Formulations, Branch-and-Cut and a Hybrid Heuristic Algorithm for an Inventory Routing Problem with Perishable Products

Aldair Alvarez[a,c,*], Jean-François Cordeau[b,c], Raf Jans[b,c], Pedro Munari[a], Reinaldo Morabito[a],

[a]*Department of Production Engineering, Federal University of São Carlos, 13565-905, São Carlos-SP, Brazil*
[b]*Department of Logistics and Operations Management, HEC Montréal, H3T 2A7, Canada*
[c]*Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT), H3T 1J4, Montréal, Canada*

Abstract

In this paper, we study an inventory routing problem in which goods are perishable. In this problem, a single supplier is responsible for delivering a perishable product to a set of customers during a given finite multiperiod planning horizon. The product is assumed to have a fixed shelf-life during which it is usable and after which it must be discarded. Age-dependent revenues and inventory holding costs are considered. We introduce four mathematical formulations for the problem, two with a vehicle index and two without a vehicle index. Branch-and-cut algorithms are proposed to solve them. In addition, we propose a hybrid heuristic solution method for the problem. The method is based on the combination of an iterated local search metaheuristic and two mathematical programming components. We present extensive computational experiments using instances from the literature as well as new larger instances. Our results indicate that, when compared to an arc-based formulation from the literature, the formulations without a vehicle index can provide a considerably larger number of optimal and feasible solutions within the imposed time limit. Additionally, a considerable speed-up is achieved for those instances solved to optimality within the time limit. The results with the hybrid method show that it is able to provide high-quality solutions in relatively short running times for small- and medium-sized instances while good quality solutions are found within reasonable running times for larger instances.

Keywords: Logistics, Inventory routing, Perishability, Hybrid method, Iterated local search.

1. Introduction

Research on the integration of multiple activities throughout the supply chain has increased considerably in the last decades. Today, it is well known that such integration can lead to significant advantages in both economic and performance terms. In particular, the integration of transportation and inventory management activities has been shown to provide substantial economic benefits and to improve the usage of the available resources. However, challenging problems can arise from this integration, one of which is the inventory routing problem (IRP). The IRP consists of defining the optimal replenishment plan of the customers of a supplier throughout a planning horizon as well as the routing schedule in each time period such that a given objective is optimized.

*Corresponding author.
Email addresses: aldair@dep.ufscar.br (Aldair Alvarez), jean-francois.cordeau@hec.ca (Jean-François Cordeau), raf.jans@hec.ca (Raf Jans), munari@dep.ufscar.br (Pedro Munari), morabito@ufscar.br (Reinaldo Morabito)

In many different industries, raw materials, as well as intermediate and final products, are often perishable. Moreover, perishability may appear in more than one activity throughout the supply chain and can influence service levels (Amorim et al., 2013). Thus, managing perishability becomes a relevant issue in the supply chain, particularly in the inventory control activities. Perishability was first studied in the context of IRPs by Federgruen et al. (1986), who addressed an inventory management and distribution problem for a product that must be discarded if it not used during a given fixed lifetime. The authors studied different patterns and policies for the distribution part of the problem. The objective was to minimize the sum of transportation and expected shortage and discarding costs.

Hemmelmayr et al. (2009) studied the problem faced by a blood bank in the distribution to hospitals. In their problem, no vehicle capacity constraints were considered (given the small size of the blood bags) but the maximum length of the routes was limited. Also, no inventory holding costs were considered since it is preferable to maintain high inventory levels rather than to experience stockouts, given the nature of the service being provided. The objective is to minimize travel costs over a finite horizon. The authors proposed different strategies for the distribution process. Le et al. (2013) studied an IRP with perishability features also motivated by a healthcare application. In their problem, it was assumed that the perishable goods have a fixed shelf-life, and they are not usable when this lifetime is exceeded. Upper bounds on the inventory levels of customers were determined only by the perishability constraints since the discarding of products is not allowed. Thus, deliveries to customers at any given time period were limited only by the shelf-life of the goods. The objective was to minimize the sum of travel and inventory holding costs. Diabat et al. (2016) addressed the same problem as Le et al. (2013) but only minimizing travel costs.

Coelho and Laporte (2014) considered an IRP with a fixed shelf-life perishable product with age-dependent revenues and holding costs. They presented a mathematical formulation and explored different strategies to model the product consumption at the customer facilities. Mirzaei and Seifi (2015) addressed an IRP for perishable goods in which the objective function included a penalty that depends on the age of the product that is used to satisfy the demands. This penalty was included in an attempt to avoid overstocking to reduce transportation costs. The objective was to minimize the sum of routing, inventory and penalty costs. Soysal et al. (2015) addressed an IRP with a fixed shelf-life perishable product. The authors proposed models that also considered fuel consumption and demand uncertainty. Split deliveries and backlogging of the demand were allowed as well. The objective was to minimize the sum of routing (driver wages and fuel consumption), inventory and spoilage costs. Azadeh et al. (2017) studied an IRP of a single perishable product with an exponentially decaying inventory. The authors included the possibility of transshipments between customers (performed by an outsourced third-party operator) since a single vehicle with limited capacity was considered. Backlogging was not allowed and the objective was to minimize the sum of inventory and travel costs (including transshipments costs) as well as spoilage costs. Crama et al. (2018) addressed an IRP for a single perishable product with stochastic demands (with a known probability distribution). A maximum time on the duration of the routes was imposed and no salvage value was included in their problem.

Finally, Shaabani and Kamalabadi (2016) and Qiu et al. (2018) studied production-routing problems (PRP) for perishable products. PRPs add production decisions to the IRP in an attempt to jointly optimize production, inventory and routing decisions (Adulyasak et al., 2015). Shaabani and Kamalabadi (2016) addressed the case with multiple products. In their problem, perishability was modeled as in Le et al. (2013), i.e., upper bounds on the inventory levels are determined only by the perishability constraints and discarding of products is forbidden. Qiu et al. (2018) addressed an PRP including deterioration rates and inventory holding costs that are both age-dependent. The authors tested different delivery and selling priority policies.

In this paper, we address the IRP for a single perishable product as proposed by Coelho and Laporte (2014), which we will refer to as the PIRP (perishable IRP). In this problem, the perishability feature is modeled by defining a fixed shelf-life for an aging product as well as setting inventory holding costs and sales revenues varying with the product freshness. The contributions of this paper are threefold. First, we present and compare four mathematical formulations of the problem, which are solved using branch-and-cut (B&C) algorithms. We also report an inconsistency in the mathematical formulation presented by Coelho and Laporte (2014) and show how we addressed it. Second, we develop a hybrid heuristic method based on the combination of an iterated local search (ILS) metaheuristic and two mathematical programming components. Third, we report the results of extensive computational experiments and introduce new large-sized problem instances. The results show the different advantages of the proposed formulations and also reveal the effectiveness of our method. We also present a further analysis of the robustness of the algorithm's behavior.

The remaining sections of this paper are organized as follows. In Section 2, we describe the problem that we address in this research. Section 3 presents the mathematical formulations introduced for the problem and the B&C algorithms used to solve them. Then, the hybrid solution method that we developed is described in detail in Section 4. Section 5 shows the computational experiments that we performed with the formulations and the hybrid method. Finally, in Section 6 we conclude the paper and discuss future research.

2. Problem description

In the PIRP (Coelho and Laporte, 2014), a supplier is responsible for delivering a single perishable product to a set of customers during a given finite multiperiod planning horizon. The product is assumed to have a fixed shelf-life during which it is usable and after which it must be discarded.

The problem can be defined on a complete undirected graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ where $\mathcal{N} = \{0, 1, \ldots, N\}$ is the vertex set and $\mathcal{E} = \{(i, j) \colon i, j \in \mathcal{N}, i < j\}$ is the edge set. Vertex 0 represents the supplier depot which has a homogeneous fleet of K vehicles of capacity Q, denoted by set $\mathcal{K} = \{1, \ldots, K\}$. The remaining vertices of set $\mathcal{N}$, denoted by $\mathcal{C} = \{1, \ldots, N\}$, represent the customers. Therefore, the vertex set $\mathcal{N}$ represents all the facilities of the distribution network.

The planning horizon is denoted by a set of time periods $\mathcal{T} = \{1, \ldots, T\}$. The perishable product under consideration spoils S time periods after becoming available at the supplier and its age increases by one unit in every time period. Thus, the age of the product belongs to a discrete set $\mathcal{S} = \{0, 1, \ldots, S\}$. The age of the product defines its value, according to the sales revenue u_{is} specified for each unit of age $s \in \mathcal{S}$ consumed by customer $i \in \mathcal{C}$. A travel cost c_{ij} is associated with every edge $(i, j) \in \mathcal{E}$ and an age-dependent inventory holding cost h_{is} is charged at both the supplier 0 and the customers $i \in \mathcal{C}$ for each unit of product of age $s \in \mathcal{S}$ at the end of every time period. Each customer $i \in \mathcal{C}$ has a limited storage capacity C_i and each facility $i \in \mathcal{N}$ has an initial inventory I^0_{i0} of fresh product (of age 0) available at the beginning of the planning horizon ($t = 0$). Thus, the initial inventory will be of age

1 in the first time period of the planning horizon ($t = 1$). Each customer $i \in \mathcal{C}$ has a known demand d_i^t for the product in every time period $t \in \mathcal{T}$, which is the minimum amount of product that the supplier must guarantee to be available at the customer at that time period. In addition, the supplier produces or receives a quantity r^t of fresh products (of age 0) in each time period $t \in \mathcal{T}$. However, this quantity is available for delivery only one time period after becoming available at the supplier's facility.

To illustrate the aging process of the inventory during the planning horizon, Figure 1 shows an example of the evolution of the end-of-period inventory for a given customer. Assume a maximum age of two time periods ($S = 2$), no consumptions during the planning horizon and two deliveries from the supplier of 70 and 50 units of age 1 ($s = 1$) in time periods two and three, respectively. The initial inventory of the customer consists of 100 units ($s = 0$ at $t = 0$), which become of age 1 in the first time period ($s = 1$ at $t = 1$) and then of age 2 in the second time period ($s = 2$ at $t = 2$) of the planning horizon. Notice that these 100 units of the product reached the maximum age ($s = S = 2$) in time period 2, in which they are still usable to satisfy potential demand in period 2. These units of maximum age will still be held in inventory at the end of period 2, but they will be discarded in period 3 and hence do not appear in the inventory in time period 3. Similarly, the amount received in $t = 2$, which was of age 1 ($s = 1$), becomes of age 2 ($s = 2$) in time period 3, reaching the maximum age but still being usable to satisfy potential demand in period 3. This amount will be discarded in period 4 and will not be in the usable inventory from time period 4 onwards.

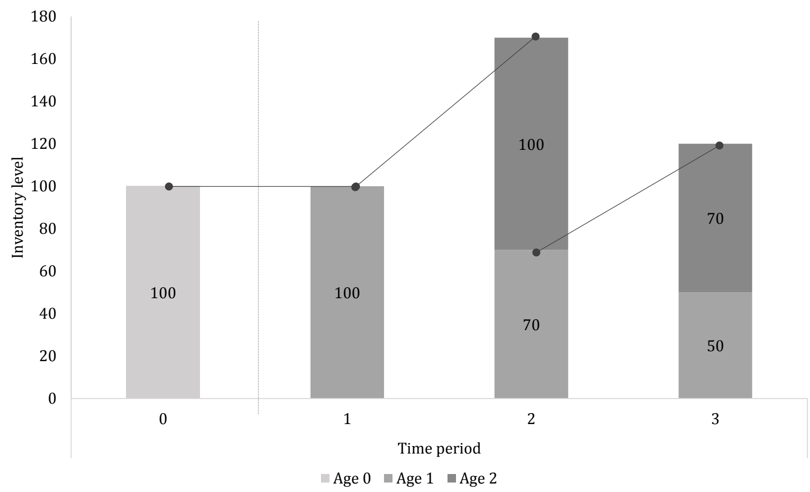

Figure 1: Aging process of the inventory at a customer in the PIRP

The PIRP consists of determining the time periods in which the customers will be visited; the quantity of product of each available age that will be delivered in every visit; the quantity of product of each available age that will be used to satisfy the demand; and the delivery routes to perform those visits. The objective is to maximize the total profit, given by the sales revenue minus the sum of inventory holding and routing costs. The holding costs are charged on the inventories at the end of each time period at both the supplier and customers. We consider that products of different ages share the same joint holding space at all facilities. It is also assumed that the supplier holding capacity is unbounded. In addition, according to the usual practice in the literature, we assume that the customers who receive a delivery in a given time period can use this to fulfill the demand in the same time period.

As in Coelho and Laporte (2014), we assume that the product that has reached its maximum age ($s = S$) at the end of a time period, will not go into the regular storage area, but will be kept separately in inventory to be discarded in the next period. Thus, these amounts incur the inventory holding costs but do not limit the quantity that the customer can receive in the next time period. Table 1 summarizes all the previously introduced notation as well as the one that will be used in the following sections.

Sets:	
$\mathcal{C}$	Set of customers
$\mathcal{N}$	Set of vertices/facilities
$\mathcal{E}$	Set of edges
$\mathcal{T}$	Set of time periods
$\mathcal{S}$	Set of ages of the product
$\mathcal{K}$	Set of vehicles
Indices:	
i, j, h	Vertices/facilities
s	Ages of the product
k	Vehicles
t, m, p	Time periods
Parameters:	
u_{is}	Unit revenue for product of age s at customer i
h_{is}	Unit inventory holding cost for product of age s at facility i
c_{ij}	Transportation cost between facilities i and j
d_i^t	Demand of customer i in time period t
r^t	Amount made available at the supplier in time period t
C_i	Storage capacity of customer i
I_{i0}^0	Initial inventory at facility i
Q	Capacity of the vehicles

Table 1: Sets, indices and parameters of the problem

3. Mathematical formulations

This section presents the mathematical formulations we introduce for the PIRP. First, we present a corrected version of the arc-based formulation introduced by Coelho and Laporte (2014) and show why there is an inconsistency in their formulation. Then, in the subsequent sections we present several reformulations of the problem.

3.1. Arc-based formulation

To formulate the PIRP using arc variables as in Coelho and Laporte (2014), consider the following notation. First, we introduce the set $\mathcal{S}^t = \{s \in \mathcal{S} \colon 1 \leq s \leq t\}$, which is the subset of product ages that can be available at all facilities in time period t. This set indicates the ages that can be delivered by the supplier in each time period and also specifies the ages that can be used to satisfy the demand of the customer in the given time period. Notice that this set does not contain age 0, which is also part of the ages set $\mathcal{S}$ and is available at the supplier in each time period, given that the supplier never delivers products of age 0 to the customers because the amount made available at the supplier facility in a certain period can only be delivered in the following period. Also, let $U_i = \min\{Q, C_i\}$ be an upper bound on the amount that can be delivered to customer i in time period t. Finally, consider the following decision variables:

$x_{ij}^{kt} \in \{0, 1, 2\}$: number of times vehicle $k \in \mathcal{K}$ traverses edge $(i, j) \in \mathcal{E}$ in time period $t \in \mathcal{T}$,

$y_i^{kt} \in \{0,1\}$: 1 if facility $i \in \mathcal{N}$ is visited by vehicle $k \in \mathcal{K}$ in period $t \in \mathcal{T}$, 0 otherwise;

$I_{is}^{t} \geq 0$: inventory of age $s \in \mathcal{S}$ at facility $i \in \mathcal{N}$ at the end of time period $t \in \mathcal{T}$;

$q_{is}^{kt} \geq 0$: quantity of product of age $s \in \mathcal{S}$ delivered to customer $i \in \mathcal{C}$ by vehicle $k \in \mathcal{K}$ in period $t \in \mathcal{T}$;

$d_{is}^{t} \geq 0$: quantity of product of age $s \in \mathcal{S}$ used to fulfill the demand of customer $i \in \mathcal{C}$ in period $t \in \mathcal{T}$.

