follows the cheapest removal rule.

- b_{jl}^{kt}: routing cost if customer l is inserted in the route of vehicle k rented by plant j in period t, where $Y_{jl}^{kt} = 0$. This cost follows the cheapest insertion rule.

- ψ_{jl}^{kt}: a binary parameter equal to 1, if customer l is served in the current route of vehicle k rented by plant j in period t, where $Y_{jl}^{kt} = 1$, 0 otherwise.

Our SIP model keeps all decision variables from the 2E-IRPFM, except for the y routing variables. Besides that, visiting customer variables Y are replaced by the following new ones:

- $\delta_{jl}^{kt} = 1$ if customer l is removed from the existing route of vehicle k rented by plant j in period t, which obviously has $\psi_{jl}^{kt} = 1$, and 0 otherwise;

- $\omega_{jl}^{kt} = 1$ if customer l is inserted in the route of vehicle k rented by plant j in period t, which obviously has $\psi_{jl}^{kt} = 0$, and 0 otherwise.

The SIP model is then formulated as:

$$\min \sum_{t \in \mathcal{T}} \left(\sum_{j \in \mathcal{P}} h_j I_j^t + \sum_{l \in \mathcal{C}} h_l I_l^t + \sum_{k \in \mathcal{K}} \sum_{(i,j) \in \mathcal{E}'} 2c_{ij} X_{ij}^{kt} + \sum_{j \in \mathcal{P}} \sum_{k \in \mathcal{K}} \left[f_w W_j^{kt} + f_s S_j^{kt} + \sum_{l \in C} b_{jl}^{kt} \omega_{jl}^{kt} - \sum_{l \in C} a_{jl}^{kt} \delta_{jl}^{kt} \right] \right) \tag{64}$$

subject to $(2)-(9)$, (11), $(15)-(39)$, and to:

$$\omega_{jl}^{kt} \leq 1 - \psi_{jl}^{kt} \qquad j \in \mathcal{P}, l \in \mathcal{C}, k \in \mathcal{K}, t \in \mathcal{T} \tag{65}$$

$$\delta_{jl}^{kt} \leq \psi_{jl}^{kt} \qquad j \in \mathcal{P}, l \in \mathcal{C}, k \in \mathcal{K}, t \in \mathcal{T} \tag{66}$$

$$q_{jl}^{kt} \leq \left(\psi_{jl}^{kt} - \delta_{jl}^{kt} + \omega_{jl}^{kt} \right) U_l \qquad j \in \mathcal{P}, l \in \mathcal{C}, k \in \mathcal{K}, t \in \mathcal{T} \tag{67}$$

$$\sum_{j \in \mathcal{P}} \sum_{k \in \mathcal{K}} \left(\psi_{jl}^{kt} - \delta_{jl}^{kt} + \omega_{jl}^{kt} \right) \leq 1 \qquad l \in \mathcal{C}, t \in \mathcal{T} \tag{68}$$

$$\psi_{jl}^{kt} - \delta_{jl}^{kt} + \omega_{jl}^{kt} \leq Y_{jj}^{kt} \qquad j \in \mathcal{P}, l \in \mathcal{C}, k \in \mathcal{K}, t \in \mathcal{T} \tag{69}$$

$$\sum_{j \in \mathcal{P}} \sum_{l \in \mathcal{C}} \left(\delta_{jl}^{kt} + \omega_{jl}^{kt} \right) \leq G \qquad k \in \mathcal{K}, t \in \mathcal{T} \tag{70}$$

$$\omega_{jl}^{kt}, \delta_{jl}^{kt} \in \{0,1\} \qquad j \in \mathcal{P}, l \in \mathcal{C}, k \in \mathcal{K}, t \in \mathcal{T}. \tag{71}$$

The objective function (64) minimizes the inventory holding, pickups, fleet management, removal and insertion costs. Constraints (65) forbid inserting a customer on a route that already serves it, while (66) guarantee that a customer can only be removed if it is served by the route. Constraints (67) link removal and insertion variables with quantities delivered. Constraints (68) avoid split deliveries, while constraints (69) force the departure of a given vehicle k rented by plant j, if any customers are assigned to it. Constraints (70) limit the total number of removals and insertions by a constant $G \in \mathbb{Z}^+$. This condition is valid only for deliveries, while pickups are free to be optimized. This is less strict than the one used by Guimarães et al. [17]. New variables domain are defined by (71).

The performance of SIP is strongly dependent on G. When G is small, feasibility may not be recovered when a destroy operator is applied on the ALNS stage. Otherwise, when its value is large, the removals and/or insertions on the routes may not lead to any improvement, because of the approximated transportation costs. We consider a dynamic arrangement to set up an appropriate value for G, detailed in Section 5.3. The OU policy is handled through the constraints (72).

$$q_{jl}^{kt} \geq \left(\psi_{jl}^{kt} - \delta_{jl}^{kt} + \omega_{jl}^{kt} \right) U_l - I_l^{t-1} \qquad j \in \mathcal{P}, l \in \mathcal{C}, k \in \mathcal{K}, t \in \mathcal{T}. \qquad (72)$$

As a final polishing, each time the SIP model is solved, the TSP procedure is applied to provide an optimal sequence of visits for each vehicle route, allowing us to compute the cost of a new solution exactly.

5.2.3 General improvement routing procedure

To overcome the drawback arising from the approximated transportation costs present in many ALNS operators, we introduce a novel way to optimize inventory holding and fleet management costs, while performing routing improvements considering real transportation costs. To the best of our knowledge, this is a new approach introduced in this paper

and useful for other routing problems.

The general improvement routing procedure (GIRP) can apply any movement in all existing routes in a given solution (either a removal, an insertion, or a swap), taking into account the true transportation costs, unlike SIP. Let $\mathcal{A}_{jkt}$ be the set of customers $l \in C$ served by vehicle k rented by plant j in period t, which obviously has $Y_{jj}^{kt} = 1$. If $\mathcal{A}_{jkt} = \emptyset$, it means that there are no customers being served by that vehicle, i.e., $Y_{jj}^{kt} = 0$. Consider also a new set of arcs $\mathcal{E}_{jkt} \subseteq \mathcal{E}''$, with $\mathcal{E}_{jkt} = \{(u, v) : u, v \in \{j\} \cup \mathcal{A}_{jkt} \cup \mathcal{A}_{jkt}^c\}$, where $\mathcal{E}_{jkt}$ represents edges adjacent to plant j, the set of customers in the existing routes $\mathcal{A}_{jkt}$, and its complement $\mathcal{A}_{jkt}^c$, where $|\mathcal{A}_{jkt}| > 0$. We note that the set generated by $\mathcal{A}_{jkt} \cup \mathcal{A}_{jkt}^c = \mathcal{C}$.

Since all the decisions concerning fleet management, inventory flows, pickup, and deliveries are interdependent, GIRP works as the 2E-IRPFM model, with a much smaller search space. In this sense, removals, insertions, and customers swaps are allowed only on established routes, avoiding customers from being served by new ones. Formally, variables q_{jl}^{kt}, Y_{jl}^{kt} and y_{uv}^{kjt} are free to be optimized if and only if their associated routes exist, where $(u, v) \in \mathcal{E}_{jkt}$ and $|\mathcal{A}_{jkt}| > 0$. Otherwise, q_{jl}^{kt}, Y_{jl}^{kt} and y_{uv}^{kjt} are set to zero, where $(u, v) \in \mathcal{E}_{jkt}$ and when $\mathcal{A}_{jkt} = \emptyset$.

