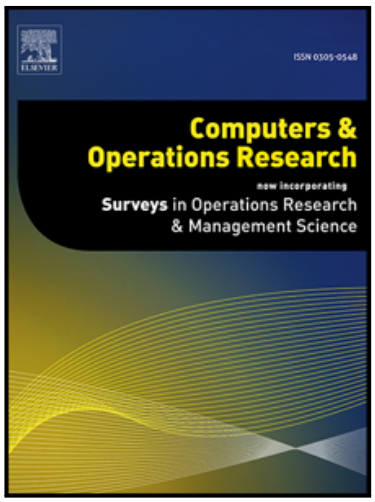

The two-echelon multi-depot inventory-routing problem

Thiago A. Guimarães, Leandro C. Coelho,
Cleder M. Schenekemberg, Cassius T. Scarpin

PII: S0305-0548(18)30214-4
DOI: https://doi.org/10.1016/j.cor.2018.07.024
Reference: CAOR 4533

To appear in: *Computers and Operations Research*

Received date: 12 February 2018
Revised date: 22 April 2018
Accepted date: 27 July 2018

Please cite this article as: Thiago A. Guimarães, Leandro C. Coelho, Cleder M. Schenekemberg, Cassius T. Scarpin, The two-echelon multi-depot inventory-routing problem, *Computers and Operations Research* (2018), doi: https://doi.org/10.1016/j.cor.2018.07.024

Highlights

- We introduce a new inventory-routing problem with three echelons
- We include the replenishment decisions from plants in the decision making process
- We study many inventory policies, model and solve the problem exactly
- We develop a hybrid algorithm combining neighborhood search and a mathematical model
- We show that replenishment from plants represent a significant portion of total costs

The two-echelon multi-depot inventory-routing problem

Thiago A. Guimarães[a,*], Leandro C. Coelho[b], Cleder M. Schenekemberg[a], Cassius T. Scarpin[a]

[a] *Research Group of Technology Applied to Optimization (GTAO), Federal University of Paraná (UFPR), Curitiba, Brazil*
[b] *Canada Research Chair in Integrated Logistics and CIRRELT, Université Laval, Québec, Canada*

Abstract

This paper studies the application of the Vendor-Managed Inventory paradigm, when a vendor manages their own inventory in addition to those of its customers. We extend this concept by applying it to a two-echelon (2E) supply chain, in which the middle layer is responsible for managing pickups of inputs from suppliers and deliveries of final product to its customers. Inspired by a real case, we introduce 2E Multi-Depot Inventory-Routing Problem (2E-MDIRP). Different inventory policies are considered for managing the inventory of input and final products. We propose a mathematical formulation capable of handling all decisions of the system, and design a branch-and-cut algorithm to solve it. Moreover, we propose and implement a rich matheuristic algorithm to solve the problem efficiently even for very large instances. A flexible metaheuristic phase handles vehicle routes, while input pickups, product deliveries and routing improvements are performed by solving a subproblem exactly. We perform extensive computational experiments in order to evaluate the performance of our method, both by comparing the two algorithms, the effectiveness of the different inventory policies, the cost structure of the solutions, and the performance of different parts of the proposed heuristic algorithm. The results show that a more strict inventory policy leads to higher costs but fewer vehicle routes, which makes the overall problem easier to be solved by the approximated algorithm.

Keywords: Two-echelon supply chain, multi-depot inventory-routing, vendor managed inventory, optimization, matheuristic, branch-and-cut.

1. Introduction

Vendor-Managed Inventory (VMI) is widely recognized as a successful managerial practice, reducing costs and improving the service level of supply chains [23]. Under a VMI system, the supplier controls

*Corresponding author
Email addresses: thiagoandre@ufpr.br (Thiago A. Guimarães), leandro.coelho@cirrelt.ca (Leandro C. Coelho), cledercms@hotmail.com (Cleder M. Schenekemberg), cassiusts@ufpr.br (Cassius T. Scarpin)

the inventory of customers, deciding when, how much to deliver and how to combine deliveries into vehicle routes over a planning horizon. Customers need not spend resources to control their inventories, while suppliers can reduce distribution costs through better coordination and synchronization. From an operational viewpoint, a VMI system requires solving a complex combinatorial optimization problem known as the Inventory-Routing Problem (IRP), which integrates into the same framework inventory management and multi-period routing decisions [18]. A number of applications of IRPs is presented in Andersson et al. [3].

Despite the large number of variants [18], some considerations are common in most IRP studies. The most traditional case in road-based supply chains involves a single plant serving a set of customers [5, 8], known as one-to-many, while cases in which multiple plants serve some customers (many-to-many) are more frequent in maritime applications [2, 24, 30]. In addition to being scarce in road-based problems, the many-to-many structure is simplified in this context, so that a single plant serves the same subset of customers in each period, reducing a many-to-many setting into several one-to-many structures [31]. Inventory considerations typically assume that a given quantity is available at the beginning of each period at the plant and at the customers [6, 16, 17].

A more complex supply chain structure arises when more echelons are considered. In many industries, major players having several production facilities can control both the upstream and the downstream flows, thus allowing them to manage a rich two echelon (2E) supply chain. Moreover, if this decision maker buys its input from several suppliers and sells its product to many customers, the supply chain contains many possible product flows, from different sources to different destinations in all echelons.

An example of such an integrated system arises in some South American countries such as Argentina, Brazil, and Colombia, where ethanol is produced from sugar cane and must be used, according to local regulation, as an additive in the gasoline blend [33]. Energy companies acquire this input from ethanol plants, transporting it to their production plants to mix with refined gasoline, which is then sold to gas stations. Both the ethanol pickup and gasoline delivery are carried out by the fleet of vehicles housed at the plants. This example shows that the implementation of a VMI system in large companies requires not only integration in the delivery side, but also on the sourcing side of production. Moreover, this example highlights a many-to-many structure over a 2E supply chain, with ethanol producers−plants as the first echelon and plants−gas stations as the second one, all controlled by the middle layer.

To the best of our knowledge, few IRP studies consider 2E networks, and all of them consider only a single supplier, who is the decision maker positioned at the first level of the supply chain. In our case, the plants, at the second level, are in charge of all decisions. Our setting is much more complex and in

the following we review the relevant literature. In Chan and Simchi-Levi [11], a system with a supplier, multiple warehouses and a set of customers is considered. The decision-making process is centralized at the supplier, which can use intermediate deposits to consolidate deliveries to customers. The paper of Li et al. [25] addresses a similar system, but with a single intermediate facility. Deliveries from the supplier to the customers can be direct or pass through this intermediate location. The structure of the problem studied by Zhao et al. [34] deals with a supply chain identical to the previous one, except that no direct delivery is allowed. More recently Rahim et al. [28] consider a VMI system containing one supplier, one warehouse and multiple customers. The paper compares direct deliveries with joint deliveries. The decision-making process is also centered on the supplier, who must ship the products to the warehouse, and then, schedule customer deliveries. For more details of 2E routing problems, we refer to Cuda et al. [21].

Usually, the decision maker is free to determine the quantity of product delivered to each customer, the only constraints pertaining to inventory and vehicle capacities. These features define the maximum level (ML) inventory policy, this being the dominant and less constrained one [18]. Several heuristics [6, 20], exact algorithms [5, 13, 22] and matheuristics [16, 17] have been proposed to solve IRPs under the ML policy. Based on target-levels, a tactical policy in the context of VMI systems is proposed by [15], under which when the supplier visits a customer, the quantity delivered is such that the final inventory will always be at the same customer-dependent target-level. In additions, others inventory policies emerged on dynamic and stochastic IRP, like big orders first, lowest storage firs, and equal quantity discount [29]. A very common alternative, the order-up-to level (OU) policy, requires the plant to deliver enough to the customers so that their inventory level reaches the maximum capacity whenever a delivery occurs. Despite this being a common feature for the one-to-many structure [7, 12], we did not find studies that applied the OU policy to the multi-depot IRP (MDIRP).

The problem addressed in this paper covers these gaps in the literature as we propose, model and solve an MDIRP for a 2E logistics system, which we call 2E-MDIRP. The study is motivated by the real case of gasoline distribution in urban environments under a VMI system. In this case, a set of plants controls the inventory of gasoline of a set of gas stations, in addition to its inventory of ethanol (input) picked up from its suppliers. The aim is to minimize routing (pickups of the input and deliveries of gasoline blend) and inventories (input and final product) costs, avoiding stock outs over a finite planning horizon.

The scientific contributions of this work are:

1. we are the first to consider an IRP over 2E with multiple nodes per echelon; the decision maker lies in the middle layer, requiring it to control both the upstream supply of inputs and the downstream

delivery of final products;

2. we describe, model, solve, and compare all four combinations of ML and OU inventory policies, for the input and the final product. We analyze the performance of these configurations in each component of the total cost of the VMI system;
3. we propose a hybrid heuristic algorithm coupled with the exact solution of an integer programming model. We also design and implement a branch-and-cut (B&C) algorithm to evaluate the performance of our heuristic;
4. we perform extensive computational experiments assessing the combination of inventory policies and the algorithm on instances inspired by real-life information. We derive many insights for this new problem.

The remainder of the paper is organized as follows. In Section 2 we formally describe the 2E-MDIRP and a mixed-integer linear programming (MILP) formulation is proposed in Section 3, where we present sets of existing and new valid inequalities, followed by the description of a B&C algorithm in Section 4. In Section 5 we describe the hybrid matheuristic algorithm we propose to solve the 2E-MDIRP. Section 6 presents the results of extensive computational experiments performed to assess the quality of the algorithms and to derive business insights for this rich and new problem. Conclusions are presented in Section 7.

2. Problem description

The 2E-MDIRP is defined over a graph $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ represents the vertex set, formed by 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. A non-negative cost c_{uv} is associated with each edge $(u, v) \in \mathcal{E}$. Each plant $j \in \mathcal{P}$ houses a set $\mathcal{K}$ of homogeneous vehicles with capacity Q. Each plant could house different sets of vehicles $\mathcal{K}_j$, but for simplicity we assume they are all equal and use simply $\mathcal{K}$.

The planning horizon is defined over a set $\mathcal{T}$ of periods, and in each period t, a plant can use a vehicle to pickup a certain amount of input from a supplier. Each unit of product delivered to customers needs a certain quantity φ of input in its blend, and the demand d_l^t of each customer l in each period is known a priori. Each plant j has a maximum inventory capacity U_j for the input, and its level cannot be lower than L_j; one unit of the input incurs an inventory holding cost h_j per period. Likewise, U_l and L_l denote the maximum and minimum inventory levels at customer l, with each unit incurring a holding cost h_l per period. The total availability of input at all suppliers is not constrained. However, each supplier i

disposes of a total input of Φ_i units for the whole planning horizon $\mathcal{T}$. At $t = 0$, 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 the input is always collected at the beginning of each period t, when a pickup is performed and can be used in the final product in the same period. The objective of the 2E-MDIRP is to minimize the total inventory and transportation cost, determining for each plant:

- when, how much and from which supplier to perform an input pickup;
- when and how much to deliver to a customer;
- how to combine customers into vehicle routes.