Given these variables, the arc-based (AB) formulation of the problem can be stated as:

$$\max \quad \sum_{i\in\mathcal{C}}\sum_{t\in\mathcal{T}}\sum_{s\in\mathcal{S}^t} u_{is}d_{is}^t - \sum_{(i,j)\in\mathcal{E}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} c_{ij}x_{ij}^{kt} - \sum_{i\in\mathcal{N}}\sum_{t\in\mathcal{T}}\sum_{s\in\mathcal{S}^t} h_{is}I_{is}^t - \sum_{t\in\mathcal{T}} h_{00}r^t \tag{1}$$

$$\text{s.t.} \quad I_{0s}^t = r^t \qquad t\in\mathcal{T}, s=0, \tag{2}$$

$$I_{0s}^t = I_{0,s-1}^{t-1} - \sum_{i\in\mathcal{C}}\sum_{k\in\mathcal{K}} q_{is}^{kt} \qquad t\in\mathcal{T}, s\in\mathcal{S}^t, \tag{3}$$

$$I_{is}^t = I_{i,s-1}^{t-1} + \sum_{k\in\mathcal{K}} q_{is}^{kt} - d_{is}^t \qquad i\in\mathcal{C}, t\in\mathcal{T}, s\in\mathcal{S}^t, \tag{4}$$

$$d_i^t = \sum_{s\in\mathcal{S}^t} d_{is}^t \qquad i\in\mathcal{C}, t\in\mathcal{T}, \tag{5}$$

$$\sum_{s\in\mathcal{S}^{t-1}\setminus\{S\}} I_{is}^{t-1} + \sum_{k\in\mathcal{K}}\sum_{s\in\mathcal{S}^t} q_{is}^{kt} \leq C_i \qquad i\in\mathcal{C}, t\in\mathcal{T}, \tag{6}$$

$$\sum_{s\in\mathcal{S}^t} q_{is}^{kt} \leq U_i y_i^{kt} \qquad i\in\mathcal{C}, k\in\mathcal{K}, t\in\mathcal{T}, \tag{7}$$

$$\sum_{i\in\mathcal{C}}\sum_{s\in\mathcal{S}^t} q_{is}^{kt} \leq Q y_0^{kt} \qquad k\in\mathcal{K}, t\in\mathcal{T}, \tag{8}$$

$$\sum_{j\in\mathcal{N}:j<i} x_{ji}^{kt} + \sum_{j\in\mathcal{N}:j>i} x_{ij}^{kt} = 2y_i^{kt} \qquad i\in\mathcal{N}, k\in\mathcal{K}, t\in\mathcal{T}, \tag{9}$$

$$\sum_{i\in\mathcal{B}}\sum_{j\in\mathcal{B}:j>i} x_{ij}^{kt} \leq \sum_{i\in\mathcal{B}} y_i^{kt} - y_e^{kt} \qquad \forall\mathcal{B}\subseteq\mathcal{C}, |\mathcal{B}|\geq 2, k\in\mathcal{K}, t\in\mathcal{T}, e\in\mathcal{B}, \tag{10}$$

$$\sum_{k\in\mathcal{K}} y_i^{kt} \leq 1 \qquad i\in\mathcal{C}, t\in\mathcal{T}, \tag{11}$$

$$I_{is}^t \geq 0 \qquad i\in\mathcal{N}, t\in\mathcal{T}, s\in\mathcal{S}^t, \tag{12}$$

$$q_{is}^{kt} \geq 0 \qquad i\in\mathcal{C}, k\in\mathcal{K}, t\in\mathcal{T}, s\in\mathcal{S}^t, \tag{13}$$

$$d_{is}^t \geq 0 \qquad i\in\mathcal{C}, t\in\mathcal{T}, s\in\mathcal{S}^t, \tag{14}$$

$$y_i^{kt} \in \{0,1\} \qquad i\in\mathcal{N}, k\in\mathcal{K}, t\in\mathcal{T}, \tag{15}$$

$$x_{ij}^{kt} \in \{0,1\} \qquad (i,j)\in\mathcal{E}\colon i\neq 0, k\in\mathcal{K}, t\in\mathcal{T}, \tag{16}$$

$$x_{ij}^{kt} \in \{0,1,2\} \qquad (i,j)\in\mathcal{E}\colon i=0, k\in\mathcal{K}, t\in\mathcal{T}. \tag{17}$$

The objective function (1) consists of maximizing the total profit, given by the total revenue minus the sum of transportation and inventory holding costs. The last term of the objective function accounts for the inventory holding cost incurred by the amount of fresh product made available at the supplier in each time period of the planning horizon. This term can be ignored as it is a constant, but for the sake of completeness, we decided to keep it in the objective function. Constraints (2)–(3) define the inventory conservation at the supplier, where the first constraint set explicitly defines the inventory of age 0 in each time period and the second constraint set defines the inventory for ages in set $\mathcal{S}^t$. Constraints (4) define the inventory conservation at the customers. Constraints (5) guarantee the fulfillment of the customer demands, which can be done with products of different ages (but only with the ages that are available in the time period of the demand). Constraints (6) impose that the inventory level after delivery at the customer facilities cannot exceed their storage capacity. Notice that products of different ages share the same storage space. Also, note that products of age S still available at the end of time period $t-1$ will not enter into the storage space and hence do not limit the amount that can be delivered in period t. Constraints (7) permit a vehicle to perform a delivery to a specific customer only if this customer is visited by the vehicle. Constraints (8) guarantee that the capacity of each vehicle is respected. Constraints (9) ensure the flow conservation. Constraints (10) are subtour elimination constraints (SECs). Constraints (11) impose that each customer can be visited at most once in each time period. Finally, the domain of the decision variables is defined in constraints (12)–(17). Notice that when $i \neq 0$ and $j > i$, x_{ij}^{kt} can only take the values 0 or 1; if $i = 0$, then x_{ij}^{kt} can also be equal to 2, indicating that vehicle k makes a round trip between the depot and customer j in time period t.

This formulation has two main differences with respect to the one proposed by Coelho and Laporte (2014). First of all, in their formulation, sums over variables of different ages (as in constraints (5)–(8) and in the objective function) consider the whole set of ages $\mathcal{S}$, instead of the subset $\mathcal{S}^t$, which we introduce here. Second, the authors define inventory conservation constraints for products of age 0 for the customers (in the form $I_{i0}^t = \sum_{k\in\mathcal{K}} q_{i0}^{kt} - d_{i0}^t, \forall i \in \mathcal{C}, t \in \mathcal{T}$), although the supplier cannot deliver these products and the customers never receive products of age 0, according to the assumptions of the problem as defined by Coelho and Laporte (2014) via their supplier inventory constraints. These two differences can lead to an internal inconsistency in their formulation. More specifically, in the Coelho and Laporte (2014) formulation, the variable q_{i0}^{kt} is defined, but only appears in the inventory conservation constraint for products of age 0 at the customers. Furthermore, their demand fulfillment constraints enable the satisfaction of the demand using products of age 0. As a result, the solutions of their formulation can have consumptions (d_{is}^t) and deliveries (q_{is}^t) of products of age 0 without subtracting these amounts from the supplier's inventory. This can be beneficial in a solution because the products of age 0 have a high revenue in the instances proposed by those authors. Notice that if deliveries of products of age 0 were to be allowed, the term $\sum_{i\in\mathcal{C}} \sum_{k\in\mathcal{K}} q_{i0}^{kt}$ should be subtracted from the right-hand side of constraints (2) and the set $\mathcal{S}^t$ should include 0 as well.

3.2. Transportation formulation I

The first reformulation we propose uses decision variables that explicitly indicate the detailed use of the deliveries of each age, i.e., the time periods in which the delivery will cover all or part of the demand, as in the facility location formulation of the single item uncapacitated lot sizing problem, introduced by Krarup and Bilde (1977). For this, we introduce some additional notation. Let $T_s^t = \min\{T, t - s + S\}$ be the last time period in which product that is of age s in time period t can be used to satisfy any demand. We also consider an additional fictitious time period $T + 1$ in order to handle inventories at the end of the planning horizon. Consider the following decision variables:

$q_{is}^{ktm} \geq 0$: quantity of product of age $s \in \mathcal{S}$ delivered to customer $i \in \mathcal{C}$ by vehicle $k \in \mathcal{K}$

in period $t \in \mathcal{T}$ to cover the demand of period $m \in \{t, \ldots, T_s^t + 1\}$;

$b_i^t \geq 0$: amount of the initial inventory of customer $i \in \mathcal{C}$ used to fulfill its own demand in time period $t \in \mathcal{T}$.

Note that in the definition of the variable q_{is}^{ktm}, the index age (s) refers to the age of the product at the time of the delivery. Notice that when $m = T + 1$ in the delivery variables (q), it indicates that the quantity delivered will remain in the customer inventory at the end of the planning horizon. Also, when $m = (t - s + S) + 1$ it means that the product will spoil and will be discarded at the customer facility in period m.

Using the introduced notation and variables, the transportation formulation (TP-I) can be stated as follows:

$$\max \quad (1)$$

$$\text{s.t.} \quad I_{0s}^t = r^{t-s} - \sum_{i \in \mathcal{C}} \sum_{k \in \mathcal{K}} \sum_{t'=0}^{s-1} \sum_{m=t-t'}^{T_s^t+1} q_{i,s-t'}^{k,t-t',m} \qquad t \in \mathcal{T}, s \in \mathcal{S}^t, \tag{18}$$

$$I_{is}^t = \sum_{k \in \mathcal{K}} \sum_{t'=0}^{s-1} \sum_{m=t+1}^{T_s^t+1} q_{i,s-t'}^{k,t-t',m} \qquad i \in \mathcal{C}, t \in \mathcal{T}, s \in \mathcal{S}^t : s < t, \tag{19}$$

$$I_{is}^t = \sum_{k \in \mathcal{K}} \sum_{t'=0}^{s-1} \sum_{m=t+1}^{T_s^t+1} q_{i,s-t'}^{k,t-t',m} + I_{i0}^0 - \sum_{t'=1}^{t} b_i^{t'} \qquad i \in \mathcal{C}, t \in \mathcal{T} : t \leq S, s = t, \tag{20}$$

$$d_{is}^t = \sum_{k \in \mathcal{K}} \sum_{t'=0}^{s-1} q_{i,s-t'}^{k,t-t',t} \qquad i \in \mathcal{C}, t \in \mathcal{T}, s \in \mathcal{S}^t : s < t, \tag{21}$$

$$d_{is}^t = \sum_{k \in \mathcal{K}} \sum_{t'=0}^{s-1} q_{i,s-t'}^{k,t-t',t} + b_i^t \qquad i \in \mathcal{C}, t \in \mathcal{T} : t \leq S, s = t, \tag{22}$$

$$\sum_{s \in \mathcal{S}^{t-1} \setminus \{S\}} I_{is}^{t-1} + \sum_{k \in \mathcal{K}} \sum_{s \in \mathcal{S}^t} \sum_{m=t}^{T_s^t+1} q_{is}^{ktm} \leq C_i \qquad i \in \mathcal{C}, t \in \mathcal{T}, \tag{23}$$

$$\sum_{s \in \mathcal{S}^t} \sum_{m=t}^{T_s^t+1} q_{is}^{ktm} \leq U_i y_i^{kt} \qquad i \in \mathcal{C}, k \in \mathcal{K}, t \in \mathcal{T}, \tag{24}$$

$$\sum_{i \in \mathcal{C}} \sum_{s \in \mathcal{S}^t} \sum_{m=t}^{T_s^t+1} q_{is}^{ktm} \leq Q y_0^{kt} \qquad k \in \mathcal{K}, t \in \mathcal{T}, \tag{25}$$

$$q_{is}^{ktm} \geq 0 \qquad i \in \mathcal{C}, k \in \mathcal{K}, t \in \mathcal{T}, s \in \mathcal{S}^t, m \in \{t, \ldots, T_s^t + 1\}, \tag{26}$$

$$b_i^t \geq 0 \qquad i \in \mathcal{C}, t \in \mathcal{T} : t \leq S, \tag{27}$$

$$(5), (9) - (12) \text{ and } (14) - (17).$$

Constraints (18) define the inventory at the supplier for all time periods and available ages (where $r^0 = I_{00}^0$). Constraints (19)–(20) define the inventory at the customers for all the different ages of the product, with (20) including the amount from the initial inventory that is not used to satisfy any demand. These constraints can be easily generalized for the case when the initial inventory is composed of products of different ages. Constraints (21)–(22) define the amount of product of each different age used to fulfill the demand of the customers. Notice that constraints (22) include the demand that can be fulfilled using the initial inventory as well. Constraints (23) impose that the inventory level after delivery at the customers cannot exceed their storage capacity. Constraints (24) allow a vehicle to perform a delivery to a specific customer only if this customer is visited by the vehicle. Constraints (25) guarantee that the capacity of each vehicle is respected. Finally, constraints (26)–(27) define the domain of the new decision variables.

