Computers and Operations Research 113 (2020) 104751

Contents lists available at ScienceDirect

Computers and Operations Research

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

Benders decomposition for the inventory vehicle routing problem with perishable products and environmental costs

Faisal Alkaabneh[a], Ali Diabat[b,c], Huaizhu Oliver Gao[d,*]

[a] Systems Engineering, Cornell University, Ithaca, NY 14853, United States
[b] Division of Engineering, New York University Abu Dhabi, Saadiyat Island, Abu Dhabi 129188, United Arab Emirates
[c] Department of Civil and Urban Engineering, Tandon School of Engineering, New York University, Brooklyn, NY 11201, United States
[d] School of Civil and Environmental Engineering, Cornell University, Ithaca, NY 14853, United States

ARTICLE INFO

Article history:
Received 23 May 2018
Revised 31 May 2019
Accepted 16 July 2019
Available online 22 August 2019

Keywords:
Food/agriculture supply chain
Inventory-routing
Perishable products
Benders decomposition
Last mile logistics
GRASP meta-heuristic

ABSTRACT

We consider the problem of inventory routing in the context of perishable products and find near-optimal replenishment scheduling and vehicle routes when the objective is to maximize the supplier's profit and minimize the costs due to fuel consumption, inventory holding, and greenhouse gas emissions. Greenhouse gas emissions are calculated as a function of fuel consumed, and fuel consumption levels are accurately calculated as a function of vehicle speed, load and traveled distance. To solve the problem efficiently, we develop an exact method based on Benders decomposition to find high-quality solutions in reasonable time. To enhance the convergence rate of the Benders decomposition algorithm, we present several acceleration strategies, such as addition of valid inequalities to the master problem and warm-up start. The warm-up start acceleration strategy itself is a meta-heuristic based on the greedy random adaptive search procedure (GRASP) and mathematical programming formulations. We present computational results which illustrate the superior performance of the proposed solution methodology in solving large-scale instances with 60 customers and 6 planning periods with 4 vehicles using Benders decomposition. Additionally, we show that utilizing a more comprehensive model to calculate the fuel cost results in fuel savings of 2 to 11% on the tested instances compared to traditional models that assume that the fuel cost is solely a function of the distance traveled during delivery.

1. Introduction

A supply chain is a system of organizations, people, activities, information, and resources involved in moving a product or service from supplier to customer. Supply chains for perishable products constitute an important subclass of supply chains globally, though they have distinct features and characteristics that differentiate them from other supply chains. Essentially, the fundamental difference between perishable and non-perishable products is the continuous and significant change in the quality of perishable products throughout the entire supply chain until the point of final consumption (Akkerman et al., 2010; Yu and Nagurney, 2013). Common perishable goods include food products, flowers, live animals, ready-mixed concrete, packaged fresh produce, fruit juice, pharmaceuticals, and frozen products. They are characterized as perishable because their quality worsens over time and their value declines as a result.

Among the most critical and important logistics decisions companies face at the operational level are inventory management, vehicle routing, and scheduling of vehicles for delivery. Efficient management of vehicle routes for delivery of products from a supplier to a set of customers can result in significant savings in both operational cost and greenhouse gas (GHG) emissions (Bektaş and Laporte, 2011). Likewise, inventory control constitutes an important logistics operation, especially when products have a limited shelf life (Coelho and Laporte, 2014). The integration of routing, inventory, and replenishment scheduling decisions yields a new logistics problem called the Inventory Routing Problem (IRP). In the context of IRP, the supplier manages inventory replenishment on behalf of the customers. Once the customer provides the supplier with its demand for each time period during a certain planning horizon, the supplier is responsible for maintaining the desired level of inventory of products at the customer site. The application of an IRP system leads to an overall reduction in logistics costs and is often described as a win–win situation (Lee and Seungjin, 2008).

Given the considerable body of scientific evidence linking global warming with the increase in GHG emissions (Smith et al., 2013),

* Corresponding author.
E-mail address: hg55@cornell.edu (H.O. Gao).

https://doi.org/10.1016/j.cor.2019.07.009
0305-0548/

companies and organizations worldwide are stepping up their efforts to reduce their GHG emissions (Huang et al., 2009). For instance, Walmart set out in earnest to find a more sustainable approach to their retail business and set long-term goals to operate with 100% renewable energy, to create zero waste in their own operations, and to sell products that sustain people and the environment. Among the areas where Walmart is making progress toward this goal is their supply chain operations. The company improved the way it loads, routes, and drives its trucks,[1] and Walmart claims that it will realize savings of $1 billion a year as a result. Likewise, other companies, such as Amazon.com, are pursuing innovative plans to improve their sustainability measures toward an environment in which logistics plays a major role.[2] Hence it is important for companies seeking better environmental sustainability measures to accurately estimate their GHG emissions due to logistics operations. Given the importance and large scale of perishable products supply chains, this study is motivated by the importance of reducing GHG emissions within the perishable products supply chain industry in IRP settings.

1.1. Related work and contributions

Bell et al. (1983) introduced the formal definition of IRP in a seminal paper dealing with IRP at Air Products, a producer of industrial gases. Since then, a great deal of work has been added to the IRP literature. Studies that have contributed to this literature can essentially be classified as one of two main types: studies that have developed algorithms to solve IRP models efficiently, and studies that have extended previously published IRP models to account for practical considerations and to relax some assumptions found in previous models. However, some studies have made algorithmic and modeling contributions simultaneously. For an extensive review of the IRP literature, the interested reader is referred to Coelho et al. (2013).

Within the IRP literature, there are studies that have succeeded in developing efficient algorithms to solve IRP models. For instance, Archetti et al. (2007) developed a branch-and-cut algorithm for a single-vehicle IRP, while Solyalı and Süral (2011) developed a branch-and-cut algorithm enhanced with an upper bound heuristic to solve a single-vehicle IRP. Among the algorithms developed to solve IRP with multiple vehicles, Coelho and Laporte (2013b) proposed a branch-and-cut algorithm, Adulyasak et al. (2013) developed a heuristic based on the adaptive large neighborhood search (ALNS) technique combined with branch-and-cut, and Desaulniers et al. (2015) developed a branch-price-and-cut algorithm. Other studies have developed meta-heuristics to solve IRP models efficiently. For instance, Archetti et al. (2017) developed a meta-heuristic that combines tabu search and mathematical programming formulations to solve IRP for the case of multiple vehicles. However, it should be noted that these studies are dedicated to the classical IRP setting, where there is a single supplier, a single product, multiple customers, and a finite planning horizon.

Apart from studies that have focused on developing efficient algorithms to solve IRP models, several studies in the IRP literature have aimed to extend traditional IRP models to solve IRP instances with some new or unique characteristic that traditional IRP models fail to accommodate. Among these extensions are the multi-product IRP (Speranza and Ukovich, 1994), stochasticity in demand (Qu et al., 1999), infinite planning time horizon (Minkoff, 1993) and many-suppliers-to-many-customers distribution systems (Christiansen, 1999). Yet another interesting practical extension of the classical IRP models is extending IRP models of supply chain systems to perishable products.

Incorporating characteristics of perishable products adds complexity to the IRP, which is already quite complex; hence relatively few studies have addressed the perishable inventory routing problem (PIRP). Generally speaking, perishable products can be classified into two general categories, based on how their quality changes over time. The first category contains products such as drugs and milk products that are considered fit for consumption until their stated expiration date, while the second category contains products whose quality changes over time, possibly in a non-monotone fashion. Here we provide a brief review of some of the studies that considered PIRP. Custódio and Oliveira (2006) described a real-world application of IRP for the distribution of frozen products in Portugal and used a heuristic to solve the developed model. Hemmelmayr et al. (2010) developed a mathematical model to plan delivery routes for the supply of blood products to hospitals by a blood bank given fixed routes and stochastic demand. Nagurney et al. (2012) developed a supply chain network optimization model for the management of human blood, while Le et al. (2013) developed a column generation algorithm for PIRP. Coelho and Laporte (2014) presented a mathematical model for PIRP and proposed a branch-and-cut algorithm to solve the developed model; this study was the first to consider the impact of product age on its quality. Their mathematical model incorporates the fact that the revenue from—and inventory holding costs surrounding—perishable products are functions of their age and that these functions are not necessarily flat as suggested in earlier studies.

It has been suggested by a number of studies in the vehicle routing literature that there are opportunities for reducing GHG emissions by extending the traditional vehicle routing problem (VRP) objectives to account for environmental and social impacts other than just the economic costs (Sbihi and Eglese, 2007). To this end, several researchers have extended traditional VRP models to account for environmental and social factors. For instance, Bektaş and Laporte (2011) developed a VRP mathematical model that minimizes the distance traveled, greenhouse gas emissions, fuel consumption, and travel times and their costs. Greenhouse gas emissions were modeled as a function of fuel consumption, which is itself a function of vehicle load, speed, and distance traveled. They showed that there is great potential for realizing savings in the total cost of routing and GHG emissions when fuel consumption is modeled as a function of vehicle speed, load, and distance traveled. Thus, there is strong evidence from the VRP literature that using different models to calculate the fuel cost yields different routing decisions.

The only study in the IRP literature that utilized accurate estimation of the fuel cost is that by Soysal et al. (2018). They considered PIRP in the case of multiple products, a many-to-many distribution system, and accurate estimation of the fuel cost, and they included the cost of greenhouse gas emissions in the objective function. They were interested in analyzing the value of horizontal collaboration between suppliers within the context of PIRP and found that such collaboration leads to reductions in emissions, driving time, inventory, and costs of waste given an uncertain demand. They were able to solve small instances with 5 customers using a commercial mixed integer programming (MIP) solver, but they did not provide an algorithm or heuristic for solving large-scale instances of their model.

We summarize the aforementioned prior work on PIRP in Table 1. The first column of the table shows the article/study considered for discussion. The second column indicates whether the model presented in each study considered revenue and holding costs as a function of product age. The third through fifth columns indicate which factors were considered when calculating the fuel

[1] https://www.theguardian.com/sustainable-business/2015/nov/18/walmart-climate-change-carbon-emissions-renewabe-energy-environment

[2] https://www.amazon.com/p/feature/wnsdvqqghme982o

Table 1
Literature on PIRP.

Study	Aging Effects	Fuel Cost Function: Distance Traveled	Vehicle Load	Vehicle Speed	Stoch. Demand	Approach
Custódio and Oliveira (2006)	-	✔	-	-	-	Heuristic
Hemmelmayr et al. (2010)	-	✔	-	-	✔	Heuristic
Nagurney et al. (2012)	-	✔	-	-	✔	Heuristic
Le et al. (2013)	-	✔	-	-	-	Column gen.
Coelho and Laporte (2014)	✔	✔	-	-	-	Branch-and-cut
Soysal et al. (2018)	✔	✔	✔	✔	✔	Solver
This work	✔	✔	✔	✔	-	Benders decomp.

(transportation) cost in each study. The sixth column represents demand uncertainty and indicates whether the demand was assumed to be stochastic or deterministic. Lastly, the seventh column indicates the methodology developed in each study.

To the best of our knowledge, solving large-scale instances of PIRP with GHG emissions costs and accurate estimation of the fuel cost has not been addressed before. Our study makes four main contributions. First, we introduce the perishable inventory routing problem in conjunction with accurate estimation of the fuel cost and GHG emissions mathematical model. Second, we propose a Benders decomposition algorithm with several computational enhancements: valid inequalities, warm-up start, and Pareto-optimal cuts. Third, we develop an effective meta-heuristic to obtain good-quality solutions based on solving a mixed integer programming (MIP) formulation and a greedy random adaptive search procedure (GRASP). Fourth, we demonstrate the benefits of utilizing a model that accurately estimates fuel consumption levels as a function of vehicle load, speed and distance traveled in the context of PIRP, as opposed to traditional models that rely on distance traveled as the sole measure of fuel consumption. The overall benefits can be summarized as savings in the fuel cost and lower GHG emissions.