Split deliveries at the customers are not allowed meaning that a customer can be visited by at most one vehicle per period. Likewise, split pickups are also not allowed given that plant can pickup input from one supplier using one vehicle per period. In both pickup and delivery activities vehicles are capacitated, and all vehicles departing from a plant must return to the same plant by the end of period. Finally, the same vehicle can perform a pickup and a delivery in the same period, as long as it returns to the original plant after collecting input from the supplier. This is due to the required mixing of the input to the product, and also because a cleaning activity must be performed in order to switch products within the vehicle.

3. Mathematical formulation

We now present a mathematical formulation to the 2E-MDIRP. We introduce a subset $\mathcal{V}' \subseteq \mathcal{V}$, with $\mathcal{V}' = \mathcal{P} \cup \mathcal{C}$, and let $\mathcal{E} = \{(i, j) : i, j \in \mathcal{V}' \wedge i, j \oplus \mathcal{P}\}$.

Each plant j must determine the quantity of product q_{jl}^{kt} delivered to customer l and the total amount of input r_{ij}^{kt} pickup from supplier i, using vehicle k in period t. At the end of each period, the inventory level of product at customer l is given by I_l^t, while the inventory level of input at plant j is I_j^t. The set of binary variables of our model is:

- $X_{ij}^{kt} = 1$ if vehicle k from plant j picks up input from supplier i in period t, 0 otherwise;
- $Y_{jl}^{kt} = 1$ if vehicle k from plant j serves customer l in period t, 0 otherwise;
- $Z_j^{kt} = 1$ if vehicle k from plant j performs a delivery in period t, 0 otherwise;

- $y_{uv}^{kjt} = 1$ if vehicle k from plant j travels directly from vertex u to vertex v ($(u, v) \in \mathcal{E}$) in period t, 0 otherwise;

Figure 1 illustrates a 2E-MDIRP instance with two suppliers, four plants and nine customers, with operational decisions given for a single period. Regarding to binary variables for a given t, exemplifying from plant $j = 1$ and considering a single vehicle $k = 1$, we have: $X_{21}^{1t} = 1$ (vehicle $k = 1$ performs a pickup from supplier $i = 2$), $Z_1^{1t} = 1$ (vehicle $k = 1$ performs a delivery in period t), $Y_{11}^{1t} = 1, Y_{12}^{1t} = 1, Y_{13}^{1t} = 1$ (customers $l = 1$, $l = 2$ and $l = 3$ are served by vehicle $k = 1$ from plant $j = 1$ in period t), and finally $y_{15}^{11t} = 1$, $y_{56}^{11t} = 1$, $y_{67}^{11t} = 1$ for routing binary variables, setting customer $l = 1$ as the node five after $n = 4$ plants.

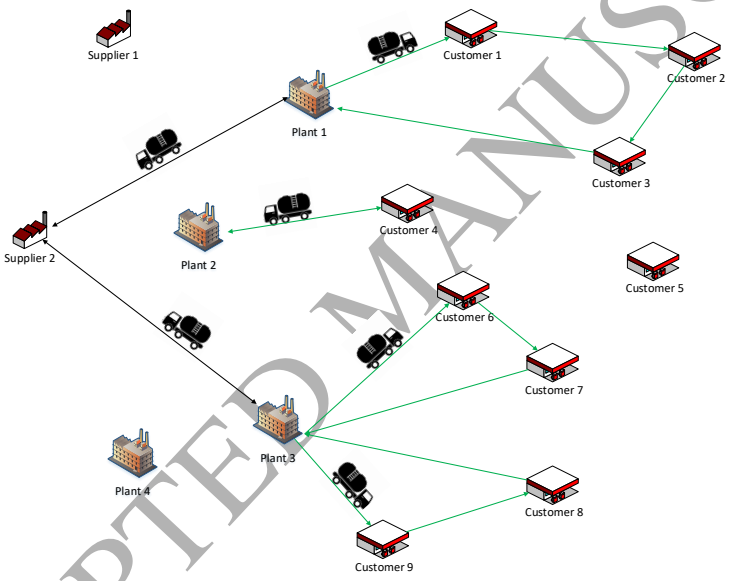

Figure 1: Graphical representation of the 2E-MDIRP

The 2E-MDIRP is formulated by (1)−(22).

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

subject to

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

$$I_j^t = I_j^{t-1} + \sum_{i\in\mathcal{F}} \sum_{k\in\mathcal{K}} r_{ij}^{kt} - \sum_{l\in\mathcal{C}} \sum_{k\in\mathcal{K}} \varphi q_{jl}^{kt} \qquad j \in \mathcal{P}, t \in \mathcal{T}\setminus\{0\} \tag{3}$$

$$L_j \leq I_j^t \leq U_j \qquad j \in \mathcal{P}, t \in \mathcal{T} \tag{4}$$

$$I_l^t = I_l^{t-1} + \sum_{j \in \mathcal{F}} \sum_{k \in \mathcal{K}} q_{jl}^{kt} - d_l^t \qquad l \in \mathcal{C}, t \in \mathcal{T} \setminus \{0\} \tag{5}$$

$$L_l \leq I_l^t \leq U_l \qquad l \in \mathcal{C}, t \in \mathcal{T} \tag{6}$$

$$\sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{F}} X_{ij}^{kt} \leq 1 \qquad j \in \mathcal{P}, t \in \mathcal{T} \tag{7}$$

$$\sum_{i \in \mathcal{F}} \sum_{k \in \mathcal{K}} r_{ij}^{kt} \leq U_j - I_j^{t-1} \qquad j \in \mathcal{P}, t \in \mathcal{T} \setminus \{0\} \tag{8}$$

$$r_{ij}^{kt} \leq Q X_{ij}^{kt} \qquad j \in \mathcal{P}, i \in \mathcal{F}, k \in \mathcal{K}, t \in \mathcal{T} \tag{9}$$

$$\sum_{j \in \mathcal{P}} \sum_{k \in \mathcal{K}} q_{jl}^{kt} \leq U_l - I_l^{t-1} \qquad l \in \mathcal{C}, t \in \mathcal{T} \setminus \{0\} \tag{10}$$

$$q_{jl}^{kt} \leq U_l Y_{jl}^{kt} \qquad l \in \mathcal{C}, j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T} \tag{11}$$

$$\sum_{l \in \mathcal{C}} q_{jl}^{kt} \leq Q Z_j^{kt} \qquad j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T} \tag{12}$$

$$\sum_{j \in \mathcal{P}} \sum_{k \in \mathcal{K}} Y_{jl}^{kt} \leq 1 \qquad l \in \mathcal{C}, t \in \mathcal{T} \tag{13}$$

$$\sum_{l \in \mathcal{C}} y_{ul}^{kut} \leq 1 \qquad u \in \mathcal{P}, l \in \mathcal{C}, k \in \mathcal{K}, t \in \mathcal{T} \tag{14}$$

$$\sum_{u \in \mathcal{V}'} y_{ul}^{kjt} + \sum_{u \in \mathcal{V}'} y_{lu}^{kjt} = 2 Y_{ij}^{kt} \qquad i \in \mathcal{F}, l \in \mathcal{C}, j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T} \tag{15}$$

$$\sum_{l.u \in S} y_{lu}^{kjt} \leq \sum_{l \in S} Y_{jl}^{kt} - Y_{ju}^{kt} \qquad S \subseteq \mathcal{C}, |S| \geq 2, u \in S, j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T} \tag{16}$$

$$r_{ij}^{kt} \geq 0 \qquad j \in \mathcal{P}, i \in \mathcal{F}, k \in \mathcal{K}, t \in \mathcal{T} \tag{17}$$

$$q_{jl}^{kt} \geq 0 \qquad l \in \mathcal{C}, j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T} \tag{18}$$

$$X_{ij}^{kt} \in \{0,1\} \qquad j \in \mathcal{P}, i \in \mathcal{F}, k \in \mathcal{K}, t \in \mathcal{T} \tag{19}$$

$$Y_{jl}^{kt} \in \{0,1\} \qquad l \in \mathcal{C}, j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T} \tag{20}$$

$$Z_j^{kt} \in \{0,1\} \qquad j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T} \tag{21}$$

$$y_{uv}^{kjt} \in \{0,1\} \qquad u, v \in \mathcal{V}', j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T}. \tag{22}$$

The objective function (1) minimizes the total cost, given by four parts: inventory of input at the plants, inventory of products at the customers, transportation pickup and transportation delivery costs. Constraints (2) incorporate the supply contract between suppliers and plants. Constraints (3) balance the inventory flow at plants (note that the output flow is calculated with respect to the input quantity) and constraints (4) ensure that input inventory level at plants is bounded. Constraints (5) balance the inventory flow at customers while constraints (6) impose inventory bounds. Constraints (7) ensure that at most one pickup per plant is performed in each period. Constraints (8) model the ML policy for the input while constraints (9) ensure that the total amount of input picked up by the plant does not exceed the vehicle capacity. The product ML policy is formulated by constraints (10). Constraints (11) link

delivery and assignment variables. Constraints (12) ensure that the vehicle capacity is not exceeded in a delivery, while constraints (13) prevent split deliveries to customers. Constraints (14)–(16) impose linking and routing conditions. In particular, the degree of the vertices is controlled by constraints (15), and subtours elimination is enforced via constraints (16). Finally, constraints (17)–(22) define the domain of the variables.

Note that this model imposes the ML policy for the inventory management of both input and final product. In what follows we model the OU policy for these two commodities.