The remainder of the 2E-IRPFM formulation is not affected, and all other decision variables are free to be optimized. Moreover, for a given solution and a positive integer parameter $\mathcal{B}$, we add the following constraints to the GIRP model, inspired by the local branching constraints of Fischetti and Lodi [16].

$$\sum_{l \in \mathcal{A}_{jkt}} \left(1 - Y_{jl}^{kt}\right) + \sum_{l \in \mathcal{A}_{jkt}^c} Y_{jl}^{kt} \leq \mathcal{B} \qquad j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T}, |\mathcal{A}_{jkt}| > 0. \qquad (73)$$

The left hand side of constraints (73) counts the number of binary variables exchanging their value with respect to each existing route from a solution $\bar{s}$, either from 1 to 0 or from 0 to 1, respectively. The set of 2E-IRPFM solutions satisfying (73) define the $\mathcal{B}$-OPT neighborhood $\mathcal{N}(\bar{s}, \mathcal{B})$ of $\bar{s}$. The neighborhood size $\mathcal{B}$ should be properly chosen,

so that the neighborhood $\mathcal{N}(\bar{s}, \mathcal{B})$ must be small enough to be explored thoroughly in a reasonable time, but sufficiently large to maximize the probability of finding solutions better than $\bar{s}$. Beyond that, we note that all valid inequalities $(44)-(60)$ are applicable to GIRP and the OU inventory policy is ensured by imposing constraints (43).

Figure 2 shows an example of how the GIRP works. Figure 2a shows a solution for which the sets $\mathcal{A}_{11t} = \{1, 2, 3\}$, $\mathcal{A}_{21t} = \{7, 8, 9\}$ and $\mathcal{A}_{22t} = \{5, 6\}$ represent all customers served by vehicle $k = 1$ rented by plant $j = 1$, and $k = 1$ and $k = 2$ from plant $j = 2$, respectively. Setting $\mathcal{B} = 2$, GIRP is able to perform up to two movements on each route. As shown in Figure 2b, unserved customer $l = 4$, and also $l = 5$ who was served by $k = 2$ from $j = 2$, are inserted on route $k = 1$ from plant $j = 1$. This last insertion removes $l = 5$ from $k = 2$. As customer $l = 6$ is inserted in route $k = 1$, these two removals empty vehicle $k = 2$, while the removal of $l = 8$ leads to one insertion and one removal in vehicle $k = 1$ from plant $j = 2$. Finally, $\mathcal{A}_{11t} = \{1, 2, 3, 4, 5\}$ and $\mathcal{A}_{21t} = \{6, 7, 9\}$ are the new sets of customers served in t, while all others $\mathcal{A}_{jkt} = \emptyset$.

5.3   Math-ALNS general framework

Our Math-ALNS starts by solving the ISP model and all associated TSPs, which provide an initial solution s_{ini}. When an ALNS operator is applied on a given solution s (s_{ini} in the first iteration), a neighboring solution s' is obtained by solving the SIP model. We initially define $G = n + m$ where $n = |\mathcal{P}|$ and $m = |\mathcal{C}|$, which guarantees feasibility is recovered if it was lost by a destroy operator. While $z(s') < z(s)$, G is decreased by one unit until $G = 1$, and the SIP model is solved after each ALNS iteration. Otherwise, if $z(s') \leq (1 + \epsilon)\, z(s)$, where $\epsilon \sim U[0.05, 0.15]$, we accept s' as a new incumbent solution. Then, we also define $G = max\,\{\xi * (n + m), 1\}$, with $\xi \sim U[0.1, 0.2]$ and solve SIP once again. After that, all individual routes are optimized by solving their associated TSPs.

Whenever a new best solution s_{best} is found, we enumerate all its routes. Then, the GIRP model is generated and solved, yielding s'''. Due to the critically of neighborhood-size $\mathcal{B}$,
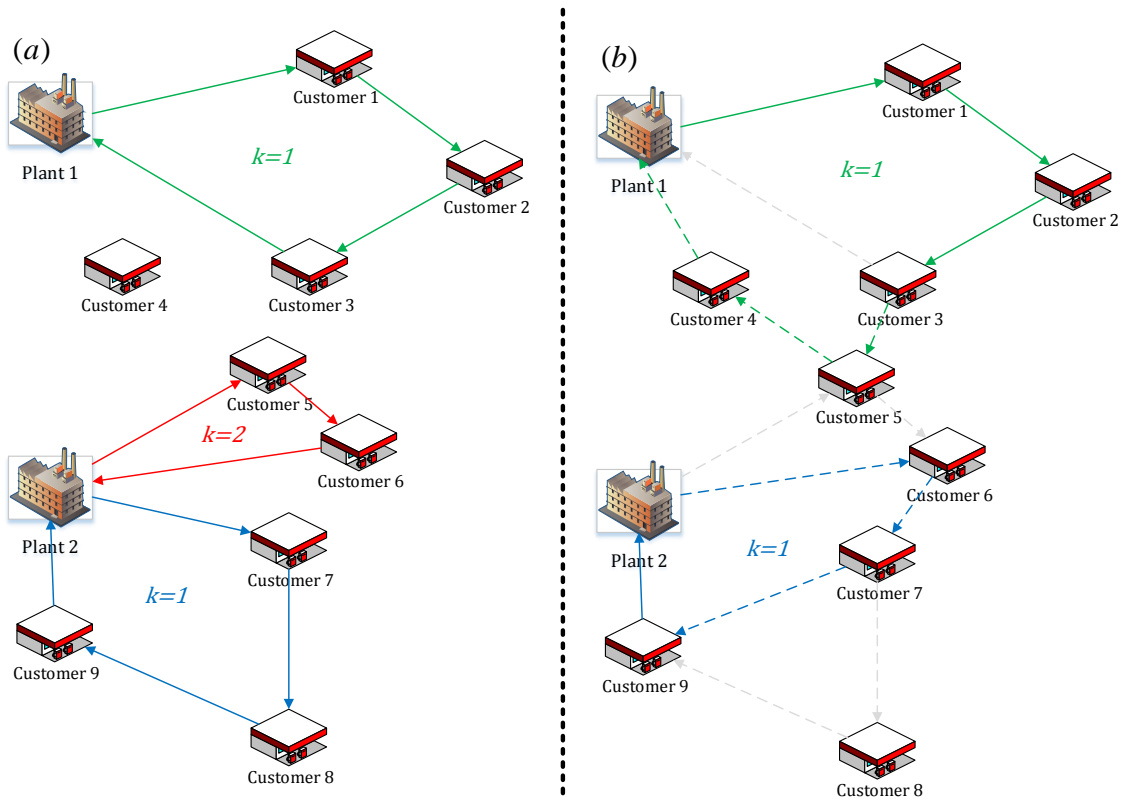
Figure 2: An example for the general improvement routing procedure (GIRP), before (a) and after (b) all movements

we only accept the neighbor solution if $z(s''') < z(s_{best})$, and then the optimization process continues. Because of the complexity of SIP and GIRP subproblems, our Math-ALNS is executed for about 1000 iterations. In order to roughly generate this number of iterations, we set $\tau_{start} = 8000$ and $\phi = 0.989$. Scores are updated with $\sigma_1 = 10$, $\sigma_2 = 5$ and $\sigma_3 = 2$, and the reaction factor η is set to 0.8. We define $\Delta = 20$, when weights are updated, scores are set to zero, and the value of ϵ is redrawn. A pseudocode of our matheuristic is provided in **Algorithm 2**.