3.3. Formulation TP-I without a vehicle index

The previous formulation can be reformulated by dropping the vehicle index of the variables, as we consider a homogeneous vehicle fleet (in both capacity and travel cost terms) and assume at most one visit to each customer in each time period. All the variables maintain the same meaning, except for y_0^t which becomes an integer variable indicating the number of vehicles used in period $t \in \mathcal{T}$. The formulation, which we will refer to as TP-I-nk, can be stated as:

$$\max \quad \sum_{i\in\mathcal{C}}\sum_{t\in\mathcal{T}}\sum_{s\in\mathcal{S}^t} u_{is}d_{is}^t - \sum_{(i,j)\in\mathcal{E}}\sum_{t\in\mathcal{T}} c_{ij}x_{ij}^t - \sum_{i\in\mathcal{N}}\sum_{t\in\mathcal{T}}\sum_{s\in\mathcal{S}^t} h_{is}I_{is}^t - \sum_{t\in\mathcal{T}} h_{00}r^t \tag{28}$$

$$\text{s.t.} \quad I_{0s}^t = r^{t-s} - \sum_{i\in\mathcal{C}}\sum_{t'=0}^{s-1}\sum_{m=t-t'}^{T_s^t+1} q_{i,s-t'}^{t-t',m} \qquad t\in\mathcal{T}, s\in\mathcal{S}^t, \tag{29}$$

$$I_{is}^t = \sum_{t'=0}^{s-1}\sum_{m=t+1}^{T_s^t+1} q_{i,s-t'}^{t-t',m} \qquad i\in\mathcal{C}, t\in\mathcal{T}, s\in\mathcal{S}^t : s<t, \tag{30}$$

$$I_{is}^t = \sum_{t'=0}^{s-1}\sum_{m=t+1}^{T_s^t+1} q_{i,s-t'}^{t-t',m} + I_{i0}^0 - \sum_{t'=1}^{t} b_i^{t'} \qquad i\in\mathcal{C}, t\in\mathcal{T} : t\le S, s=t, \tag{31}$$

$$d_{is}^t = \sum_{t'=0}^{s-1} q_{i,s-t'}^{t-t',t} \qquad i\in\mathcal{C}, t\in\mathcal{T}, s\in\mathcal{S}^t : s<t, \tag{32}$$

$$d_{is}^t = \sum_{t'=0}^{s-1} q_{i,s-t'}^{t-t',t} + b_i^t \qquad i\in\mathcal{C}, t\in\mathcal{T} : t\le S, s=t, \tag{33}$$

$$\sum_{s\in\mathcal{S}^{t-1}\setminus\{S\}} I_{is}^{t-1} + \sum_{s\in\mathcal{S}^t}\sum_{m=t}^{T_s^t+1} q_{is}^{tm} \le C_i \qquad i\in\mathcal{C}, t\in\mathcal{T}, \tag{34}$$

$$\sum_{s\in\mathcal{S}^t}\sum_{m=t}^{T_s^t+1} q_{is}^{tm} \le U_i y_i^t \qquad i\in\mathcal{C}, t\in\mathcal{T}, \tag{35}$$

$$\sum_{j\in\mathcal{N}:j<i} x_{ji}^t + \sum_{j\in\mathcal{N}:j>i} x_{ij}^t = 2y_i^t \qquad i\in\mathcal{N}, t\in\mathcal{T}, \tag{36}$$

$$Q\sum_{i\in\mathcal{B}}\sum_{j\in\mathcal{B}:j>i} x_{ij}^t \le \sum_{i\in\mathcal{B}}(Qy_i^t - \sum_{s\in\mathcal{S}^t}\sum_{m=t}^{T_s^t+1} q_{is}^{tm}) \qquad \forall\mathcal{B}\subseteq\mathcal{C}, |\mathcal{B}|\ge 2, t\in\mathcal{T}, \tag{37}$$

$$y_0^t \le K \qquad t\in\mathcal{T}, \tag{38}$$

$$q_{is}^{tm} \ge 0 \qquad i\in\mathcal{C}, t\in\mathcal{T}, s\in\mathcal{S}^t, m\in\{t,\dots,T_s^t+1\}, \tag{39}$$

$$y_0^t \in \mathbb{Z}^+ \qquad t\in\mathcal{T}, \tag{40}$$

$$y_i^t \in \{0,1\} \qquad i\in\mathcal{C}, t\in\mathcal{T}, \tag{41}$$

$$x_{ij}^t \in \{0,1\} \qquad (i,j) \in \mathcal{E}\colon i \neq 0, t \in \mathcal{T}, \quad (42)$$

$$x_{ij}^t \in \{0,1,2\} \qquad (i,j) \in \mathcal{E}\colon i = 0, t \in \mathcal{T}, \quad (43)$$

$(5), (12), (14)$ and (27).

The objective function (28) consists of maximizing the total profit. Constraints (29) and (30)–(31) define the inventory level for the different ages of the product at the supplier and customers, respectively, where $r^0 = I_{00}^0$ in constraints (29). Constraints (32)–(33) define the amount of each different age used to fulfill the demand of the customers. Constraints (34) impose the maximum storage capacity at the customers. Constraints (35) allow to perform deliveries to a specific customer only if it is visited by a vehicle. Constraints (36) ensure the conservation of the flow. Constraints (37) are SECs and ensure that the vehicle capacities are respected as well. Constraints (38) limit the number of vehicles that can be used in each time period. Constraints (39)–(43) define the domain of the decision variables.

Notice that, as pointed out by Adulyasak et al. (2014), if one divides the inequalities (37) by Q, they have a form similar to the generalized fractional SECs (GFSECs) for the vehicle routing problem (VRP) (Toth and Vigo, 2002). However, GFSECs in the form (37) are numerically more stable than the original GFSECs, which contain a fractional right-hand side.

3.4. Transportation formulation II

This reformulation, similar to TP-I, uses decision variables that explicitly indicate the detailed use of the deliveries of each age. However, in this case we consider implicitly the age of the product being delivered. Thus, the delivery variable is defined as follows:

$q_i^{ktpm} \geq 0$: amount of product that was made available at the supplier in time period $t \in \{0\} \cup \mathcal{T}$ and was delivered to customer $i \in \mathcal{C}$ by vehicle $k \in \mathcal{K}$ in period $p \in \mathcal{T}$ to cover the demand of period $m \in \{p, \ldots, T_0^t + 1\}$.

Note that in the definition of the variable q_i^{ktpm}, the age of the product at delivery and consumption periods is given by the difference between index t and indices p and m, respectively. Notice that when $t = 0$, the amount delivered comes from the initial inventory of the supplier. Notice also that, analogously to formulation TP-I, when $m = T + 1$ in the delivery variables (q) the quantity delivered will remain in the customer inventory at the end of the planning horizon and when $m = t + S + 1$ the product delivered will spoil and be discarded at the customer facility in period m. The formulation, which we will refer to as TP-II, can be stated as:

$$\max \quad (1)$$

$$\text{s.t.} \quad I_{0s}^t = r^{t-s} - \sum_{i\in\mathcal{C}} \sum_{k\in\mathcal{K}} \sum_{p=t-s+1}^{t} \sum_{m=p}^{T_s^t+1} q_i^{k,t-s,p,m} \qquad t \in \mathcal{T}, s \in \mathcal{S}^t, \quad (44)$$

$$I_{is}^t = \sum_{k\in\mathcal{K}} \sum_{p=t-s+1}^{t} \sum_{m=t+1}^{T_s^t+1} q_i^{k,t-s,p,m} \qquad i \in \mathcal{C}, t \in \mathcal{T}, s \in \mathcal{S}^t : s < t, \quad (45)$$

$$I_{is}^t = \sum_{k\in\mathcal{K}} \sum_{p=1}^{t} \sum_{m=t+1}^{T_s^t+1} q_i^{k0pm} + I_{i0}^0 - \sum_{t'=1}^{t} b_i^{t'} \qquad i \in \mathcal{C}, t \in \mathcal{T} : t \leq S, s = t, \quad (46)$$

$$d_{is}^t = \sum_{k\in\mathcal{K}} \sum_{p=t-s+1}^{t} q_i^{k,t-s,p,t} \qquad i\in\mathcal{C}, t\in\mathcal{T}, s\in\mathcal{S}^t : s<t, \quad (47)$$

$$d_{is}^t = \sum_{k\in\mathcal{K}} \sum_{p=1}^{t} q_i^{k0pt} + b_i^t \qquad i\in\mathcal{C}, t\in\mathcal{T} : t\leq S, s=t, \quad (48)$$

$$\sum_{s\in\mathcal{S}^{t-1}\setminus\{S\}} I_{is}^{t-1} + \sum_{k\in\mathcal{K}} \sum_{s\in\mathcal{S}^t} \sum_{m=t}^{T_s^t+1} q_i^{k,t-s,tm} \leq C_i \qquad i\in\mathcal{C}, t\in\mathcal{T}, \quad (49)$$

$$\sum_{s\in\mathcal{S}^t} \sum_{m=t}^{T_s^t+1} q_i^{k,t-s,tm} \leq U_i y_i^{kt} \qquad i\in\mathcal{C}, k\in\mathcal{K}, t\in\mathcal{T}, \quad (50)$$

$$\sum_{i\in\mathcal{C}} \sum_{s\in\mathcal{S}^t} \sum_{m=t}^{T_s^t+1} q_i^{k,t-s,tm} \leq Q y_0^{kt} \qquad k\in\mathcal{K}, t\in\mathcal{T}, \quad (51)$$

$$q_i^{ktpm} \geq 0 \qquad i\in\mathcal{C}, k\in\mathcal{K}, t\in\{0\}\cup\mathcal{T}\setminus\{T\}, p\in\{t+1,\ldots,T_0^t\}, m\in\{p,\ldots,T_0^t+1\}, \quad (52)$$

$(5), (9)-(12), (14)-(17)$ and (27).

Constraints (44) calculate the inventory at the supplier for the available ages, where $r^0 = I_{00}^0$. Constraints (45)–(46) define the inventory at the customers for all the different ages of the product. Constraints (47)–(48) state that the demand of the customers can be satisfied using products of all the available ages. Notice that constraints (48) include the demand that can be fulfilled using the initial inventory as well. Constraints (49) impose the maximum storage capacity after delivery at the customer facilities. Constraints (50) allow deliveries to a customer by a specific vehicle only if it is visited by the same vehicle. Constraints (51) guarantee that the capacity of each vehicle is respected. Finally, constraints (52) define the domain of the new decision variable.

3.5. Formulation TP-II without a vehicle index

As in Section 3.3, an additional formulation can be obtained by dropping the vehicle index of the variables for cases in which the vehicle fleet is considered to be homogeneous and at most a single visit is allowed to each customer in each time period. This formulation (TP-II-nk) can be stated as:

$$\max \quad (28)$$

$$\text{s.t.} \quad I_{0s}^t = r^{t-s} - \sum_{i\in\mathcal{C}} \sum_{p=t-s+1}^{t} \sum_{m=p}^{T_s^t+1} q_i^{t-s,p,m} \qquad t\in\mathcal{T}, s\in\mathcal{S}^t, \quad (53)$$

$$I_{is}^t = \sum_{p=t-s+1}^{t} \sum_{m=t+1}^{T_s^t+1} q_i^{t-s,p,m} \qquad i\in\mathcal{C}, t\in\mathcal{T}, s\in\mathcal{S}^t : s<t, \quad (54)$$

$$I_{is}^t = \sum_{p=1}^{t} \sum_{m=t+1}^{T_s^t+1} q_i^{0pm} + I_{i0}^0 - \sum_{t'=1}^{t} b_i^{t'} \qquad i\in\mathcal{C}, t\in\mathcal{T} : t\leq S, s=t, \quad (55)$$

$$d_{is}^t = \sum_{p=t-s+1}^{t} q_i^{t-s,p,t} \qquad i\in\mathcal{C}, t\in\mathcal{T}, s\in\mathcal{S}^t : s<t, \quad (56)$$

$$d_{is}^t = \sum_{p=1}^{t} q_i^{0pt} + b_i^t \qquad i\in\mathcal{C}, t\in\mathcal{T} : t\leq S, s=t, \quad (57)$$

$$\sum_{s \in \mathcal{S}^{t-1} \setminus \{S\}} I_{is}^{t-1} + \sum_{s \in \mathcal{S}^t} \sum_{m=t}^{T_s^t+1} q_i^{t-s,tm} \leq C_i \qquad i \in \mathcal{C}, t \in \mathcal{T}, \tag{58}$$

$$\sum_{s \in \mathcal{S}^t} \sum_{m=t}^{T_s^t+1} q_i^{t-s,tm} \leq U_i y_i^t \qquad i \in \mathcal{C}, t \in \mathcal{T}, \tag{59}$$

$$Q \sum_{i \in \mathcal{B}} \sum_{j \in \mathcal{B}: j > i} x_{ij}^t \leq \sum_{i \in \mathcal{B}} (Q y_i^t - \sum_{s \in \mathcal{S}^t} \sum_{m=t}^{T_s^t+1} q_i^{t-s,tm}) \qquad \forall \mathcal{B} \subseteq \mathcal{C}, |\mathcal{B}| \geq 2, t \in \mathcal{T}, \tag{60}$$

$$q_i^{tpm} \geq 0 \qquad i \in \mathcal{C}, t \in \{0\} \cup \mathcal{T} \setminus \{T\}, p \in \{t+1, \ldots, T_0^t\}, m \in \{p, \ldots, T_0^t + 1\}, \tag{61}$$

$$(5), (12), (14), (27), (36), (38) \text{ and } (40) - (43).$$

Constraints (53) and (54)–(55) define the inventory levels of the available ages of the product at the supplier and customers, respectively, where $r^0 = I_{00}^0$ in constraints (53). Constraints (56)–(57) specify that the demand of the customers can be satisfied using products of all the available ages. Constraints (58) enforce the maximum storage capacity at the customer facilities. Constraints (59) allow deliveries to customers in a given time period if they are visited by a vehicle in the same time period. Constraints (60) are the GFSECs and guarantee that the capacity of each vehicle is respected as well. Finally, constraints (61) define the domain of the new decision variable.

3.6. Branch-and-cut algorithms

Given that all the presented formulations contain an exponentially large number of SECs, we must apply a B&C algorithm to solve them. These constraints are dropped from the formulations and added in an iterative fashion every time they are violated at the nodes of the branch-and-bound (B&B) tree. In this section we provide the details of our B&C approaches for both the formulations with and without a vehicle index as well as further improvements. Additionally, in the supplementary material of this paper, we show some valid inequalities that we used to further strengthen the formulations and provide detailed results of all the formulations when including them.

3.6.1. Branch-and-cut for the vehicle index formulations

To solve the formulations with a vehicle index, we use an exact separation algorithm that solves a series of minimum $s-t$ cut problems to detect violated SECs for each vehicle in each time period of the planning horizon. At a given node of the B&B tree, let $\bar{y}_i^{kt}$ and $\bar{x}_{ij}^{kt}$ denote the values of the visit (y) and flow variables (x) of the solution, respectively. A graph for vehicle k in time period t is constructed from the set of nodes where $\bar{y}_i^{kt} > 0$, setting the weights of the graph edges to $\bar{x}_{ij}^{kt}$, $\forall (i,j) \in \mathcal{E}$. Then, for each customer node of the constructed graph, we solve a minimum $s-t$ cut problem, setting the supplier node as the source node (s) and the customer node as the sink node (t). A violated SEC is identified if the capacity of the minimum cut is less than $2\bar{y}_i^{kt}$ (Adulyasak et al., 2014). If a subtour on a set of nodes $\mathcal{B} \subseteq \mathcal{C}$ is found for vehicle k in period t, we add constraints (10) with $e = \arg\max_{i \in \mathcal{B}}\{\bar{y}_i^{kt}\}$ to the formulation, for all vehicles and time periods of the planning horizon. To solve the minimum $s-t$ cut problem, we used the Concorde solver (Applegate et al., 2018).

These SECs are separated only at the root node and then every time an integer solution is found at a node of the B&B tree, to avoid generating too many cuts in the tree. Notice that constraints (10) can be added to the formulation in many different ways, among which we tested: adding the cut only for the specific vehicle and time period for which it was violated; adding the cut for all vehicles in the same time period in which the violated cut was identified; and, finally, adding the cut for all vehicles and time periods. The latter strategy resulted in a slightly better performance. Regarding the selection of the customer $e \in \mathcal{B}$ for which the cut would be set, we tried including the cut only for the customer e such that $e = \arg\max_{i \in \mathcal{B}}\{\bar{y}_i^{kt}\}$ and, for every customer in the identified subset of customers $\mathcal{B}$. In this case, the former strategy resulted in a better performance of the B&C algorithms.

3.6.2. Branch-and-cut for the formulations without a vehicle index

To separate GFSECs (37) and (60) of formulations TP-I-nk and TP-II-nk, respectively, we use the separation package developed by Lysgaard et al. (2004) for the VRP, as in Adulyasak et al. (2014). This algorithm consists of four heuristic algorithms, which are applied sequentially. One of these heuristics is an exact separation algorithm when all the flow variables (x) take integer values.