Under an OU policy, whenever a plant performs an input pickup, the total quantity loaded must fill its input inventory capacity. This condition links the *when* to the *how much* decisions regarding input inventory control. The input OU policy is guaranteed by adding constraints (23) to the model (1)–(22).

$$\sum_{i\in\mathcal{P}}\sum_{k\in\mathcal{K}} r_{ij}^{kt} \geq U_j \sum_{k\in\mathcal{K}}\sum_{i\in\mathcal{F}} X_{ij}^{kt} - I_j^{t-1} \qquad j\in\mathcal{P}, t\in\mathcal{T}\setminus\{0\}\,. \tag{23}$$

Equivalently, the OU policy for the final product at the customers is imposed by constraints (24).

$$\sum_{j\in\mathcal{P}}\sum_{k\in\mathcal{K}} q_{jl}^{kt} \geq U_l \sum_{j\in\mathcal{F}}\sum_{k\in\mathcal{K}} Y_{jl}^{kt} - I_l^{t-1} \qquad l\in\mathcal{C}, t\in\mathcal{T}\setminus\{0\}\,. \tag{24}$$

We also present valid inequalities, used to strengthen the formulation and improve the quality of root node lower bound as well. Several classes of valid inequalities for the single-vehicle IRP have been introduced by Archetti et al. [5], where some of them were extended to multi-vehicle IRP by Coelho and Laporte [14]. We also include the valid inequalities of the multi-depot multi-vehicle IRP, proposed by Bertazzi et al. [10]. These are described next.

$$y_{jl}^{kjt} + y_{lj}^{kjt} \leq 2Y_{jl}^{kt} \qquad l\in\mathcal{C}, j\in\mathcal{P}, k\in\mathcal{K},\ t\in\mathcal{T} \tag{25}$$

$$y_{lu}^{kjt} \leq Y_{jl}^{kt} \qquad l\in\mathcal{C}, u\in\mathcal{V}', j\in\mathcal{P}, k\in\mathcal{K},\ t\in\mathcal{T} \tag{26}$$

$$Y_{jl}^{kt} \leq Z_j^{kt} \qquad l\in\mathcal{C}, j\in\mathcal{P}, k\in\mathcal{K}, t\in\mathcal{T}. \tag{27}$$

Constraints (25)–(27) are known as logical inequalities. Precisely, inequalities (25) impose that if customer l is the predecessor or the successor of plant j in the route performed by vehicle k in period t, then customer l must be served by the same vehicle departing from plant j in period t. A similar situation is enforced by inequalities (26), where a customer is preceded or succeeded by another customer. Inequalities (27) ensure that customer l is served by vehicle k starting from plant j in period t, only if this vehicle performs some product delivery.

$$Z_j^{kt} \leq Z_j^{k-1.t} \qquad j \in \mathcal{P}, k \in \mathcal{K}\backslash\{1\}, t \in \mathcal{T} \tag{28}$$

$$Y_{jl}^{kt} \leq \sum_{\substack{u \in \mathcal{C} \\ u<l}} Y_{ju}^{k-1.t} \qquad l \in \mathcal{C}, j \in \mathcal{P}, k \in \mathcal{K}\backslash\{1\}, t \in \mathcal{T}. \tag{29}$$

Inequalities (28) break symmetry by ensuring that vehicle k cannot leave plant j if vehicle $k-1$ is not used. The same priority rule is extended to the customer vertices by (29): if customer l is served in period t by vehicle k starting from plant j, then vehicle $k-1$ departing from the same plant j must serve a customer with an index smaller than l in the same period.

$$\sum_{j \in \mathcal{P}} \sum_{t=t_1}^{t_2} Y_{jl}^{kt} \geq \left\lceil \frac{\sum_{t=t_1}^{t_2} d_l^t - U_l}{min\{Q.U_l\}} \right\rceil \qquad l \in \mathcal{C}, k \in \mathcal{K}, t_1, t_2 \in \mathcal{T}, t_2 > t_1. \tag{30}$$

Finally, we adapt inequalities (30) proposed by Coelho and Laporte [14] for the multi-depot multi-vehicle IRP. In this case, if the demand between $[t_1, t_2]$ is higher than the inventory capacity at customer l or the vehicle capacity, the right-hand side defines the minimum number of deliveries to avoid stock-out at customer l. Notably, these inequalities are useful to identify the smallest range $[t_1, t_2]$ for which customer l must be served.

Based on the structure of the 2E-MDIRP, we introduce the following new inequalities.

$$\sum_{i \in \mathcal{F}} \sum_{k \in \mathcal{K}} \sum_{t=t_1}^{t_2} X_{ij}^{kt} \geq \frac{\sum_{l \in \mathcal{C}} \sum_{k \in \mathcal{K}} \sum_{t=t_1}^{t_2} (\varphi) q_{jl}^{kt} - I_j^{t_1-1}}{min\{Q, U_j\}} \qquad j \in \mathcal{P}, t_1, t_2 \in \mathcal{T}\backslash\{0\}, t_2 > t_1 \tag{31}$$

$$\sum_{i \in \mathcal{F}} X_{ij}^{kt} \leq \sum_{l \in \mathcal{C}} Y_{jl}^{k,t} \qquad j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T}. \tag{32}$$

Inequalities (31) are similar to (30), being useful to identify the smallest range $[t_1.t_2]$ for which plant j must perform an input pickup. Inequalities (32) impose the condition that if vehicle k is used in an input pickup route from plant j, this same vehicle must be used to serve a set of customers in a final product delivery route, before any additional vehicle is required. This is valid because input can be mixed to the product in the same period and whenever a plant picks up some amount of input from a supplier, this same plant must serve at least one customer in the same period, otherwise it will incur an unnecessary inventory cost.

4. Branch-and-cut algorithm

Due to the combinatorial number of subtour elimination constraints (16), the model presented in Section 3 can only be completely described and all variables and constraints can be explicitly generated only for small size instances. Under these conditions, a branch-and-bound algorithm can be employed to solve the 2E-MDIRP. For more realistic instances, constraints (16) must be dynamically generated and applied along the optimization process.

In this sense, we designed a B&C algorithm to insert only the constraints that are violated during optimization. Initially, an incomplete model is generated, excluding constraints (16) and adding valid inequalities (25)−(32). When a feasible solution that satisfies all constraints of the incomplete model is found, each vehicle route in each period is obtained and analyzed. If no subtour exists, the solution is then accepted as incumbent to the complete model (2)–(22) and the process of cost optimization continues. Otherwise, the customers belonging to a subtour are identified, and the associated constraints (16) are generated and added to the model. This process continues until no route contains a subtour. We provide the pseudocode of this B&C scheme in Algorithm 1.

Algorithm 1 Pseudocode of the proposed B&C algorithm

```
At the root node of the search tree, generate (2)–(15), (17)–(22) and all valid inequalities.
Subproblem solution, Solve the LP relaxation of the node.
Termination check:
if there are no more nodes to evaluate then
    Stop.
else
    Select one node from the B&C tree.
end if
while the solution of the current LP relaxation contains subtours do
    Add violated subtour elimination constraints [26].
    Subproblem solution. Solve the LP relaxation of the node.
end while
if the solution of the current LP relaxation is integer then
    Go to the termination check.
else
    Branching: branch on one of the fractional variables.
    Go to the termination check.
end if
```

5. Heuristic algorithm

The 2E-MDIRP is $\mathcal{NP}$-hard, since its generalizes the capacitated vehicle routing problem (CVRP), restricting the use of exact techniques to small size instances, without the processing time increasing dramatically to obtain optimal solutions. Alternatively, large size problems can be solved heuristically in a reasonable time, without compromising the quality of the solutions.

We propose a hybrid heuristic approach capable of solving the 2E-MDIRP under any combined inventory policy for input and the final product. Our method combines mathematical programming techniques with an adaptive large neighborhood search (ALNS) procedure, in a framework known as a matheuristic. The concept of ALNS was introduced by Pisinger and Ropke [27] in the context of the CVRP and has since been successfully applied to several related inventory-routing problems [1, 19, 32]. Bertazzi et al. [9] solve a stochastic IRP with procurement, combining a rollout algorithm and mixed-integer linear programming models. In Coelho et al. [16], the authors solve a multi-vehicle IRP with consistency, comparing both ML and OU policies in a two-echelon system with a single depot. The authors proposed an ALNS scheme which operates on the vehicle routes, while the quantities delivered are determined via mixed-integer programming (MIP) models. The same idea was employed by Coelho et al. [17] in an IRP with transshipment. To see more applications of matheuristics on routing problems, we refer to Archetti and Speranza [4].

Our ALNS is responsible for handling the delivery vehicle routes, while the quantities and the pickup of input are defined in a MIP subproblem (see Section 5.2). The search procedure is divided into Δ segments. At each iteration, some customers are either removed or inserted from their current route by destroy and repair operators. Each operator i contains three information:

1. ω_i: the weight, whose value depends on past performance;
2. π_i: the score, which measures the impact on the solution when the operator is chosen. 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 incumbent, and by σ_3 if the solution is worse but accepted;
3. ς_{ij}: the number of times it has been chosen in the last segment j, with $j \in \Delta$.

Initially, all weighs are equal to one and all scores are equal to zero, and a roulette-wheel controls the choice of operators. Given h operators, the probability of using an operator i is $\omega_i / \sum_{j=1}^{h} \omega_j$. After each iteration, a reaction factor $\eta \in [0, 1]$ is applied to balance the present and the past operator performance,

where the weights are updated according to rule (33). After that, all scores are reset to zero.

$$\omega_i := \begin{cases} \omega_i & \text{if } \varsigma_{ij} = 0 \\ (1-\eta)\omega_i + \eta\pi_i/\varsigma_{ij} & \text{if } \varsigma_{ij} \neq 0. \end{cases} \tag{33}$$

As in Pisinger and Ropke [27], we consider an acceptance criterion based on simulated annealing. Given a solution s, a neighbor solution s' is accepted if $z(s) > z(s')$ or if $e^{(z(s)-z(s'))/\tau} > \Theta$, where $z(\cdot)$ is the solution cost, $\tau > 0$ is the current temperature and Θ is a random value from the interval $[0, 1]$. The initial temperature is τ_{start}, decreasing at each iteration by a cooling rate factor ϕ.

Typically, a neighborhood is defined by the operation of removing and reinserting customers in the route set [27]. Eventually, the customers removed may not be reinserted, and yet a feasible solution can be found, provided there is some mechanism that can recover the feasibility. As the model proposed in Section 5.2 has this mechanism, we chose not to use the traditional removal and reinsertion framework. Thus, only one operator is selected at each iteration, which can be either a removal, a reinsertion or a swap.

5.1. List of operators

1. ***Randomly remove ρ***: This operator selects one period, one plant and one vehicle and removes one randomly selected customer from it. It is repeated ρ times.
2. ***Remove worst ρ***: First, a transportation saving cost is calculated by not visiting each customer. The operator is applied ρ times, removing the visited customers yielding the highest saving.
3. ***Shaw removal route based***: This operator randomly selects a period t, a plant j, a vehicle k and a customer l served by this vehicle and computes the distance $min(c_{lu})$ to the closest customer u also being served by the same route. Then, all customers within $2min(c_{lu})$ from customer l are removed.
4. ***Shaw removal greedy***: Similar to the previous operator, differing only in the customer selection as the basis for the calculation of the minimum distance. In this case, the shortest distance between any two customers is considered.
5. ***Avoid consecutive visits***: This operator empties the second visit of all customers with deliveries in two consecutive periods. It is based on observation that good solutions often do no contain visits to the same customer on two consecutive periods, [16].
6. ***Empty one period***: Randomly selects a period and empties all its deliveries.
7. ***Empty one vehicle***: Randomly selects a vehicle and empties all its pickups and deliveries from all plants and all periods.