6   Branch-and-cut-based ALNS algorithm

During preliminary experiments, we observed that our B&C algorithm is very effective in solving small size instances, but as expected from an exact procedure, the upper bound quality deteriorates dramatically for medium and large instances. On the other hand, our Math-ALNS is very powerful to find good upper bounds in very short computing, even for very large instances. Based on that, we hybridize a branch-and-cut-based ALNS scheme, which we name H-B&C, in order to allow B&C algorithm to take advantage of the good solutions of our Math-ALNS and vice-versa.

Employing parallel computing, H-B&C starts from two fronts, one solving the pure B&C of Section 4, and the other one executing our Math-ALNS described in **Algorithm 2**. Whenever a new best solution is found by one of the algorithms, it is immediately provided to other one. This strategy is used to provide better upper bounds to B&C, especially in large instances. Since B&C stops when the optimal solution is found, it avoids wasting time exploring ALNS neighborhoods over and over again on small instances. The algorithm runs until a time limit is reached or an optimal solution is proved, which characterizes it as an exact method. Figure 3 illustrates the dynamics of the proposed H-B&C algorithm.

Algorithm 2 Math-ALNS pseudocode

1: Initialize weights to 1 and scores to 0 for all operators;

2: Solve ISP and all associated TSPs, yielding s_{ini}.

3: $s_{best} \leftarrow s \leftarrow s_{ini}$; $\tau \leftarrow \tau_{start}$; $iter = 0$; Generate ϵ;

4: **while** $\tau \geq 0.01$ and $iter \leq 1,000$ **do**

5:     Select and apply an operator i to s; set $G = n + m$;

6:     Solve SIP and all associated TSPs, yielding s';

7:     **if** $z(s') < z(s)$ **then**

8:         $s \leftarrow s'$;

9:         Set $G = max\{G - 1, 1\}$;

10:         Solve SIP and all associated TSPs, yielding s'';

11:         **if** $z(s'') < z(s)$ **then**

12:             $s \leftarrow s''$ and go to step (9);

13:         **else**

14:             **if** $z(s'') \leq (1 + \epsilon)\, z(s)$ **then**

15:                 Set $G = max\{\xi * (n + m), 1\}$ and go to step (10);

16:             **end if**

17:         **end if**

18:         **if** $z(s) < z(s_{best})$ **then**

19:             $s_{best} \leftarrow s$;

20:             $\pi_i \leftarrow \pi_i + \sigma_1$;

21:             Solve GIRP, yielding s''';

22:             **if** $z(s''') < z(s_{best})$ **then**

23:                 $s_{best} \leftarrow s \leftarrow s'''$;

24:             **end if**

25:             Go to step (9);

26:         **else**

27:             $\pi_i \leftarrow \pi_i + \sigma_2$;

28:             **if** $z(s'') \leq (1 + \epsilon)\, z(s_{best})$ **then**

29:                 Set $G = max\{\xi * (n + m), 1\}$ and go to step (10);

30:             **end if**

31:         **end if**

32:     **else**

33:         **if** $e^{(z(s) - z(s'))/\tau} > \Theta$, where $\Theta \sim U[0, 1]$ **then**

34:             $s \leftarrow s'$;

35:             $\pi_i \leftarrow \pi_i + \sigma_3$;

36:         **end if**

37:     **end if**

38:     **if** $(iter \mod \Delta) = 0$ **then**

39:         Generate ϵ, update the weights of all operators and reset their scores;

40:         $s \leftarrow s_{best}$;

41:     **end if**

42:     $iter \leftarrow iter + 1$;

43:     $\tau \leftarrow \phi\tau$;

44: **end while**

45: **return** s_{best};
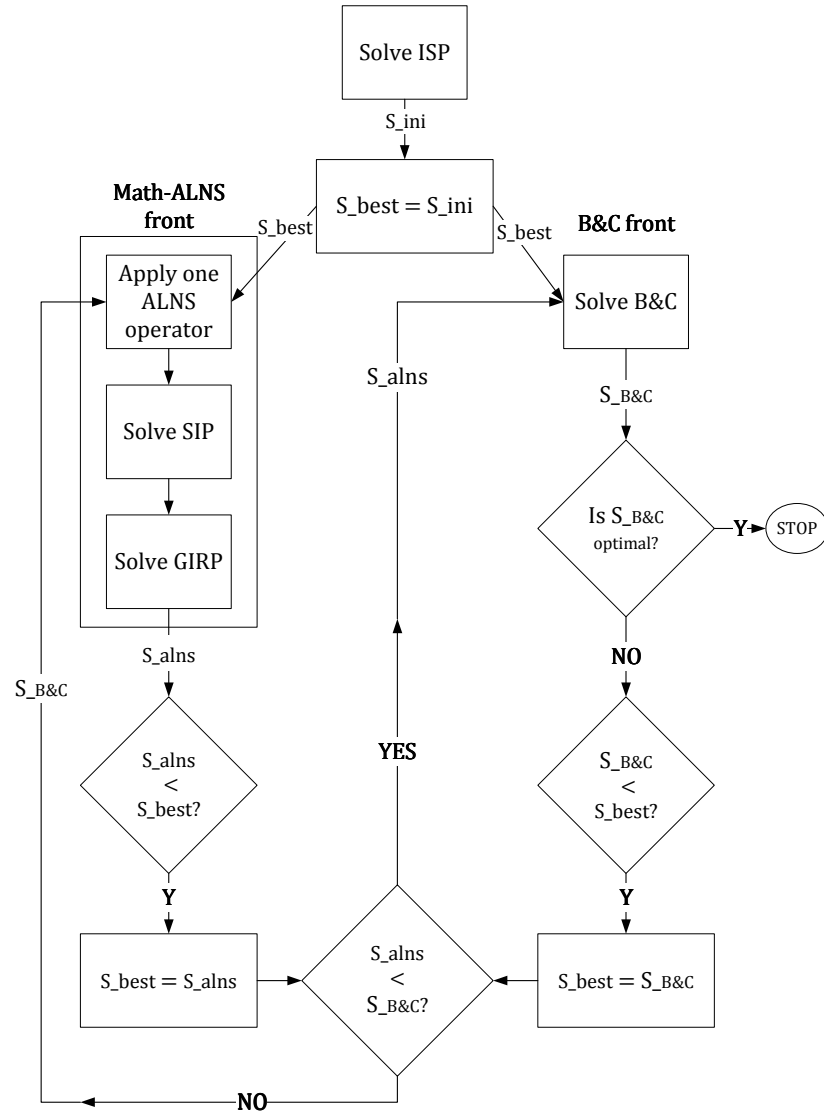
Figure 3: Scheme of our H-B&C parallel algorithm

7 Computational experiments

All algorithms were coded in C++, executed on a grid of Intel(R) Xeon(R) processors at 2.60GHz with up to 16 GB of RAM per node, running in CentOS Linux operating system. All MIPs were solved by Gurobi 8.1.0 and both pure B&C and Math-ALNS were processed using six threads. After preliminary tests, we split the hybrid algorithm H-B&C in four threads dedicated to B&C front, and two threads to Math-ALNS front. ISP and SIP are solved to optimality while GIRP is executed for 200s. The algorithms ran up to 7200s on each experiment.