For a given solution, we call the algorithm for each time period $t \in \mathcal{T}$. At a given node of the B&B tree, let $\bar{y}_i^{kt}$, $\bar{x}_{ij}^{kt}$ and $\bar{q}_{is}^{tm}$ ($\bar{q}_i^{tpm}$) denote the values of variables y_i^{kt}, x_{ij}^{kt} and q_{is}^{tm} (q_i^{tpm}) for the formulation TP-I-nk (TP-II-nk). The input required by the separation package (a VRP solution) in period t is constructed considering customer nodes with $\bar{y}_i^{kt} > 0$, setting the weight of each edge (i, j) to $\bar{x}_{ij}^{kt}$ and setting the delivery quantity for customer i to $\sum_{s \in \mathcal{S}^t} \sum_{m=t}^{T_s^t+1} q_{is}^{tm}$ and to $\sum_{s \in \mathcal{S}^t} \sum_{m=t}^{T_s^t+1} q_i^{t-s,tm}$ for the formulations TP-I-nk and TP-II-nk, respectively.

Similar to the formulations with a vehicle index, we separate GFSECs only at the root node and then whenever an integer solution is found at a given node of the B&B tree to avoid generating too many cuts. Every time we identify a violated cut, we add the corresponding constraint for all time periods. In addition, to further strengthen formulations TP-I-nk and TP-II-nk, we included the following SECs, as used in the formulations with a vehicle index (AB, TP-I, TP-II):

$$\sum_{i \in \mathcal{B}} \sum_{j \in \mathcal{B}: j > i} x_{ij}^t \leq \sum_{i \in \mathcal{B}} y_i^k - y_e^k \qquad \forall \mathcal{B} \subseteq \mathcal{C}, |\mathcal{B}| \geq 2, k \in \mathcal{K}, e \in \mathcal{B}. \tag{62}$$

Using these constraints together with GFSECs resulted in an improved performance by the formulations without a vehicle index. These last SECs are separated as described in Section 3.6.1.

4. Optimization-based iterated local search

In this section, we present the hybrid algorithm that we propose to solve the PIRP. This algorithm is based on the ILS metaheuristic for the basic variant of the IRP presented by Alvarez et al. (2018). Additionally, we include two mathematical programming components within its structure. The basic idea of ILS is to iteratively apply a local search algorithm to solutions resulting from the perturbation of the previously visited local optima, which leads to a randomized walk in the space of local optimal solutions (Lourenço et al., 2003). In the proposed method, the various decisions of the problem are handled by different components. On the one hand, routing decisions (x) are managed by the local search phase of the method while the visit variables (y) are mostly handled in the perturbation phase. On the other hand, a multicommodity flow (MCF) problem formulation is used to determine the optimal values of the continuous variables (q, d and I) for a given set of visit variables (y). Additionally, a mixed-integer programming (MIP) formulation that can remove and insert customers from a solution given as input is used as a solution improvement (SI) step in the final phase of the method. An overview of the proposed method is shown in Algorithm 1.

Algorithm 1: Optimization-based iterated local search

1 **begin**
2     $O^0 \leftarrow$ `construction_heuristic()`;
3   **if** $O^0 \neq \emptyset$ **then**
4         $O^* \leftarrow$ `rvnd_heuristic`(O^0);
5   **while** *stop criterion is not met* **do**
6             $O' \leftarrow$ `perturbation`(O^*);
7             $O' \leftarrow$ `rvnd_heuristic`(O');
8             $O' \leftarrow$ `optimize_amounts`(O');
9   **if** $f(O') > f(O^*)$ **then** $O^* \leftarrow O'$;
10   **end**
11         $O^* \leftarrow$ `SI_formulation`(O^*);
12   **end**
13 **end**

The algorithm starts with an initial feasible solution (line 2), which is generated using the construction heuristic that will be described in Section 4.1. If the construction heuristic cannot find a feasible solution, the algorithm stops; otherwise, the search process continues. A randomized variable neighborhood descent (RVND) heuristic is used as local search algorithm (lines 4 and 7), and a multi-operator algorithm is used as a perturbation mechanism (line 6). The continuous variables (deliveries, consumptions and inventories) of the solution are then optimized by solving a MCF formulation (line 8). The acceptance criterion admits the resulting solution only if it is better than the current best solution (line 9). Finally, after reaching a stopping criterion, the method applies the SI formulation (line 11). All components of the hybrid method are described in detail in the following sections.

In the description of the hybrid method, we use the subsequent notation. Given a solution O, we denote by $\bar{I}^t_{is}$, $\bar{q}^{kt}_{is}$, $\bar{d}^t_{is}$ and $\bar{y}^{kt}_i$ the values of its inventory, delivery, consumption and visit variables, respectively. In addition,

- $\mathcal{R}(O)$ is the set of all vehicle routes of the solution;
- $\mathcal{C}^t(O) = \{i \in \mathcal{C} \colon \sum_{k \in \mathcal{K}} \bar{y}^{kt}_i = 1\}$ is the set of customers visited by routes of the solution in time period t;
- $\mathcal{T}_i(O) = \{t \in \mathcal{T} \colon \sum_{k \in \mathcal{K}} \bar{y}^{kt}_i = 1\}$ is the set of time periods in which customer i is visited by routes of the solution.

Also, given a route $r \in \mathcal{R}(O)$ of the solution,

- $t(r)$ is the time period of the route; and
- $\mathcal{V}(r)$ is the set of customers visited by the route.

4.1. A construction heuristic for the PIRP

To obtain feasible solutions, we devised a decomposition construction heuristic which iteratively separates the decisions of the problem into two phases. In the first phase, the heuristic defines the size of the potential delivery to each customer and assigns a priority to each one of them. Then, in the second phase, feasible delivery routes are designed to deliver the amounts set in the first phase.

The heuristic starts by using the initial inventory of each customer to satisfy the maximum number of demands. The values for the respective consumption variables (d_{is}^t) are set as follows:

$$\bar{d}_{is}^t = \begin{cases} \min\{\bar{I}_i^{t-1}, d_i^t\} & \text{if } t \le S, s = t, \\ 0 & \text{otherwise}, \end{cases} \qquad \forall i \in \mathcal{C}, t \in \mathcal{T}, s \le S^t. \tag{63}$$

Then, the heuristic computes the aggregated inventory levels I_i^t for each customer $i \in \mathcal{C}$ at the end of each time period $t \in \mathcal{T}$ given the initial consumptions, as follows:

$$I_i^t = \begin{cases} I_{i0}^0 - \sum_{p=1}^{t} \bar{d}_{ip}^p & \text{if } t < S, \\ 0 & \text{otherwise}, \end{cases} \qquad \forall i \in \mathcal{C}, t \in \mathcal{T}. \tag{64}$$

These inventory levels will be updated at the end of each iteration based on the deliveries and consumptions. They are used to determine the delivery sizes in each period given that the amount a customer can receive is bounded by the holding capacity and the inventory level at the end of the previous time period. Notice that, initially, $I_i^t = 0$ for $t = S$ since the initial inventory will spoil at the end of this time period and, as stated in Section 2, the spoiled inventory does not limit the amount that the customer can receive in the next period.

Using these values, the heuristic performs one iteration for each time period $t \in \mathcal{T}$, starting from $t = 1$. In the first phase of iteration t, the heuristic sets a potential delivery quantity to each customer by computing the difference between its capacity and aggregated inventory level in the previous time period, also respecting the vehicle capacity. To simplify the heuristic, we apply a greedy approach in which all the deliveries are of the freshest possible product, which in our case is product of age $s = 1$. Therefore, the potential delivery ($\tilde{q}$) to each customer i is set as:

$$\tilde{q}_{i1}^t = \min\{\texttt{ratio\_demand} \times (C_i - I_i^{t-1}), Q\}, \tag{65}$$

where `ratio_demand` $\in (0, 1]$ is a parameter that defines the proportion of the maximum possible quantity that will be actually delivered. Next, the priority π_i of customer i is set as the number of upcoming `look_ahead` periods (including t) in which its demand is not fully covered yet, i.e., π_i is the number of times in which $\sum_{s \in \mathcal{S}^p} \bar{d}_{is}^p < d_i^p$, for $p = t, \ldots, \min\{T, t + \texttt{look\_ahead}\}$, $\forall i \in \mathcal{C}$. The value of `look_ahead` determines how much to look forward in the planning horizon, trying to anticipate forthcoming stockouts.

After defining these deliveries and priorities, the second phase of iteration t starts. It consists of determining one or more vehicle routes using a nearest-neighbor insertion heuristic that first routes customers with higher priority as long as the insertion satisfies the vehicle capacity. At most K routes can be defined in this phase. Then, given the deliveries actually performed, we set the values of $\bar{q}_{i1}^{kt}$ and update the values of the consumption (using the first-in first-consumed (FIFC) rule) and inventory variables. Finally, a new iteration is started for the next period ($t + 1$), until reaching time period T.

A pseudo-code of the heuristic is given in Algorithm 2. Since the heuristic runs in a short time, it was defined inside two outer loops, exploring different values for `ratio_demand` and `look_ahead`, aiming at finding a reasonably good feasible solution, as in Alvarez et al. (2018). Furthermore, at the end of the execution of the heuristic, the continuous variables of the best feasible solution found (if any) are optimized using the MCF formulation of Section 4.4. Finally, as will be shown in Section 5, this heuristic was able to find feasible solutions for all the benchmark instances used in this paper.

Algorithm 2: Construction heuristic for the PIRP

begin
  $O^* \leftarrow \emptyset$;
  Use initial inventory to set as many consumptions ($\bar{d}$) as possible, using (63);
  Compute aggregated inventory levels I_i^t as in (64), for all $i \in \mathcal{C}$ and $t \in \mathcal{T}$;
  `ratio_demand` $\leftarrow 1.0$;
  while *ratio_demand* > 0 **do**
    `look_ahead` $\leftarrow 0$;
    while *look_ahead* $\leq S$ **do**
      for $t \in \mathcal{T}$ **do**
        for $i \in \mathcal{C}$ **do**
          $\tilde{q}_{i1}^t \leftarrow \min\{\texttt{ratio\_demand} \times (C_i - I_i^{t-1}), Q\}$;
          $\pi_i \leftarrow 0$;
          for $p = t, \ldots, \min\{T, t + \textit{look\_ahead}\}$ **do**
            if $\sum_{s \in \mathcal{S}^p} \bar{d}_{is}^p < d_i^p$ **then** $\pi_i \leftarrow \pi_i + 1$;
          end
        end
        Apply a nearest-neighbor insertion heuristic, routing customers with higher π_i first;
        For all routed customers, set the corresponding $\bar{q}_{i1}^{kt}$ values and compute the corresponding $\bar{d}_{is}^t$ values using the FIFC rule and update $\bar{I}_{is}^t$ and I_i^t;
      end
      Update best feasible solution O^*;
      `look_ahead` $\leftarrow$ `look_ahead` $+ 1$;
    end
    `ratio_demand` $\leftarrow$ `ratio_demand` $- 0.1$;
  end
  if $O^* \neq \emptyset$ **then** $O^* \leftarrow$ `optimize_amounts`(O^*) ;
end

4.2. Randomized variable neighborhood descent heuristic

For the local search procedure of the proposed method, we use a variable neighborhood descent heuristic (Mladenović and Hansen, 1997) with random neighborhood ordering. In this algorithm, local search operators are selected randomly from a predefined set and applied to the incumbent solution until none of them can improve it (Subramanian et al., 2012; Alvarez and Munari, 2017; Alvarez et al., 2018). The randomized behavior of RVND enhances the diversification of the method because different local search operators can provide distinct local optimal solutions. Therefore, a different final solution can be obtained every time RVND is applied over the same starting solution. Moreover, the randomized order leads to a more balanced exploration of the neighborhoods, given that when a fixed sequential order is adopted most of the effort is spent on the first operators (Deng and Bard, 2011). In our implementation, a set containing several local search operators is initialized at the beginning of the procedure. Then, while the set is not empty, an operator is chosen at random and applied to the incumbent solution. If the operator improves the solution, the set is re-established to its initial configuration (containing all the local search operators). Otherwise, the operator is removed from the set and the process continues with the remaining operators. A pseudo-code of the RVND heuristic is shown in Algorithm 3.

Algorithm 3: Randomized variable neighborhood descent heuristic

begin
    $O^* \leftarrow O^0$ (save initial solution);
    $\mathcal{L} \leftarrow$ initialize the set of local search operators;
    while $|\mathcal{L}| > 0$ **do**
        $l \leftarrow$ select at random a local search operator from $\mathcal{L}$;
        Apply l to O^*;
        if l *improved* O^* **then**
            $\mathcal{L} \leftarrow$ reinitialize the set of local search operators;
        else
            remove l from $\mathcal{L}$;
        end
    end
end

In our method, we use the local search phase to handle the routing decisions of the solution. For this, the RVND heuristic uses the following classical VRP operators: Or-opt-k, $k \in \{1, 2, 3\}$; Shift(k), $k \in \{1, 2, 3\}$; Swap(k_1, k_2), $k_1, k_2 \in \{1, 2\}, k_1 \geq k_2$; and k-opt, $k \in \{2, 3\}$. All operators explore the search space using the first improvement strategy, allowing only feasible solutions in the search process.

4.3. Perturbation mechanism

Since in the PIRP there are different decisions that must be made simultaneously, we designed a perturbation algorithm that can change multiple attributes of a solution in a single call. The algorithm uses the following operators to modify the visit and delivery decisions of an input solution O.

1. Insert visits: choose randomly a route $r \in \mathcal{R}(O)$ and customer i such that $i \notin \mathcal{C}^{t(r)}(O)$. The customer is inserted into the cheapest insertion position in the route. Then the values of $\bar{q}$, $\bar{d}$ and $\bar{I}$ are re-optimized using the MCF formulation;
2. Remove visits: choose a random route $r \in \mathcal{R}(O)$ and a customer $i \in \mathcal{V}(r)$ and then remove i from r. After that, the values of $\bar{q}$, $\bar{d}$ and $\bar{I}$ are re-optimized using the MCF formulation;
3. Move visit: choose a random route $r \in \mathcal{R}(O)$ and a customer $i \in \mathcal{V}(r)$ such that $|\mathcal{T}_i(O)| < T$, i.e., a customer that is not visited in every time period of the planning horizon. Then, the visit to i is removed from r and inserted into the cheapest position of a route of a period $p \in \mathcal{T} \setminus \mathcal{T}_i(O)$, choosing both, p and the route, at random. Finally, the values of $\bar{q}$, $\bar{d}$ and $\bar{I}$ are re-optimized using the MCF formulation;
4. Reduce deliveries: choose a random route $r \in \mathcal{R}(O)$ and a delivery (of a certain age s) to a customer $i \in \mathcal{V}(r)$ such that the amount delivered is not completely consumed by the customer. This can happen, for instance, when it is profitable to accumulate inventory at the customer to save holding costs at the supplier. Then, the delivery is reduced by the amount not consumed by the customer. Both the customer and the delivery to be reduced are chosen at random.