2. Problem description and mathematical formulation

We now formally introduce the perishable inventory routing problem (PIRP). For the sake of convenience and consistency, we use the conventional notation for IRP that is found in the literature. Our IRP model is presented as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, where $\mathcal{V} = \{0, \ldots, n\}$ is the vertex set and $\mathcal{A}$ is the edge set. The vertex 0 is the supplier node, and the vertices in the set $\mathcal{V}' = \mathcal{V}\backslash\{0\}$ are the customer nodes. There is a routing cost c_{ij} associated with edge $(i, j) \in \mathcal{A}$. This cost c_{ij} can be viewed as the cost of traveling along edge (i, j) when the vehicle is not loaded. Our model assumes that the edge length matrix is symmetric (i.e., $c_{ij} = c_{ji}$).

Given the nature of perishable products, the age of a product is equal to some element of the discrete, finite set $\mathcal{S} = \{0, 1, 2, \ldots, s\}$. For a product that becomes spoiled after s time periods, at age $s+1$ it is deemed to be spoiled and not suitable for consumption, and hence it vanishes from the inventory. The revenue from an item depends on its age g and is denoted by u^g. An inventory holding cost is incurred for every product in the inventory of the supplier and every product in the inventory of a customer. This cost depends on the age of the product, g, and is denoted by h_i^g. The inventory capacity of customer i is known and is denoted by C_i; this holding capacity cannot be exceeded at any time.

The set of time periods is $\mathcal{T} = \{1, \ldots, p\}$, where p is the planning horizon. In each time period, the quantity of items made available at the supplier is r^t. We assume that the supplier always acquires the freshest products—an assumption that is consistent with that presented in Coelho and Laporte (2014). Let d_i^t be the demand of customer i in each time period t. As in the model presented in Coelho and Laporte (2014), we assume that customers do not require the supplier to supply them with the freshest items, and that in any time period, the supplier has the choice of delivering products of any age to customers; nevertheless, the revenue of the supplier depends on the age of the products delivered. As in other studies in the literature on IRP, we assume that the supplier has enough inventory to satisfy the demands of all the customers during the planning horizon. Furthermore, neither the supplier nor the customers are allowed to have a negative inventory in any time period. We decompose the demand d_i^t of customer i in time period t as $\sum_{g\in\mathcal{S}} d_i^{gt}$, since the demand can be satisfied by products of any age g, as long as the products are not spoiled. Lastly, a fleet of vehicles is available at the supplier site to deliver products to the customers. We denote the set of vehicles by $\mathcal{K}$ and the capacity of vehicle k by Q_k.

When the supplier decides to replenish a customer's inventory, the following constraints are imposed:

- The customer's inventory level can never exceed its capacity.
- Inventory levels are not allowed to be negative.
- Each of the supplier's vehicles must start its delivery task at the supplier site and return there at the end of the delivery task.
- The vehicle capacities cannot be exceeded.

The solution to the PIRP provides the following information to the supplier: (1) when to supply each customer during the planning horizon, (2) the quantities and ages of the products to be delivered to each customer, and (3) how to combine visits to different customers into vehicle routes.

We define a binary variable x_{ij}^{kt} that is equal to 1 if and only if vehicle k travels along edge (i, j) in time period t, and we define a binary variable y_i^{kt} that is equal to 1 if and only if node i (the supplier or a customer) is visited by vehicle k in time period t. Let I_i^t denote the inventory level at node i at the end of time period t. Just as with the demand d_i^t we decompose the inventory level decision variable I_i^t by the ages of the items; hence $I_i^t = \sum_{g\in\mathcal{S}} I_i^{gt}$, where I_i^{gt} is the quantity of product of age g in the inventory at node i at the end of time period t. We denote by q_i^{gkt} the quantity of product of age g that is delivered in vehicle k to customer i in time period t. The problem can then be formulated as follows, which is adapted from Coelho and Laporte (2014):

$$\min_{I,d,q,y,x} \; -\sum_{i\in\mathcal{V}'}\sum_{g\in\mathcal{S}}\sum_{t\in\mathcal{T}} u^g d_i^{gt} + \sum_{i\in\mathcal{V}}\sum_{g\in\mathcal{S}}\sum_{t\in\mathcal{T}} h_i^g I_i^{gt} + \sum_{(i,j)\in\mathcal{A}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} c_{ij} x_{ij}^{kt} \tag{1}$$

$$\text{s.t.} \quad I_0^{gt} = I_0^{g-1,t-1} - \sum_{i\in\mathcal{V}'}\sum_{k\in\mathcal{K}} q_i^{gkt} \quad \text{for all } g \in \mathcal{S}\backslash\{0\}, \; t \in \mathcal{T} \tag{2}$$

$$I_0^{0t} = r^t \quad \text{for all } t \in \mathcal{T} \tag{3}$$

$$I_i^{gt} = I_i^{g-1,t-1} + \sum_{k\in\mathcal{K}} q_i^{gkt} - d_i^{gt} \quad \text{for all } i \in \mathcal{V}', \; g \in \mathcal{S}\backslash\{0\}, t \in \mathcal{T} \tag{4}$$

$$I_i^{0t} = \sum_{k\in\mathcal{K}} q_i^{0kt} - d_i^{0t} \quad \text{for all } i \in \mathcal{V}', t \in \mathcal{T} \tag{5}$$

$$\sum_{g\in\mathcal{S}} I_i^{gt} \le C_i \quad \text{for all } i \in \mathcal{V}', t \in \mathcal{T} \tag{6}$$

$$d_i^t = \sum_{g\in\mathcal{S}} d_i^{gt} \quad \text{for all } i \in \mathcal{V}', t \in \mathcal{T} \tag{7}$$

$$\sum_{g\in\mathcal{S}}\sum_{k\in\mathcal{K}} q_i^{gkt} \le C_i - \sum_{g\in\mathcal{S}} I_i^{g,t-1} \quad \text{for all } i \in \mathcal{V}', t \in \mathcal{T} \tag{8}$$

$$q_i^{gkt} \le C_i y_i^{kt} \quad \text{for all } i \in \mathcal{V}', g \in \mathcal{S}, k \in \mathcal{K}, t \in \mathcal{T} \tag{9}$$

$$\sum_{i\in\mathcal{V}'}\sum_{g\in\mathcal{S}} q_i^{gkt} \le Q_k y_0^{kt} \quad \text{for all } k \in \mathcal{K}, t \in \mathcal{T} \tag{10}$$

$$\sum_{j\in\mathcal{V}:j\neq i} x_{ij}^{kt} + \sum_{j\in\mathcal{V}:j\neq i} x_{ji}^{kt} = 2y_i^{kt} \quad \text{for all } i \in \mathcal{V}, k \in \mathcal{K}, \; t \in \mathcal{T} \tag{11}$$

$$\sum_{i\in\mathcal{ST}}\sum_{j\in\mathcal{ST}} x_{ij}^{kt} \le \sum_{i\in\mathcal{ST}} y_i^{kt} - y_m^{kt} \quad \text{for all } \mathcal{ST} \subseteq \mathcal{V}', k \in \mathcal{K}, \; t \in \mathcal{T} \text{ and some } m \in \mathcal{ST} \tag{12}$$

$$\sum_{k\in\mathcal{K}} y_i^{kt} \le 1 \quad \text{for all } i \in \mathcal{V}', t \in \mathcal{T} \tag{13}$$

$$I_i^{gt}, d_i^{gt}, q_i^{gkt} \ge 0 \quad \text{for all } i \in \mathcal{V}', k \in \mathcal{K}, t \in \mathcal{T} \tag{14}$$

$$x_{ij}^{kt} \in \{0, 1\} \quad \text{for all } (i, j) \in \mathcal{A}, k \in \mathcal{K}, t \in \mathcal{T} \tag{15}$$

$$y_i^{kt} \in \{0, 1\} \quad \text{for all } i \in \mathcal{V}, k \in \mathcal{K}, t \in \mathcal{T}. \tag{16}$$

The objective function (1) maximizes the total sales revenue minus the total cost of inventory and routing; for the sake of convenience, it is expressed in terms of minimization. Constraints (2) define the supplier's inventory levels and the aging of the products by one unit in each time period. Constraints (3) ensure that the supplier always receives the freshest products. Constraints (4) and (5) define the customers' inventory levels and the aging of the products they hold. Constraints (6) ensure that the inventory capacities of the customers are respected. Constraints (7) require the demand of each customer in each period to be the sum of the quantities of products of different ages. Note that spoiled products cannot be used to satisfy the demand, and hence are eliminated from the inventory. Constraints (8) and (9) ensure that each vehicle delivers products only to those customers that the vehicle has been assigned to. Constraints (10) ensure that the vehicle capacities are not exceeded. Constraints (11) ensure that each vertex on a route has exactly one incoming edge and one outgoing edge, while Constraints (12) eliminate sub-tours ($\mathcal{ST}$ denotes the set of subtours). Inequalities (13) prohibit split deliveries, by ensuring that each customer is served by at most one vehicle in each time period. Constraints (14) require the continuous decision variables to be non-negative and Constraints (15)–(16) require the integer decision variables to be binary.

2.1. Calculating GHG emissions

Calculating fuel consumption of a vehicle during routing is a critical step in accurately estimating GHG emissions and operational costs associated with routing. Recently, there has been growing interest in providing simulation models that accurately predict fuel consumption of vehicles in different settings and for different values of parameters. For instance, the United States Environmental Protection Agency (USEPA) developed a simulator, called the MOtor Vehicle Emissions Simulator (MOVES), to estimate emissions of mobile sources at the national, county, and project levels (MOVES, 2018). Likewise, several studies in the scientific literature have proposed models for the estimation of GHG emissions of moving vehicles based on different factors. One such study, by Bektaş and Laporte (2011), presented a mathematical model for the optimization of vehicle routes that takes fuel consumption into account. In their model, vehicle speed and load were among the factors considered in estimating GHG emissions of vehicles during routing. In this paper, we adopt their model to estimate fuel consumed by a vehicle during routing.

Bektaş and Laporte's model is a comprehensive emissions model in which the rate of fuel consumption is a function of engine characteristics, road conditions, vehicle speed, and vehicle load. For further information on the comprehensive model used to calculate fuel consumption levels and hence GHG emissions, the interested reader is referred to Bektaş and Laporte (2011). To simplify the presentation in this paper and to keep the discussion concise, we introduce a parameter α_{ij} to represent the effective value of all the parameters associated with fuel consumption due to vehicle load; this parameter is defined as the cost associated with traveling 1 m when the vehicle load is 1 kg. That parameter will be multiplied by the vehicle load to calculate the fuel cost per meter. Also, we introduce a new decision variable f_{ij}^{kt} to represent the load (in kg) of vehicle k when traveling along edge (i, j) in time period t. Thus the routing cost is $\sum_{(i,j)\in\mathcal{A}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}}(c_{ij}x_{ij}^{kt} + \alpha_{ij}f_{ij}^{kt})$. While our work explicitly considers vehicle speed in the calculation of greenhouse gas emissions, vehicle speed in our model is an exogenous parameter to be set by the decision maker.