7.1 Overall results for 2E-IRPFM

We have adapted the 2E-MDIRP instances proposed by Guimarães et al. [17], wich one derived from Archetti et al. [4]. Four settings are considered: one supplier-one plant, two suppliers-two plants, two suppliers-three plants, and three suppliers-two plants. The number of customers ranges from 5 to 50. Inventory costs and planning horizon make up four groups: low inventory cost with three (absH3low) and six (absH6low) periods, and high inventory cost with three (absH3high) and six (absH6high) periods. To generate a variety of cleaning and rental costs scenarios, we created four additional groups: low rental-low cleaning (LRLS), low rental-high cleaning (LRHS), high rental-low cleaning (HRLS), and high rental-high cleaning (HRHS) costs. We generate a total of 256 instances. Due to the 2E-IRPFM multi-vehicle topology, we consider only the three-vehicle instances from the 2E-MDIRP, which are available for each plant at the rental company. The parameters are calculated as follows and all instances generated and detailed results are available from https://www.leandro-coelho.com/two-echelon-inventory-routing-problem-with-fleet-management/.

- High rental cost (HR): $f_w = \lceil \Omega Q \rceil$, where $\Omega \sim U[0.4, 0.6]$

- Low rental cost (LR): $f_w = \lceil \Omega Q \rceil$, where $\Omega \sim U[0.2, 0.3]$

- High cleaning cost (HS): $f_s = \lceil \Omega Q \rceil$, where $\Omega \sim U[0.3, 0.5]$

- Low cleaning cost (LS): $f_s = \lceil \Omega Q \rceil$, where $\Omega \sim U[0.1, 0.2]$

We start our analysis by presenting the results obtained with the exact algorithms. Table 1 shows a comparison between B&C and H-B&C for ML and the OU policies. On each supply chain considered, we have 16 instances, with four (one instance absH3low, one absH6low, one absH3high, and one absH6high) on each fleet management cost combination (LRLS, LRHS, HRLS, HRHS). The first three columns describe the supply chain structure, where $|\mathcal{C}|$, $|\mathcal{P}|$ and $|\mathcal{F}|$ represent the number of customers, plants and suppliers, respectively. For each method considered, columns **OPT** and **SF** show the number of optimal and feasible solutions found. Column $\overline{LB}$ presents the average lower bound, while $\overline{T(s)}$ shows the run time.

Due to the difficulty of the 2E-IRPFM, the performance of the exact methods clearly deteriorate as the supply chain structure becomes more complex, especially for the OU policy. Given that the H-B&C gets an initial solution from the Math-ALNS front, the algorithm finds a solution for every instance. Furthermore, H-B&C was able to find three additional optimal solutions for the ML policy. The overall results do not compromise the LB quality, showing the efficiency of B&C and Math-ALNS combination. Among the 512 instances evaluated on both policies, H-B&C found 212 better solutions than B&C and there were 300 ties.

We also compare the performance of B&C with H-B&C, on the subset of instances where B&C obtained a solution. Table 2 has up to 32 instances on each supply chain structure, making up 128 instances per row (64 ML and 64 OU). As observed, H-B&C yielded tighter bounds, in an equivalent running time.

Table 3 presents the mean results among 64 instances on each inventory policy, for Math-ALNS and H-B&C. As H-B&C took advantage from the B&C front, it was able to solve small instances in much shorter running time. The UBs were equivalent, showing the quality of our heuristic method. Considering both inventory policies, we highlight that

Table 1: Comparison between B&C and H-B&C on the 2E-IRPFM

			ML							OU						
			B&C				H−B&C			B&C				H−B&C		
$|\mathcal{C}|$	$|\mathcal{P}|$	$|\mathcal{F}|$	OPT	SF	$\overline{LB}$	$\overline{T(s)}$	OPT	$\overline{LB}$	$\overline{T(s)}$	OPT	SF	$\overline{LB}$	$\overline{T(s)}$	OPT	$\overline{LB}$	$\overline{T(s)}$
	1	1	16	16	4423.4	0.9	16	4423.4	3.4	16	16	4672.8	1.6	16	4672.8	4.7
	2	2	16	16	4383.5	375.2	16	4383.5	313.1	16	16	4719.0	363.6	16	4719.0	210.9
5	2	3	16	16	4343.1	225.7	16	4343.1	417.2	16	16	4679.7	232.1	16	4679.7	233.3
	3	2	14	16	4151.9	1628.4	16	4191.3	1359.1	15	16	4464.8	1276.4	16	4495.1	1024.2
	1	1	16	16	6128.4	5.9	16	6128.4	10.0	16	16	6460.1	12.8	16	6460.1	23.6
	2	2	8	16	5959.7	3823.2	8	5901.6	3731.6	8	16	6268.0	3812.1	8	6237.0	3857.2
10	2	3	8	16	5942.3	3689.3	8	5861.1	3712.1	8	16	6202.9	3773.0	8	6192.5	3814.9
	3	2	8	16	5646.0	3997.0	8	5650.8	4044.8	8	16	5966.7	3733.5	8	5989.5	3693.4
	1	1	16	16	7898.8	128.9	16	7898.8	144.2	16	16	8635.7	479.2	16	8635.7	633.7
	2	2	8	16	6932.5	4613.3	8	6962.5	4817.4	3	16	7433.2	6803.7	2	7469.8	6966.9
25	2	3	8	16	6940.5	4422.6	8	6938.3	4678.1	3	16	7425.5	6668.5	3	7474.6	6816.3
	3	2	4	13	6909.5	6536.3	5	6913.1	6534.9	0	13	7368.5	7200.0	0	7404.9	7200.0
	1	1	7	16	13239.4	5271.1	7	13317.1	5469.1	0	16	13915.2	7200.0	0	14305.4	7200.0
	2	2	0	8	10923.4	7200.0	0	10964.5	7200.0	0	11	11549.5	7200.0	0	11750.3	7200.0
50	2	3	0	8	10771.6	7200.0	0	10921.2	7200.0	0	9	11455.2	7200.0	0	11664.6	7200.0
	3	2	0	8	10891.9	7200.0	0	10928.2	7200.0	0	6	11886.8	7200.0	0	11978.0	7200.0
Total			145	229			148			125	231			125		
Avg					7217.9	3519.9		7232.9	3552.2			7694.0	3947.3		7758.1	3954.9

Table 2: Average results for B&C and H-B&C to 2E-IRPFM, where B&C found a solution

$|\mathcal{C}|$	SF	B&C			H−B&C		
		$\overline{Z}$	$\overline{GAP}$	$\overline{T(s)}$	$\overline{Z}$	$\overline{GAP}$	$\overline{T(s)}$
5	128	4488.5	0.2	513.0	4488.5	0.0	445.7
10	128	6520.2	5.3	2855.8	6476.8	5.1	2860.9
25	122	9138.3	12.5	4587.4	8253.3	8.2	4704.7
50	82	12092.9	20.3	6958.9	10583.1	8.0	6983.6
Avg		**8060.0**	**9.6**	**3728.8**	**7450.4**	**5.3**	**3748.8**

H-B&C found better solutions in 54 cases, against 45 from Math-ALNS. In another perspective, H-B&C reached the BKS in 467 cases while Math-ALNS did in 458, having 51 exclusive BKS, six more than Math-ALNS.