8. ***Empty one plant***: Randomly selects a plant and empties all deliveries of all vehicles from that plant over the planning horizon.
9. ***Farthest customer***: This operator randomly selects a period, a plant and a vehicle and removes its farthest customer. It is repeated ρ times.
10. ***Randomly insert ρ***: Randomly selects a period, a plant and a vehicle and inserts a randomly selected customer that has not yet been serviced during that period in this route, following the cheapest insertion rule. The operator is repeated ρ times.
11. ***Assignment to the nearest plant***: This operator randomly selects a period and a customer not served on that period, and inserts it on the nearest plant, following the cheapest insertion rule. It is repeated ρ times.
12. ***Insert best ρ***: Similar to the previous one, this operator inserts the customer with the smallest transportation cost. The operator is applied ρ times.
13. ***Shaw insertion***: This operator randomly selects a period t and a customer l not served in t. Then it is calculates $min(c_{lv}), \forall l \in \mathcal{C}$ with $v \in \mathcal{V}'$, and all customers not yet served in that period, distant up to $2min(c_{lv})$ are inserted in a route in t, according to cheapest insertion rule.
14. ***Swap ρ customers intra-routes***: Randomly selects a period t and two customers l, u, served by the same plant j in different vehicle routes, and swaps their assignments, following the cheapest insertion rule.
15. ***Swap ρ customers intra-plants***: Randomly selects a period t and two customers l, u, served by different plants, and swaps their assignments, following the cheapest insertion rule.

5.2. Exact subproblem solutions

Our matheuristics uses a MIP subproblem, called Initial Solution and Improvements (ISI), which plays a central role in the optimization process. This model works on three fronts: first, it generates a partial initial solution, scheduling the input pickups according to product deliveries, while minimizing the inventory and approximated transportation costs. The optimization of each delivery route is done exactly using a traveling salesman problem algorithm (TSP) employing the branch-and-cut technique proposed by Padberg and Rinaldi [26]. At the same time, the ISI is able to optimize the input pickups and the final product deliveries associated with a given set of vehicle routes. Finally, ISI is useful to improve solutions from the first phase of our ALNS, by removing or inserting pickups and deliveries, and swapping customers among routes in the same period as well. This approach simplifies the idea proposed by Coelho et al. [16], which employs one model to optimize quantity deliveries and another one to improve solutions.

As in Coelho et al. [16], we also exploit an idea proposed by Archetti et al. [6] to simplify and approximate the routing costs resulting from vertex removals and reinsertion as follows. For a given period t, let a_l^{kjt} be the routing cost reduction if customer l is removed from vehicle k departed from plant j, which obviously has $Y_{jl}^{kt} = 1$; let also b_l^{kjt} be the routing cost, according to cheapest insertion rule, if customer l is inserted in vehicle k departed from plant j, which obviously has $Y_{jl}^{kt} = 0$. As these routing costs are evaluated before any removal or insertion procedure, the exact costs are valid only for one movement, and represent approximations otherwise.

In the case of input pickups, the routing costs are exact since the routes are direct. In this case, we consider c_{ij} as the associated cost with a pickup insertion (when $X_{ij}^{kt} = 0$) or pickup removal ($X_{ij}^{kt} = 1$) for plant j with vehicle k at supplier i in a given period t. Finally, let the binary parameters $\psi_l^{kjt} = 1$ if customer l is served in the current route of vehicle k from plant j in period t, and $\Psi_{ij}^{kt} = 1$ if plant j picks up some input from supplier i with vehicle k in period t. The variables of the ISI model are:

- $\delta_l^{kjt} = 1$ if customer l is removed from the route of vehicle k departing from plant j in period t, and 0 otherwise;
- $\omega_l^{kjt} = 1$ if customer l is inserted in the route of vehicle k departing from plant j in period t, and 0 otherwise;
- $\Theta_{ij}^{kt} = 1$ if input pickup of vehicle k from plant j at supplier i in period t is removed from the solution, and 0 otherwise;

The ISI model is formulated by:

$$\sum_{t\in\mathcal{T}}\sum_{j\in\mathcal{P}} h_j I_j^t + \sum_{t\in\mathcal{T}}\sum_{l\in\mathcal{C}} h_l I_l^t + \sum_{t\in\mathcal{T}}\sum_{j\in\mathcal{P}}\sum_{k\in\mathcal{K}}\sum_{i\in\mathcal{F}} 2c_{ij} X_{ij}^{kt} + \sum_{t\in\mathcal{T}}\sum_{j\in\mathcal{P}}\sum_{k\in\mathcal{K}}\sum_{l\in\mathcal{C}} b_l^{kjt}\omega_l^{kjt} - \\ - \sum_{t\in\mathcal{T}}\sum_{j\in\mathcal{P}}\sum_{k\in\mathcal{K}}\sum_{i\in\mathcal{F}} 2c_{ij}\Theta_{ij}^{kt} - \sum_{t\in\mathcal{T}}\sum_{j\in\mathcal{P}}\sum_{k\in\mathcal{K}}\sum_{l\in\mathcal{C}} a_l^{kjt}\delta_l^{kjt} \tag{34}$$

subject to (2)−(6) and to

$$q_{jl}^{kt} \le U_l - I_l^{t-1} \qquad l \in \mathcal{C}, j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T}\setminus\{0\} \tag{35}$$

$$q_{jl}^{kt} \le \left(\psi_l^{kjt} - \delta_l^{kjt} + \omega_l^{kjt}\right) U_l \qquad l \in \mathcal{C}, j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T} \tag{36}$$

$$\omega_l^{kjt} \le 1 - \psi_l^{kjt} \qquad l \in \mathcal{C}, j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T} \tag{37}$$

$$\delta_l^{kjt} \le \psi_l^{kjt} \qquad l \in \mathcal{C}, j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T} \tag{38}$$

$$r_{ij}^{kt} \le U_j - I_j^{t-1} \qquad j \in \mathcal{P}, i \in \mathcal{F}, k \in \mathcal{K}, t \in \mathcal{T}\setminus\{0\} \tag{39}$$

$$r_{ij}^{kt} \leq \left(\Psi_{ij}^{kt} - \Theta_{ij}^{kt} + X_{ij}^{kt}\right) U_j \qquad j \in \mathcal{P}, i \in \mathcal{F}, k \in \mathcal{K}, t \in \mathcal{T} \tag{40}$$

$$X_{ij}^{kt} \leq 1 - \Psi_{ij}^{kt} \qquad j \in \mathcal{P}, i \in \mathcal{F}, k \in \mathcal{K}, t \in \mathcal{T} \tag{41}$$

$$\Theta_{ij}^{kt} \leq \Psi_{ij}^{kt} \qquad j \in \mathcal{P}, i \in \mathcal{F}, k \in \mathcal{K}, t \in \mathcal{T} \tag{42}$$

$$\sum_{j \in \mathcal{P}} \sum_{l \in \mathcal{C}} \left(\delta_l^{kjt} + \omega_l^{kjt}\right) + \sum_{i \in \mathcal{F}} \sum_{j \in \mathcal{P}} \left(\Theta_{ij}^{kt} + X_{ij}^{kt}\right) \leq G \qquad k \in \mathcal{K}, t \in \mathcal{T} \tag{43}$$

$$\sum_{l \in \mathcal{C}} q_{jl}^{kt} \leq Q \qquad j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T} \tag{44}$$

$$r_{ij}^{kt} \leq Q \qquad j \in \mathcal{P}, i \in \mathcal{F}, k \in \mathcal{K}, t \in \mathcal{T} \tag{45}$$

$$\Theta_{ij}^{kt}, \delta_l^{kjt}, \omega_l^{kjt} \in \{0, 1\} \qquad l \in \mathcal{C}, j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T}. \tag{46}$$

The objective function (34) minimizes the total inventory, removal and insertion cost. The ML policy is formulated by constraints (35) and (36) for customers, and by (39) and (40) for plants. Constraints (37) prevent a customer from being inserted on a route already containing it. On the other hand, constraints (38) ensure that removing a customer from a route is only possible if that customer belongs to the route. Constraints (41) and (42) model the same situation for plants. The limit of removals and insertions is done by constraints (43). Vehicle capacity is imposed by constraints (44) and (45). Constraints (46) define the domain of the variables.

As pointed out by Coelho et al. [16], if $G > 1$, the model only provides an approximation for the current routing cost, since the ISI model removes and reinserts vertices, taking both decisions at the same time. Hence, if G is large, the changes on the routes do not accurately measure the improvement of the incumbent solution cost. Likewise, if G is small, the model may not be able to recover feasibility, in case an ALNS operator yields an infeasible solution. Thus, determining a suitable value for G is critical so that the ISI model can improve the current solution.

The input OU policy for the input is guaranteed by adding constraints (47) to the model, while the OU policy for the final product at the customers is imposed by constraints (48).

$$r_{ij}^{kt} \geq \left(\Psi_{ij}^{kt} - \Theta_{ij}^{kt} + X_{ij}^{kt}\right) U_j - I_j^{t-1} \qquad j \in \mathcal{P}, i \in \mathcal{F}, k \in \mathcal{K}, t \in \mathcal{T} \setminus \{0\} \tag{47}$$

$$q_{jl}^{kt} \geq \left(\psi_l^{kjt} - \delta_l^{kjt} + \omega_l^{kjt}\right) U_l - I_l^{t-1} \qquad l \in \mathcal{C}, j \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T} \setminus \{0\}. \tag{48}$$

5.3. Parameter setting

During an adjustment phase, we observed that the ISI model is quite efficient exploring potential neighborhoods, when iteratively invoked for a small number of exchanges. After each ALNS iteration, a

neighborhood s' is obtained by solving the ISI model with $G = n + m$, where n is the number of plants and m is the number of customers, optimizing the pickups and deliveries for a feasible set of routes. If $z(s') \leq (1 + \epsilon)\, z(s)$, where ϵ is randomly generated between $[0.05, 0.15]$, we accept s' as a potential neighborhood. Then, we define $G = max\{\xi * (n + m), 1\}$, where ξ is a random value from the interval $[0.1, 0.2]$, and the ISI model is solved again. The process continues whenever a potential neighborhood is generated, decreasing G by one unit until $G = 1$. As a final polishing, each time the ISI model is solved, a separate procedure solves the TSP associated with each existing route.

Due to the number of times the ISI model is solved, the heuristic stopping criteria is 1000 iterations. We set $\tau_{start} = 8000$, and $\phi = 0.989$, which generates approximately 1,000 iterations when the temperature reaches 0.1. The reaction factor η was set to 0.8. Scores are updated with $\sigma_1 = 10$, $\sigma_2 = 5$ and $\sigma_3 = 2$. The segment length is $\Delta = 20$ and whenever the number of iterations reaches this value, all weights are updated, the scores are set to zero and the value of ϵ is renewed. The potential neighborhood exploration is also bounded by Δ, when the ISI model is solved again to explore new directions from s_{best}. Algorithm 2 provides the pseudocode of our matheuristic.

6. Computational experiments

All algorithms were coded in Visual Basic.NET with Visual Studio Community 2015, executed in an Intel Core i5-6200U with 2.40GHz and 8 GB of RAM installed, running in Windows 10. The B&C and ISI were solved by Gurobi 7.0.2. The B&C was run for 7200 seconds.