After applying each operator, the objective function value of the solution is recomputed. The aim of these operators is twofold. First, helping to determine the periods in which each customer must be visited and, second, creating slack in the routes for the local search heuristic. In the Remove and Move operators, infeasible solutions are rejected. In such a case, the operator chooses another customer of the same route. If all customers of the chosen route are unsuccessfully explored (resulting in infeasible solutions), the operator chooses another route and the process is applied in the same fashion.

Note that the performance of an ILS-based algorithm is strongly related to the strength of its perturbation mechanism given that it defines much of the behavior of the method. This mechanism must be able to diversify the search process without turning it into a randomized restart search. For this purpose, we use the parameter `max_perturb`, which defines the maximum number of elements of the solution that can be changed each time the perturbation mechanism is called. Thus, similar to the RVND heuristic, our perturbation algorithm can use multiple operators in a single call, applying one operator at a time (changing at most one element of the solution) until either the number of changes performed to the solution reaches `max_perturb` or none of the operators can change the solution.

4.4. Multicommodity flow (MCF) formulation

Given the values of $\bar{y}$ from a solution O, one can determine the optimal values for the delivery, consumption and inventory variables that maximize the total profit by solving the following MCF problem formulation:

$$\max \quad \sum_{i\in\mathcal{C}}\sum_{t\in\mathcal{T}}\sum_{s\in\mathcal{S}^t} u_{is}d_{is}^t - \sum_{i\in\mathcal{N}}\sum_{t\in\mathcal{T}}\sum_{s\in\mathcal{S}^t} h_{is}I_{is}^t \tag{66}$$

$$\text{s.t.} \quad \sum_{s\in\mathcal{S}^t} q_{is}^{kt} \le U_i\bar{y}_i^{kt} \qquad i\in\mathcal{C}, k\in\mathcal{K}, t\in\mathcal{T}, \tag{67}$$

$$\sum_{i\in\mathcal{C}}\sum_{s\in\mathcal{S}^t} q_{is}^{kt} \le Q\bar{y}_0^{kt} \qquad k\in\mathcal{K}, t\in\mathcal{T}, \tag{68}$$

$$(2) - (6) \text{ and } (12) - (14),$$

where $\bar{y}_0^{kt} = 1$ indicates that vehicle k is used in time period t and $\bar{y}_i^{kt} = 1$ indicates that vehicle k visits customer i in time period t. The objective function (66) consists of maximizing the total profit, given by the total revenue minus the total inventory cost. Constraints (67) allow a vehicle to perform a delivery to a specific customer in a given time period only if the customer is visited by the vehicle in that time period in the solution O. Finally, constraints (68) impose the vehicle capacity. This is a linear program (LP) that can be solved using a general-purpose solver.

Notice that empty visits can result from this phase, i.e., cases with $\sum_{s\in\mathcal{S}^t} q_{is}^{kt} = 0$ and $\bar{y}_i^{kt} = 1$ for a given $i\in\mathcal{C}$, $k\in\mathcal{K}$, $t\in\mathcal{T}$. In such a case, the customer is removed from the route and the objective function value of the solution is updated.

4.5. Solution improvement (SI) formulation

As an improvement step, we use a MIP formulation for a customer assignment problem, as in Archetti et al. (2012). This model can be used to remove and insert customers into a given solution O. Let Δ_i^{kt} be the savings in the travel cost when customer i is removed from the route of vehicle k in time period t. This value is computed as $c_{hi} + c_{ij} - c_{hj}$, where h and j are the predecessor and successor of the customer in the route, respectively. We set Δ_i^{kt} as 0 when the customer is not visited by vehicle k in time period t (i.e., $\Delta_i^{kt} = 0$ if $\bar{y}_i^{kt} = 0$). Similarly, let Γ_i^{kt} be the cost of inserting customer i into its cheapest position in the route of vehicle k in time period t. Γ_i^{kt} equals 0 for those customers that are already visited by vehicle k in time period t. The formulation uses two binary decision variables. Let δ_i^{kt} be a binary variable equal to 1 if and only if customer i is removed from the route of vehicle k in period t, and let γ_i^{kt} be a binary variable equal to 1 if and only if customer i is inserted into the route of vehicle k in time period t. Then, the solution improvement formulation can be stated as follows:

$$\begin{aligned}
\max \quad & \sum_{i\in\mathcal{C}}\sum_{t\in\mathcal{T}}\sum_{s\in\mathcal{S}^t} u_{is}d_{is}^t - \sum_{i\in\mathcal{N}}\sum_{t\in\mathcal{T}}\sum_{s\in\mathcal{S}^t} h_{is}I_{is}^t \\
& + \sum_{i\in\mathcal{C}}\sum_{t\in\mathcal{T}}\sum_{k\in\mathcal{K}} \Delta_i^{kt}\delta_i^{kt} - \sum_{i\in\mathcal{C}}\sum_{t\in\mathcal{T}}\sum_{k\in\mathcal{K}} \Gamma_i^{kt}\gamma_i^{kt} && && (69)\\
\text{s.t.} \quad & \sum_{s\in\mathcal{S}^t}\sum_{m=t}^{T_s^t+1} q_{is}^{ktm} \le U_i(\bar{y}_i^{kt} - \delta_i^{kt} + \gamma_i^{kt}) && i\in\mathcal{C}, k\in\mathcal{K}, t\in\mathcal{T}, && (70)\\
& \sum_{i\in\mathcal{C}}\sum_{s\in\mathcal{S}^t}\sum_{m=t}^{T_s^t+1} q_{is}^{ktm} \le Q\bar{y}_0^{kt} && k\in\mathcal{K}, t\in\mathcal{T}, && (71)\\
& \gamma_i^{kt} \le 1 - \bar{y}_i^{kt} && i\in\mathcal{C}, k\in\mathcal{K}, t\in\mathcal{T}, && (72)\\
& \delta_i^{kt} \le \bar{y}_i^{kt} && i\in\mathcal{C}, k\in\mathcal{K}, t\in\mathcal{T}, && (73)\\
& \gamma_i^{kt} \le \bar{y}_0^{kt} && i\in\mathcal{C}, k\in\mathcal{K}, t\in\mathcal{T}, && (74)\\
& \sum_{i\in\mathcal{C}}(\delta_i^{kt} + \gamma_i^{kt}) \le \beta && k\in\mathcal{K}, t\in\mathcal{T}, && (75)\\
& \delta_i^{kt} \in \{0,1\} && i\in\mathcal{C}, k\in\mathcal{K}, t\in\mathcal{T}, && (76)\\
& \gamma_i^{kt} \in \{0,1\} && i\in\mathcal{C}, k\in\mathcal{K}, t\in\mathcal{T}, && (77)\\
& (5), (12), (14), (18) - (23) \text{ and } (26) - (27).
\end{aligned}$$

The objective function (69) consists of maximizing the total profit, given by the total revenue minus the total inventory cost plus the difference between the savings and additional travel cost given by the removal and insertion operations, respectively. Constraints (70) allow vehicle k to perform deliveries to customer i in period t only if either this customer is already visited by the vehicle in the solution and it was not removed from the route or if the customer was inserted into the route of the vehicle in the given time period. Constraints (71) impose the vehicle capacity. Constraints (72) ensure that if a customer is already visited by a route, it cannot be reinserted into the same route. Analogously, constraints (73) allow the removal of a customer from a route only if it is visited by the route. Constraints (74) forbid the insertion of customers into routes of vehicles that are not used in the given time period. Notice that if more than one visit is removed or inserted from a vehicle route, then the values of Δ and Γ provide only an approximation of the actual routing costs. For this reason, we impose constraints (75) which limit the number of changes that can be performed to every single route to a value β. Finally, constraints (76) and (77) define the domain of the removal and insertion decision variables.

Note that to model the inventory part of this model, we use delivery variables as in the formulation TP-I (Section 3.2) given that we need delivery variables with a vehicle index and, as will be shown in the computational experiments, formulation TP-I had a slightly better performance than TP-II.

4.6. Details on the computational implementation

In this section we provide some details on the computational implementation of the optimization-based ILS. We tested many different additional strategies and components to further improve the performance of the method, among which it is worth mentioning the following. First, we tried including the MCF formulation to optimize the continuous amounts of every feasible solution found by the construction heuristic. This resulted in considerably better initial solutions but the gain in solution quality did not pay off the additional computational effort required for solving many LPs, particularly for the largest problem instances. Additionally, using these better initial solutions did not result in better final solutions. Therefore, we decided to use the construction heuristic as stated in Section 4.1.

For the perturbation phase, we developed biased operators and rules for the selection of the components of the solutions that would be perturbed. However, using more sophisticated operators resulted in only marginal gains and, therefore, we opted for setting simple random selection rules.

Finally, for each instance we only construct the MCF formulation once and then we update the right-hand side of constraints (67) and (68) every time the formulation is solved. It is worth highlighting that this allowed us to obtain a speed-up of five times when compared to a method that constructs the entire MCF formulation every time it is called.

5. Computational experiments

In this section we describe the computational experiments performed with the proposed formulations and the hybrid method. All the algorithms were coded in C++ and run on a 2.67 GHz Intel Xeon X5650 Westmere processor with one thread and 36 GB of RAM. We used CPLEX 12.8 as MIP and LP solver. We turned off CPLEX's parallel mode and set the CPLEX MIP tolerance parameter to 10^{-6}. All other CPLEX parameters were set to their default values. The test instances and computational experiments are discussed in the subsequent sections.

5.1. Test instances

In our computational experiments we used two sets of problem instances. The first one, proposed by Coelho and Laporte (2014) (which we will refer to as set CL), is composed of 60 instances divided into 12 subsets with 5 instances each. In addition, we propose a second set of instances, containing 55 (larger) problem instances divided into 11 subsets of 5 instances each. We will refer to this new set as set A. The sizes of the instances of both sets vary in terms of the number of customers (N), the maximum age of the product (S), the number of vehicles (K) and the size of the planning horizon (T). The ranges of these dimensions for both sets are summarized in Table 2. Notice that the instances that we propose are larger than those of set CL in terms of the length of the planning horizon, number of vehicles and maximum age of the product, rather than in terms of the number of customers.

Parameter	Set CL	Set A
N	From 10 to 50	From 10 to 40
S	From 2 to 5	From 5 to 10
K	From 1 to 3	From 2 to 8
T	From 3 to 10	From 10 to 20

Table 2: Dimensions of the problem instances

In our new instances, the parameters were generated according to the procedure described in Coelho and Laporte (2014), except for r^t and I_{00}^0 (amount made available and initial inventory at the supplier, respectively) given that the authors do not describe how the values of these parameters are generated. Therefore, following the common practice in the IRP literature (Archetti et al., 2007, 2012) we assume that the supplier has a supply of products large enough to always be able to serve all its customers in each time period. Thus, we set $r^t = 1.5 \sum_{i \in \mathcal{C}} d_i^1, \ \forall t \in \mathcal{T}$ and $I_{00}^0 = r^1$. It is worth mentioning that the instances have revenues that are nonincreasing with the age, i.e., $u_{is} \geq u_{i,s+1}, \ \forall s \in \mathcal{S} \backslash \{S\}$, and the inventory holding costs are random for both the supplier and the customers (using a formula that includes an increase with the age of the product). Notice that random inventory holding costs represent the most general case for these values. Travel costs correspond to Euclidean distances rounded to the nearest integer. Additionally, in Section 5.4, we analyze the results of some further experiments performed with different configurations for the values of revenue, holding and travel costs.

5.2. Comparison of the formulations

In this section, we compare the results obtained with the proposed formulations. Recall that we have stated five formulations: arc-based (AB), transportation-based I and II with and without a vehicle index (TP-I, TP-II, TP-I-nk and TP-II-nk, respectively). It is worth remembering that, for these experiments, we did not include any additional valid inequality to the formulations. Experiments including additional valid inequalities are presented in the supplementary material of this paper.

First, in Table 3, we report the values of the LP relaxations provided by the formulations and the running times required to solve them. When solving the LP relaxations, in addition to removing the integrality conditions on the variables, we drop the SECs (and GFSECs) of all the formulations. We do this because the separation algorithm that we use for the formulations without a vehicle index is a heuristic procedure for fractional solutions, i.e., it may fail to find a violated inequality. However, as we will show in the upcoming experiments, despite the fact that we use this heuristic procedure, the formulations without a vehicle index had a better performance than the formulations with a vehicle index (which use an exact separation algorithm). In the table, columns 1 and 2 display the instance set and the number of instances in the respective set, respectively. Columns labeled with "Difference to AB" show the relative difference between the values of the LP relaxation of the respective formulations and formulation AB, computed using the formula $100 \times (z^* - z^{AB})/z^{AB}$, where z^* is the LP relaxation value of the formulation and z^{AB} is the value of the LP relaxation of formulation AB. Columns labeled with "Running time" report the time in seconds that the solver required to solve the LP. The results are reported separately in the same table for the two sets of instances (sets CL and A).

It is possible to observe that the LP relaxations of the formulations without a vehicle index can be considerably worse than the other formulations since, on average, the LP relaxation bounds provided by them are 3.85% and 13.30% greater than the ones of formulation AB, for set CL and A, respectively (with a maximum difference of 24.56%). The LP relaxation of the reformulations with a vehicle index is only slightly worse than the ones of formulation AB (less than 0.5%, for all sets). Notice also that the bounds of formulations TP-I and TP-II (and their respective counterparts without a vehicle index) are identical. In addition, as a result of the considerably larger number of variables in the reformulations with a vehicle index, the times to solve their LP relaxation are large when compared to formulation AB and to the reformulations without a vehicle index. These differences are remarkably larger for those sets with the largest values of S and T in set A, given that the number of delivery variables (q) in those formulations grows quickly with increasing values for these parameters.