Table 3: Comparison between Math-ALNS and H-B&C on the 2E-IRPFM

	ML				OU			
	Math−ALNS		**H−B&C**		**Math−ALNS**		**H−B&C**	
$|\mathcal{C}|$	$\overline{Z}$	$\overline{T(s)}$	$\overline{Z}$	$\overline{T(s)}$	$\overline{Z}$	$\overline{T(s)}$	$\overline{Z}$	$\overline{T(s)}$
5	4335.4	2144.6	4335.4	523.2	4641.6	1807.3	4641.6	368.3
10	6262.2	2771.7	6263.2	2874.6	6690.3	2860.6	6690.5	2847.2
25	8081.4	2873.6	8079.0	4043.6	8852.1	3108.1	8860.1	5404.2
50	12861.3	4388.6	12882.8	6767.3	14289.9	4674.3	14298.4	7200.0
Avg	**7885.1**	**3044.6**	**7890.1**	**3552.2**	**8618.5**	**3112.6**	**8622.7**	**3954.9**

Since H-B&C and Math-ALNS reached similar results without a clear dominance, all following analyses are performed taking into account the BKS on each instance on each inventory policy.

7.2 Cost analysis for 2E-IRPFM

We evaluate the impact of choosing each of the inventory policies from the perspective of total cost. As shown in Table 4, imposing the OU policy at customers increases the total cost in almost 8.6% on average, and its impact is greater in more complex supply chain structures. These results are consistent with the findings of Archetti et al. [4] for the basic IRP, Coelho et al. [11] for multi-vehicle IRP, Coelho et al. [12] for the IRP with transshipment, and Guimarães et al. [17] for the 2E-MDIRP.

Tables 5 and 6 make it possible to evaluate the effect of supply chain complexity on average costs. For the ML policy, it is interesting to observe that in a more complex system with more plants and suppliers, cleaning costs are less representative. This is due to the fact that deliveries can be better coordinated among different plants, reducing the

Table 4: Comparison for inventory policies on BKS of the 2E-IRPFM

$|\mathcal{C}|$	ML	OU	GAP %
5	4335.4	4641.6	7.1
10	6262.2	6688.6	6.8
15	8070.0	8846.2	9.6
50	12851.1	14257.3	11.0
Avg	**7879.7**	**8608.5**	**8.6**

need for vehicle cleaning. Another interesting point is that the share of rental costs does not change among the policies, while cleaning costs are not influenced by the complexity of the system. This fact can be explained by the loss of delivery flexibility when the inventory policy is more strict.

Table 7 shows a sensitivity analysis when the rental and cleaning costs change, with respect to the LRLS case. The total cost increases 4.5% (ML) and 4.9% (OU) on average, when the cleaning cost changes from low to high. Due to the higher coordination, it is relevant mentioning that the system complexity can mitigate this variation, especially when instances are large. For both policies, the total cost increases around 12% when rental costs shift from low to high, and more than 18% when both rental and cleaning costs are high.

7.3 Fleet management analysis

We also carry out an analysis regarding the fleet management decision. Aiming to provide a compact evaluation, we investigate the quantity-distance $(\boldsymbol{q/dist})$ ratio, which computes the volume delivered per distance traveled. Equation (74) shows the index.

$$\frac{q}{dist} = \frac{\sum\limits_{t\in\mathcal{T}}\sum\limits_{j\in\mathcal{P}}\sum\limits_{k\in\mathcal{K}}\sum\limits_{l\in\mathcal{C}} q_{jl}^{kt}}{\sum\limits_{t\in\mathcal{T}}\sum\limits_{j\in\mathcal{P}}\sum\limits_{k\in\mathcal{K}}\sum\limits_{(u,v)\in\mathcal{E}''} c_{uv}y_{uv}^{kjt}}. \tag{74}$$

Table 5: Fleet management cost as % of total cost for the ML policy

$|\mathcal{C}|$	$|\mathcal{P}|$	$|\mathcal{F}|$	Rental Cost %				Cleaning Cost %			
			LRLS	LRHS	HRLS	HRHS	LRLS	LRHS	HRLS	HRHS
5	1	1	5.2	7.2	10.7	10.1	3.4	3.3	3.1	6.6
	2	2	6.0	7.0	11.1	11.5	1.6	2.8	1.5	3.4
	2	3	6.1	6.5	10.6	11.2	2.1	3.5	2.0	3.3
	3	2	6.5	8.7	13.1	12.6	4.0	3.6	3.7	6.2
10	1	1	10.2	12.0	19.0	19.2	3.1	4.6	2.7	5.3
	2	2	8.9	10.7	15.8	15.3	2.4	2.6	2.6	4.5
	2	3	9.0	11.0	16.1	15.6	2.4	2.6	2.8	4.0
	3	2	8.6	10.1	15.6	15.5	2.8	3.1	3.0	3.6
25	1	1	13.9	15.7	24.9	22.4	7.8	7.9	6.8	11.7
	2	2	12.7	19.9	21.9	20.4	6.4	2.9	6.1	7.5
	2	3	13.8	17.3	22.4	19.6	4.7	4.7	6.1	6.9
	3	2	12.6	16.9	21.3	21.5	4.9	4.6	5.2	6.7
50	1	1	19.9	25.1	30.7	26.3	9.2	10.5	7.9	19.5
	2	2	19.5	21.1	31.7	30.5	6.3	6.2	5.3	6.4
	2	3	19.2	19.7	33.2	31.9	5.5	7.0	6.0	6.7
	3	2	19.6	20.2	32.7	31.7	2.9	4.9	2.5	5.3
Avg			**12.0**	**14.3**	**20.7**	**19.7**	**4.3**	**4.7**	**4.2**	**6.7**

Table 6: Fleet management cost as % of total cost for the OU policy

$|\mathcal{C}|$	$|\mathcal{P}|$	$|\mathcal{F}|$	Rental Cost %				Cleaning Cost %			
			LRLS	LRHS	HRLS	HRHS	LRLS	LRHS	HRLS	HRHS
5	1	1	6.0	7.7	12.4	11.9	2.1	3.2	2.0	5.8
	2	2	5.8	7.9	10.1	11.5	2.7	2.8	2.6	4.4
	2	3	6.5	7.6	9.7	11.1	2.0	3.2	2.7	3.6
	3	2	7.1	10.1	13.5	12.1	3.7	3.4	3.2	6.1
10	1	1	9.7	11.3	18.2	18.3	3.2	5.2	2.9	6.5
	2	2	8.7	10.6	15.5	15.6	2.7	3.0	3.0	4.6
	2	3	9.0	10.4	16.4	16.4	3.4	2.8	3.1	4.7
	3	2	8.7	11.1	15.6	15.6	3.5	3.3	3.1	6.2
25	1	1	14.8	17.0	24.4	22.4	5.7	7.2	4.5	12.4
	2	2	11.9	16.6	22.0	21.8	6.6	5.6	5.3	7.1
	2	3	12.5	14.9	21.6	20.4	6.2	4.8	5.3	8.0
	3	2	12.7	15.9	22.5	21.9	5.4	4.2	5.2	6.8
50	1	1	22.5	24.6	32.0	29.0	5.6	9.7	4.8	12.2
	2	2	17.8	20.6	29.0	28.3	5.2	6.6	4.7	9.0
	2	3	17.8	21.1	30.4	29.7	6.3	7.1	5.7	8.4
	3	2	18.2	20.5	30.6	29.7	5.1	4.9	4.5	7.8
Avg			**11.9**	**14.2**	**20.2**	**19.7**	**4.3**	**4.8**	**3.9**	**7.1**