To calculate the value of f_{ij}^{kt} for edge (i, j), we need to impose the following constraints:

$$\sum_{j\in\mathcal{V}} f_{ij}^{kt} - \sum_{j\in\mathcal{V}} f_{ji}^{kt} = \sum_{g\in\mathcal{S}} q_i^{gkt} \quad \text{for all } i \in \mathcal{V}, k \in \mathcal{K}, t \in \mathcal{T} \tag{17}$$

$$\sum_{g\in\mathcal{S}} q_j^{gkt} x_{ij}^{kt} \le f_{ij}^{kt} \quad \text{for all } j \in \mathcal{V}, (i, j) \in \mathcal{A}, k \in \mathcal{K}, t \in \mathcal{T} \tag{18}$$

$$f_{ij}^{kt} \le \left(Q_k - \sum_{g\in\mathcal{S}} q_i^{gkt}\right) x_{ij}^{kt} \quad \text{for all } i \in \mathcal{V}, (i, j) \in \mathcal{A}, \; k \in \mathcal{K}, t \in \mathcal{T} \tag{19}$$

$$f_{ij}^{kt} \ge 0 \quad \text{for all } (i, j) \in \mathcal{A}, k \in \mathcal{K}, t \in \mathcal{T}. \tag{20}$$

Constraints (18) and (19) are non-linear, because they each involve a product of two decision variables; however, these constraints can be linearized simply and effectively using the big M technique, resulting in the following set of linear constraints:

$$\sum_{g\in\mathcal{S}} q_j^{gkt} - f_{ij}^{kt} \le M(1 - x_{ij}^{kt}) \quad \text{for all } j \in \mathcal{V}, (i, j) \in \mathcal{A}, \; k \in \mathcal{K}, t \in \mathcal{T} \tag{21}$$

$$-Mx_{ij}^{kt} \le f_{ij}^{kt} \quad \text{for all } (i, j) \in \mathcal{A}, k \in \mathcal{K}, t \in \mathcal{T} \tag{22}$$

$$f_{ij}^{kt} \le Q_k x_{ij}^{kt} - \sum_{g\in\mathcal{S}} q_i^{gkt} + M(1 - x_{ij}^{kt}) \quad \text{for all } i \in \mathcal{V}, (i, j) \in \mathcal{A}, \; k \in \mathcal{K}, t \in \mathcal{T} \tag{23}$$

$$f_{ij}^{kt} \leq Q_k x_{ij}^{kt} + M x_{ij}^{kt} \quad \text{for all } (i,j) \in \mathcal{A}, k \in \mathcal{K}, t \in \mathcal{T}. \tag{24}$$

We restate the PIRP formulation for convenience:

$$\min_{I,d,q,f,y,x} -\sum_{i\in\mathcal{V}}\sum_{g\in\mathcal{S}}\sum_{t\in\mathcal{T}} u_i^g d_i^{gt} + \sum_{i\in\mathcal{V}}\sum_{g\in\mathcal{S}}\sum_{t\in\mathcal{T}} h_i^g I_i^{gt} + \sum_{(i,j)\in\mathcal{A}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} c_{ij} x_{ij}^{kt} + \sum_{(i,j)\in\mathcal{A}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} \alpha_{ij} f_{ij}^{kt} \tag{25}$$

$$\text{s.t.} \quad (2)-(16), (17), (20)-(24). \tag{26}$$

3. Benders decomposition

In 1962, Benders (1962) proposed a decomposition strategy for solving a large-scale optimization problem based on partitioning the problem into two simpler problems: a master problem (MP) and a subproblem (SP). Benders' algorithm works by solving the MP, which is a simpler version of the original problem, and passing the values of the MP solution to the SP. The algorithm solves each of the two simpler problems iteratively, one at a time. In each iteration, a new constraint, known as a Benders cut, is added to the MP. The algorithm keeps iterating until a pre-defined stopping criterion is met.

Several studies in the literature have reported the success of implementing Benders decomposition to solve difficult combinatorial problems (Contreras et al., 2011; Cordeau et al., 2001; Côté et al., 2014; Hooker, 2007). In the context of logistics, Adulyasak et al. (2015) implemented Benders decomposition to solve the production inventory routing problem with stochastic demand, which is a more general problem than the IRP, since it considers decision variables for both production quantities and production scheduling. However, the authors did not incorporate environmental considerations and considered only non-perishable products.

3.1. Benders reformulation

If the decision variables for visit scheduling and vehicle routing are held fixed, the PIRP reduces to a simple problem that aims to determine the quantities to be delivered to each customer on each visit and the ages of those products delivered. Thus we define variables $\bar{y}$ and $\bar{x}$ to represent the fixed, complicating variables. Once the values of the decision variables y and x are fixed, the Benders SP is obtained:

$$\min_{I,d,q,f} -\sum_{i\in\mathcal{V}'}\sum_{g\in\mathcal{S}}\sum_{t\in\mathcal{T}} u^g d_i^{gt} + \sum_{i\in\mathcal{V}}\sum_{g\in\mathcal{S}}\sum_{t\in\mathcal{T}} h_i^g I_i^{gt} + \sum_{(i,j)\in\mathcal{A}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} \alpha_{ij} f_{ij}^{kt} \tag{27}$$

$$\text{s.t.} \quad I_0^{gt} = I_0^{g-1,t-1} - \sum_{k\in\mathcal{K}}\sum_{i\in\mathcal{V}'} q_i^{gkt} \quad \text{for all } g \in \mathcal{S}\backslash\{0\}, t \in \mathcal{T} \tag{28}$$

$$I_0^{0t} = r^t \quad \text{for all } t \in \mathcal{T} \tag{29}$$

$$I_i^{gt} = I_i^{g-1,t-1} + \sum_{k\in\mathcal{K}} q_i^{gkt} - d_i^{gt} \quad \text{for all } i \in \mathcal{V}', g \in \mathcal{S}\backslash\{0\}, t \in \mathcal{T} \tag{30}$$

$$I_i^{0t} = \sum_{k\in\mathcal{K}} q_i^{0kt} - d_i^{0t} \quad \text{for all } i \in \mathcal{V}', t \in \mathcal{T} \tag{31}$$

$$\sum_{g\in\mathcal{S}} I_i^{gt} \leq C_i \quad \text{for all } i \in \mathcal{V}', t \in \mathcal{T} \tag{32}$$

$$d_i^t = \sum_{g\in\mathcal{S}} d_i^{gt} \quad \text{for all } i \in \mathcal{V}', t \in \mathcal{T} \tag{33}$$

$$\sum_{g\in\mathcal{S}}\sum_{k\in\mathcal{K}} q_i^{gkt} \leq C_i - \sum_{g\in\mathcal{S}} I_i^{g,t-1} \quad \text{for all } i \in \mathcal{V}', t \in \mathcal{T} \tag{34}$$

$$q_i^{gkt} \leq C_i \bar{y}_i^{kt} \quad \text{for all } i \in \mathcal{V}', g \in \mathcal{S}, k \in \mathcal{K}, t \in \mathcal{T} \tag{35}$$

$$\sum_{i\in\mathcal{V}'}\sum_{g\in\mathcal{S}} q_i^{gkt} \leq Q_k \bar{y}_0^{kt} \quad \text{for all } k \in \mathcal{K}, t \in \mathcal{T} \tag{36}$$

$$\sum_{g\in\mathcal{S}} q_j^{gkt} - f_{ij}^{kt} \leq M(1 - \bar{x}_{ij}^{kt}) \quad \text{for all } j \in \mathcal{V}', (i,j) \in \mathcal{A}, k \in \mathcal{K}, t \in \mathcal{T} \tag{37}$$

$$-M\bar{x}_{ij}^{kt} \leq f_{ij}^{kt} \quad \text{for all } (i,j) \in \mathcal{A}, k \in \mathcal{K}, t \in \mathcal{T} \tag{38}$$

$$f_{ij}^{kt} \leq Q_k \bar{x}_{ij}^{kt} - \sum_{g\in\mathcal{S}} q_i^{gkt} + M(1 - \bar{x}_{ij}^{kt}) \quad \text{for all } i \in \mathcal{V}', (i,j) \in \mathcal{A}, k \in \mathcal{K}, t \in \mathcal{T} \tag{39}$$

$$f_{ij}^{kt} \leq Q_k \bar{x}_{ij}^{kt} + M\bar{x}_{ij}^{kt} \quad \text{for all } (i,j) \in \mathcal{A}, k \in \mathcal{K}, t \in \mathcal{T} \tag{40}$$

$$I_i^{gt}, d_i^{gt}, q_i^{gkt}, f_{ij}^{kt} \geq 0 \quad \text{for all } i \in \mathcal{V}, k \in \mathcal{K}, k \in \mathcal{K}, t \in \mathcal{T}. \tag{41}$$

However, SP(I, d, q, f) might be infeasible if the quantity of items supplied to some customer during the planning horizon falls short of that customer's demand. This problem can be addressed by introducing a continuous variable to represent the amount of the shortage. To this end, we introduce a variable s_i^t to represent the amount by which the product supplied to customer i in time period t falls short of that customer's demand. This positive variable is added to constraints (33) to serve as an artificial quantity of product provided by the supplier to a customer to prevent having unmet demand. In addition, a large penalty is associated with this variable to prevent having any shortage in the optimal solution of the PIRP. Therefore, (33) is replaced by the following constraint:

$$d_i^t = \sum_{g\in\mathcal{S}} d_i^{gt} + s_i^t \quad \text{for all } i \in \mathcal{V}', t \in \mathcal{T}. \tag{42}$$

The coefficient of the decision variable s_i^t in the objective function, σ_i, which is the cost of the unmet demand of customer i, is set to a very large number. The objective function of SP(I, d, q, f, s) is

$$\min_{I,d,q,f,s} -\sum_{i\in\mathcal{V}'}\sum_{g\in\mathcal{S}}\sum_{t\in\mathcal{T}} u^g d_i^{gt} + \sum_{i\in\mathcal{V}}\sum_{g\in\mathcal{S}}\sum_{t\in\mathcal{T}} h_i^g I_i^{gt} + \sum_{(i,j)\in\mathcal{A}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} \alpha_{ij} f_{ij}^{kt} + \sum_{i\in\mathcal{V}'}\sum_{t\in\mathcal{T}} \sigma_i s_i^t. \tag{43}$$

Proposition 3.1. *For any fixed values of the decision variables y and x, SP(I, d, q, f, s) is feasible and bounded.*

Proof. The proof follows from the proposition presented in Adulyasak et al. (2015). □

The associated dual variables to constraints (28)–(32), (34)–(40), and (42) are $\boldsymbol{E} = \{E^{gt} | g \in \mathcal{S}\backslash\{0\}, t \in \mathcal{T}\}$, $\boldsymbol{H} = \{H^t | t \in \mathcal{T}\}$, $\boldsymbol{\Gamma} = \{\Gamma_i^{gt} | i \in \mathcal{V}', g \in \mathcal{S}\backslash\{0\}, t \in \mathcal{T}\}$, $\boldsymbol{\Xi} = \{\Xi_i^t | i \in \mathcal{V}', t \in \mathcal{T}\}$, $\boldsymbol{K} = \{K_i^t \geq 0 | i \in \mathcal{V}', t \in \mathcal{T}\}$, $\boldsymbol{M} = \{M_i^t \geq 0 | i \in \mathcal{V}', t \in \mathcal{T}\}$, $\boldsymbol{N} = \{N_i^{gkt} \geq 0 | i \in \mathcal{V}', k \in \mathcal{K}, g \in \mathcal{S}, t \in \mathcal{T}\}$, $\boldsymbol{\Omega} = \{\Omega^{kt} \geq 0 | k \in \mathcal{K}, t \in \mathcal{T}\}$, $\boldsymbol{O} = \{O_{ij}^{kt} \geq 0 | (i,j) \in \mathcal{A}, k \in \mathcal{K}, t \in \mathcal{T}\}$, $\boldsymbol{\Phi} = \{\Phi_{ij}^{kt} \geq 0 | (i,j) \in \mathcal{A}, k \in \mathcal{K}, t \in \mathcal{T}\}$, $\boldsymbol{\Pi} = \{\Pi_{ij}^{kt} \geq$