The performance of the algorithms was evaluated on a set of 128 instances adapted from single vehicle IRP instances of Archetti et al. [5], being 64 instances with a fleet of $K = 1$ and 64 with $K = 3$ vehicles. Each of 128 instances was solved to four different inventory policies, totaling 512 tests. The instances are divided into four groups: low inventory cost with three (absH3low) and six (absH6low) periods, and high inventory cost with three (absH3high) and six (absH6high) periods. The number of customers ranges from 5 to 50.

We kept the number of periods and all information about the customers on the instance (number, location, demand, initial inventory, inventory cost and inventory capacity), as well the location of the depot. To configure 2E multi-depot multi-vehicle scenarios, we generated the parameters as follows.

- Suppliers
  - Coordinates: X and Y follow a discrete uniform distribution in the interval $U[500, 1000]$ and $U[0, 1000]$, respectively;

Algorithm 2 Matheuristic pseudocode

Initialize weights of removal and insertion operators to 1 and scores to 0.
Set $G = n + m$ and solve ISI and all associated TSPs, yielding s_{ini}.
$s_{best} \leftarrow s \leftarrow s_{ini}$; $\tau \leftarrow \tau_{start}$; $iterations = 0$.
Generate ϵ.
while $\tau > 0.01$ and $iterations < 1.000$ **do**
  Select and apply an operator to s; Set $G = n + m$, solve ISI and all associated TSPs, yielding s'.
  if $z(s') < z(s)$ **then**
    $s \leftarrow s'$;
    Set $G = max\{G - 1, 1\}$.
    Solve ISI for s and all associated TSPs, yielding s''.
    if $z(s'') < z(s)$ **then**
      $s \leftarrow s''$;
      Set $G = max\{G - 1, 1\}$ and go to step (10).
    else if $z(s'') \leq (1 + \epsilon)\, z(s)$ **then**
      Set $G = max\{\xi * (n + m), 1\}$ and go to step (10).
    end if
    if $z(s) < z(s_{best})$ **then**
      $s_{best} \leftarrow s$;
      increase the score of the operators by σ_1;
      Set $G = max\{G - 1, 1\}$ and go to step (10).
    else
      increase the score of the operators by σ_2;
      if $z(s'') \leq (1 + \epsilon)\, z(s_{best})$ **then**
        Set $G = max\{\xi * (n + m), 1\}$ and go to step (10).
      end if
    end if
  else if s' is accepted by the simulated annealing criterion **then**
    $s \leftarrow s'$;
    increase the score of the operators by σ_3.
  end if
  if the iteration count is a multiple of Δ **then**
    Generate ϵ, update the weights of all operators and reset their scores;
    $s \leftarrow s_{best}$.
  end if
  $iterations \leftarrow iterations + 1$.
end while
return s_{best};

  - Input available (Φ_i): given by $U[500, 1000]\left(\frac{r^0\varphi}{|\mathcal{F}|}\right)$, where r^0 is the initial inventory at depot of each instance.

- Plants
  - Coordinates: when $|P| > 1$, X and Y follow a discrete uniform distribution in the interval $U[0, 500]$;
  - Inventory capacity (U_j): given by $U[0.2, 0.3]\left(\frac{\sum_{l\in\mathcal{C}} U_l}{|\mathcal{P}|}\right)$;
  - Initial inventory $\left(I_j^0\right)$: given by $U[0.2, 0.3]\,(U_j)$;
  - Inventory cost (h_j): follow a discrete uniform distribution in the interval $U[0.2, 0.3]$ for high cost and $U[0.02, 0.03]$ for low cost instances.

- Vehicles
  - Capacity (Q): given by $max\{U_j\}$ in order to avoid infeasibility of the OU policy for input.

In order to analyze the effect of the logistic system structure, whenever the number of suppliers or plants is greater than one, the location of the previous ones is preserved. All instances generated and detailed results are available upon request.

6.1. Branch-and-cut and hybrid heuristic comparison

We start our analysis by presenting the results obtained with the exact B&C algorithm and with our hybrid heuristic. Tables 1 and 2 shows the B&C performance for instances with one and three vehicles, respectively. The first three columns describe the 2E structure with the number of customers $(\boldsymbol{C})$, plants $(\boldsymbol{P})$ and suppliers $(\boldsymbol{S})$, followed by the percentage of optimal solutions found in each inventory policy (input-final product), and the running time. The B&C was able to find optimal solutions for all instances up to 25 customers for one vehicle in a simple 2E system (one plant and one supplier), and up to 10 customers for three vehicles. The B&C performance clearly deteriorates as the system becomes more complex. Moreover, the results show that more restricted inventory policies for products tend to lead to more difficult cases with fewer optimal solutions found, especially for $K = 3$. In addition, we highlight the symmetry when the inventory policy for input changes. Since replenishment of input is done by direct pickups, the input inventory policy does not compromise the B&C performance. In all 512 tests, considering both $K = 1$ and $K = 3$, B&C found optimal solutions for 298 cases (58.2%), could not prove optimality for for 99 cases (19.3%), and did not find any feasible solution for 115 cases (22.5%).

Table 1: Average percentage of optimal solutions for $K = 1$ found by B&C algorithm

$K = 1$			ML-ML		ML-OU		OU-ML		OU-OU	
C	P	S	% Opt	Time (s)	% Opt	Time (s)	% Opt	Time (s)	% Opt	Time (s)
5	1	1	100	0.43	100	0.63	100	0.55	100	0.52
	2	2	100	2.27	100	2.49	100	1.52	100	3.54
	2	3	100	61.54	100	6.10	100	1.93	100	4.08
	3	2	100	6.36	100	6.18	100	4.20	100	5.88
10	1	1	100	1.30	100	1.28	100	1.40	100	1.41
	2	2	100	42.40	100	88.66	100	30.18	100	108.35
	2	3	100	32.17	100	88.52	100	41.09	100	94.55
	3	2	100	90.14	100	337.75	100	67.08	100	574.29
25	1	1	100	33.90	100	381.47	100	116.24	100	445.19
	2	2	50	3604.80	50	3662.88	75	3356.07	50	3578.27
	2	3	75	3132.14	50	3672.48	50	3606.57	50	3657.86
	3	2	50	3612.68	50	3857.47	50	3612.43	50	3785.60
50	1	1	50	3630.28	0	7200.63	50	3623.17	0	7200.60
	2	2	50	3706.84	0	7200.19	50	3678.27	0	7200.32
	2	3	50	3720.21	0	7200.13	50	3794.06	0	7200.21
	3	2	50	4301.78	0	7200.11	50	4084.11	0	7200.13
Avg			**79.69**	**1623.70**	**65.63**	**2556.68**	**79.69**	**1626.18**	**65.63**	**2577.55**

Table 2: Average percentage of optimal solutions for $K = 3$ found by B&C algorithm

$K = 3$			ML-ML		ML-OU		OU-ML		OU-OU	
C	P	S	% Opt	Time (s)	% Opt	Time (s)	% Opt	Time (s)	% Opt	Time (s)
5	1	1	100	1.85	100	2.08	100	1.77	100	2.46
	2	2	100	81.88	100	431.61	100	96.70	100	236.38
	2	3	100	66.88	100	320.39	100	98.30	100	293.32
	3	2	100	238.65	100	1057.05	100	221.53	100	851.44
10	1	1	100	54.08	100	114.12	100	36.18	100	180.01
	2	2	50	3716.25	50	3700.43	50	3703.00	50	3684.40
	2	3	50	3697.00	50	3801.29	50	3704.44	50	3700.91
	3	2	50	3812.53	50	3647.50	50	3755.09	50	3661.62
25	1	1	50	4314.51	25	5663.38	75	4114.42	25	5773.09
	2	2	0	7200.03	0	7200.05	0	7200.06	0	7200.24
	2	3	25	6920.08	0	7200.04	0	7200.04	0	7200.04
	3	2	0	7200.03	0	7200.04	0	7200.09	0	7200.03
50	1	1	0	7200.14	0	7200.17	0	7200.14	0	7200.77
	2	2	0	7209.74	0	7200.06	0	7200.13	0	7200.09
	2	3	0	7200.17	0	7200.07	0	7200.09	0	7200.53
	3	2	0	7200.09	0	7200.08	0	7200.11	0	7200.07
Avg			**45.31**	**4132.12**	**42.19**	**4321.15**	**45.31**	**4133.25**	**42.19**	**4299.09**

Tables 3 and 4 summarize the results for one and three vehicles, respectively, and compares the performance of the hybrid heuristic and B&C, for the 298 cases for which the exact algorithm was able to prove optimality. Column **Policy** provides all combinations for inventory policies, while **B&C** and **Total %** give the number and proportion out of the 128 instances in each inventory policy for which B&C found an optimal solution. Column **Heuristic** gives the number of optimal solutions found by our method for each inventory policy, followed by the average and maximum gap between the heuristic solution and the B&C upper bound (UB). Even for the few cases for which the heuristic did not find the optimal solution, the average gap was less than 0.010% for $K = 1$ and 0.015% for $K = 3$, while the maximum gap was no greater than 0.4%, showing great robustness and efficiency.

Table 3: Number of B&C optimal solutions and comparison against the heuristic for $K = 1$

Policy	B&C	% Total	Time B&C (s)	Heuristic	Time Heuristic (s)	Avg gap %	Max gap %
ML-ML	51	79.69	202.22	49	21.50	0.006	0.20
ML-OU	42	65.63	124.36	41	21.91	0.005	0.20
OU-ML	51	79.69	205.28	50	23.41	0.004	0.20
OU-OU	42	65.63	156.14	40	26.59	0.019	0.40
Avg		**72.66**	**172.00**		**23.35**	**0.008**	**0.25**

Table 4: Number of B&C optimal solutions and comparison against the heuristic for $K = 3$

Policy	B&C	% Total	Time B&C (s)	Heuristic	Time Heuristic (s)	Avg gap %	Max gap %
ML-ML	29	45.31	428.08	28	186.08	0.010	0.30
ML-OU	27	42.19	375.96	26	225.31	0.007	0.20
OU-ML	29	45.31	431.92	27	90.20	0.014	0.30
OU-OU	27	42.19	323.49	25	88.07	0.022	0.30
Avg		**43.75**	**389.86**		**147.41**	**0.013**	**0.28**

We also evaluated the heuristic performance over 99 remaining open cases, being 23 instances to ML-ML, 29 to ML-OU, 21 to OU-ML and 26 to OU-OU policy. Tables 5 and 6 provide, for one and three vehicles respectively, the average result for each inventory policy. Column **Total %** indicates the percentage of open instances in relation to total, followed by the average gaps between upper and lower bounds provided by B&C, the heuristic result in comparison with the B&C lower bound (LB) and finally, the average reduction with impact to the B&C UB. We note that the heuristic has reduced the gaps of all instances of this category, with more significant results for the more restricted inventory policies.