Table 7: Comparison of fleet management costs

$|\mathcal{C}|$	$|\mathcal{P}|$	$|\mathcal{F}|$	ML				OU			
			LRLS	Δ%HS	Δ%HR	Δ%HRHS	LRLS	Δ%HS	Δ%HR	Δ%HRHS
5	1	1	4168.7	3.8	7.2	13.4	4395.2	3.1	8.8	13.4
	2	2	4197.9	2.6	6.1	9.0	4508.8	2.6	6.0	10.0
	2	3	4171.2	2.4	5.7	8.3	4482.2	2.5	5.8	9.3
	3	2	3941.8	3.1	8.9	13.3	4217.0	3.1	9.4	13.9
10	1	1	5693.8	3.9	10.5	16.0	6004.0	4.4	10.0	15.9
	2	2	5857.7	3.3	9.6	13.7	6282.0	3.0	9.2	13.6
	2	3	5841.1	2.6	9.7	13.2	6289.7	2.3	9.4	13.2
	3	2	6105.9	2.8	8.6	11.9	6519.1	3.2	8.6	13.1
25	1	1	6951.7	9.1	16.1	29.3	7654.7	7.1	16.8	27.4
	2	2	7201.8	3.9	13.7	21.0	7892.0	5.9	14.0	21.6
	2	3	7346.2	3.6	11.2	17.7	7941.4	5.1	12.5	18.3
	3	2	7862.4	4.0	12.2	18.3	8600.8	4.0	13.6	18.8
50	1	1	11473.9	13.3	16.3	37.4	12771.7	9.8	18.2	31.3
	2	2	11181.6	5.9	18.8	23.5	12230.5	8.3	18.9	27.2
	2	3	10980.4	4.7	22.6	27.5	12259.0	6.8	19.3	27.1
	3	2	11694.7	2.7	18.7	22.9	13011.3	6.3	17.3	24.5
Avg			**7166.9**	**4.5**	**12.3**	**18.5**	**7816.2**	**4.9**	**12.4**	**18.7**

As pointed by Song and Savelsbergh [28], $q/dist$ is not effective to measure absolute performance. We then introduce two new performance indices, in order to evaluate fleet management strategies among different costs combinations. The gross fleet usage (GFU) shown in (75) computes the average occupancy of the rented fleet. The numerator calculates the total volume delivered to customers, while the denominator computes the total rented capacity. GFU is particularly useful to compute rented fleet idleness, derived from delivery and fleet management strategy facing different rental and cleaning costs combinations.

$$GFU = \frac{\sum\limits_{t\in\mathcal{T}}\sum\limits_{j\in\mathcal{P}}\sum\limits_{k\in\mathcal{K}}\sum\limits_{l\in\mathcal{C}} q_{jl}^{kt}}{\left(\sum\limits_{t\in\mathcal{T}}\sum\limits_{j\in\mathcal{P}}\sum\limits_{k\in\mathcal{K}} W_j^{kt}\right) Q}. \tag{75}$$

Given that rented vehicles can be housed at plants and not being used, the net fleet usage (NFU) presented in (76) computes the average occupancy of the fleet. Thus, it is possible to split the delivery strategy from fleet management decisions in our analyses.

$$NFU = \frac{\sum\limits_{t\in\mathcal{T}}\sum\limits_{j\in\mathcal{P}}\sum\limits_{k\in\mathcal{K}}\sum\limits_{l\in\mathcal{C}} q_{jl}^{kt}}{\left(\sum\limits_{t\in\mathcal{T}}\sum\limits_{j\in\mathcal{P}}\sum\limits_{k\in\mathcal{K}} Y_{jj}^{kt}\right) Q}. \tag{76}$$

In Table 8, each row reports the average of 128 instances (64 for each inventory policy), according to the supply chain structure. As expected, the $q/dist$ index does not reveal anything distinctive among cost combinations. Overall, we observed that for large instances, the fleet carries a greater volume per traveled distance. The clearest conclusion lies on GFU and NFU comparison on the LRHS combination. We note that the rented fleet average occupancy is 66%, but for vehicles actually assigned it rises up to 81%. When cleaning cost is low or both are high, there is no remarkable difference between GFU and NFU.

Finally, Tables 9 clarifies the single fleet management decision as cost parameters change.

Table 8: Performance indices

$|\mathcal{C}|$	GFU				NFU				q/dist			
	LRLS	LRHS	HRLS	HRHS	LRLS	LRHS	HRLS	HRHS	LRLS	LRHS	HRLS	HRHS
5	0.72	0.58	0.75	0.73	0.72	0.72	0.75	0.75	0.28	0.28	0.28	0.28
10	0.80	0.66	0.83	0.81	0.80	0.80	0.83	0.84	0.48	0.48	0.48	0.48
25	0.88	0.67	0.91	0.90	0.88	0.86	0.91	0.91	0.72	0.70	0.71	0.68
50	0.88	0.74	0.88	0.86	0.88	0.87	0.88	0.87	1.27	1.23	1.26	1.19
Avg	**0.82**	**0.66**	**0.84**	**0.83**	**0.82**	**0.81**	**0.84**	**0.84**	**0.69**	**0.68**	**0.68**	**0.66**

Column **#W** brings the average of rentals, **#S** the average number of cleanings, **#R** the average return, and **#(X+Y)** the average pickup plus delivery routes. For low rental and low cleaning cost (LRLS) we observe that five rentals, three cleanings, two returns and six routes are performed. When cleaning cost switched to high, it becomes more advantageous to keep the vehicles at the plants, instead of returning them. The number of returns is 86.6% smaller than before, and the number of cleanings is 55.6% smaller as well. When rental costs are high, the cleaning cost is less relevant and the average number of cleanings dropped 32% when cleaning costs changed from low to high. At the same time, we observe a decrease in the number of rentals, especially when cleaning costs are high, decreasing from 6.3 on LRHS to 5.0 on HRHS.

7.4 Results for the 2E-MDIRP

We have also compared the performance of our methods on instances proposed by Guimarães et al. [17] for the 2E-MDIRP. We assessed all 64 instances with three vehicles and solved them under the ML and the OU policies on the second echelon, making a total of 128 experiments for each algorithm.

The mathematical model proposed in Section 3 is flexible enough to handle all 2E-MDIRP features. To this end, it is sufficient set to zero all of fleet management costs, i.e., $\boldsymbol{f_s} = \boldsymbol{0}$ and $\boldsymbol{f_w} = \boldsymbol{0}$. We highlight that all valid inequalities (44)−(60) and the OU policy (43) apply to both problems.