$0 | (i,j) \in \mathcal{A}, k \in \mathcal{K}, t \in \mathcal{T}\}$, $\boldsymbol{\Psi} = \{\Psi_{ij}^{kt} \geq 0 | (i,j) \in \mathcal{A}, k \in \mathcal{K}, t \in \mathcal{T}\}$, and $\boldsymbol{\Lambda} = \{\Lambda_i^t | i \in \mathcal{V}', t \in \mathcal{T}\}$, respectively. After associating the dual variables, the dual linear Benders subproblem can be written as

$$\begin{aligned} \max \quad & -\sum_{t\in\mathcal{T}} r^t H^t - \sum_{i\in\mathcal{V}'}\sum_{t\in\mathcal{T}} C_i K_i^t - \sum_{i\in\mathcal{V}'}\sum_{t\in\mathcal{T}} d_i^t \Lambda_i^t - \sum_{i\in\mathcal{V}'}\sum_{t\in\mathcal{T}} C_i M_i^t \\ & -\sum_{i\in\mathcal{V}'}\sum_{g\in\mathcal{S}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} (C_i \bar{y}_i^{kt}) N_i^{gkt} - \sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} (Q_k \bar{y}_0^{kt}) \Omega^{kt} \\ & - \sum_{(i,j)\in\mathcal{A}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} \big(M(1-\bar{x}_{ij}^{kt})\big) O_{ij}^{kt} - \sum_{(i,j)\in\mathcal{A}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} (M\bar{x}_{ij}^{kt}) \Phi_{ij}^{kt} \\ & - \sum_{(i,j)\in\mathcal{A}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} \big((Q_k - M)\bar{x}_{ij}^{kt} + M\big) \Pi_{ij}^{kt} \\ & - \sum_{(i,j)\in\mathcal{A}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} \big((Q_k + M)\bar{x}_{ij}^{kt}\big) \Psi_{ij}^{kt} \end{aligned} \tag{44}$$

$$\text{s.t.} \quad \boldsymbol{E}, \boldsymbol{H}, \boldsymbol{\Gamma}, \boldsymbol{I}, \boldsymbol{K}, \boldsymbol{\Lambda}, \boldsymbol{M}, \boldsymbol{N}, \boldsymbol{\Omega}, \boldsymbol{O}, \boldsymbol{\Phi}, \boldsymbol{\Pi}, \boldsymbol{\Psi} \in \Upsilon, \tag{45}$$

where Υ is the polyhedron defined by the constraints of the dual problem. Let $\mathcal{P}_\Upsilon$ be the set of extreme points defined by Υ. Since SP is feasible and bounded by Proposition 3.1, strong duality ensures that the dual problem of the primal SP is feasible and bounded as well. Therefore, we can add a Benders cut to the MP:

$$\begin{aligned} & \eta + \sum_{i\in\mathcal{V}'}\sum_{g\in\mathcal{S}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} C_i N_i^{gkt} y_i^{kt} + \sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} Q_k \Omega^{kt} y_0^{kt} \\ & \quad + \sum_{(i,j)\in\mathcal{A}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} \Big(M(O_{ij}^{kt} - \Phi_{ij}^{kt} - \Pi_{ij}^{kt} + \Psi_{ij}^{kt}) \\ & \qquad\qquad + Q_k(\Pi_{ij}^{kt} + \Psi_{ij}^{kt}) \Big) x_{ij}^{kt} \\ & \geq -\sum_{t\in\mathcal{T}} r^t H^t - \sum_{i\in\mathcal{V}'}\sum_{t\in\mathcal{T}} C_i K_i^t - \sum_{i\in\mathcal{V}'}\sum_{t\in\mathcal{T}} d_i^t \Lambda_i^t \\ & \quad -\sum_{i\in\mathcal{V}'}\sum_{t\in\mathcal{T}} C_i M_i^t - \sum_{(i,j)\in\mathcal{A}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} M \Pi_{ij}^{kt}, \end{aligned} \tag{46}$$

where η is the underestimation variable for revenue and holding, shortages, and GHG emissions costs. Let

$$\begin{aligned} \chi(\boldsymbol{H}, \boldsymbol{K}, \boldsymbol{\Lambda}, \boldsymbol{M}) = & -\sum_{t\in\mathcal{T}} r^t E^t - \sum_{i\in\mathcal{V}'}\sum_{t\in\mathcal{T}} C_i K_i^t - \sum_{i\in\mathcal{V}'}\sum_{t\in\mathcal{T}} d_i^t \Lambda_i^t \\ & -\sum_{i\in\mathcal{V}'}\sum_{t\in\mathcal{T}} C_i M_i^t - \sum_{(i,j)\in\mathcal{A}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} M \Psi_{ij}^{kt}. \end{aligned}$$

Then the Benders MP can be written as

$$\max_{y,x,\eta} \; -\sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{V}, i<j}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} c_{ij} x_{ij}^{kt} + \eta \tag{47}$$

s.t. (11)–(13) and (15)–(16) are satisfied and

$$\begin{aligned} & \eta + \sum_{(i,j)\in\mathcal{A}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} \Big(M(O_{ij}^{kt} - \Phi_{ij}^{kt} - \Pi_{ij}^{kt} + \Psi_{ij}^{kt}) \\ & \qquad + Q_k(\Pi_{ij}^{kt} + \Psi_{ij}^{kt}) \Big) x_{ij}^{kt} \\ & + \sum_{i\in\mathcal{V}}\sum_{g\in\mathcal{S}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} C_i N_i^{gkt} y_i^{kt} + \sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} Q_k \Omega^{kt} y_0^{kt} \\ & \qquad \geq \chi(\boldsymbol{H}, \boldsymbol{K}, \boldsymbol{\Lambda}, \boldsymbol{M}) \\ & \text{for all } \boldsymbol{E}, \boldsymbol{H}, \boldsymbol{\Gamma}, \boldsymbol{I}, \boldsymbol{K}, \boldsymbol{\Lambda}, \boldsymbol{M}, \boldsymbol{N}, \boldsymbol{\Omega}, \boldsymbol{O}, \boldsymbol{\Phi}, \boldsymbol{\Pi}, \boldsymbol{\Psi} \in \mathcal{P}_\Upsilon. \end{aligned} \tag{48}$$

Note that the Benders MP includes the sub-tour elimination Constraints (12), which are too stringent to allow full enumeration within the Benders MP, hence we relax these constraints and add them dynamically once a sub-tour is detected and hence resolve the master problem iteratively within each iteration to ensure that the solution provided by the master problem does not have any sub-tours.

3.2. Benders decomposition enhancements

Despite the great success of Benders decomposition, the algorithm may require a large number of iterations to converge. To overcome this deficiency in the classical algorithm, several techniques were introduced to accelerate the convergence of the algorithm or reduce the number of iterations. For an extensive review of the literature on the Benders decomposition algorithm, as well as its applications, accelerating techniques, and algorithmic refinements, the interested reader is referred to Rahmaniani et al. (2017). In the following subsections, we provide several techniques and strategies to accelerate the convergence or reduce the number of iterations of the Benders decomposition for the PIRP.

3.2.1. Valid inequalities and symmetry breaking

Valid inequalities can be added to the Benders MP to help it find near-optimal solutions by providing it with some information projected out in the SP. The inclusion of valid inequalities strengthens the Benders MP and accelerates its convergence.

In this paper, we consider adding several valid inequalities and symmetry-breaking constraints to the Benders MP. First, we adapt the families of inequalities presented in Coelho and Laporte (2013a):

$$x_{0i}^{kt} \leq y_i^{kt}, \quad \text{for all } i \in \mathcal{V}, k \in \mathcal{K}, t \in \mathcal{T} \tag{49}$$

$$x_{ij}^{kt} \leq y_i^{kt}, \quad \text{for all } i, j \in \mathcal{V}, k \in \mathcal{K}, t \in \mathcal{T} \tag{50}$$

$$y_i^{kt} \leq y_0^{kt}, \quad \text{for all } i \in \mathcal{V}', k \in \mathcal{K}, t \in \mathcal{T} \tag{51}$$

$$y_0^{kt} \leq y_0^{k-1,t}, \quad \text{for all } i \in \mathcal{V}, k \in \mathcal{K}\backslash\{1\}, t \in \mathcal{T} \tag{52}$$

Inequalities (49) and (50) impose the condition that if the supplier is the immediate successor of customer i along the route planned for vehicle k in time period t, then i must be visited by vehicle k in that period. Constraints (51) ensure that the supplier is visited if any customer is visited by vehicle k in time period t. Constraints (52), which can only be used when the fleet of vehicles is homogeneous, ensure that vehicle k cannot leave the depot if vehicle $k-1$ is not used. Coelho and Laporte (2014) used Constraints (52) in their work; since the vehicle fleet is homogeneous in all of our test instances, we are able to use these constraints as well.

The total number of visits for customer i during the planning horizon must be at least as large as the ratio of its total demand to the lesser of its storage capacity and the capacity of the vehicle with the maximum capacity. Furthermore, the total number of visits for all customers must be at least as large as the ratio of the sum of their total demands to the capacity of the vehicle with maximum capacity. Thus we have the following set of valid inequalities:

$$\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} y_i^{kt} \geq \left\lceil \frac{\sum_{t\in\mathcal{T}} d_i^t}{\min\{\max_k Q_k, C_i\}} \right\rceil \quad \text{for all } i \in \mathcal{V}' \tag{53}$$

$$\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} y_0^{kt} \geq \left\lceil \frac{\sum_{i\in\mathcal{V}}\sum_{t\in\mathcal{T}} d_i^t}{\max_k Q_k} \right\rceil \tag{54}$$

Lastly, for perishable products, the number of visits for a customer should be at least as large as the number of life cycles of the product (the life cycle is the amount of time elapsed between being fresh and expiring) during the planning horizon. Thus a valid inequality for PIRP can be stated as

$$\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} y_i^{kt} \geq \left\lceil \frac{|\mathcal{T}|}{|\mathcal{S}|} \right\rceil, \quad \text{for all } i \in \mathcal{V}' \tag{55}$$

Proposition 3.2. *Constraints (55) are valid cuts for PIRP if $d_i^t > 0$ for $i \in \mathcal{V}'$ and $t \in \mathcal{T}$.*

Proof. Assume that customer i always receives products that are fresh, and assume without loss of generality that the current time period is 1 and the life cycle of the product is s. Then after s time periods (i.e., at $t = 1 + s$) the inventory level of customer i is zero because of the perishability constraint. Thus there should be a supply to customer i at time $t = 1 + s$, and hence a visit to satisfy the constraint that the entire demand of customer i has to be covered. Since $s \le |S|$, a lower bound on the number of visits for each customer can be expressed as $\left\lceil \frac{|\mathcal{T}|}{|\mathcal{S}|} \right\rceil$. □