There were 115 cases without a solution found by the B&C algorithm, being 25 instances to ML-ML, 30 to ML-OU, 27 to OU-ML and 33 to OU-OU policy. Table 7 presents the results for the group of more

Table 5: Comparison between B&C and heuristic for open instances with $K = 1$

Policy	Open instances (%)	Avg gap B&C (%)	Time (s)	Avg gap (%) Heuristic / LB	Avg gap (%) Heuristic / UB	Time (s)
ML-ML	10.94	8.27	7200.32	3.94	−5.09	137.92
ML-OU	21.88	12.66	7200.28	5.34	−9.26	129.75
OU-ML	9.38	5.10	7200.69	4.07	−1.10	191.07
OU-OU	17.19	19.86	7200.25	8.22	−16.10	126.64
Avg	**14.84**	**11.47**	**7200.39**	**5.39**	**−7.89**	**146.35**

Table 6: Comparison between B&C and heuristic for open instances with $K = 3$

Policy	Open instances (%)	Avg gap B&C (%)	Time (s)	Avg gap (%) Heuristic / LB	Avg gap (%) Heuristic / UB	Time (s)
ML-ML	25.00	15.09	7200.06	11.21	−5.05	933.17
ML-OU	23.44	27.06	7200.05	19.23	−11.57	1567.67
OU-ML	23.44	15.97	7200.05	12.39	−4.43	753.13
OU-OU	23.44	24.68	7200.26	18.15	−9.73	1145.98
Avg	**23.83**	**20.70**	**7200.10**	**15.24**	**−7.69**	**1099.99**

difficult instances. For this case, the average gap between the solution obtained by the heuristic and the LB of the B&C was approximately 13% for one vehicle and 26% for three vehicles, without sensitive difference between inventory policies.

Table 7: Comparison between B&C and ALNS for unsolved instances

	$K = 1$			$K = 3$		
Policy	**Unsolved instances (%)**	**Avg gap Heuristic / LB (%)**	**Time (s)**	**Unsolved instances (%)**	**Avg gap Heuristic / LB (%)**	**Time (s)**
ML-ML	9.38	11.43	631.98	29.69	23.56	1810.43
ML-OU	12.50	13.26	881.31	34.38	25.90	2289.50
OU-ML	10.94	15.43	586.53	31.25	25.58	1713.67
OU-OU	17.19	12.82	786.48	34.38	28.72	2763.35
Avg	**12.50**	**13.24**	**721.57**	**32.42**	**25.94**	**2144.24**

In order to illustrate the efficiency of our hybrid heuristic, we show in Table 8 how long it takes to find a solution as good as the best one achieved by B&C to be obtained by our method. Note that the times indicated in this table refer to how long it took B&C to find its best solution, and how long it took our heuristic to find a solution just as good. In addition to achieving significantly better results, our heuristic is substantially faster, requiring, on average, 16.8% of the B&C time to beat it. Another interesting point is observed in the OU-OU policy, for which the heuristic was significantly faster. This is due to the consolidation that this policy promotes, generating fewer routes, since the resolution of associated TSPs requires more processing time.

Table 8: Comparison between B&C and heuristic for CPU processing time

Policy	N. Instances	Avg B&C (s)	Avg Heuristic (s)	Avg Time %
ML-ML	103	1235.63	215.24	17.42
ML-OU	98	1742.10	292.45	16.79
OU-ML	101	1237.00	301.89	24.41
OU-OU	95	1474.88	127.02	8.61
Avg		**1422.40**	**234.15**	**16.81**

Given the superior performance of our hybrid heuristic, all following analysis are performed with respect to its results.

6.2. Analysis of inventory policies

In this section, we evaluate the impact of choosing each of the inventory policies from the perspective of the 2E system. The results of Table 9 compare the average increase in total cost with respect to the ML-ML policy, switching to OU for input at plants **(Δ % OU-Inp)**, followed by OU for final product at customers **(Δ % OU-Prd)** and for both **(Δ % OU-Prd OU-Inp)**. The last column brings the effect of switching to OU in the customer inventory policy, taking as reference the OU-ML policy. Imposing of the OU policy at customers increases the total cost in almost 9% on average, with respect to ML-ML, and 8% with respect to to OU-ML, which is consistent with the findings of Archetti et al. [5] for the basic IRP, Coelho et al. [16] for multi-vehicle IRP and Coelho et al. [17] for the IRP with transshipment. It is also noteworthy that the average cost increase when the OU policy is imposed to plants tends to be smaller when the system becomes more complex, because the costs are distributed between the plants and there are more options to pickup input.

Table 10 makes it possible to evaluate the effect of system complexity on average costs. As the location of the nodes is preserved, comparisons are made in relation to the structure with one plant and one supplier. It is interesting to note that a system with more plants is more efficient than a system with more suppliers. This can be explained because inventory levels are smaller and the delivery to the customers is more balanced between plants. This becomes even more evident when the number of customers is higher and the inventory policy is less flexible.

6.3. Analysis of the cost structure

We also analyzed each component of the logistics costs separately. For a better analysis we separated the results according to the magnitude of the inventory cost and the number of periods in the planning horizon.

Table 9: Comparison of inventory policies' costs

			Policy				ML-ML			OU-ML
C	***P***	***S***	**ML-ML**	**ML-OU**	**OU-ML**	**OU-OU**	**Δ % OU-Inp**	**Δ % OU-Prd**	**Δ % OU-Inp OU-Prd**	**Δ % OU-Prd**
5	1	1	3657.83	3808.55	3792.41	3943.14	3.68	4.12	7.80	3.97
	2	2	3520.26	3716.41	3563.71	3758.59	1.23	5.57	6.77	5.47
	2	3	3546.19	3750.17	3587.68	3791.59	1.17	5.75	6.92	5.68
	3	2	3266.86	3406.45	3288.54	3420.06	0.66	4.27	4.69	4.00
10	1	1	4699.01	4983.46	4949.59	5257.96	5.33	6.05	11.90	6.23
	2	2	4637.11	5006.35	4728.19	5082.01	1.96	7.96	9.59	7.48
	2	3	4590.74	4951.21	4690.57	5045.79	2.17	7.85	9.91	7.57
	3	2	4724.18	5132.38	4739.38	5071.54	0.32	8.64	7.35	7.01
25	1	1	5272.51	5761.33	5660.41	6175.89	7.36	9.27	17.13	9.11
	2	2	5442.36	5956.93	5646.72	6135.36	3.76	9.45	12.73	8.65
	2	3	5421.83	5992.46	5613.74	6143.29	3.54	10.52	13.31	9.43
	3	2	5641.63	6198.97	5722.70	6208.57	1.44	9.88	10.05	8.49
50	1	1	7875.93	8826.66	8892.42	9872.11	12.91	12.07	25.35	11.02
	2	2	7853.67	8753.43	8295.96	9220.99	5.63	11.46	17.41	11.15
	2	3	7691.66	8711.69	8112.76	9045.21	5.47	13.26	17.60	11.49
	3	2	8115.44	9182.81	8238.24	9119.21	1.51	13.15	12.37	10.69
AVG			**5372.33**	**5883.70**	**5595.19**	**6080.71**	**3.63**	**8.71**	**11.93**	**7.97**

Table 10: Average decrease on cost with respect to the case with one plant and one supplier

C	***P***	***S***	**ML-ML %**	**ML-OU %**	**OUML %**	**OU-OU %**
5	2	2	−3.76	−2.42	−6.03	−4.68
	2	3	−3.05	−1.53	−5.40	−3.84
	3	2	−10.69	−10.56	−13.29	−13.27
10	2	2	−1.32	0.46	−4.47	−3.35
	2	3	−2.30	−0.65	−5.23	−4.04
	3	2	0.54	2.99	−4.25	−3.55
25	2	2	3.22	3.40	−0.24	−0.66
	2	3	2.83	4.01	−0.82	−0.53
	3	2	7.00	7.60	1.10	0.53
50	2	2	−0.28	−0.83	−6.71	−6.60
	2	3	−2.06	−0.48	−8.77	−8.38
	3	2	5.51	5.41	−7.36	−7.63

Table 11 provides the average ratio of each cost component with respect to total cost. From top to bottom, **InvP** gives the ratio of input inventory cost at plants, **InvC** the inventory cost at customers, **S-P** the transportation cost by the input pickups and **P-C** the ratio of distribution costs to customers. Since the input-output production ratio is 1 to 10, the ratio of the cost of input inventory at the plant is no greater then 1.12% when low inventory cost is considered. On the other hand, the transportation cost by the input pickups represents around 20% of the total cost. Even for the OU-OU policy, for which inventory levels are higher, and with high storage costs, this component accounts for 14.57%. Regardless of the policy, the replenishment of inventory of the plants is more representative than the customers' inventory cost, confirming the need to incorporate this link in the decision making in VMI systems.

Table 11: Average ratio of each cost dimension according to inventory cost levels

Inventory cost level	Component	ML-ML % of cost	ML-OU % of cost	OU-ML % of cost	OU-OU % of cost
Low	**InvP**	0.32	0.28	1.12	1.02
	InvC	1.65	2.40	1.64	2.40
	S-P	20.58	19.41	20.41	19.32
	P-C	77.45	77.91	76.83	77.27
High	**InvP**	2.07	1.82	8.33	7.34
	InvC	12.69	19.69	11.95	18.56
	S-P	18.10	16.21	16.35	14.57
	P-C	67.14	62.28	63.38	59.53

Table 12 presents the results when the planning horizon changes. When $H = 3$, the ratio of transportation cost by the input pickups exceeds $\frac{1}{4}$ of the total cost, achieving almost 30% for the ML-ML policy. As the horizon is short, the plants perform just one pickup, regardless of the inventory policy considered. When the time horizon is longer, this cost ends up being diluted between the other components, but even so, it is the second most representative one.

Table 12: Average proportion of each cost dimension according to time horizon

Time horizon	Component	ML-ML % of cost	ML-OU % of cost	OU-ML % of cost	OU-OU % of cost
H3	**InvP**	0.40	0.28	2.97	2.46
	InvC	7.51	11.08	7.30	10.82
	S-P	29.45	26.01	28.73	25.55
	P-C	62.64	62.63	61.00	61.17
H6	**InvP**	1.69	1.57	6.16	2.80
	InvC	7.66	12.47	7.38	10.58
	S-P	14.17	13.32	13.02	15.54
	P-C	76.48	72.64	73.44	71.07

6.4. Detailed analysis of the ISI model

Finally, we analyzed the strategy to explore potential neighborhoods due to the value of G. To this end, we modified the strategy to set the value of G in constraints (43) of the ISI model. Our first experiment is to remove all improvement opportunities due to the ISI model, by only invoking it to obtain pickup and delivery quantities, but setting $G = 0$ forbids any change to be performed on the vehicle routes. As a result, the ISI model is reduced to a network flow model, optimizing the delivery quantities associated with a given set of routes. This simplified version of the heuristic is described in Algorithm 3.