Table 9: Average fleet usage

Rental Cost	$|\mathcal{C}|$	Low Cleaning Cost				High Cleaning Cost				Comparative Analysis			
		#W	#S	#R	#(X+Y)	#W	#S	#R	#(X+Y)	Δ% W	Δ% S	Δ% R	Δ% (X+Y)
Low	5	4.2	2.8	1.7	5.2	5.4	1.3	0.3	5.3	30.5	-51.6	-82.4	0.6
	10	5.7	3.0	2.0	6.7	7.1	1.3	0.3	6.8	23.2	-56.1	-87.5	0.8
	25	5.0	3.5	2.4	6.1	6.7	1.3	0.2	6.2	33.1	-63.7	-91.7	1.8
	50	5.1	2.7	1.6	6.2	6.1	1.3	0.3	6.2	20.3	-51.0	-85.0	0.9
Avg		**5.0**	**3.0**	**1.9**	**6.0**	**6.3**	**1.3**	**0.2**	**6.1**	**26.8**	**-55.6**	**-86.6**	**1.0**
High	5	4.0	2.8	1.8	5.1	4.2	2.1	1.0	5.2	5.3	-27.0	-43.5	1.9
	10	5.5	3.3	2.3	6.5	5.7	2.1	1.1	6.5	4.2	-34.4	-50.9	0.5
	25	4.8	3.7	2.6	5.9	4.9	2.2	1.2	5.9	1.0	-38.0	-52.9	-0.2
	50	5.0	2.8	1.8	6.1	5.1	2.0	0.9	6.1	1.7	-28.9	-42.7	0.0
Avg		**4.8**	**3.2**	**2.1**	**5.9**	**5.0**	**2.1**	**1.1**	**5.9**	**3.0**	**-32.1**	**-47.5**	**0.5**

Guimarães et al. [17] proposed an asymmetric formulation for the 2E-MDIRP, employing symmetry breaking constraints. Our formulation is subtly different, since we consider edge (i, j) only if $i < j$. This reformulation is useful to enable direct deliveries through a single-edge, which substantially reduces the number of decision variables, before the search tree is structured. In addition, we propose a new set of valid inequalities $(58)-(60)$. Therefore, we are able to analyze the performance of the B&C and H-B&C, derived from this new formulation, against the B&C proposed by [17].

Table 10 shows a comparison among exact algorithms for the ML policy. We highlight that our B&C was able to find a solution to all instances up to 25 customers, and 36 optimal solutions were proved, with seven new ones. Our B&C and H-B&C had an equivalent performance in this regard, but H-B&C obtained tighter bounds. Analogously, Table 11 presents the results to OU policy, with similar performance.

To handle large instances, Guimarães et al. [17] designed a matheuristic under the ALNS mechanism. Although we follow a similar way, our Math-ALNS embeds reformulated MIPs. The main innovation lies in the GIRP model, which has a neighborhood exploration

Table 10: Comparison between B&C from Guimarães et al. [17], new B&C and H-B&C to 2E-MDIRP, ML policy.

$\|\mathcal{C}\|$	$\|\mathcal{P}\|$	$\|\mathcal{F}\|$	B&C [17]				B&C				H-B&C			
			OPT	SF	$\overline{LB}$	$\overline{T(s)}$	OPT	SF	$\overline{LB}$	$\overline{T(s)}$	OPT	SF	$\overline{LB}$	$\overline{T(s)}$
5	1	1	4	4	3781.5	1.9	4	4	3781.5	1.2	4	4	3781.5	2.7
	2	2	4	4	3822.5	81.9	4	4	3822.6	52.7	4	4	3822.6	278.4
	2	3	4	4	3786.2	66.9	4	4	3786.2	57.4	4	4	3786.2	430.3
	3	2	4	4	3484.6	238.7	4	4	3484.6	478.3	4	4	3484.6	1461.6
10	1	1	4	4	4897.7	54.1	4	4	4897.7	20.5	4	4	4897.7	48.4
	2	2	2	4	4661.8	3716.2	2	4	4695.6	4326.9	2	4	4674.5	3734.9
	2	3	2	4	4702.7	3697.0	2	4	4707.4	3823.5	2	4	4683.8	3644.8
	3	2	2	4	4421.1	3812.5	2	4	4682.6	5286.7	2	4	4606.9	4120.3
25	1	1	2	4	5305.0	4314.5	4	4	5403.4	626.7	4	4	5403.4	838.7
	2	2	0	2	4538.7	7200.0	2	4	4998.0	3862.0	2	4	4951.8	4126.8
	2	3	1	2	4574.6	6920.0	2	4	4962.2	3854.6	2	4	5012.2	3868.1
	3	2	0	2	4399.8	7200.0	2	4	4971.2	5521.2	2	4	4899.8	5065.9
50	1	1	0	2	7365.1	7200.0	0	3	7468.1	7200.0	0	4	7454.8	7200.0
	2	2	0	0	6500.5	7200.0	0	1	6752.0	7200.0	0	4	6815.2	7200.0
	2	3	0	1	6693.7	7200.0	0	2	6810.2	7200.0	0	4	6729.9	7200.0
	3	2	0	0	6143.6	7200.0	0	1	6727.8	7200.0	0	4	6819.6	7200.0
Total			29	45			36	55			36	64		
Avg					4942.4	4131.5			5121.9	3544.5			5114.0	3526.3

Table 11: Comparison between B&C from Guimarães et al. [17], new B&C and H-B&C to 2E-MDIRP, OU policy.

$\mathcal{C}$	$\mathcal{P}$	$\mathcal{F}$	B&C [17]				B&C				H-B&C			
			OPT	SF	$\overline{LB}$	$\overline{T(s)}$	OPT	SF	$\overline{LB}$	$\overline{T(s)}$	OPT	SF	$\overline{LB}$	$\overline{T(s)}$
5	1	1	4	4	3998.2	2.1	4	4	3998.2	2.3	4	4	3998.2	5.3
	2	2	4	4	4085.4	431.6	4	4	4085.4	416.9	4	4	4085.4	219.6
	2	3	4	4	4065.7	320.4	4	4	4065.7	319.3	4	4	4065.7	189.6
	3	2	4	4	3733.4	1057.0	4	4	3733.3	1261.1	4	4	3733.5	778.9
10	1	1	4	4	5204.2	114.1	4	4	5204.2	33.1	4	4	5204.2	235.2
	2	2	2	4	4799.2	3700.4	2	4	4927.8	4460.6	2	4	4999.8	3746.3
	2	3	2	4	4811.2	3801.3	2	4	4933.4	3715.0	2	4	4920.8	3765.6
	3	2	2	4	4597.9	3647.5	2	4	4817.8	3660.8	2	4	4831.9	3666.6
25	1	1	1	3	5262.3	5663.3	3	4	5800.4	3462.7	2	4	5767.5	4207.6
	2	2	0	2	4664.8	7200.0	1	4	5452.7	6453.8	1	4	5317.2	6637.6
	2	3	0	2	4548.0	7200.0	0	4	5453.1	7200.0	0	4	5408.5	7200.0
	3	2	0	2	4344.7	7200.0	0	4	5320.9	7200.0	1	4	5297.3	6508.6
50	1	1	0	1	7773.6	7200.0	0	3	8016.2	7200.0	0	4	8044.7	7200.0
	2	2	0	0	7034.9	7200.0	0	1	7413.5	7200.0	0	4	7442.8	7200.0
	2	3	0	0	7178.6	7200.0	0	1	7358.2	7200.0	0	4	7359.7	7200.0
	3	2	0	0	6627.3	7200.0	0	2	7489.8	7200.0	0	4	7539.1	7200.0
Total			27	42			30	55			30	64		
Avg					5170.6	4321.1			5504.4	4186.6			5501.0	4122.6

strategy based on real transport costs (see Section 5.2.3). Table 12 compares our Math-ALNS and the matheuristic from [17]. Each row shows the average of sixteen instances, four instances (absH3low, absH6low, absH3high, and absH6high) on each supply chain structure (one supplier one plant, two suppliers two plants, two suppliers three plants, and three suppliers two plants). For each inventory policy, columns $\overline{Z}$ [17] and $\overline{T(s)}$ [17] show the average results for total cost and processing time for the matheuristic of Guimarães et al. [17], while our Math-ALNS is presented in $\overline{Z}$ and $\overline{T(s)}$. The gap between the average total cost is obtained by $\left(\frac{\overline{z} - \overline{z}[17]}{\overline{z}[17]} \right) \times 100$. As Math-ALNS has been adapted from 2E-IRPFM, it preserves the full set of fleet management constraints, requiring more time to solve the associated MIPs. Although the overall relative improvement on $\overline{Z}$ suggests an equivalent performance, Math-ALNS was superior especially on large instances. In general, there was improvement in 46 (23 for ML and 23 for OU) out of the 128 instances, with a tie in 81 and only one worse result.