Instance set		Difference to AB (%)				Running time				
(N-S-K-T)	#	TP-I	TP-I-nk	TP-II	TP-II-nk	AB	TP-I	TP-I-nk	TP-II	TP-II-nk
10-2-1-3	5	0.11	1.72	0.11	1.72	0.00	0.00	0.00	0.00	0.00
10-3-1-6	5	0.26	5.10	0.26	5.10	0.01	0.01	0.01	0.01	0.01
10-5-1-10	5	0.28	4.10	0.28	4.10	0.02	0.03	0.03	0.03	0.03
20-2-2-3	5	0.39	4.55	0.39	4.55	0.01	0.01	0.00	0.02	0.01
20-3-2-6	5	0.25	4.47	0.25	4.47	0.03	0.05	0.01	0.04	0.02
20-5-2-10	5	0.13	4.78	0.13	4.78	0.18	0.33	0.08	0.34	0.07
30-2-2-3	5	0.16	2.06	0.16	2.06	0.01	0.02	0.01	0.02	0.01
30-3-2-6	5	0.26	3.85	0.26	3.85	0.07	0.07	0.04	0.08	0.03
30-5-2-10	5	0.14	4.35	0.14	4.35	0.32	0.48	0.10	0.47	0.10
40-2-3-3	5	0.45	3.42	0.45	3.42	0.06	0.07	0.01	0.06	0.01
40-3-3-6	5	0.26	4.21	0.26	4.21	0.19	0.25	0.04	0.26	0.04
50-2-3-3	5	0.27	3.59	0.27	3.59	0.06	0.09	0.02	0.08	0.02
All	60	0.25	3.85	0.25	3.85	0.08	0.12	0.03	0.12	0.03
10-7-2-15	5	0.41	11.20	0.41	11.20	0.14	0.33	0.09	0.32	0.09
10-10-2-15	5	0.30	13.72	0.30	13.72	0.19	0.64	0.18	0.75	0.16
10-10-2-20	5	0.36	11.89	0.36	11.89	0.41	1.33	0.34	1.49	0.32
20-7-4-15	5	0.19	11.89	0.19	11.89	1.15	4.18	0.26	3.63	0.22
20-10-4-15	5	0.18	10.78	0.18	10.78	1.52	7.35	0.43	6.52	0.38
20-10-6-15	5	0.17	24.56	0.17	24.56	1.98	11.40	0.37	11.33	0.38
30-7-4-15	5	0.14	8.84	0.14	8.84	2.24	9.75	0.35	8.45	0.36
30-7-8-15	5	0.12	16.71	0.12	16.71	6.29	25.11	0.37	27.07	0.33
30-10-8-15	5	0.15	18.38	0.15	18.38	8.40	52.17	0.67	57.97	0.60
40-5-4-10	5	0.08	7.37	0.08	7.37	1.14	2.85	0.15	2.52	0.13
40-5-8-10	5	0.13	10.93	0.13	10.93	2.87	9.08	0.15	6.97	0.13
All	55	0.20	13.30	0.20	13.30	2.39	11.29	0.31	11.55	0.28

Table 3: Comparison of the LP relaxation of the formulations

Table 4 reports the results when we impose a time limit of two hours to solve the MIP formulations. In the table, column "#" indicates the number of instances in each set. Then, for each formulation, column "#O" shows the number of instances of the set solved to optimality, column "#F" shows the number of instances of the set for which a feasible solution was found. In addition, we report the average relative optimality gap ("Opt gap") of the solutions of the set (as a percentage) and the average total execution time ("Total time") in seconds for all the instances of the set. Unfilled cells ("–") indicate that no feasible solution was found for all the instances of the set. The results are reported separately in the same table for the two sets of instances. It is possible to observe that all the reformulations were able to find a number of feasible solutions that is larger than or equal to the number of feasible solutions found by formulation AB. The differences are especially remarkable for the formulations without a vehicle index (TP-I-nk and TP-II-nk), which were able to find feasible solutions to 67% and 71% more instances than formulation AB, respectively. Additionally, the reformulations with a vehicle index (TP-I and TP-II) found three and two optimal solutions more than formulation AB, respectively, while both reformulations without a vehicle index found eight optimal solutions more than formulation AB (which represent 30% more optimal solutions). It is worth mentioning that when for a given set no optimal solution was found but at least a feasible solution was obtained and the average time is less than 7200 seconds, it means that the optimizer ran out of memory before reaching the time limit. Specifically, that happened once (one instance of subset 20-5-2-10) and three times (one instance of subset 20-5-2-10, 30-5-2-10 and 10-7-2-15) for the formulation TP-I-nk and TP-II-nk, respectively. Finally, notice that when the instance size grows, the solver starts failing to find feasible solutions within the time limit, especially for the formulations with a vehicle index. As expected, this degradation in the performance is more noticeable when we increase T, S and K than when we increase N. This is because the number of variables and constraints of the formulations increases faster with increasing values of those former parameters.

Table 5 summarizes the relative optimality gaps of the solutions but only considering instances for which all the formulations found at least a feasible solution. Column 2 ("#") displays the number of instances of the set (column 1) for which at least a feasible solution was found by all the formulations. The first set of columns ("B&C optimality gap") shows, for each formulation, the relative optimality gaps of the computed feasible solutions using the upper (dual) bound of the corresponding B&C algorithm. "Best dual bound gap" shows the relative difference of the solutions of the respective formulations compared to the best upper (dual) bound among all the formulations, computed as $100 \times (\bar{z} - z)/z$, where $\bar{z}$ is the best dual upper bound computed at the end of the B&C algorithm over all five formulations and z is the objective value of the solution of the model. The results show that, despite the fact that the formulations without a vehicle index can have a considerably worse LP relaxation, better optimality gaps can be obtained at the end of their B&C algorithms. These formulations can provide solutions of better quality for the largest instances of the table, as shown by the values of the gaps to the best dual bounds. Notice that for sets 10-7-2-15 and 10-10-2-15 (which are part of set A), the formulations without a vehicle index (TP-I-nk and TP-II-nk) found solutions with gaps to the best dual bounds significantly better than the ones provided by the other three formulations. Also, the results highlight the sensitivity of the formulations to increases in the values of S, K and T.

Finally, Table 6 shows the results for only those instances that were solved to optimality within the time limit by all the formulations. Column 2 ("#") displays the number of instances of the set (column 1) for which all formulations found the optimal solution. The results reveal the speed-up obtained with the reformulations, especially the ones without a vehicle index. In particular, formulations TP-I-nk and TP-II-nk use, in total, only 6.85% and 4.93% of the time required by formulation AB, respectively. The results also reveal that a significant portion of the total running time ("Total time") is spent proving the optimality of the solutions since in most of the cases the optimal solutions are found early ("Time to optimal") in the B&C tree.

5.3. Results with the optimization-based ILS

Next, we analyze the performance of the optimization-based ILS hybrid method. In all tables of this section, column "Best sol gap" shows the relative difference of the solutions obtained with our heuristic method (z^h) to the best feasible solution found by all the MIP formulations (z^f), computed as $100 \times (z^f - z^h)/z^f$; "Opt gap" shows the relative optimality gap of the obtained solutions, computed using the formula $100 \times (\bar{z} - z^h)/\bar{z}$, where $\bar{z}$ is the best dual upper bound computed at the end of the B&C algorithm over all five formulations (Section 5.2); "Total time" displays the total time (in seconds) required by the algorithm; and "Imp" shows the relative improvement in the objective function value (profit) obtained by including the SI formulation in the method, computed as $100 \times (z^{h_2} - z^{h_1})/z^{h_1}$, where z^{h_1} and z^{h_2} are the solutions found by the heuristic before and after applying the SI formulation in the method, respectively. Notice that negative values of "Best sol gap" indicate that our method found a feasible solution that is better than the best solution provided by the five formulations since we are maximizing profit.

Tables 7 and 8 show the results obtained with the optimization-based ILS. The stopping criterion

Instance set		AB				TP-I				TP-I-nk				TP-II				TP-II-nk			
(N-S-K-T)	#	#O	#F	Opt gap (%)	Total time	#O	#F	Opt gap (%)	Total time	#O	#F	Opt gap (%)	Total time	#O	#F	Opt gap (%)	Total time	#O	#F	Opt gap (%)	Total time
10-2-1-3	5	5	5	0.00	0.05	5	5	0.00	0.04	5	5	0.00	0.32	5	5	0.00	0.04	5	5	0.00	0.35
10-3-1-6	5	5	5	0.00	0.29	5	5	0.00	0.40	5	5	0.00	1.01	5	5	0.00	0.37	5	5	0.00	0.51
10-5-1-10	5	5	5	0.00	3.00	5	5	0.00	3.19	5	5	0.00	4.33	5	5	0.00	3.32	5	5	0.00	4.71
20-2-2-3	5	5	5	0.00	43.96	5	5	0.00	21.85	5	5	0.00	36.08	5	5	0.00	25.11	5	5	0.00	21.83
20-3-2-6	5	1	5	1.14	6,137.72	3	5	0.47	4,205.82	4	5	0.43	1,906.76	2	5	0.67	5,044.31	4	5	0.39	1,918.78
20-5-2-10	5	0	5	7.37	7,200.02	0	5	6.88	7,200.02	0	5	3.00	7,036.44	0	4	5.56	7,200.03	0	5	2.33	6,964.66
30-2-2-3	5	4	5	0.23	2,295.86	5	5	0.00	579.75	4	5	0.05	1,461.97	5	5	0.00	868.97	4	5	0.02	1,455.03
30-3-2-6	5	1	5	1.43	5,845.56	1	5	1.22	5,877.06	2	5	0.90	4,955.59	1	5	1.28	5,857.42	2	5	0.73	5,339.87
30-5-2-10	5	0	3	4.02	7,200.03	0	3	4.21	7,200.05	0	5	1.22	7,200.03	0	3	1.51	7,200.07	0	5	1.13	6,964.58
40-2-3-3	5	0	5	3.47	7,200.02	0	5	2.73	7,200.02	3	5	0.18	3,727.92	0	5	2.33	7,200.01	3	5	0.33	4,007.69
40-3-3-6	5	0	0	–	–	0	1	11.50	7,200.02	0	5	3.71	7,200.04	0	0	–	–	0	5	2.97	7,200.02
50-2-3-3	5	0	5	2.78	7,200.02	0	5	2.25	7,200.01	1	5	1.07	5,770.43	0	5	2.63	7,200.01	1	5	0.91	5,805.27
All	60	26	53			29	54			34	60			28	52			34	60		
10-7-2-15	5	0	4	14.34	7,200.01	0	5	11.81	7,200.03	0	5	4.82	7,200.05	0	5	16.94	7,200.02	0	5	5.23	6,935.15
10-10-2-15	5	0	2	9.99	7,200.01	0	4	22.71	7,200.04	0	5	6.51	7,200.07	0	2	18.95	7,200.05	0	5	6.45	7,200.03
10-10-2-20	5	0	0	–	–	0	0	–	–	0	5	13.25	7,200.05	0	0	–	–	0	5	10.13	7,200.07
20-7-4-15	5	0	0	–	–	0	0	–	–	0	5	14.16	7,200.06	0	0	–	–	0	5	13.90	7,200.06
20-10-4-15	5	0	0	–	–	0	0	–	–	0	2	10.91	7,200.04	0	0	–	–	0	2	10.26	7,200.02
20-10-6-15	5	0	0	–	–	0	0	–	–	0	3	10.62	7,200.04	0	0	–	–	0	3	11.44	7,200.02
30-7-4-15	5	0	0	–	–	0	0	–	–	0	2	11.25	7,200.15	0	0	–	–	0	3	14.54	7,200.03
30-7-8-15	5	0	0	–	–	0	0	–	–	0	3	18.83	7,200.07	0	0	–	–	0	4	18.63	7,200.06
30-10-8-15	5	0	0	–	–	0	0	–	–	0	1	15.51	7,200.11	0	0	–	–	0	0	–	–
40-5-4-10	5	0	0	–	–	0	0	–	–	0	3	13.55	7,200.05	0	0	–	–	0	5	15.70	7,200.06
40-5-8-10	5	0	0	–	–	0	0	–	–	0	5	15.23	7,200.02	0	0	–	–	0	4	13.41	7,200.07
All	55	0	6			0	9			0	39			0	7			0	41		

Table 4: Comparison of the MIP results obtained with the formulations

Instance set		B&C optimality gap (%)					Best dual bound gap (%)				
(N-S-K-T)	#	AB	TP-I	TP-I-nk	TP-II	TP-II-nk	AB	TP-I	TP-I-nk	TP-II	TP-II-nk
10-2-1-3	5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10-3-1-6	5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10-5-1-10	5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
20-2-2-3	5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
20-3-2-6	5	1.14	0.47	0.43	0.67	0.39	0.46	0.33	0.41	0.41	0.39
20-5-2-10	4	5.75	5.58	2.88	5.56	2.29	4.36	4.37	2.52	4.31	2.27
30-2-2-3	5	0.23	0.00	0.05	0.00	0.02	0.01	0.00	0.01	0.00	0.00
30-3-2-6	5	1.43	1.22	0.90	1.28	0.73	0.86	0.74	0.89	0.71	0.73
30-5-2-10	2	3.71	5.35	0.75	1.30	0.51	3.14	4.86	0.75	0.82	0.50
40-2-3-3	5	3.47	2.73	0.18	2.33	0.33	1.91	1.31	0.18	0.92	0.33
50-2-3-3	5	2.78	2.25	1.07	2.63	0.91	1.97	1.43	1.05	1.89	0.90
10-7-2-15	4	14.34	12.49	4.59	17.20	4.76	9.06	7.53	4.56	11.96	4.66
10-10-2-15	2	9.99	10.51	3.40	18.95	3.44	6.43	7.00	3.31	14.94	3.40
All	57	2.68	2.41	0.90	2.91	0.84	1.74	1.58	0.86	2.04	0.83

Table 5: Results for instances with feasible solutions found by all the formulations

		AB		TP-I		TP-I-nk		TP-II		TP-II-nk	
Instance set		Total	Time	Total	Time	Total	Time	Total	Time	Total	Time
(N-S-K-T)	#	time	to best	time	to best	time	to best	time	to best	time	to best
10-2-1-3	5	0.05	0.05	0.04	0.04	0.32	0.31	0.04	0.04	0.35	0.33
10-3-1-6	5	0.29	0.24	0.40	0.35	1.01	0.98	0.37	0.32	0.51	0.45
10-5-1-10	5	3.00	2.62	3.19	2.55	4.33	4.05	3.32	2.66	4.71	4.48
20-2-2-3	5	43.96	23.34	21.85	12.82	36.08	28.02	25.11	13.57	21.83	17.32
20-3-2-6	1	1,888.53	213.08	753.52	51.25	12.09	11.41	1,984.50	77.44	10.66	10.34
30-2-2-3	3	368.49	151.97	61.23	28.75	5.80	4.40	85.25	45.30	6.24	5.49
30-3-2-6	1	427.71	182.28	585.22	304.36	12.19	11.73	487.02	58.48	13.84	13.75
All	25	146.33	39.30	65.99	20.83	10.02	8.13	114.86	14.19	7.21	6.14

Table 6: Results for instances with optimal solutions found by all the formulations

was the number of iterations, which we set to 500. The value of `max_perturb` was set to $\lceil 0.7 \times N \rceil$. This value was determined through empirical preliminary experiments using a random subset of the problem instances. Recall that this parameter defines the number of elements of a solution that will be changed in each call of the perturbation mechanism. For the SI formulation, we defined a time limit of 60 seconds and set β to 1, i.e., we allow at most one removal or insertion per route. We present both results, with and without the SI formulation. We run the algorithm only once for each instance given that, as will be shown in Section 5.3.1, the results are relatively consistent between different runs. We separated the results into two tables. Each row displays the average result of the five instances of the given set (column 1), except for "Best sol gap", whose values were calculated over the instances for which we could find a feasible solution with at least one of the formulations. Recall that for set A none of the formulations was able to find a feasible solution for all the instances. Column 2 ("#") shows the number of instances of the set (column 1) for which at least one of the formulations could find a feasible solution.

The results on the instance set CL (Table 7) show that the proposed method is able to find high-quality solutions in relatively short running times. Specifically, solutions with an average optimality gap of 2.12% (1.90%) were obtained within 5.81 (6.39) seconds, on average, for all the instances of this set without (with) the SI formulation. Note that the average CPU time on this data set using the B&C algorithm with the TP-I-nk (TP-II-nk) formulation is 3275 (3306) seconds. The results also reveal the sensitivity of the method to an increase in the number of periods (T), mainly given that the effort required in the local search phase of the method, as well as the size of the MCF formulation, depend primarily on T. The inclusion of the SI formulation as a post-optimization phase provided an average relative improvement on the objective function value (profit) of the solutions of 0.23% with an increase of 8.43% in the running time.