Algorithm 3 Simplified matheuristic pseudocode

1: Initialize weights of removal and insertion operators to 1 and scores to 0.
2: Solve modified ISI and all associated TSPs, yielding s_{ini}.
3: $s_{best} \leftarrow s \leftarrow s_{ini}$; $\tau \leftarrow \tau_{start}$; $iterations = 0$.
4: **while** $\tau > 0.01$ and $iterations < 1.000$ **do**
5: Select an operator and apply it to s; Solve modified ISI and all associated TSPs, yielding s'.
6: **if** $z(s') < z(s)$ **then**
7: $s \leftarrow s'$;
8: **if** $z(s) < z(s_{best})$ **then**
9: $s_{best} \leftarrow s$;
10: increase the score of the operators by σ_1;
11: **else**
12: increase the score of the operators by σ_2;
13: **end if**
14: **else if** s' is accepted by the simulated annealing criterion **then**
15: $s \leftarrow s'$;
16: increase the score of the operators by σ_3.
17: **end if**
18: **if** the iteration count is a multiple of Δ **then**
19: update the weights of all operators and reset their scores;
20: **end if**
21: **end while**
22: **return** s_{best};

Next, we simplify the potential neighborhoods exploration, considering only the real costs due to the insertion or removal of customers on the route. For this, we set the value of $G = 1$ on steps 9, 13, 16, 22 and 26 on Algorithm 2, also exploiting a potential neighborhood s' whenever $z(s') \leq (1 + \epsilon)\, z(s)$. We proceed on the same way, also for intermediate decreasing fixed values of G, being $G_A = n + m$, $G_B = \lceil \frac{n+m}{2} \rceil$ and $G_C = \lceil \frac{n+m}{3} \rceil$.

To compare these changes with the original flexible strategy to define G employed at Algorithm 2, we select a subset of instances ranging from small, medium and large sizes and solve them by Algorithm 3 and Algorithm 2 with $G = 1$, G_A, G_B and G_C, for each inventory policy. Table 13 shows the results for ML-ML policy. From left to right, the first column describes the instance according to $\boldsymbol{F}$(number of suppliers), $\boldsymbol{P}$(number of plants), $\boldsymbol{H}$(number of periods)(inventory level cost), $\boldsymbol{n}$(number of customers) and $\boldsymbol{v}$(number of vehicles). Sequentially, the table displays the solution and running time for each strategy considered, where $\boldsymbol{G_{flex}}$ and all fixed values of G refers to Algorithm 2, while $\boldsymbol{G = 0}$ to Algorithm 3.

Table 13 confirms the positive effect of the flexible choice for G. When G is fixed, the heuristic ends up requiring more time to polish solutions to a level as good as when G is flexible. There are also cases in which fixing the value G did not allow the heuristic to find the same final solutions. These five strategies are equivalent only for small-sized instances.

When removals and insertions are not allowed in the ISI model ($G = 0$), the result is very poor. Although faster, there were cases for which the heuristic was not even able to improve the initial solution, as in instance *F2_P2_H3low_n25*. We also emphasize that in more than 50% of the iterations the solutions found by the heuristic were infeasible. Since the modified ISI model does not include the possibility of inserting customers, there is no guarantee that it will recover the feasibility when a removal operator is selected.

Table 13: Analysis of the value of G for the ML-ML policy

Instances	Solution						CPU					
	G_{flex}	G_A	G_B	G_C	$G=1$	$G=0$	G_{flex}	G_A	G_B	G_C	$G=1$	$G=0$
F.1_P.1_absH3high_1n10_v.1	5075.32	5075.32	5075.32	5075.32	5075.32	5816.41	4.36	2.83	2.65	0.84	14.09	10.66
F.1_P.1_absH3high_1n25_v.3	3391.79	3391.79	3391.79	3391.79	3391.79	5127.51	16.54	126.65	57.65	4.93	71.63	15.26
F.1_P.1_absH3high_1n50_v.1	5562.99	5562.99	5562.99	5562.99	6002.91	6625.68	37.22	27.43	35.95	59.86	86.41	40.4
F.1_P.1_absH6high_1n10_v.3	5859.72	5867.85	5867.85	5867.85	5878.41	6414.3	118.23	10.28	43.63	31.85	15.73	33.86
F.1_P.1_absH6high_1n50_v.1	11198.43	11807.93	11198.43	11198.43	12184.55	13620.83	54.12	171.58	78.29	14.72	328.68	100.62
F.1_P.1_absH6low_1n10_v.3	4698.04	4698.04	4698.04	4698.04	4703.71	6868.65	38.64	88.61	43.01	42.78	51.36	54.14
F.2_P.2_absH3high_1n25_v.1	3088.79	3088.79	3088.79	3088.79	3088.79	4768.14	2.62	15.04	1.06	20.55	6.86	10.15
F.2_P.2_absH3high_1n50_v.1	4960.5	4960.50	4960.50	4960.50	4960.5	6567.71	71.96	28.21	22.65	43.93	267.74	24.58
F.2_P.2_absH3low_1n25_v.1	4639.85	4639.85	4639.85	4639.85	4641.8	5945.18	44.6	3.28	1.99	1.42	11.29	32.49
F.2_P.2_absH6low_1n10_v.1	2593.5	4639.85	4639.85	4639.85	2593.5	3884.05	8.35	19.23	16.89	43.29	4.28	28.85
F.2_P.2_absH6low_1n50_v.1	8982.35	8983.92	8982.35	9227.77	9722.65	12390.99	214.01	666.71	728.99	545.29	563.79	4.11
F.2_P.3_absH3high_1n10_v.3	4167.38	4167.38	4167.38	4167.38	4167.38	5347.33	16.15	39.48	582.62	252.72	92.35	14.97
F.2_P.3_absH3low_1n25_v.3	3201.66	3201.66	3221.49	3221.49	3221.49	5665.51	171.76	418.96	227.90	182.06	196.79	81.2
F.3_P.2_absH3high_1n25_v.1	3034.73	3034.73	3034.73	3034.73	3034.73	4712.49	4.68	10.05	4.95	0.95	23.91	6.28
F.3_P.2_absH6low_1n50_v.1	9388.55	9388.55	9388.55	9388.55	9617.53	11217.92	301.73	591.46	548.22	705.81	526.35	13.42
Average	**5322.91**	**5500.61**	**5461.19**	**5477.56**	**5485.67**	**6998.18**	**73.66**	**147.99**	**159.76**	**130.07**	**150.75**	**31.40**

The same pattern is found for other policies. Table 14 presents the results for the ML-OU inventory

policy, while Tables 15 and 16 present the solutions for the OU-ML and OU-OU ones. The results show that the flexible strategy of Section 5.3 designed for setting the value of G is robust enough to handle all inventory policies and obtain the best results.

Table 14: Analysis of the value of G for the ML-OU policy

Instances	Solution						CPU					
	G_{flex}	G_A	G_B	G_C	$G=1$	$G=0$	G_{flex}	G_A	G_B	G_C	$G=1$	$G=0$
F.1_P.1_absH3high_1n10_v.1	5218.62	5218.62	5218.62	5218.62	5218.62	5218.62	0.47	0.58	0.34	0.31	0.07	0.13
F.1_P.1_absH3high_1n25_v.3	4005.77	4005.77	4079.71	4005.77	4005.77	4794.19	113.67	118.07	110.94	190.44	312.67	14.31
F.1_P.1_absH3high_1n50_v.1	6697.13	6697.13	6697.13	6697.13	6832.55	7488.4	18.66	7.91	26.00	10.28	14.7	21.91
F.1_P.1_absH6high_1n10_v.3	6294.15	6294.15	6294.15	6294.15	6294.15	8351.62	76.66	19.93	26.79	26.36	145.25	28.1
F.1_P.1_absH6high_1n50_v.1	13282.1	13282.10	13284.10	13282.10	13757.75	14815.54	42.31	196.15	196.09	178.84	299.66	50.57
F.1_P.1_absH6low_1n10_v.3	4995.04	4995.04	4995.04	4995.04	4995.04	5738.37	225.42	84.29	63.43	111.60	172.55	30.35
F.2_P.2_absH3high_1n25_v.1	3593.13	3593.13	3593.13	3593.13	3593.13	5675.64	5.57	26.77	8.84	26.87	20.54	0.62
F.2_P.2_absH3high_1n50_v.1	5933.39	5933.39	5933.39	5933.39	6261.28	7784.48	17.85	183.18	13.55	118.14	174.45	41.36
F.2_P.2_absH3low_1n25_v.1	4948.67	4948.67	4948.67	4948.67	4948.67	5977.49	5.38	29.23	51.13	32.39	37.79	21.31
F.2_P.2_absH6low_1n10_v.1	3015.02	4948.67	4948.67	3015.02	3015.02	4320.46	19.72	39.04	83.07	138.56	130.46	26.55
F.2_P.2_absH6low_1n50_v.1	9271.47	9313.08	9271.47	9271.47	10105.68	11830.16	609.81	815.68	962.16	986.18	600.35	55.1
F.2_P.3_absH3high_1n10_v.3	4495.12	4495.12	4495.12	4495.12	4495.12	6310.6	84.32	37.73	109.87	38.24	4.76	0.08
F.2_P.3_absH3low_1n25_v.3	3547.87	3547.87	3547.87	3547.87	3564.44	6084.2	532.11	371.14	285.40	408.15	1268.11	0.3
F.3_P.2_absH3high_1n25_v.1	3539.06	3539.06	3539.06	3539.06	3539.06	5096.62	14.22	7.94	4.71	46.52	66.96	0.48
F.3_P.2_absH6low_1n50_v.1	9599.12	9599.12	9645.77	9599.12	10050.02	12334.77	510.11	1362.16	536.91	556.61	493.7	87.47
Average	**5895.71**	**6027.40**	**6032.79**	**5895.71**	**6045.09**	**7454.74**	**151.75**	**219.99**	**165.28**	**191.30**	**249.47**	**25.24**

7. Conclusions

In this study, we have introduced the 2E-MDIRP, a new variant of IRP problem which incorporates the inventory replenishment process of plants in a two echelon logistics system. This problem, inspired from a real life case, has the middle layer player controlling the decisions regarding pickups of input from plants and deliveries of final product to customers. Each echelon contains many nodes, yielding a complex many-to-many supply chain. We have proposed a MILP formulation and a B&C algorithm to solve the problem, considering several different inventory policies for input and final product management. We have developed an effective matheuristic algorithm composed of an ALNS enhanced by the exact solution of a MILP subproblem. The matheuristic is sufficiently flexible to handle all inventory policies. Extensive computational tests over a data set of instances adapted from literature with real-life parameters have shown that the matheuristic is able to find better solutions than the exact algorithm in shorter computational time. We have shown that the process of inventory replenishing at plants represents a significant portion of logistics costs and that a more complex logistics system allows greater efficiency in logistics operations.