Table 12: Comparison between the matheuristic of Guimarães et al. [17] and our Math-ALNS

$|\mathcal{C}|$	ML					OU				
	$\overline{Z}$ [17]	$\overline{T(s)}$ [17]	$\overline{Z}$	$\overline{T(s)}$	GAP % $(\overline{Z})$	$\overline{Z}$ [17]	$\overline{T(s)}$ [17]	$\overline{Z}$	$\overline{T(s)}$	GAP % $(\overline{Z})$
5	3718.73	280.77	3718.73	1160.18	0.00	3970.70	313.41	3970.70	874.16	0.00
10	5131.25	874.21	5121.70	2075.28	-0.18	5475.60	1320.02	5474.37	1743.51	-0.02
25	5846.60	902.26	5824.61	2633.23	-0.36	6429.30	1029.30	6350.48	2457.46	-1.19
50	8167.46	1246.01	8068.61	3942.77	-1.19	9105.79	2335.23	9076.98	4486.34	-0.32
Avg	5716.0	825.8	5683.4	2452.9	-0.4	6245.3	1249.5	6218.1	2390.4	-0.4

8 Conclusions

In this study, we have introduced the 2E-IRPFM, a new variant of IRP which incorporates fleet planning in a two-echelon logistics system. This problem has a complex many-to-many supply chain structure, where the plants are in the middle layer and must control the inventory and routing decisions regarding input pickup and final product delivery, while

managing tactical and operational fleet decisions. We have proposed a MILP formulation and a B&C algorithm to solve the problem, taking into account different inventory policies. We have also designed a matheuristic algorithm and an exact hybrid parallel approach to efficiently solve large instances. Validation experiments, performed on 2E-MDIRP instances from the literature, showed that our algorithms are very effective, yielding BKSs for the whole set of problems evaluated. We have shown that rentals, cleanings and vehicle returns represent a significant portion of logistics costs. Moreover, a more complex logistics system proved more resilient regarding changing in rental and cleaning costs, leading to greater efficiency in logistics operations.

References

[1] D. Aksen, O. Kaya, F. Sibel Salman, and Ö. Tüncel. An adaptive large neighborhood search algorithm for a selective and periodic inventory routing problem. *European Journal of Operational Research*, 239(2):413–426, 2014.

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

[3] C. Archetti and M. G. Speranza. A survey on matheuristics for routing problems. *EURO Journal on Computing Optimization*, 2:223–246, 2014.

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

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

[6] W. J. Bell, L. M. Dalberto, M. L. Fisher, A. J. Greenfield, R. Jaikumar, P. Kedia, R. G. Mack, and P. J. Prutzman. Improving the distribution of industrial gases with

an on-line computerized routing and scheduling optimizer. *Interfaces*, 13(6):4–23, 1983.

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

[8] M. Bielli, A. Bielli, and R. Rossi. Trends in models and algorithms for fleet management. *Procedia - Social and Behavioral Sciences*, 20:4–18, 2011.

[9] T. Chabot, F. Bouchard, A. Legault-Michaud, J. Renaud, and L. C. Coelho. Service level, cost and environmental optimization of collaborative transportation. *Transportation Research Part E: Logistics and Transportation Review*, 110:1–14, 2018.

[10] L. C. Coelho and G. Laporte. Improved solutions for inventory-routing problems through valid inequalities and input ordering. *International Journal of Production Economics*, 155:391–397, 2014.

[11] L. C. Coelho, J.-F. Cordeau, and G. Laporte. Consistency in multi-vehicle inventory-routing. *Transportation Research Part C: Emerging Technologies*, 24:270–287, 2012.

[12] L. C. Coelho, J.-F. Cordeau, and G. Laporte. The inventory-routing problem with transshipment. *Computers & Operations Research*, 39(11):2537–2548, 2012.

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

[14] G. Desaulniers, J. G. Rakke, and L. C. Coelho. A branch-price-and-cut algorithm for the inventory-routing problem. *Transportation Science*, 1655:1–17, 2015.

[15] K. Fagerholt and H. Lindstad. Turborouter: An interactive optimisation-based decision support system for ship routing and scheduling. *Maritime Economics & Logistics*, 9(3):214–233, 2007.

[16] M. Fischetti and A. Lodi. Local branching. *Mathematical Programming*, 98(1-3): 23–47, 2003.

[17] T. A. Guimarães, L. C. Coelho, C. M. Schenekemberg, and C. T. Scarpin. The two-echelon multi-depot inventory-routing problem. *Computers & Operations Research*, 101:220–233, 2019.

[18] L. M. Hvattum, K. Fagerholt, and V. A. Armentano. Tank allocation problems in maritime bulk shipping. *Computers & Operations Research*, 36(11):3051–3060, 2009.

[19] R. Lahyani, L. C. Coelho, M. Khemakhem, G. Laporte, and F. Semet. A multi-compartment vehicle routing problem arising in the collection of olive oil in Tunisia. *Omega*, 51:1–10, 2015.

[20] W. Lefever. *Stochastic and robust optimization algorithms for the inventory-routing problem and its extensions*. PhD thesis, Ghent University, 2018.

[21] W. Lefever, E.-H. Aghezzaf, K. Hadj-Hamou, and B. Penz. Analysis of an improved branch-and-cut formulation for the inventory-routing problem with transshipment. *Computers & Operations Research*, 98:137–148, 2018.

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

[23] J. Oppen, A. Løkketangen, and J. Desrosiers. Solving a rich vehicle routing and inventory problem using column generation. *Computers & Operations Research*, 37 (7):1308–1317, 2010.

[24] M. Padberg and G. Rinaldi. A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM Review*, 33(1):60–100, 1991.

[25] D. Pisinger and S. Ropke. A general heuristic for vehicle routing problems. *Computers & Operations Research*, 34(8):2403–2435, 2007.

[26] Windle R. Dresner M. Rabinovich, E. and T. Corsi. Outsourcing of integrated logistics functions. *International Journal of Physical Distribution & Logistics Management*, 29(6):353–374, 1999.

[27] K. Shintani, R. Konings, and A. Imai. The impact of foldable containers on container fleet management costs in hinterland transport. *Transportation Research Part E: Logistics and Transportation Review*, 46(5):750–763, 2010.

[28] J.-H. Song and M. W. P. Savelsbergh. Performance measurement for inventory routing. *Transportation Science*, 41(1):44–54, 2007.

[29] Y. Yuliang and M. Dresner. The inventory value of information sharing, continuous replenishment, and vendor-managed inventory. *Transportation Research Part E: Logistics and Transportation Review*, 44(3):361–378, 2008.

[30] Q.-H. Zhao, S. Chen, S. Leung, and Lai K.-K. Integration of inventory and transportation decisions in a logistics system. *Transportation Research Part E: Logistics and Transportation Review*, 46(6):913–925, 2010.