The results on the new problem instances (set A, Table 8) show the impact of solving larger instances since, as expected, significantly larger running times are required to perform 500 iterations. Also, larger optimality gaps are obtained for the solutions on these instances. This fact does not necessarily reflect a degradation in the quality of the solutions that our method can provide for these instances, but can reflect the low quality of the dual upper bounds provided by the formulations on these instances. Recall that, using the B&C algorithms, even finding feasible solutions was difficult for these instances. On average, the hybrid method finds solutions with objective function values 0.47% (1.86%) better than the best solutions found by all the formulations without (with) the SI formulation, for the instances for which the formulations could provide a feasible solution. An average relative improvement on the objective function value (profit) of 1.15% was obtained when the SI formulation was included in the method at a cost of an increase of around 60% on the total running time. Note that the average CPU time of the B&C algorithm with the TP-I-nk (TP-II-nk) formulation (for those instances of set A for which a feasible solution was found) was 7200 (7167) seconds.

		Without SI			With SI			
Instance set		Best sol	Opt	Total	Best sol	Opt	Total	Imp
(N-S-K-T)	#	gap (%)	gap (%)	time	gap (%)	gap (%)	time	(%)
10-2-1-3	5	0.00	0.00	0.22	0.00	0.00	0.23	0.00
10-3-1-6	5	0.12	0.12	0.35	0.12	0.12	0.39	0.00
10-5-1-10	5	0.64	0.64	0.71	0.54	0.54	0.90	0.10
20-2-2-3	5	0.97	0.97	0.62	0.50	0.50	0.66	0.49
20-3-2-6	5	2.13	2.42	1.31	1.88	2.18	1.55	0.25
20-5-2-10	5	0.97	3.11	13.08	0.56	2.70	15.11	0.42
30-2-2-3	5	0.43	0.43	1.20	0.37	0.37	1.26	0.06
30-3-2-6	5	1.26	1.74	2.74	1.13	1.61	3.19	0.14
30-5-2-10	5	1.19	2.04	23.84	0.86	1.72	26.34	0.33
40-2-3-3	5	5.22	5.40	2.06	4.92	5.10	2.25	0.32
40-3-3-6	5	1.71	4.00	20.22	1.35	3.67	21.27	0.34
50-2-3-3	5	3.94	4.54	3.36	3.65	4.25	3.55	0.30
All	60	1.55	2.12	5.81	1.32	1.90	6.39	0.23

Table 7: Results with the optimization-based ILS on instance set CL

		Without SI			With SI			
Instance set		Best sol	Opt	Total	Best sol	Opt	Total	Imp
(N-S-K-T)	#	gap (%)	gap (%)	time	gap (%)	gap (%)	time	(%)
10-7-2-15	5	4.02	8.04	12.60	2.11	6.20	69.34	2.02
10-10-2-15	5	3.16	8.55	16.10	0.99	6.51	76.11	2.24
10-10-2-20	5	2.54	10.57	21.51	0.65	8.82	81.53	1.98
20-7-4-15	5	-1.03	10.81	51.39	-3.04	9.05	111.44	1.99
20-10-4-15	2	-0.64	8.98	69.11	-1.13	8.80	129.18	0.20
20-10-6-15	3	3.36	15.81	114.12	3.36	15.81	174.25	0.00
30-7-4-15	4	-5.03	7.66	99.74	-6.65	6.15	159.81	1.64
30-7-8-15	4	-0.79	14.30	222.11	-0.79	14.30	282.25	0.00
30-10-8-15	1	-0.06	14.64	303.53	-0.06	14.64	363.84	0.00
40-5-4-10	5	-8.14	6.27	62.47	-9.34	5.23	122.50	1.11
40-5-8-10	5	-1.78	10.65	122.14	-3.30	9.30	182.19	1.50
All	44	-0.47	10.57	99.53	-1.86	9.53	159.31	1.15

Table 8: Results with the optimization-based ILS on instance set A

5.3.1. Evaluation of the hybrid method with different configurations

The purpose of this section is to assess the performance of the optimization-based ILS under different configurations. First, Table 9 shows the results of the method when we change the number of iterations used as stopping criterion (column 1). The results show that, for set CL, even for a small number of iterations (e.g., 100), the method is able to find relatively good quality solutions in very short running times. For set A, notice that from 250 iterations onwards the method is able to find solutions with an average objective function value that is better than the objective function value of the best solutions provided by all the formulations within two hours, which further shows the ability of the method to find good feasible solutions in relatively short running times.

	Set CL			Set A		
# of iter	Best sol gap (%)	Opt gap (%)	Total time	Best sol gap (%)	Opt gap (%)	Total time
100	2.32	2.87	1.83	0.87	12.15	79.95
250	1.69	2.24	3.83	-0.57	10.71	110.61
500	1.32	1.90	6.39	-1.86	9.53	159.31
1000	1.14	1.70	12.37	-2.72	8.72	262.79
2000	1.02	1.57	27.46	-3.62	7.79	497.17

Table 9: Results of increasing number of iterations for the hybrid method

To evaluate the impact of the randomness in our method, we executed it five times and computed the percent coefficient of variation (%CV) of the results. The experiment resulted in a %CV of 0.57% for the total profit considering both sets of instances. This result highlights the consistency of the results obtained in different runs of the method. It is worth mentioning that the %CV for set CL is 0.37% while for set A it was 0.78%. Although these values are both very small, the difference between them reveals the impact of the size of the instances of set A as, on average, the results obtained for these instances are (slightly) more dispersed than the results for instances of set CL.

5.4. Experiments changing the parameters of the instances

To verify the impact of different scenarios on the solution structure, we perform additional experiments changing the values of the parameters of the problem instances. In these experiments, we used the first four subsets from set CL (10-2-1-3, 10-3-1-6, 10-5-1-10 and 20-2-2-3) given that these are the only sets in which all instances can be solved to optimality within the time limit of two hours by all the formulations. The results of the experiment are shown in Table 10. "TC" shows the results when we increase the travel cost of each edge of the graph by 10 ($c_{ij} = 10c_{ij},\ \forall (i,j) \in \mathcal{E}$), "SR" shows the results when we set the same revenue for all the ages, using the formula $u_{is} = u_{i0},\ \forall i \in \mathcal{C}, s \in \mathcal{S}$. Notice that the revenues may still be different between customers. Finally, "SH" shows the results when we set the same value for the holding costs of all facilities and ages ($h_{is} = 0.5,\ \forall i \in \mathcal{N}, s \in \mathcal{S}$). "Base case" shows the results with the original values of the parameters. For all the changes performed to the problem instances, we report the total profit ("Profit"), which corresponds to the objective function value, the revenue ("Revenue") and the total routing ("Routing") and inventory holding cost ("Inventory") of the optimal solutions of the considered sets. Each cell shows the average for all the instances of the respective set (indicated in the column header). It is worth remembering that the original revenue values are nonincreasing with an increasing age ($u_{is} \geq u_{i,s+1},\ \forall s \in \mathcal{S} \backslash \{S\}$) and the holding costs are random. The impact of the changes on the CPU times of the solver is shown in Section 3 of the supplementary material.

The results show that when we increase the travel costs (TC) (compared to the base case), the total revenue is reduced given that the sales revenue values do not pay off the cost of delivering fresh products and consequently the demands tend to be satisfied using older products, which provide less revenue per consumed unit. Analogously, a slight increase in the total inventory cost was observed since extra deliveries, which provide savings in the inventory cost, are not performed anymore given that they are no longer profitable. In this scenario, the total routing cost increases substantially, compared to the base case, resulting in negative profits. On the other hand, when we set the same revenue for all the ages (SR), the total revenue increases (compared to the base case) given that the income of the sales revenue when satisfying the demand is always maximum. In this scenario (SR), the total revenue becomes constant because the total demand must be satisfied. Additionally, a reduction in the total routing cost is observed, when compared to the base case, given that in this scenario it is not necessary to perform deliveries in each period to obtain the maximum revenue and only when it is necessary to satisfy a demand or when it can provide savings in the total inventory holding cost. Finally, when the holding costs are set to a constant value (SH) it is possible to observe a slight reduction in the total routing cost in most of the sets when compared to the base case. This can be explained by the fact that deliveries that provided savings in the inventory cost are no longer performed. Again, analogously to the total revenue in scenario SR, the total inventory cost becomes a constant term in scenario SH.

		Instance subset (N-S-K-T)				
		10-2-1-3	10-3-1-6	10-5-1-10	20-2-2-3	All
Profit	Base case	18,597.61	42,434.46	70,897.15	37,712.51	42,410.43
	TC	-14,753.83	-29,809.26	-55,722.49	-18,927.16	-29,803.19
	SR	35,156.47	66,593.79	81,603.14	72,124.65	63,869.51
	SH	18,178.50	44,247.90	89,138.20	43,147.50	48,678.03
Revenue	Base case	33,580.40	81,343.00	164,797.40	72,144.60	87,966.35
	TC	29,276.00	67,448.80	143,224.80	61,916.80	75,466.60
	SR	47,846.80	100,993.40	168,464.20	104,097.20	105,350.40
	SH	33,653.00	81,258.80	164,571.60	72,141.00	87,906.10
Routing	Base case	5,661.20	13,096.20	22,000.40	8,718.00	12,368.95
	TC	34,658.00	68,976.00	121,646.00	53,824.00	69,776.00
	SR	3,445.80	7,177.80	13,853.00	5,688.00	7,541.15
	SH	5,727.20	12,905.80	21,435.40	8,556.20	12,156.15
Inventory	Base case	9,321.59	25,812.34	71,899.85	25,714.09	33,186.97
	TC	9,371.83	28,282.06	77,301.29	27,019.96	35,493.79
	SR	9,244.53	27,221.81	73,008.06	26,284.55	33,939.74
	SH	9,747.30	24,105.10	53,998.00	20,437.30	27,071.93

Table 10: Results when the parameters of the problem instances are changed

6. Conclusions and future research

In this paper, we addressed an inventory routing problem in which goods are perishable. We present four new mathematical formulations for the problem, two with a vehicle index and two without a vehicle index, and present branch-and-cut algorithms to solve them. We also developed a hybrid solution method for the problem by combining an iterated local search metaheuristic with two mathematical programming components. Additionally, we introduced new instances for the problem. The results of the computational experiments show that the formulations without a vehicle index provide a considerably larger number of feasible solutions within two hours when compared to the other formulations, in addition to a significant speed-up for instances solved to optimality within the time limit by all the formulations. Furthermore, our hybrid heuristic solution method was able to provide high-quality solutions within relatively short running times on small- and medium-sized problem instances. When applied to larger instances, the method provides good feasible solutions within reasonable running times.

An interesting perspective for future research is to extend the formulation and adapt the hybrid method to solve richer extensions of the problem, such as the case with multiple perishable products or multiple sources of the products, as well as considering production as a decision within the problem, which would lead to a production-routing problem with perishability considerations.

Acknowledgments

This work was supported by the São Paulo Research Foundation (FAPESP); the Brazilian National Council for Scientific and Technological Development (CNPq); and the Coordination for the Improvement of Higher Education Personnel (CAPES). This support is gratefully acknowledged.

References

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

Adulyasak, Y., Cordeau, J.-F., and Jans, R. (2015). The production routing problem: A review of formulations and solution algorithms. *Computers & Operations Research*, 55:141–152.

Alvarez, A. and Munari, P. (2017). An exact hybrid method for the vehicle routing problem with time windows and multiple deliverymen. *Computers & Operations Research*, 83:1–12.

Alvarez, A., Munari, P., and Morabito, R. (2018). Iterated local search and simulated annealing algorithms for the inventory routing problem. *International Transactions in Operational Research*, 25:1785–1809.

Amorim, P., Meyr, H., Almeder, C., and Almada-Lobo, B. (2013). Managing perishability in production-distribution planning: A discussion and review. *Flexible Services and Manufacturing Journal*, 25(3):389–413.

Applegate, D., Bixby, R., Chvátal, V., and Cook, W. (2018). Concorde TSP solver. `http://www.math.uwaterloo.ca/tsp/concorde.html`. Accessed: 2018-07-20.

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

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

Azadeh, A., Elahi, S., Farahani, M. H., and Nasirian, B. (2017). A genetic algorithm-Taguchi based approach to inventory routing problem of a single perishable product with transshipment. *Computers & Industrial Engineering*, 104:124–133.

Coelho, L. and Laporte, G. (2014). Optimal joint replenishment, delivery and inventory management policies for perishable products. *Computers & Operations Research*, 47:42 – 52.

Crama, Y., Rezaei, M., Savelsbergh, M., and Woensel, T. V. (2018). Stochastic inventory routing for perishable products. *Transportation Science*, 52(3):526–546.

Deng, Y. and Bard, J. F. (2011). A reactive GRASP with path relinking for capacitated clustering. *Journal of Heuristics*, 17(2):119–152.

Diabat, A., Abdallah, T., and Le, T. (2016). A hybrid tabu search based heuristic for the periodic distribution inventory problem with perishable goods. *Annals of Operations Research*, 242(2):373–398.

Federgruen, A., Prastacos, G., and Zipkin, P. H. (1986). An allocation and distribution model for perishable products. *Operations Research*, 34(1):75–82.

Hemmelmayr, V., Doerner, K., Hartl, R., and Savelsbergh, M. (2009). Delivery strategies for blood products supplies. *OR Spectrum*, 31(4):707–725.

Krarup, J. and Bilde, O. (1977). Plant location, set covering and economic lot size: An 0 (mn)-algorithm for structured problems. In Collatz, L., Meinardus, G., and Wetterling, W., editors, *Numerische Methoden bei Optimierungsaufgaben Band 3: Optimierung bei graphentheoretischen und ganzzahligen Problemen*, pages 155–180. Birkhäuser Basel.

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

Lourenço, H., Martin, O., and Stützle, T. (2003). Iterated local search. In Glover, F. and Kochenberger, G., editors, *Handbook of Metaheuristics*, pages 321–353. Springer US.

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

Mirzaei, S. and Seifi, A. (2015). Considering lost sale in inventory routing problems for perishable goods. *Computers & Industrial Engineering*, 87:213–227.

Mladenović, N. and Hansen, P. (1997). Variable neighborhood search. *Computers & Operations Research*, 24:1097–1100.

Qiu, Y., Qiao, J., and Pardalos, P. M. (2018). Optimal production, replenishment, delivery, routing and inventory management policies for products with perishable inventory. *Omega*. (Online first).

Shaabani, H. and Kamalabadi, I. N. (2016). An efficient population-based simulated annealing algorithm for the multi-product multi-retailer perishable inventory routing problem. *Computers & Industrial Engineering*, 99:189–201.

Soysal, M., Bloemhof-Ruwaard, J., Haijema, R., and Van Der Vorst, J. (2015). Modeling an inventory routing problem for perishable products with environmental considerations and demand uncertainty. *International Journal of Production Economics*, 164:118–133.

Subramanian, A., Penna, P., Uchoa, E., and Ochi, L. S. (2012). A hybrid algorithm for the heterogeneous fleet vehicle routing problem. *European Journal of Operational Research*, 221(2):285–295.

Toth, P. and Vigo, D. (2002). An overview of vehicle routing problems. In Toth, P. and Vigo, D., editors, *The Vehicle Routing Problem*, pages 1–26. SIAM.