3.2.2. Pareto-optimal cuts.

Magnanti and Wong (1981) proposed a strategy to accelerate the convergence of the Benders algorithm by adding more stringent, undominated cuts, known as Pareto-optimal cuts. Let $\boldsymbol{Y}$ and $\boldsymbol{X}$ be the set of vectors associated with the decision variables y_i^{kt} and x_{ij}^{kt} obtained after solving the MP. The cut generated from the dual of SP(I, d, q, f, s) from the extreme point

$$(\boldsymbol{E^1}, \boldsymbol{H^1}, \boldsymbol{\Gamma^1}, \boldsymbol{I^1}, \boldsymbol{K^1}, \boldsymbol{\Lambda^1}, \boldsymbol{M^1}, \boldsymbol{N^1}, \boldsymbol{\Omega^1}, \boldsymbol{O^1}, \boldsymbol{\Phi^1}, \boldsymbol{\Pi^1}, \boldsymbol{\Psi^1})$$

dominates the cut generated from the extreme point

$$(\boldsymbol{E^2}, \boldsymbol{H^2}, \boldsymbol{\Gamma^2}, \boldsymbol{I^2}, \boldsymbol{K^2}, \boldsymbol{\Lambda^2}, \boldsymbol{M^2}, \boldsymbol{N^2}, \boldsymbol{\Omega^2}, \boldsymbol{O^2}, \boldsymbol{\Phi^2}, \boldsymbol{\Pi^2}, \boldsymbol{\Psi^2})$$

if and only if the inequality for at least one of these points is strict. Generating Pareto-optimal cuts requires two more steps. The first is to find core points of the convex hull of MP, and the second is to solve an LP problem to find good dual solutions. This extra computational effort is usually compensated for by faster convergence of the Benders algorithm and hence reduces the overall number of iterations. Nevertheless, finding the core points of the convex hull of the MP is challenging. To circumvent this difficulty, researchers have developed several strategies to find the core points or approximations to them. The second step in finding a Pareto-optimal cut is to solve the following subproblem:

$$\begin{aligned}
\max \quad & -\sum_{t\in\mathcal{T}} r^t H^t - \sum_{i\in\mathcal{V}'}\sum_{t\in\mathcal{T}} C_i K_i^t - \sum_{i\in\mathcal{V}'}\sum_{t\in\mathcal{T}} d_i^t \Lambda_i^t \\
& - \sum_{i\in\mathcal{V}'}\sum_{t\in\mathcal{T}} C_i M_i^t - \sum_{i\in\mathcal{V}'}\sum_{g\in\mathcal{S}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} (C_i \tilde{y}_i^{kt}) N_i^{gkt} \\
& - \sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} (Q_k \tilde{y}_0^{kt})\Omega^{kt} - \sum_{(i,j)\in\mathcal{A}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} \left(M(1-\tilde{x}_{ij}^{kt})\right) O_{ij}^{kt} \\
& - \sum_{(i,j)\in\mathcal{A}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} (M\tilde{x}_{ij}^{kt})\Phi_{ij}^{kt} \\
& - \sum_{(i,j)\in\mathcal{A}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} \left((Q_k - M)\tilde{x}_{ij}^{kt} + M\right)\Psi_{ij}^{kt}
\end{aligned} \tag{56}$$

$$\begin{aligned}
\text{s.t.} \quad & -\sum_{t\in\mathcal{T}} r^t H^t - \sum_{i\in\mathcal{V}'}\sum_{t\in\mathcal{T}} C_i K_i^t - \sum_{i\in\mathcal{V}'}\sum_{t\in\mathcal{T}} d_i^t \Lambda_i^t \\
& - \sum_{i\in\mathcal{V}'}\sum_{t\in\mathcal{T}} C_i M_i^t - \sum_{i\in\mathcal{V}'}\sum_{g\in\mathcal{S}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} (C_i \bar{y}_i^{kt}) N_i^{gkt} \\
& - \sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} (Q_k \bar{y}_0^{kt})\Omega^{kt} - \sum_{(i,j)\in\mathcal{A}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} \left(M(1-\bar{x}_{ij}^{kt})\right) O_{ij}^{kt} \\
& - \sum_{(i,j)\in\mathcal{A}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} (M\bar{x}_{ij}^{kt})\Phi_{ij}^{kt} \\
& - \sum_{(i,j)\in\mathcal{A}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} \left((Q_k - M)\bar{x}_{ij}^{kt} + M\right)\Psi_{ij}^{kt} = z_{SP}^*(\bar{y}, \bar{x})
\end{aligned} \tag{57}$$

$$\text{and } \boldsymbol{E}, \boldsymbol{H}, \boldsymbol{\Gamma}, \boldsymbol{I}, \boldsymbol{K}, \boldsymbol{\Lambda}, \boldsymbol{M}, \boldsymbol{N}, \boldsymbol{\Omega}, \boldsymbol{O}, \boldsymbol{\Phi}, \boldsymbol{\Pi}, \boldsymbol{\Psi} \in \Upsilon, \tag{58}$$

where $\tilde{y}$ and $\tilde{x}$ are the core points of the MP convex hull. Constraints (57) and (58) ensure that we select a feasible dual solution that was optimal for the original dual SP objective function (i.e., the objective function (44)).

In addition to these acceleration techniques, we implement some of the classical acceleration strategies, such as a trust region and knapsack inequalities, to accelerate the convergence rate of the Benders decomposition. For further information on these techniques, the interested reader is referred to Santoso et al. (2005). Note that the original dual subproblem is solved first and then the modified subproblem is solved.

4. A meta-heuristic for PIRP

The aforementioned acceleration strategies proved to be effective in significantly closing the gap obtained by the Benders decomposition. Nonetheless, that was the case only for PIRP with a single vehicle. In the case of multiple vehicles, the gap would still be large even with a large number of iterations. Understandably, developing an efficient, exact algorithm to solve PIRP with multiple vehicles can be very challenging; however, developing an exact algorithm is still desired.

Our ultimate objective of developing the Benders decomposition framework is to get high-quality solutions of the PIRP with a guarantee of their quality. However, since Benders decomposition is not able to yield good solutions in and of itself, we can still utilize Benders decomposition to evaluate the quality of a solution or — even better — to improve on an existing solution by running the Benders decomposition algorithm for a certain number of iterations before termination. To this end, we develop a powerful heuristic to provide good solutions that will later be evaluated by Benders decomposition and improved upon. Note that combining Benders decomposition with heuristics is not new in the literature; see Wentges (1996) and Easwaran and Üster (2009). Instead of initially solving the master problem with no Benders cuts, which is the standard implementation of the Benders decomposition algorithm, we start with a feasible solution and generate Benders cuts based on that solution. Specifically, we use the solution generated by a heuristic to generate the initial set of Benders cuts.

The two-stage meta-heuristic we develop is inspired by the meta-heuristic developed by Archetti et al. (2017) to solve the IRP, but with major changes due to the differences between the PIRP with accurate estimation of the fuel cost and the IRP. In the following subsections, we describe the main components of the hybrid two-stage meta-heuristic.

4.1. Phase 1: generation of an initial set of solutions

Even obtaining a feasible solution for PIRP within a reasonable computing time may be a challenging task. The basic idea of Phase 1 is to solve a relaxation of the original PIRP model similar to the meta-heuristic developed by Archetti et al. (2017). To this end, we solve the PIRP model by assuming only a single vehicle, and we relax the integrality requirement of the routing variables (i.e., we allow $x_{ij}^t \in [0, 1]$ instead of requiring that $x_{ij}^t \in \{0, 1\}$). Also, note that the vehicle index is dropped, since all the vehicles are aggregated under one vehicle that has the capacity of all the vehicles combined. Implementing such a procedure provides only one feasible solution; nonetheless, we can get a set of feasible solutions by adding cuts to the relaxed problem to cut the current schedule of visits and hence obtain a new solution. To this end, after each iteration of Phase 1, we add the following cut to the relaxed PIRP formulation and use it to get a new feasible solution:

$$\left(\sum_{i\in\mathcal{V}, t\in\mathcal{T} | \bar{y}_i^t = 1} (1 - y_i^t) + \sum_{i\in\mathcal{V}, t\in\mathcal{T} | \bar{y}_i^t = 0} y_i^t \right) \ge 1, \tag{59}$$

where $\bar{y}_i^t$ is the value of the scheduling variable y_i^t found in the previous iteration. Inequality (59) simply implies that each iteration in Phase 1 will provide a solution different than the previous solution by at least one visit to a customer. Adding the cut (59) iteratively provides a set of feasible solutions. In our implementation, we solve the Phase 1 mathematical model $\lfloor 1.5|\mathcal{K}| \rfloor$ times by adding $\lfloor 1.5|\mathcal{K}| \rfloor - 1$ cuts iteratively. Thus at the end of Phase 1, we have a set of $|\mathcal{T}|$ feasible solutions. Since each solution yields a scheduling decision and a routing decision by assuming that there is one vehicle, we actually obtain a feasible solution to the original multi-vehicle problem. Obtaining a feasible solution to the original problem is done simply by running the adaptive large neighborhood search (ALNS) heuristic due to Demir et al. (2012) on the vehicle routing problem (VRP) instances generated from each solution. For each solution, a VRP instance is generated based on the values of the variables y_i^t and q_i^{gt}. For each $t \in \mathcal{T}$, we identify the set of customers with $y_i^t = 1$ and use the values of $\sum_{g \in \mathcal{S}} q_i^{gt}$ as the demand of customer i. In this way a VRP instance is generated for each $t \in \mathcal{T}$. Thus the ALNS heuristic generates the values of the decision variables y_i^{kt}, x_{ij}^{kt}, and q_i^{gkt}, and hence a feasible solution for the original PIRP formulation.

4.2. Phase 2: GRASP

The greedy randomized adaptive search procedure (GRASP) is an iterative procedure developed in the late 1980s by Feo and Resende (1989) in which each iteration consists of two phases. In the construction phase a feasible solution is produced, and in the local search phase a local optimum in the neighborhood of the constructed solution is sought. More information on GRASP can be found in Festa and Resende (2011).

For the construction phase, we use the set of feasible solutions found at the end of Phase 1, hence we can focus on developing the second phase, with the GRASP algorithm, to get a high-quality solution. We define a set of moves from the current solution to a solution in the neighborhood thereof. In what follows, we provide a brief description of the neighborhood moves inspired by the search procedure used by Archetti et al. (2017), which are basically remove, insert, and move operations that change the scheduling of visits to customers.

We now describe the remove operations used in our implementation (i.e., operations that lead to removal of a visit to one customer).

1. **Most visited:** The customers are ranked by the frequency of visits scheduled for each of them (i.e., the ratio of the customer's total number of visits to the number of time periods in the planning horizon). This operator starts with an empty removal list. It fills the removal list of the most visited with the 40% of customers that have the highest visit frequencies. Then the algorithm picks one of the customers in the list at random. The selected customer will get a new schedule of visits similar to the previous schedule but with one visit removed. A removable visit is defined as a visit whose removal maintains the feasible solution with respect to all constraints. In case there is more than one visit that could be removed, one of those visits is picked at random to be removed from the set of scheduled visits.
2. **Routing cost:** This operation removes a visit of a customer that has a high routing cost. The routing cost of each customer is defined as the ratio of the sum of the customer's routing costs for individual time periods to the number of time periods in the planning horizon. The routing cost of each customer in each time period is calculated as the difference between the total routing cost when the customer visit is scheduled and the total routing cost when the customer visit is removed. Note that the routing cost is a vector with $|\mathcal{T}| * |\mathcal{V}'|$ components. After constructing the routing cost vectors, the routing cost removal list of customers along with their visits is sorted and the 40% of customers with the highest routing costs are selected. Then one of the customers in that list is selected for a change in its schedule by having a visit removed.
3. **Minimum delivered quantity:** For each time period, the two customers with the lowest quantities of product to be delivered are selected and added to the minimum delivered quantity removal list. After constructing that removal list, the heuristic picks one of the customers at random and removes one visit from its schedule. Two customers are selected (rather than one) in order to increase the size of the list.
4. **Random selection:** This operator selects a customer at random and finds a visit that can be removed. The selected visit to be removed is the visit that leads to the greatest improvement in the objective function.

Note that each removal is implemented provided that the new schedule of the selected customer maintains feasibility with respect to all constraints (vehicle capacity, demand satisfaction, and inventory capacity). When a visit is removed, the quantity of product that was to be delivered on that visit has to be delivered on some other visit(s). In our implementation, we first try to move the delivery to the visit preceding the removed visit while respecting all the constraints. Otherwise, we try to split the delivery between the visits preceding and following the removed visit.

We now describe the insert operations used in our implementation.