Table 15: Analysis of the value of G for the OU-ML policy

Instances	Solution						CPU					
	G_{flex}	G_A	G_B	G_C	$G=1$	$G=0$	G_{flex}	G_A	G_B	G_C	$G=1$	$G=0$
F.1_P.1_absH3high_1n10_v.1	5298.03	5298.03	5298.03	5298.03	5298.03	6101.4	0.23	1.42	0.46	2.67	0.27	7.73
F.1_P.1_absH3high_1n25_v.3	3793.24	3793.24	3793.24	3793.24	3793.24	4621.56	159.1	44.67	31.74	120.58	275.43	5.2
F.1_P.1_absH3high_1n50_v.1	6142.01	6142.01	6142.01	6142.01	6581.92	7204.7	1.22	26.69	25.00	6.77	68.86	29.16
F.1_P.1_absH6high_1n10_v.3	6562.91	6562.91	6562.91	6562.91	6562.91	8062.63	11.95	43.79	15.91	16.54	143.19	25.76
F.1_P.1_absH6high_1n50_v.1	14228.58	14228.58	14228.58	14228.58	14228.58	14228.58	9.2	7.00	6.33	15.62	167.1	51.12
F.1_P.1_absH6low_1n10_v.3	4775.02	4775.02	4775.02	4775.02	4775.02	6214.49	98.38	208.11	123.11	173.02	289.79	22.05
F.2_P.2_absH3high_1n25_v.1	3223.18	3223.18	3223.18	3223.18	3223.18	4327.63	0.42	2.69	3.70	12.59	7.57	26.44
F.2_P.2_absH3high_1n50_v.1	5273.65	5273.65	5273.65	5273.65	5775.81	6613.14	2.78	60.20	36.14	130.72	61.24	50.09
F.2_P.2_absH3low_1n25_v.1	4666.83	4666.83	4666.83	4666.83	4668.79	5793.4	99.35	1.66	0.67	1.94	36.32	2.09
F.2_P.2_absH6low_1n10_v.1	2609.21	4666.83	4666.83	2609.21	2609.21	3264.54	1.64	20.36	4.35	7.39	1.14	0.27
F.2_P.2_absH6low_1n50_v.1	9118.35	9125.41	9330.50	9118.35	9868.21	12070.5	850.31	717.77	358.67	907.36	553.3	136.49
F.2_P.3_absH3high_1n10_v.3	4212.67	4212.67	4212.67	4212.67	4212.67	5374.31	193.2	381.28	74.65	563.61	15.08	11.68
F.2_P.3_absH3low_1n25_v.3	3210.54	3230.54	3349.90	3210.54	3330.51	5338.82	281.52	289.05	376.85	220.46	428.48	67.34
F.3_P.2_absH3high_1n25_v.1	3186.18	3186.18	3186.18	3186.18	3186.18	3975.83	0.32	16.02	0.43	0.55	0.34	1
F.3_P.2_absH6low_1n50_v.1	9461.7	9461.70	9502.70	9461.70	10165.94	12170.04	651.64	284.07	625.50	482.14	432.44	119.89
Average	**5717.47**	**5856.45**	**5880.82**	**5717.47**	**5885.35**	**7024.10**	**157.42**	**140.32**	**112.23**	**177.46**	**165.37**	**37.09**

Table 16: Analysis of the value of G for the OU-OU policy

Instances	Solution						CPU					
	G_{flex}	G_A	G_B	G_C	$G=1$	$G=0$	G_{flex}	G_A	G_B	G_C	$G=1$	$G=0$
F.1_P.1_absH3high_1n10_v.1	5520.1	5520.10	5520.10	5520.10	5520.1	6147.01	0.03	2.25	3.29	8.78	0.69	5.87
F.1_P.1_absH3high_1n25_v.3	4374.04	4521.04	4374.04	4374.04	4374.04	4659.56	82.98	144.65	101.99	125.77	436.04	0.09
F.1_P.1_absH3high_1n50_v.1	7249.22	7249.22	7249.22	7249.22	7388.97	8023.48	17.74	28.18	37.96	15.60	3.4	3.63
F.1_P.1_absH6high_1n10_v.3	7096.83	7096.83	7096.83	7096.83	7096.83	8243.91	26.54	79.95	124.25	55.06	94.43	5.13
F.1_P.1_absH6high_1n50_v.1	16231.08	16231.08	16231.08	16231.08	16231.08	17773.74	261.15	75.30	36.64	211.13	5.98	69.55
F.1_P.1_absH6low_1n10_v.3	5069.5	5069.50	5076.86	5069.50	5076.86	5901.23	62.76	301.36	68.49	435.28	319.05	0.53
F.2_P.2_absH3high_1n25_v.1	3720.53	3720.53	3720.53	3720.53	3720.53	4817.73	0.55	31.62	15.96	12.52	11.36	6.94
F.2_P.2_absH3high_1n50_v.1	6243.59	6243.59	6243.59	6243.59	6583.74	7163.33	31.04	83.87	155.36	144.81	17.69	1.65
F.2_P.2_absH3low_1n25_v.1	4972.72	4972.72	4972.72	4972.72	4972.72	5349.3	61.68	4.57	25.50	120.64	322.75	40.35
F.2_P.2_absH6low_1n10_v.1	3042.32	4972.72	4972.72	3042.32	3062.74	4173.72	4.02	44.00	12.12	177.24	29.18	11.53
F.2_P.2_absH6low_1n50_v.1	9636.2	9636.20	9636.20	9636.20	9989.67	12113.86	532.74	790.09	911.14	668.75	284.9	76.34
F.2_P.3_absH3high_1n10_v.3	4535.9	4556.25	4565.16	4535.90	4535.9	5374.31	78.16	138.12	163.14	257.34	70.88	0.09
F.2_P.3_absH3low_1n25_v.3	3555.86	3555.86	3555.86	3555.86	3668.54	5374.98	458.9	741.94	296.94	201.07	983.29	0.24
F.3_P.2_absH3high_1n25_v.1	3683.54	3683.54	3683.54	3683.54	3683.54	5547.21	4.48	24.54	3.26	9.24	18.7	2.6
F.3_P.2_absH6low_1n50_v.1	9557.56	9594.82	9972.06	9557.56	10364.35	11654.08	1103.46	848.93	101.54	576.32	514.8	75.19
Average	**6299.27**	**6441.60**	**6458.03**	**6299.27**	**6417.97**	**7487.83**	**181.75**	**222.63**	**137.17**	**201.30**	**207.54**	**19.98**

Acknowledgments

This project was partly funded by the Natural Sciences and Engineering Research Council of Canada (NSERC) under grant 2014-05764. This support is greatly acknowledged. We thank an associate editor and two anonymous referees for their valuable comments on an earlier version of this paper.

References

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] F. Al-Khayyal and S. Hwang. Inventory constrained maritime routing and scheduling for multi-commodity liquid bulk, part I: applications and model. *European Journal of Operational Research*, 176(1):106–130, 2007.

[3] 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.

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

[5] 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.

[6] C. Archetti, L. Bertazzi, A. Hertz, and M. G. Speranza. A hybrid heuristic for an inventory routing problem. *INFORMS Journal on Computing*, 24(1):101–116, 2012.

[7] L. Bertazzi and M. G. Speranza. Inventory routing problems: an introduction. *EURO Journal on Transportation and Logistics*, 1(4):307–326, 2012.

[8] L. Bertazzi, G. Paletta, and M. G. Speranza. Deterministic order-up-to level policies in an inventory routing problem. *Transportation Science*, 36(1):119–132, 2002.

[9] L. Bertazzi, A. Bosco, and D. Laganà. Managing stochastic demand in an inventory routing problem with transportation procurement. *Omega*, 56:112–121, 2015.

[10] L. Bertazzi, L. C. Coelho, A. Maio, and D. Laganà. The multi-depot inventory routing problem: an application of vendor-management inventory in city logistics. Technical Report CIRRELT-2017-54, Montréal, Canada, 2017.

[11] L. Chan and D. Simchi-Levi. Probabilistic analyses and algorithms for three-level distribution systems. *Management Science*, 44(11):1562–1576, 1998.

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

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

[14] 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.

[15] L. C. Coelho and G. Laporte. An optimised target-level inventory replenishment policy for vendor-managed inventory systems. *International Journal of Production Research*, 53(12):3651–3660, 2015.

[16] 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.

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

[18] L. C. Coelho, J.-F. Cordeau, and G. Laporte. Thirty years of inventory routing. *Transportation Science*, 48(1):1–19, 2013.

[19] L. C. Coelho, J.-F. Cordeau, and G. Laporte. Heuristics for dynamic and stochastic inventory-routing. *Computers & Operations Research*, 52(part A):55–67, 2014.

[20] J.-F. Cordeau, D. Laganà, R. Musmanno, and F. Vocaturo. A decomposition-based heuristic for the multiple-product inventory-routing problem. *Computers & Operations Research*, 55:153–166, 2014.

[21] R. Cuda, G. Guastaroba, and M. G. Speranza. A survey on two-echelon routing problems. *Computers & Operations Research*, 55:185–199, 2015.

[22] 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.

[23] K. Govindan. Vendor-managed inventory: a review based on dimensions. *International Journal of Production Research*, 51(13):3808–3835, 2013.

[24] M. Hewitt, G. L. Nemhauser, M. W. P. Savelsbergh, and J. Song. A branch-and-price guided search approach to maritime inventory routing. *Computers & Operations Research*, 40(5):1410–1419, may 2013.

[25] J. Li, F. Chu, and H. Chen. A solution approach to the inventory routing problem in a three-level distribution system. *European Journal of Operational Research*, 210(3):736–744, 2011.

[26] 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.

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

[28] M. Rahim, E.-H. Aghezzaf, V. Limère, and B. Raa. Analysing the effectiveness of vendor-managed inventory in a single-warehouse, multiple-retailer system. *International Journal of Systems Science*, 47(8):1953–1965, 2016.

[29] R. F. Roldán, R. Basagoiti, and L. C. Coelho. Robustness of inventory replenishment and customer selection policies for the dynamic and stochastic inventory-routing problem. *Computers & Operations Research*, 74:14–20, 2016.

[30] D. Ronen. Marine inventory routing: shipments planning. *Journal of the Operational Research Society*, 53(1):108–114, 2002.

[31] M. W. P. Savelsbergh and J. Song. Inventory routing with continuous moves. *Computers & Operations Research*, 34:1744–1763, 2007.

[32] V. Schmid, K.-F. Doerner, and G. Laporte. Rich routing problems arising in supply chain management. *European Journal of Operational Research*, 224(3):435–448, 2013.

[33] G. Sorda, M. Banse, and C. Kemfert. An overview of biofuel policies across the world. *Energy Policy*, 38(11):6977–6988, 2010.

[34] Q. H. Zhao, S. Chen, and C. X. Zang. Model and algorithm for inventory/routing decision in a three-echelon logistics system. *European Journal of Operational Research*, 191(3):623–635, 2008.