Online Supplementary Material: Formulations, Branch-and-Cut and a Hybrid Heuristic Algorithm for an Inventory Routing Problem with Perishable Products

Aldair Alvarez[a,c,*], Jean-François Cordeau[b,c], Raf Jans[b,c], Pedro Munari[a], Reinaldo Morabito[a]

[a] *Department of Production Engineering, Federal University of São Carlos, 13565-905, São Carlos-SP, Brazil*
[b] *Department of Logistics and Operations Management, HEC Montréal, H3T 2A7, Canada*
[c] *Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT), H3T 1J4, Montréal, Canada*

Abstract

This is the online supplement of the paper "Formulations, Branch-and-Cut and a Hybrid Heuristic Algorithm for an Inventory Routing Problem with Perishable Products". Section 1 shows the valid inequalities that we used to further strengthen the formulations. In Section 2, we provide some detailed results of all the formulations when including these inequalities, while Section 3 shows the CPU times required to solve to optimality all the instances considered in the experiments in which some parameters of the instances were changed.

1. Valid inequalities

We can further strengthen the formulations (AB, TP-I, TP-II, TP-I-nk and TP-II-nk) by including some valid inequalities. All these inequalities have been used in previous works (Archetti et al., 2007; Engineer et al., 2012; Coelho and Laporte, 2014; Desaulniers et al., 2016), and can also be used in our formulations. Constraints (1) and (2) enforce the relation between the routing variables (x) and the visit variables (y):

$$x_{0i}^{kt} \leq 2y_i^{kt} \qquad i \in \mathcal{C}, k \in \mathcal{K}, t \in \mathcal{T}, \tag{1}$$

$$x_{ij}^{kt} \leq y_i^{kt} \qquad i, j \in \mathcal{E} \colon i \neq 0, k \in \mathcal{K}, t \in \mathcal{T}. \tag{2}$$

The respective counterparts of these inequalities for the formulations without a vehicle index are obtained by dropping the vehicle index from the variables.

Symmetry breaking constraints can be included in the vehicle index formulations in the presence of identical vehicles, as follows:

$$y_i^{kt} \leq \sum_{j \in \mathcal{C}: j < i} y_j^{k-1,t} \qquad i \in \mathcal{C} \backslash \{1\}, k \in \mathcal{K} \backslash \{1\}, t \in \mathcal{T}, \tag{3}$$

$$y_0^{kt} \leq y_0^{k-1,t} \qquad k \in \mathcal{K} \backslash \{1\}, t \in \mathcal{T}. \tag{4}$$

*Corresponding author.

Email addresses: aldair@dep.ufscar.br (Aldair Alvarez), jean-francois.cordeau@hec.ca (Jean-François Cordeau), raf.jans@hec.ca (Raf Jans), munari@dep.ufscar.br (Pedro Munari), morabito@ufscar.br (Reinaldo Morabito)

We can use three additional sets of valid inequalities. The first set corresponds to inequalities on the minimum number of visits to a single customer up to a given time period. Let $\bar{I}_i^t = \max\left\{0, I_{i0}^0 - \sum_{p=1}^t d_i^p\right\}$ be the minimum amount from the initial inventory that must remain at customer i at the end of time period t, with $\bar{I}_i^t = 0, \forall i \in \mathcal{C}, t \in \mathcal{T}, t > S$ and $\bar{I}_i^0 = I_{i0}^0$. Let $\bar{d}_i^t$ be the minimum residual demand of customer i in time period t, where $\bar{d}_i^t = \max\left\{0, d_i^t - \bar{I}_i^{t-1}\right\}, \forall i \in \mathcal{C}, t \in \mathcal{T}, t \leq S$ and $\bar{d}_i^t = d_i^t, \forall i \in \mathcal{C}, t \in \mathcal{T}, t > S$. Then, considering all the residual demands of a customer i up to a given time period t and the maximum size of a single delivery to the customer, it is possible to compute a lower bound on the number of visits to the customer up to that time period. This lower bound is given by $LB1_i^t = \lceil \sum_{p=1}^t \bar{d}_i^p / U_i \rceil$. Now, it follows that the following inequalities are valid:

$$\sum_{p=1}^{t} \sum_{k \in \mathcal{K}} y_i^{kp} \geq LB1_i^t, \qquad i \in \mathcal{C}, t \in \mathcal{T}. \tag{5}$$

For the formulations without a vehicle index, the left-hand side of the inequalities is replaced by the term $\sum_{p=1}^t y_i^p$.

The second set corresponds to inequalities on the minimum number of routes up to a given time period. These can be obtained by summing over the residual demands of all the customers up to a time period. Thus, a lower bound on the minimum number of routes to serve the residual demands of all customers up to time period t is given by $LB2^t = \lceil \sum_{i \in \mathcal{C}} \sum_{p=1}^t \bar{d}_i^p / Q \rceil$. Then, the following inequalities are valid:

$$\sum_{p=1}^{t} \sum_{k \in \mathcal{K}} y_0^{kp} \geq LB2^t, \qquad t \in \mathcal{T}. \tag{6}$$

Analogously, for the formulations without a vehicle index, the left-hand side of the inequalities becomes $\sum_{p=1}^t y_0^p$.

The final set generalizes inequalities (5) by considering any time interval $[t_1, t_2]$, $\forall t_1, t_2 \in \mathcal{T}, 1 < t_1 < t_2 \leq T$. Let $\bar{d}_i^{t_1 t_2} = \sum_{t=t_1}^{t_2} d_i^t$ denote the sum of demands over time periods t_1 to t_2 and $I_i^t = \sum_{s \in \mathcal{S}^t} I_{is}^t$ denote the sum of inventory variables for customer $i \in \mathcal{C}$ in time period $t \in \mathcal{T}$. Then

$$\left\lceil \frac{\bar{d}_i^{t_1 t_2} - I_i^{t_1 - 1}}{U_i} \right\rceil$$

is a lower bound on the number of visits to customer $i \in \mathcal{C}$ from t_1 to t_2. Notice that for $t_1 = 1$ the resulting inequalities correspond to (5). For $t_1 > 1$ it results in a nonlinear bound given the presence of the inventory variables. However, as shown by Engineer et al. (2012), linear inequalities can be derived by appropriately bounding $I_i^{t_1-1}$. Assume that $I_i^{t_1-1} = C_i - d_i^{t_1-1}$, i.e., the inventory at the end of period $t_1 - 1$ is at its maximum level. Then, given that $I_i^{t_1-1} \leq C_i - d_i^{t_1-1}$,

$$\sum_{t=t_1}^{t_2} \sum_{k \in \mathcal{K}} y_i^{kt} \geq \left\lceil \frac{\bar{d}_i^{t_1 t_2} - (C_i - d_i^{t_1-1})}{U_i} \right\rceil \qquad i \in \mathcal{C}, 1 < t_1 < t_2 \leq T \tag{7}$$

is a valid inequality. Similarly, for the formulations without a vehicle index, the left-hand side of (7) is changed to $\sum_{t=t_1}^{t_2} y_i^t$.

2. Impact of the valid inequalities

The purpose of this section is to analyze the results obtained when we strengthen the formulations by including the valid inequalities of Section 1. Table S1 show the number of optimal and feasible solutions found when the different valid inequalities are included in the formulations. We considered the instance set CL only. Again, we set a time limit of two hours to solve each instance by each formulation. In the cells of columns 2-6, we show the number of optimal and feasible solutions, respectively (separated by a comma), found by the formulation specified in the respective header when including the inequalities given in column 1. "Base case" shows the results when no valid inequality is considered in the formulation and "All" shows the results when we include all the inequalities simultaneously. Unfilled cells ("–") indicate that the inequalities were not applied to the formulations (symmetry breaking constraints to the formulations without a vehicle index).

The results show that none of the valid inequalities was able to improve alone the performance of all the formulations in terms of both the number of optimal and feasible solutions found within the time limit. Only inequality (1) was able to provide a number of optimal and feasible solutions that is greater than or equal to the base case for all the formulations. This could be explained by the fact that CPLEX already includes some general valid inequalities (in addition to the preprocessing operations that the solver applies to the formulation), which does not allow to reveal potential gains obtained by including the given valid inequalities. Note that some of the inequalities are able to improve the performance of some of the formulations in either the number of optimal or feasible solutions, e.g., (7) for the number of feasible solutions, but not in both at the same time. It is worth mentioning that we tried including combinations of these inequalities simultaneously, but no performance improvement was obtained.

	Formulation				
	AB	TP-I	TP-I-nk	TP-II	TP-II-nk
Base case	26, 53	29, 54	34, 60	28, 52	34, 60
(1)	26, 54	29, 54	34, 60	29, 54	34, 60
(2)	24, 54	27, 54	33, 60	27, 53	32, 60
(3)	26, 55	29, 53	–	29, 52	–
(4)	26, 54	29, 53	–	28, 53	–
(5)	26, 53	29, 53	32, 60	28, 56	32, 60
(6)	25, 53	28, 52	33, 60	29, 53	33, 60
(7)	25, 56	27, 56	32, 60	29, 54	32, 60
(1)-(4)	26, 55	29, 52	33, 60	29, 53	32, 60
(5)-(7)	26, 54	28, 54	34, 60	29, 52	33, 60
All	24, 56	29, 57	32, 60	28, 57	33, 60

Table S1: Number of optimal and feasible solutions found by the formulations when including the valid inequalities

The impact of the valid inequalities on the CPU times are presented in Table S2. In the table, for each formulation, column "#O" shows the number of instances of the set solved to optimality, column "#F" shows the number of instances of the set for which a feasible solution was found, "Opt gap" shows the average relative optimality gap of the solutions of the set (as a percentage) and "Total time" displays the average total execution time in seconds for all the instances in the set. Recall that each set contains five instances.

The results show that there is no consistency in the impact of the valid inequalities over all five formulations and all the sets of instances when compared to the results without any valid inequality.

For some sets, the valid inequalities improved the performance of some formulations (e.g., formulation AB on set 20-5-2-10) but in other cases, they worsened it (e.g., formulation TP-I-nk on set 30-2-2-3). The number of feasible solutions found by all the formulations without a vehicle index increased, while the number of optimal solutions was reduced for some of the formulations. We believe this may be in part due to the fact that CPLEX already generates some general valid inequalities that do not allow to reveal potential gains obtained by including the given valid inequalities.

3. CPU times of the solver when changing the parameters of the instances

We report the CPU times required by the solver to solve to optimality all the instances considered in the experiments in which some parameters of the instances were changed (Section 5.4 of the main paper). It is worth remembering that we used all the instances of subsets 10-2-1-3, 10-3-1-6, 10-5-1-10 and 20-2-2-3. The results are displayed in Table S3. In the table, columns 2-6 show the average CPU time (in seconds) required by CPLEX to solve all the instances using the formulation stated in the respective header and when applying the change given in column 1. Recall that "Base case" shows the result with the original values of the parameters, "TC" indicates that we increased the travel cost of each edge of the graph by 10, "SR" shows the results when we set the same revenue for all the ages, and "SH" shows the results when we set the same value for the holding costs of all facilities and ages.

All but one instance could be solved to optimality in less than two hours using all the formulations (namely, one instance of set 10-5-1-10 when solved by formulation TP-I-nk and applying change SR). In this case, we computed the average time by setting a CPU time of 7200 seconds for this instance (which is indicated by the "*" mark in the table). It is worth mentioning that we could not solve this instance with the applied change and formulation TP-I-nk even within a larger running time limit (14 hours).

In the table, we observe that when we increase the routing cost (TC) and when we set the same revenue for all the ages (SR), the total time required to solve the instances increased for all the formulations (compared to the base case). The increase is especially noticeable for the change SR. This may be due to the observation that now changes on the integer decision variables (x and y) have a reduced effect on the objective function (total profit), which directly affects the performance of the B&C algorithm. On the other hand, when we set a constant value for the holding costs (SH), the time required by the solver decreases, compared to the base case. This latter fact can be explained because now the inventory part of the problem does not have any impact on the objective function value, remaining only routing and consumption decisions as key to maximizing it. Thus, the problem now reduces to maximizing the amount of fresh product that is sent to the customers, as long as it is profitable by the revenue values.

References

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

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

Instance set	AB				TP-I				TP-I-nk				TP-II				TP-II-nk			
(N-S-K-T)	#O	#F	Opt gap (%)	Total time	#O	#F	Opt gap (%)	Total time	#O	#F	Opt gap (%)	Total time	#O	#F	Opt gap (%)	Total time	#O	#F	Opt gap (%)	Total time
10-2-1-3	5	5	0.00	0.03	5	5	0.00	0.02	5	5	0.00%	0.03	5	5	0.00	0.03	5	5	0.00	0.03
10-3-1-6	5	5	0.00	0.21	5	5	0.00	0.22	5	5	0.00	0.48	5	5	0.00	0.23	5	5	0.00	0.56
10-5-1-10	5	5	0.00	1.75	5	5	0.00	2.08	5	5	0.00	4.77	5	5	0.00	1.75	5	5	0.00	4.38
20-2-2-3	5	5	0.00	26.21	5	5	0.00	15.36	5	5	0.00	29.41	5	5	0.00	16.45	5	5	0.00	22.65
20-3-2-6	1	5	1.10	6,724.99	3	5	0.43	3,740.16	3	5	0.59	3,185.49	3	5	0.55	3,464.04	4	5	0.38	1,904.10
20-5-2-10	0	5	5.34	7,200.01	0	5	5.20	7,200.03	0	5	2.75	7,200.02	0	5	8.60	7,200.02	0	5	2.71	6,817.40
30-2-2-3	2	5	0.52	4,345.10	5	5	0.00	1,240.25	4	5	0.02	1,465.08	4	5	0.03	1,490.79	4	5	0.04	1,468.55
30-3-2-6	1	5	1.48	5,976.06	1	5	1.48	5,842.84	1	5	0.88	5,762.64	1	5	1.53	5,799.09	2	5	0.94	5,270.90
30-5-2-10	0	5	9.21	7,200.03	0	4	2.36	7,200.04	0	5	1.47	6,899.24	0	4	4.44	7,200.03	0	5	2.09	6,872.01
40-2-3-3	0	5	3.16	7,200.01	0	5	2.61	7,200.01	3	5	0.20	3,462.04	0	5	2.56	7,200.01	2	5	0.26	4,649.26
40-3-3-6	0	1	7.39	7,200.06	0	3	13.01	7,200.03	0	5	3.81	7,200.01	0	3	10.92	7,200.04	0	5	5.44	7,200.03
50-2-3-3	0	5	2.97	7,200.01	0	5	2.87	7,200.01	1	5	1.12	5,915.02	0	5	3.14	7,200.02	1	5	1.02	5,781.86
All	24	56			29	57			32	60			28	57			33	60		

Table S2: Results of the formulations including all the valid inequalities

	Formulation				
	AB	TP-I	TP-I-nk	TP-II	TP-II-nk
Base case	11.82	6.37	10.44	7.21	6.85
TC	43.17	28.71	55.57	29.02	284.88
SR	111.46	127.90	645.77*	85.84	496.27
SH	7.91	2.95	1.01	3.46	1.07

Table S3: CPU times when the parameters of the problem instances are changed

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

Engineer, F. G., Furman, K. C., Nemhauser, G., Savelsbergh, M., and Song, J.-H. (2012). A branch-price-and-cut algorithm for single-product maritime inventory routing. *Operations Research*, 60(1):106–122.