1. **Least visited:** The customers are ranked by the frequency of visits scheduled for them (i.e., the ratio of each customer's number of visits to the number of time periods in the planning horizon). This operator starts with an empty insert list. It fills the insert list with the 40% of customers that have the lowest visit frequencies. Then, the algorithm picks one of the customers in the list at random and adds a visit to that customer at random.
2. **Holding cost:** This operator adds visits to customers that have a high inventory cost. The inventory cost of each customer is defined as the ratio of the sum of the customer's holding costs for individual time periods to the number of time periods in the planning horizon. The list of customers is sorted by holding cost, and the 40% of customers with the highest costs are selected. Then one of the customers is selected for a change in its schedule by having a visit added.
3. **Maximum delivered quantity:** For each time period, the two customers with the largest quantities of product to be delivered are selected and added to the list. After constructing the list of customers with the largest quantities of product, the algorithm picks one of them at random and adds one visit to the selected customer.
4. **Random selection:** This operator selects a customer at random and adds a visit to its schedule.

For each of the insert and remove operations, the quantities of product that are to be delivered to the selected customer on the individual visits have to be updated. In our implementation we solve a small LP problem to decide the quantities delivered to each customer (respecting vehicle capacities, customer inventory capacities, and quantities available at the supplier site). The small LP problem objective function is concerned with holding costs and revenues.

Lastly, the move operation tries to change the schedule for a customer by changing the scheduling of one of its visits (i.e., changing a visit from the original time period t' to some time period t for which no visit to that customer was originally scheduled). The move operator is implemented in the same fashion as in Archetti et al. (2017): Select a customer i, and time periods t' and t; then remove the visit to i in time period t', and add a visit to i in time period t.

For insert and move operations, routing decisions are updated by inserting the new visit to the selected customer in the selected time period by choosing a route and a position within that route which is based on the minimum cost. After calculating the value of the objective function for each of the selected remove, insert, and move operations, the one with the best value of the objective function is selected and the current solution is updated accordingly.

We define a counter (called COUNT) as the number of times a solution found by GRASP so far was better than the best solution found before that, and we set an upper limit on the value of COUNT. Once the number of times the best solution found so far is equal to the maximum value we set, we run the ALNS heuristic due to Demir et al. (2012) to find near-optimal routing decisions for each time period and update the routing decisions and the value of the objective function accordingly. In our implementation, we set the upper limit on COUNT to 8. Whenever the actual value of COUNT hits 8, we reset it to 0 and run the ALNS heuristic. As to the termination criteria of GRASP, we set an upper limit on the number of iterations of GRASP to be performed when there is no improvement over the best value of the objective function found so far. We set the maximum number of iterations to $15 * |\mathcal{V}|$. Another termination criterion we use is based on the total time spent by the meta-heuristic: If the total time exceeds three hours, the search is terminated.

5. Computational study

In this section, we present computational test and analysis results with an emphasis on:

- Testing the computational efficiency and effectiveness of the proposed solution approach. Specifically, we first examine the effects of algorithmic enhancements on the overall approach to solving PIRP using Benders decomposition and the effectiveness of the developed meta-heuristic for solving large-scale instances of PIRP.
- Examination of the effects—on the solution structure—of using accurate estimation of the fuel cost and incorporating the GHG cost. In this context, we first present a PIRP model that uses the distance traveled as a measure of fuel consumption and compare it to the proposed model. Then we provide an analysis of the delivery, logistics, environmental, and inventory performance indicators.

We randomly generated instances to assess the performance of the developed algorithms for a wide range of situations. Our testbed was composed of instances generated with the following parameters:

- Number of customers, $|\mathcal{V}'|$: 10, 15, 20, 25, 30, 35, 40, 45, 50, 60
- Number of periods in the planning horizon, p: 3, 6
- Number of vehicles, $|\mathcal{K}|$: 1, 2, 3, 4
- Maximum age of the products, $|\mathcal{S}|$: 2, 3
- Demand d_i^t: randomly selected integer from the interval [30, 300]
- Positions (x, y) of the supplier and customers: randomly selected from the interval [0, 150]
- Inventory capacity of customer i, C_i: $R \max_t d_i^t$, where R is randomly selected from the set {2, 3}
- Inventory holding cost h_i: randomly generated number in the interval [0.1,1.0]
- Vehicle capacities Q_k: $2 * \sum_{i \in \mathcal{V}, t \in \mathcal{T}} d_i^t / (p * |\mathcal{K}|)$
- For each $g \in \mathcal{S}$, profit from the sale of a unit of product of age g, u^g: a random integer in the interval [7,15] for each age $g \in \mathcal{S}$
- Fuel cost: $1.5 per liter
- Vehicle speed: 50 miles per hour

The cost of emitting one ton of GHG was set to $100, which is consistent with some values found in other studies (Price et al., 2007). All the code was written using MATLAB. The master problem and dual subproblems were solved using Gurobi 7.5.2 on a laptop computer with a 2.8 GHz Intel Core processor and 16GB of RAM.

5.1. Performance of algorithmic enhancements

For computational testing of algorithmic enhancements, we generated 3 instances of data with 10 customers, 6 planning periods, and a product with a lifetime of 3 periods. We also varied the number of vehicles from 1 to 3. Table 2 reports the computational time in seconds and the calculated gap for each methodology.

Table 2 displays the computational results for different algorithms. The first column shows the label of the instance, encoded as $|\mathcal{V}'|$-p-instance#-$|\mathcal{K}|$. The second through fifth sets of columns show the computational time in seconds and the gap percentage for each setting. The gap is defined as $[(\text{UB} - \text{LB})/\text{LB}] * 100$. A gap of 100% indicates that no feasible solution was found at termination. “Standard Benders” refers to the standard Benders decomposition algorithm with no acceleration strategies. Benders+S1 refers to Benders decomposition with S1 acceleration strategies (namely, MP enhanced with a set of valid inequalities, Pareto optimality cuts, knapsack inequalities, and a trust region). Benders+S1+WU refers to Benders decomposition with S1 acceleration strategies and the warm-up start meta-heuristic. GAMS refers to the use of the GAMS modeling language to solve the PIRP directly using the Gurobi solver. As shown in Table 2, the standard Benders decomposition technique was unable to find any feasible solution within 30 iterations. When S1 acceleration strategies were utilized, there was significant improvement in solving the PIRP with a single vehicle; however, S1 acceleration strategies were unable to achieve significant improvement for more than one vehicle. For the case of Benders decomposition with S1 acceleration strategies and the warm-up start meta-heuristic, significant improvement was achieved. In the last set of columns in Table 2, we report the computational results as found by GAMS using the Gurobi solver. Note that for the cases where the number of vehicles was more than one, no feasible solution was found within 3 hours and hence the gap result is entered as N.A. Clearly, the results found by GAMS highlight the difficulty of the problem.

Note that since GAMS solver does not enable the user to add the sub-tour cuts dynamically, we used the MTZ formulation to ensure that the GAMS solution does not include sub-tours.

5.2. Computation time analyses

In this subsection, we evaluate the solution quality and the performance of the developed algorithm when the parameters and settings are changed for different sizes of instances. There were a total of 120 instances.

We solved the instances with the Benders algorithm combined with the warm-up start meta-heuristic and S1 acceleration strategies. The results are shown in Tables 3–12. The format of Tables 3–12 is as follows. Columns 1–4 display the number of nodes in the network (including the supplier) $|\mathcal{V}|$, the length of the planning horizon (p), the number of vehicles $|\mathcal{K}|$, and the maximum age of product $|\mathcal{S}|$. The fifth column displays the computational time (in seconds) spent on solving MP and SP of the Benders decomposition. The sixth column displays the computational time (in seconds) in the meta-heuristic. The seventh and eighth columns display the total time spent by the algorithm (in seconds) and the

Table 2
Summary of the computational results for different algorithms.

Instance	Standard Benders		Benders+S1		Benders+S1 +WU		GAMS	
	Time (s)	Gap (%)	Time (s)	Gap (%)	Time (s)	Gap (%)	Time (s)	Gap (%)
10-6-1-1	6.43	100	8.20	1.17	22.99	1.81	10,800	1.97
10-6-1-2	16.14	100	21.21	100	14.62	2.62	10,800	N.A.
10-6-1-3	34.93	100	55.28	28.37	22.14	2.75	10,800	N.A.
10-6-2-1	6.21	100	6.52	1.33	22.48	1.50	10,800	2.37
10-6-2-2	18.64	100	22.70	18.23	13.97	2.48	10,800	N.A.
10-6-2-3	33.6	100	43.31	100	17.15	2.95	10,800	N.A.
10-6-3-1	8.10	100	6.74	1.12	22.85	1.48	10,800	4.38
10-6-3-2	17.42	100	28.61	100	11.61	2.53	10,800	N.A.
10-6-3-3	31.04	100	49.32	100	17.13	2.81	10,800	N.A.

Table 3
Computational results for the PIRP with 10 customers.

$\|\mathcal{V}\|$	p	$\|\mathcal{K}\|$	$\|\mathcal{S}\|$	Benders (s)	WU (s)	Total time (s)	Gap (%)
11	3	2	2	8.62	4.57	13.20	1.33
11	3	2	3	9.57	6.37	15.94	1.87
11	3	3	2	11.02	6.24	17.26	2.81
11	3	3	3	11.00	6.69	17.70	3.08
11	3	4	2	11.02	8.92	19.93	2.82
11	3	4	3	12.79	7.62	20.41	3.70
11	6	2	2	9.42	7.30	16.72	2.12
11	6	2	3	11.96	9.59	21.55	3.31
11	6	3	2	30.37	8.08	38.45	2.75
11	6	3	3	13.64	11.28	24.92	2.89
11	6	4	2	20.48	13.10	33.58	2.53
11	6	4	3	32.33	15.94	48.27	4.14
Avg.						23.99	2.78
Max.						48.27	4.14
Min.						13.20	1.33

Table 4
Computational results for the PIRP with 15 customers.

$\|\mathcal{V}\|$	p	$\|\mathcal{K}\|$	$\|\mathcal{S}\|$	Benders (s)	WU (s)	Total time (s)	Gap (%)
16	3	2	2	7.64	14.56	22.19	3.61
16	3	2	3	7.31	14.18	21.49	3.94
16	3	3	2	15.30	1.92	17.22	3.02
16	3	3	3	15.89	13.16	29.04	3.28
16	3	4	2	21.94	18.49	40.43	3.29
16	3	4	3	22.42	13.42	35.84	3.28
16	6	2	2	21.38	19.89	41.27	4.61
16	6	2	3	14.50	22.06	36.56	4.69
16	6	3	2	17.91	18.58	36.49	4.09
16	6	3	3	32.46	26.23	58.69	5.78
16	6	4	2	52.27	19.02	71.29	2.93
16	6	4	3	53.85	34.11	87.96	3.15
Avg.						41.35	3.81
Max.						87.96	5.78
Min.						17.22	2.93

Table 5
Computational results for the PIRP with 20 customers.

$\|\mathcal{V}\|$	p	$\|\mathcal{K}\|$	$\|\mathcal{S}\|$	Benders (s)	WU (s)	Total time (s)	Gap (%)
21	3	2	2	10.17	27.50	37.67	6.16
21	3	2	3	11.61	17.98	29.59	5.84
21	3	3	2	36.26	34.77	71.04	3.79
21	3	3	3	30.09	20.66	50.76	4.48
21	3	4	2	26.54	43.04	69.58	3.58
21	3	4	3	30.40	22.61	53.01	4.73
21	6	2	2	39.33	236.03	275.36	6.23
21	6	2	3	22.73	66.93	89.66	7.22
21	6	3	2	53.03	38.41	91.44	4.57
21	6	3	3	60.46	52.79	113.25	4.44
21	6	4	2	78.55	120.13	198.68	3.56
21	6	4	3	64.43	59.72	124.16	6.01
Avg.						100.35	5.05
Max.						275.36	7.22
Min.						29.59	3.56

Table 6
Computational results for the PIRP with 25 customers.

$\|\mathcal{V}\|$	p	$\|\mathcal{K}\|$	$\|\mathcal{S}\|$	Benders (sec)	WU (sec)	Total time (sec)	Gap (%)
26	3	2	2	27.08	438.51	465.59	8.47
26	3	2	3	28.70	38.30	67.00	8.63
26	3	3	2	38.33	454.83	493.17	5.62
26	3	3	3	49.29	67.58	116.87	5.19
26	3	4	2	78.89	822.88	901.77	5.38
26	3	4	3	99.74	36.19	135.93	3.18
26	6	2	2	46.20	316.30	362.50	6.87
26	6	2	3	66.58	139.31	205.88	9.07
26	6	3	2	91.39	427.74	519.13	6.24
26	6	3	3	111.30	85.03	196.34	5.85
26	6	4	2	171.67	453.02	624.69	3.73
26	6	4	3	205.80	111.42	317.23	6.00
Avg.						367.17	6.19
Max.						901.77	9.07
Min.						67.00	3.18

gap, respectively. The Benders algorithm was terminated after 20 iterations.

As demonstrated by Tables 3–12, the performance of the developed algorithm is very high when it comes to computational time and quality of the solution obtained, given the complexity of the problem. Also, it is very clear that solving instances where $|\mathcal{S}| = 2$ is more difficult than solving the same size instance with $|\mathcal{S}| = 3$. The intuition behind this is that when $|\mathcal{S}|$ gets close to 1, the problem essentially reduces to solving several instances of VRP, one for each time period within the planning horizon. In the scientific literature, it is well known that algorithms such as branch-price-and-cut perform better than Benders decomposition when solving VRP.

5.3. Impact of the accurate estimation of the fuel cost

To highlight the importance of utilizing a model that accurately calculates fuel consumption levels by incorporating vehicle load and speed in PIRP settings, we compare the solutions obtained by the model presented in this study to those for traditional PIRP models that use just the distance traveled to estimate fuel consumption levels. To this end, similar to what is presented in Darvish et al. (2017), we employ some key performance indicators (KPIs). These KPIs are classified as one of four types: inventory, delivery, environmental, and logistics. Inventory KPIs include average inventory levels and costs to customers. Delivery KPIs measure the

Table 7
Computational results for the PIRP with 30 customers.

$\|\mathcal{V}\|$	p	$\|\mathcal{K}\|$	$\|\mathcal{S}\|$	Benders (s)	WU (s)	Total time (s)	Gap (%)
31	3	2	2	27.54	832.00	859.54	7.70
31	3	2	3	33.38	52.54	85.92	6.20
31	3	3	2	78.11	1570.00	1648.11	5.81
31	3	3	3	77.15	89.71	166.87	4.73
31	3	4	2	146.78	670.08	816.86	5.33
31	3	4	3	110.05	101.47	211.52	3.78
31	6	2	2	34.20	216.30	250.50	10.17
31	6	2	3	69.12	93.18	162.30	9.57
31	6	3	2	199.65	110.58	310.24	9.66
31	6	3	3	223.71	121.97	345.69	7.22
31	6	4	2	292.58	768.48	1061.06	9.42
31	6	4	3	373.36	134.27	507.63	7.13
Avg.						535.52	7.63
Max.						1648.11	10.17
Min.						85.92	3.78

Table 8
Computational results for the PIRP with 35 customers.

$\|\mathcal{V}\|$	p	$\|\mathcal{K}\|$	$\|\mathcal{S}\|$	Benders (s)	WU (s)	Total time (s)	Gap (%)
36	3	2	2	51.97	2280.00	2331.97	6.84
36	3	2	3	50.14	139.27	189.41	8.56
36	3	3	2	76.34	1620.00	1696.34	4.14
36	3	3	3	89.64	170.00	259.64	7.10
36	3	4	2	160.75	1250.00	1410.75	6.08
36	3	4	3	132.17	86.08	218.25	5.87
36	6	2	2	213.97	1430.00	1643.97	9.78
36	6	2	3	289.20	146.27	435.47	9.51
36	6	3	2	346.97	2150.00	2496.97	6.19
36	6	3	3	451.09	196.06	647.15	8.39
36	6	4	2	521.69	1400.00	1921.69	5.99
36	6	4	3	607.13	332.87	940.00	5.06
Avg.						1182.63	6.96
Max.						2496.97	9.78
Min.						189.41	4.14

Table 9
Computational results for the PIRP with 40 customers.

$\|\mathcal{V}\|$	p	$\|\mathcal{K}\|$	$\|\mathcal{S}\|$	Benders (s)	WU (s)	Total time (s)	Gap (%)
41	3	2	2	94.15	5200.00	5294.15	12.14
41	3	2	3	65.08	89.15	154.22	11.25
41	3	3	2	129.27	3180.00	3309.27	8.83
41	3	3	3	135.19	257.70	392.89	7.63
41	3	4	2	177.03	2490.00	2667.03	7.04
41	3	4	3	204.72	117.73	322.45	5.98
41	6	2	2	254.53	5160.00	5414.53	7.67
41	6	2	3	168.65	202.18	370.83	6.91
41	6	3	2	636.34	2930.00	3566.34	4.81
41	6	3	3	735.80	307.62	1043.41	4.92
41	6	4	2	622.78	812.27	1435.05	3.01
41	6	4	3	660.16	374.28	1034.44	5.70
Avg.						2083.72	7.16
Max.						5414.53	12.14
Min.						154.22	3.01

Table 10
Computational results for the PIRP with 45 customers.

$\|\mathcal{V}\|$	p	$\|\mathcal{K}\|$	$\|\mathcal{S}\|$	Benders (s)	WU (s)	Total time (s)	Gap (%)
46	3	2	2	141.28	5260.00	5401.28	14.89
46	3	2	3	77.76	134.11	211.87	13.61
46	3	3	2	214.83	6260.00	6474.83	9.86
46	3	3	3	167.03	223.62	390.65	9.92
46	3	4	2	246.68	7390.00	7636.68	9.52
46	3	4	3	286.57	619.32	905.89	7.31
46	6	2	2	564.51	5590.00	6154.51	10.44
46	6	2	3	254.43	406.97	661.40	9.71
46	6	3	2	1022.65	416.09	1438.74	9.01
46	6	3	3	1103.71	369.65	1473.35	9.41
46	6	4	2	793.78	4180.00	4973.78	12.62
46	6	4	3	1167.78	848.82	2016.59	8.87
Avg.						3144.97	10.43
Max.						7636.36	14.89
Min.						211.87	7.31

Table 11
Computational results for the PIRP with 50 customers.

$\|\mathcal{V}\|$	p	$\|\mathcal{K}\|$	$\|\mathcal{S}\|$	Benders (s)	WU (s)	Total time (s)	Gap (%)
51	3	2	2	114.71	7290.00	7404.71	14.89
51	3	2	3	135.92	161.28	297.20	13.61
51	3	3	2	266.87	10800.00	11066.87	9.86
51	3	3	3	276.03	214.14	490.17	9.92
51	3	4	2	245.19	10800.00	11045.19	9.52
51	3	4	3	309.68	206.97	516.65	7.31
51	6	2	2	346.86	6890.00	7236.86	10.44
51	6	2	3	443.79	856.57	1300.36	9.71
51	6	3	2	584.45	3450.00	4034.45	9.01
51	6	3	3	744.80	861.22	1606.02	9.41
51	6	4	2	1429.65	4690.00	6119.65	12.62
51	6	4	3	2018.53	785.99	2804.52	8.87
Avg.						4493.55	10.43
Max.						11066.87	14.89
Min.						297.20	7.31

Table 12
Computational results for the PIRP with 60 customers.

$\|\mathcal{V}\|$	p	$\|\mathcal{K}\|$	$\|\mathcal{S}\|$	Benders (s)	WU (s)	Total time (s)	Gap (%)
61	3	2	2	179.72	10800.00	10979.72	15.03
61	3	2	3	200.41	295.22	495.63	14.07
61	3	3	2	351.10	10800.00	11151.10	13.48
61	3	3	3	458.48	550.54	1009.01	12.23
61	3	4	2	459.94	10800.00	11259.94	10.25
61	3	4	3	616.12	467.74	1083.87	7.76
61	6	2	2	522.44	9290.00	9812.44	15.33
61	6	2	3	542.02	626.99	1169.01	14.27
61	6	3	2	1159.33	10800.00	11959.33	14.06
61	6	3	3	1195.67	736.21	1931.88	11.71
61	6	4	2	1952.81	10800.00	12752.81	10.70
61	6	4	3	2946.21	890.09	3836.30	9.87
Avg.						6453.42	12.40
Max.						12752.81	15.33
Min.						495.63	7.76

fuel cost and the total distance traveled by the delivery vehicles on all trips. Environmental KPIs measure GHG emissions due to delivery activities. And finally, logistics KPIs measure vehicle fill, average load, and empty running distance. Lastly, note that the generated instances use the same parameters as the instances generated in Tables 3–12, with a minor difference when it comes to the inventory holding cost. For the sake of consistency, that cost was set to $0.1 per unit of product per time period across all customers.

The vehicle fill is the ratio of the total load of the vehicle when it departs from the supplier's depot to its capacity. The average load is the ratio of the total load to the number of trips. The empty running KPI is the distance the vehicle travels to get back to the supplier's depot after the last delivery, since all the vehicles return to the depot empty. To provide a more consistent comparison of the two models, we included the cost due to GHG emissions in the traditional model. For the fuel cost, however, we used the accurate estimation procedure utilized in this study to estimate the fuel cost as a function of vehicle load, speed, and distance traveled.

We ran the comparison experiments on instances with 15 customers, 6 planning periods, a maximum product age of 3 periods, either 1, 2, or 3 vehicles, and three different instances for diversification. We labeled the instances by encoding them as $|\mathcal{V}'|$-p-

Table 13
Inventory cost (IC) and emissions KPIs.

Instance	IC		Emissions		IC	Emissions
	M1	M2	M1	M2	Dev (%)	Dev (%)
15-6-1-1	455.9	496.0	1430	1492	−8.80	−4.33
15-6-1-2	448.2	411.9	1680	1749	8.10	−4.06
15-6-1-3	583.9	451.3	1811	2015	22.71	−11.27
15-6-2-1	453.1	431.0	1673	1698	4.88	−1.50
15-6-2-2	341.4	345.5	1513	1643	−1.20	−8.64
15-6-2-3	454.0	463.7	2130	2297	−2.14	−7.80
15-6-3-1	467.1	410.7	1803	1979	12.07	−9.76
15-6-3-2	445.4	441.7	2067	2134	0.83	−3.25
15-6-3-3	559.7	576.6	2479	2527	−3.02	−1.96
Max.					22.71	−1.5
Ave.					3.71	−5.84
Min.					−8.8	−11.27

Table 14
Delivery KPIs: fuel cost (FC) and total distance traveled (TDT).

Instance	FC		TDT		FC	TDT
	M1	M2	M1	M2	Dev (%)	Dev (%)
15-6-1-1	3795	3960	3057	2995	−4.33	2.03
15-6-1-2	4461	4643	3634	3708	−4.06	−2.04
15-6-1-3	4807	5349	3622	3817	−11.27	−5.38
15-6-2-1	4442	4509	3567	3594	−1.50	−0.76
15-6-2-2	4015	4363	3266	3417	−8.64	−4.62
15-6-2-3	5656	6098	4662	4967	−7.80	−6.54
15-6-3-1	4787	5254	3801	3672	−9.76	3.39
15-6-3-2	5488	5667	4510	4648	−3.25	−3.06
15-6-3-3	6580	6709	5352	5388	−1.96	−0.67
Max.					−1.5	3.39
Ave.					−5.84	−1.96
Min.					−11.27	−6.54

Table 15
Logistics KPIs: vehicle fill (VF) and average load (AL).

Instance	VF		AL		VF	AL
	M1	M2	M1	M2	Dev (%)	Dev (%)
15-6-1-1	0.36	0.60	893	1470	−64.84	−64.6
15-6-1-2	0.76	0.71	925	868	6.23	6.23
15-6-1-3	0.75	0.75	1867	1867	0.00	0.00
15-6-2-1	0.86	0.75	1050	919	12.60	12.5
15-6-2-2	0.50	0.62	595	732	−23.21	−23.07
15-6-2-3	0.91	0.84	715	663	7.36	7.34
15-6-3-1	0.41	0.75	1013	1867	−84.28	−84.29
15-6-3-2	0.90	0.82	707	643	9.01	9.09
15-6-3-3	0.90	0.90	747	747	0.00	0.00
Max.					12.6	12.5
Ave.					−15.23	−15.20
Min.					−84.28	−84.29

instance#-$|\mathcal{K}|$. (For instance, 15-6-2-2 means the second instance with 15 customers, six planning periods, and two trucks.) Results of the comparison are displayed in Tables 13–16. The format of the tables is as follows: the first column displays the instance label, and the next set of columns display the values of the KPIs. Each KPI column is divided into two sub-columns; the first sub-column presents the KPI value found by the model presented in this study (denoted by M1), and the second sub-column presents the KPI value found by the traditional model (denoted by M2). The last set of columns show the deviation between the two models, as $[(M1 - M2)/M1] * 100$, hence a negative deviation indicates that the M1 value is less than the M2 value.

The inventory and emissions KPIs are presented in Table 13. Note that the model presented in this study does not always achieve smaller inventory levels or inventory costs; nonetheless, the levels of GHG emissions are always smaller. This point highlights the importance of utilizing a vendor-managed-inventory system, not only to minimize the overall logistics costs but also to minimize GHG emissions due to transportation. Also, it should be noted that having different inventory levels and costs obtained by the two models immediately implies that different scheduling decisions were implemented, as we show in the analysis of the delivery and logistics KPIs. Furthermore, since it is not the case that inventory costs are always lower under one of the two models, this illustrates that the cost structure of PIPR is complicated and has several dependencies, and that optimization of all of these components simultaneously is crucial.

Table 16
Logistics KPIs (continued): empty running distance (ER).

Instance	ER		ER
	M1	M2	Dev (%)
15-6-1-1	706	679	3.82
15-6-1-2	1080	988	8.52
15-6-1-3	708	369	47.88
15-6-2-1	1122	1007	10.25
15-6-2-2	560	453	19.11
15-6-2-3	1673	1005	39.93
15-6-3-1	783	403	48.53
15-6-3-2	1357	1014	25.28
15-6-3-3	1933	1458	24.57
Max.			48.53
Ave.			25.32
Min.			3.82

Table 14 shows the delivery KPIs. Clearly, solutions found by M1 always indicate a lower fuel cost than those in M2, which implies that traditional models may fail to provide an accurate estimate of the fuel cost unless the vehicle load is incorporated. On the other hand, the total distance traveled (TDT) is not always lower, which implies that minimizing fuel consumption is not always associated with minimizing the total distance traveled during delivery operations, since the fuel consumption is a function of vehicle load as well. Furthermore, this leads to the conclusion that different routing decisions were obtained by the two models, which is consistent with a previous study (Bektaş and Laporte, 2011) in the context of VRP.

Tables 15 and 16 report the logistics KPIs. As shown in Table 15, the average vehicle fill (VF) is not always lower in M1 than in M2. The fluctuation in the average vehicle fill is due to the fact in some cases we observe that a vehicle makes a trip for one delivery only and hence the average vehicle fill tends to be lower overall, whereas the average vehicle fill in the traditional model is stable and has less deviation across instances. On the other hand, Tables 15 and 16 show that the average load (AL) and empty running distance (ER) are always larger in M1, since the vehicle load plays a role in the objective function and hence the model optimizes vehicle load, number of trips, and distance traveled. ER is larger in M1 than in M2, because the vehicle load affects the objective function and a vehicle with a smaller load consumes less fuel.

6. Conclusions

We have addressed the perishable inventory routing problem with accurate estimation of the fuel cost. To solve this problem, we have proposed two different algorithms: Benders decomposition and a two-stage meta-heuristic. The Benders decomposition is further improved by several computational enhancements, namely, valid inequalities and a warm-up start heuristic. The computational results show that the Benders decomposition enhanced with several acceleration strategies is efficient in handling small to medium instances, while the two-stage meta-heuristic is capable of handling larger instances with 60 customers and 6 time periods. To demonstrate the practical benefit of our PIRP model with accurate estimation of the fuel cost, we show that savings in fuel and hence GHG can be achieved by utilizing the model.

References

Adulyasak, Y., Cordeau, J.-F., Jans, R., 2013. Formulations and branch-and-cut algorithms for multivehicle production and inventory routing problems. INFORMS J. Comput. 26 (1), 103–120.

Adulyasak, Y., Cordeau, J.-F., Jans, R., 2015. Benders decomposition for production routing under demand uncertainty. Oper. Res. 63 (4), 851–867.

Akkerman, R., Farahani, P., Grunow, M., 2010. Quality, safety and sustainability in food distribution: a review of quantitative operations management approaches and challenges. OR Spectrum 32 (4), 863–904.

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

Archetti, C., Boland, N., Grazia Speranza, M., 2017. A matheuristic for the multivehicle inventory routing problem. INFORMS J. Comput. 29 (3), 377–387.

Bektaş, T., Laporte, G., 2011. The pollution-routing problem. Transp. Res. Part B 45 (8), 1232–1250.

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

Benders, J.F., 1962. Partitioning procedures for solving mixed-variables programming problems. Numerische Mathematik 4 (1), 238–252.

Christiansen, M., 1999. Decomposition of a combined inventory and time constrained ship routing problem. Transp. Sci. 33 (1), 3–16.

Coelho, L.C., Cordeau, J.-F., Laporte, G., 2013. Thirty years of inventory routing. Transp. Sci. 48 (1), 1–19.

Coelho, L.C., Laporte, G., 2013. A branch-and-cut algorithm for the multi-product multi-vehicle inventory-routing problem. Int. J. Prod. Res. 51 (23–24), 7156–7169.

Coelho, L.C., Laporte, G., 2013. The exact solution of several classes of inventory-routing problems. Comput. Oper. Res. 40 (2), 558–565.

Coelho, L.C., Laporte, G., 2014. Optimal joint replenishment, delivery and inventory management policies for perishable products. Comput. Oper. Res. 47, 42–52.

Contreras, I., Cordeau, J.-F., Laporte, G., 2011. Benders decomposition for large-scale uncapacitated hub location. Oper. Res. 59 (6), 1477–1490.

Cordeau, J.-F., Soumis, F., Desrosiers, J., 2001. Simultaneous assignment of locomotives and cars to passenger trains. Oper. Res. 49 (4), 531–548.

Côté, J.-F., Dell'Amico, M., Iori, M., 2014. Combinatorial Benders' cuts for the strip packing problem. Oper. Res. 62 (3), 643–661.

Custódio, A.L., Oliveira, R.C., 2006. Redesigning distribution operations: a case study on integrating inventory management and vehicle routes design. Int. J. Logistics 9 (2), 169–187.

Darvish, M., Archetti, C., Coelho, L.C., 2017. Minimizing Emissions in Integrated Distribution Problems. Technical Report CIRRELT-2017-41. Interuniversity Research Centre on Enterprise Networks, Logistics, and Transportation.

Demir, E., Bektaş, T., Laporte, G., 2012. An adaptive large neighborhood search heuristic for the pollution-routing problem. Eur. J. Oper. Res. 223 (2), 346–359.

Desaulniers, G., Rakke, J.G., Coelho, L.C., 2015. A branch-price-and-cut algorithm for the inventory-routing problem. Transp. Sci. 50 (3), 1060–1076.

Easwaran, G., Üster, H., 2009. Tabu search and benders decomposition approaches for a capacitated closed-Loop supply chain network design problem. Transp. Sci. 43 (3), 301–320.

Feo, T.A., Resende, M.G., 1989. A probabilistic heuristic for a computationally difficult set covering problem. Oper. Res. Lett. 8 (2), 67–71.

Festa, P., Resende, M.G., 2011. GRASP: Basic components and enhancements. Telecommun. Syst. 46 (3), 253–271.

Hemmelmayr, V., Doerner, K.F., Hartl, R.F., Savelsbergh, M.W., 2010. Vendor managed inventory for environments with stochastic product usage. Eur. J. Oper. Res. 202 (3), 686–695.

Hooker, J.N., 2007. Planning and scheduling by logic-based benders decomposition. Oper. Res. 55 (3), 588–602.

Huang, Y.A., Weber, C.L., Matthews, H.S., 2009. Categorization of scope 3 emissions for streamlined enterprise carbon footprinting. Environ. Sci. Technol. 43 (22), 8509–8515.

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

Lee, H.L., Seungjin, W., 2008. The whose, where and how of inventory control design. Supply Chain Manage. Rev. 12 (8), 22–29.

Magnanti, T.L., Wong, R.T., 1981. Accelerating benders decomposition: algorithmic enhancement and model selection criteria. Oper. Res. 29 (3), 464–484.

Minkoff, A.S., 1993. A Markov decision model and decomposition heuristic for dynamic vehicle dispatching. Oper. Res. 41 (1), 77–90.

MOVES, 2018. Motor Vehicle Emissions Simulator. United States Environmental Protection Agency. https://www.epa.gov/moves/latest-version-motor-vehicle-emission-simulator-moves

Nagurney, A., Masoumi, A.H., Yu, M., 2012. Supply chain network operations management of a blood banking system with cost and risk minimization. Comput. Manage. Sci. 9 (2), 205–231.

Price, R., Thornton, S., Nelson, S., 2007. The Social Cost of Carbon and the Shadow Price of Carbon: What they are, and How to Use them in Economic Appraisal in the UK. Technical Report. Economics Group, Department for Environment, Food, and Rural Affairs, London.

Qu, W.W., Bookbinder, J.H., Iyogun, P., 1999. An integrated inventory–transportation system with modified periodic policy for multiple products. Eur. J. Oper. Res. 115 (2), 254–269.

Rahmaniani, R., Crainic, T.G., Gendreau, M., Rei, W., 2017. The benders decomposition algorithm: a literature review. Eur. J. Oper. Res. 259 (3), 801–817.

Santoso, T., Ahmed, S., Goetschalckx, M., Shapiro, A., 2005. A stochastic programming approach for supply chain network design under uncertainty. Eur. J. Oper. Res. 167 (1), 96–115.

Sbihi, A., Eglese, R.W., 2007. The Relationship between Vehicle Routing and Scheduling and Green Logistics – a Literature Survey. Technical Report. The Department of Management Science, Lancaster University Management School.

Smith, P., Haberl, H., Popp, A., Erb, K.-h., Lauk, C., Harper, R., Tubiello, F.N., Siqueira Pinto, A., Jafari, M., Sohi, S., et al., 2013. How much land-based greenhouse gas mitigation can be achieved without compromising food security and environmental goals? Glob. Change Biol. 19 (8), 2285–2302.

Solyalı, O., Süral, H., 2011. A branch-and-cut algorithm using a strong formulation and an a priori tour-based heuristic for an inventory-routing problem. Transp. Sci. 45 (3), 335–345.

Soysal, M., Bloemhof-Ruwaard, J.M., Haijema, R., van der Vorst, J.G., 2018. Modeling a green inventory routing problem for perishable products with horizontal collaboration. Comput. Oper. Res. 89, 168–182.

Speranza, M.G., Ukovich, W., 1994. Minimizing transportation and inventory costs for several products on a single link. Oper. Res. 42 (5), 879–894.

Wentges, P., 1996. Accelerating Benders' decomposition for the capacitated facility location problem. Math. Methods Oper. Res. 44 (2), 267–290.

Yu, M., Nagurney, A., 2013. Competitive food supply chain networks with application to fresh produce. Eur. J. Oper. Res. 224 (2), 273–282.